Paper Title:

Mutual Trust-Based Subtask Allocation for Human-Robot Collaboration in Flexible Lightweight Assembly in Manufacturing

Names, Affiliations and Addresses of the Authors:

S. M. Mizanoor Rahman
Department of Mechanical Engineering
Clemson University
Clemson, SC 29634, USA
E-mail: rsmmizanoor@gmail.com

Yue Wang*
Department of Mechanical Engineering
Clemson University
Clemson, SC 29634, USA
E-mail: yue6@clemson.edu
*Corresponding author

Mutual Trust-Based Subtask Allocation for Human-Robot Collaboration in Flexible Lightweight Assembly in Manufacturing

Abstract: A human-robot hybrid cell is developed for flexible assembly in manufacturing through the collaboration between a human and a robot. The selected task is to assemble a few LEGO blocks (parts) into a final product following specified sequence and instructions. The task is divided into several subtasks. A two-level feedforward optimization strategy is developed that determines optimum subtask allocation between the human and the robot before the assembly starts. Human's trust in robot and robot's trust in human are considered, computational models of the trust are derived and real-time trust measurement and display methods are developed. A feedback approach is integrated into the feedforward subtask allocation in the form of subtask re-allocation if trust levels reduce to below specified thresholds. It is hypothesized that subtask re-allocation may help regain trust and maintain satisfactory performance. Experiment results prove that (i) the integrated (feedforward + feedback) optimum subtask allocation is effective to maintain satisfactory trust levels of human and robot that result in satisfactory human-robot interactions (HRI) and assembly performance, and (ii) consideration of two-way trust (human's trust in robot and robot's trust in human) produces better HRI and assembly performance than that produced when one-way trust (human's trust in robot) or no trust is considered.

Keywords: Flexible manufacturing, assembly, human-robot collaboration, industrial robot, hybrid cell, subtask allocation, trust, robot vision, visual interface, prediction of robot behavior, safety, productivity, quality

1. Introduction

Global competitiveness in manufacturing is a challenging issue due to the requirements of high productivity and quality, low costs and highly skilled workforces [1]. Assembly in manufacturing significantly affects overall manufacturing productivity and quality due to extensive usages of labors, materials, utilities and maintenance in assembly operations [2]. Manual assembly is usually tedious, burdensome to workforces, inefficient and it affects worker's health and safety adversely [3]. Hence, automation of assembly should be prioritized, but it is usually expensive and inflexible [4]. We posit that appropriate collaboration between human and robot exploiting their complementary skills and competence can make the assembly more flexible, safe, cost effective and productive [5]. Recent advancements in lightweight low-cost flexible industrial robots, e.g. Baxter and Sawyer [6], Kinova [7], KUKA [8] have raised the possibility of such collaborations. This innovation is especially necessary for Small and Medium-sized Enterprises (SMEs) as such enterprises cannot afford highly expensive assembly automation due to financial limitations and frequent changes in assembly requirements.

Being motivated by the above prospects, Human-Robot Collaboration (HRC) in assembly has become an active area of research that has addressed different aspects of assembly in manufacturing, e.g. [9]-[13]. In [9], Tan *et al*. focused on design and development of HRC in cellular assembly in manufacturing. In [10], Wilcox *et al*. proposed optimization of temporal dynamics for adaptive HRC in assembly. In [11], Kaipa *et al*. presented HRC in hybrid cells for low volume assembly tasks. In [12], Sauppe and Mutlu proposed task training strategies for instructional robots for HRC in assembly. In [13], Gleeson *et al*. investigated gesture-based HRC in assembly, and so forth. The state-of-the-art initiatives (e.g. [9]-[13]) are undoubtedly helpful to promote the effectiveness of HRC in assembly. However, we still find two areas as follows that can add significant innovations and benefits to HRC in assembly, but have not received much attention yet:

1.1 Optimum Subtask Allocation

HRC in assembly can be executed in the form of a hybrid cell (a dedicated assembly space where a human and a robot can work simultaneously side by side without being separated from each other by physical cages) due to its various advantages such as convenient task allocation and scheduling and ease in resource mobilization, communication and supervision [11]. Subtask allocation in the hybrid cell for collaborative assembly divides the entire assembly task into several subtasks and assigns the subtasks to the human and the robot [14]. Subtask allocation is an important feature of the hybrid cell, which can affect overall assembly performance [18]. For subtask allocation, it is assumed that the task is divisible rather than unitary, which makes it possible that the human and the robot can work in parallel and no agent (human or robot) remains idle [15]. The main objective of subtask allocation is to minimize human's cognitive workload, improve situation awareness [16], increase team fluency [40] and safety [5], and maximize task performance (productivity, quality, etc.) [14]. Subtask allocation seems to be better than task transition between agents where one agent is always idle and thus reduces task inertia and efficiency [14].

Optimization of subtask allocation assigns right subtasks to right agents (human and robot) so that the overall performance is maximized through maximizing the utilization of resources and agent capabilities and minimizing agent

constraints [17]. Suboptimum subtask allocation in collaborative assembly can reduce safety, productivity and quality, and sometimes the assembly cannot be performed at all [14]-[17]. Such problems can be more critical for high-mix and low-volume assembly lines with frequent changes in requirements [18]. However, optimum approaches to subtask allocation for assembly in manufacturing are rare except a few initiatives, e.g. [18]-[20], [43]. Furthermore, effectiveness and practicality of these existing initiatives are not evaluated properly. A few task allocation methods have been proposed for other scenarios, e.g., space mission [14], satellite communication [46], multi-robot systems [17], [47], multiple humans working with multiple unmanned aerial vehicles (UAVs) [16], etc. Optimizations of instantaneous and resource-based task allocation for multi-agent systems have also been proposed [21]. In addition, a plethora of well-established optimization techniques are available in the state-of-the-art literatures for various other purposes such as business risk analysis [44], business decision making [45], etc. However, the state-of-the-art optimization methods and strategies for task allocation and other purposes cannot be directly applicable to HRC in assembly for a few reasons as follows: (i) the roles are switched between the agents instead of dividing the roles between the agents [14], (ii) the ideas are not tested through actual HRC [18], (iii) the task allocations are between multi-robots [17], [47], or multi-humans [16] instead of between a human and a robot, (iv) the techniques are good for business solutions, but may not be well-suited for assembly in manufacturing [44]-[45], and so forth.

The state-of-the-art task allocation strategies are of feedback type [18]-[20], [43], i.e. the optimum subtask allocation is determined after the assembly or activity is performed. This optimization approach seems to be very reliable as it takes the information of the actual assembly/activity into account. However, it is not practical in industrial applications because the optimum subtask allocation needs to be decided at the beginning of the assembly, but actual assembly information is not available at this stage [26]. This problem emphasizes the feedforward optimization of subtask allocation where the optimum subtask allocation can be determined before the assembly starts [26]. We predict that such feedforward optimization can be performed based on the potential feasibility of the subtask allocation instead of on the actual assembly information. Nonetheless, a possible drawback of the feedforward strategy may be that the optimization results are not very reliable as the optimization is conducted based on the information of feasibility analysis instead of on the actual assembly performance. Considering the above dilemma, we posit that an integrated approach combining feedforward and feedback optimization of subtask allocation can be more beneficial than an individual feedforward or a feedback approach. We argue that, in the integrated allocation, the feedforward optimization can help start the assembly with optimum subtask allocation and the feedback optimization in the form of subtask re-allocation can modify the feedforward optimization if needed while the assembly is in progress [22], [26]. However, initiatives for such integrated optimization of subtask allocation/reallocation in human-robot collaborative assembly in manufacturing have not received much attention yet except a preliminary initiative taken in [26].

1.2 Human-Robot Mutual (Bilateral) Trust in HRC in Assembly

Human's trust in the collaborating robot is the willingness of the human to rely on or to believe in the cooperation provided by the robot [23]. A satisfactory level of trust is mandatory because the human may not find interest to collaborate with the robot if the human does not trust it [23]-[24]. Again, the human may be influenced by the institutional trust towards the robots [50]. A few studies on human's trust in collaborating robots have been proposed, e.g. [25], but these studies are preliminary and are not related to human worker's trust in robot in assembly in manufacturing. We observe mutual (bilateral) trust between two or multiple humans that plays a significant role in human-human collaborative task [41]. We also observe interpersonal trust in society [51], which is bidirectional. Hence, being inspired by such bidirectional trust in nature, we in addition to considering human's trust in robot, argue that robot's trust in collaborating human should also be considered for human-robot collaborative assembly. The bilateral trust between the human and the robot can make the collaboration transparent to the partners (robot and human). Such transparency can allow the human and the robot to alter some aspects of their behaviors based on their mutual trust levels that can enhance the predictability of one agent's actions and behaviors to another agent and thus can improve the team fluency [26], [40]. Such predictions through trust can reduce human's cognitive workload as the robot's perceptions about the human become transparent through the robot's trust levels, and thus the human can devote more cognitive resources to the task instead of worrying about or trying to explain the actions and behaviors of the robot. The robot can also adjust its behaviors based on human's trust level to keep pace with the human. All these can enhance quality, productivity and safety in HRC in assembly in manufacturing [26].

In addition, bilateral trust status can be a criterion to decide subtask re-allocation between the collaborating partners, which can help regain trust. However, none have examined the possibility of giving the robot the ability to perceive its trust in its human collaborator for HRC in assembly except a few preliminary studies, e.g. [26]. As a consequence, it can cause uncertainty in the effectiveness of HRC in assembly even though the human and the robot start the assembly with optimum subtask allocation, which can affect assembly performance accordingly. Hence, bilateral trust between human and robot should be considered for HRC in assembly in conjunction with optimum subtask allocation. Again, trust depends on agent performance and faults, and thus the trust can change as the assembly progresses [27]. Hence, a computational

model of trust is necessary to include it in the execution of HRC in assembly and in triggering the trust-based subtask re-allocation [26]. However, such computational trust models for HRC in assembly have not received much priority yet except the preliminary concepts proposed in [26]. Trust in task allocation was considered in [28], but the optimization strategy was not designed properly and it was not related to assembly in manufacturing. Hence, based on above discussion, it can be understood easily that modeling of human-robot bilateral trust, real-time trust measurement and display, mutual trust-based assembly and trust-triggered subtask re-allocation during assembly demand special priority for upholding the overall assembly performance. However, such initiatives have not received much priority yet.

Being motivated by the above limitations of the state-of-the-art research works on HRC in assembly in manufacturing, we decided the objectives of this article as to: (i) derive computational models for human's trust in robot and robot's trust in human, (ii) develop methods to measure trust in real-time (or near real-time), (iii) develop an integrated (feedforward and feedback) strategy of optimum subtask allocation between the human and the robot triggered by trust, and (iv) evaluate the integrated optimum subtask allocation in terms of human-robot interaction (HRI) and performance and compare the findings among *two-way trust* (human's trust in robot and robot's trust in human), *one-way trust* (human's trust in robot) and *no trust* cases for collaborative assembly in flexible manufacturing to realize the impacts of trust on HRI and assembly performance.

2. The Case Studies, In-lab Assembly Task, Human-Robot Hybrid Cell and the Subtasks

2.1 The Case Studies

We made a few case studies in different manufacturing companies to identify some examples where HRC can be applied to enhance assembly performance. Based on our observations, we identified that the center console and the front fender assemblies and automation cell design especially layout and control element implementation in automation integration in automotive manufacturing, and assembly of wing especially wing-to-fuselage joint in aerospace manufacturing may be the cases where HRC can contribute significantly. Figures 1 and 2 illustrate the center console assembly case of automotive manufacturing. As Fig.1 shows, three main parts-face plate, I-drive and switch rows are assembled together to produce the center console product. In current practices, as Fig.2 (a) shows, the parts are stored at different shelves, the human worker collects the parts from the shelves, places on a small table and assembles the parts manually using screws and a screwdriver. We argue that this type of manual assemblies can be performed through HRC. For example, a robot can be set in a suitable location beside the human, and the human and the robot can collaborate to perform the assembly task based on an optimum subtask allocation scheme [26], as Fig.2 (b) shows. Note that this case can represent

(a) (b) (c)

Fig.1 Assembly of automotive center console, (a) the components (parts) to be assembled, (b) back, and (c) front views of the assembled product (center console) [courtesy: BMW Manufacturing Co., Spartanburg, SC, USA].

(a) (b)

Fig. 2 (a) The current practices of center console assembly. The parts are stored at different shelves in the assembly area so that the human worker can collect the parts to assemble on a table, (b) the proposed assembly through HRC where a human worker and a robot collaborate in the assembly area based on an optimum subtask allocation scheme to perform the assembly tasks [courtesy: BMW Manufacturing Co., Spartanburg, SC, USA].

all flexible assemblies in manufacturing that can be performed through HRC, and it is possible to find out more scope and opportunities in assembly lines where HRC can work. By *flexibility* we mean the ability to adapt with changes of processes and requirements. By representation of all flexible assemblies, we here want to indicate the assemblies that can be performed through HRC, whether it is rigid or puzzle-like. Similarly, it is also not an issue whether the human and the robot are physically coupled or decoupled, and whether the task is complex. The main point is whether the assembly can be performed through collaboration between human and robot. If so, then the proposed approach as follows can be applicable directly or indirectly.

2.2 The In-lab Assembly Task

To address the aforementioned types of HRC observed in actual assembly in manufacturing, we selected a simple assembly task that can be performed in our laboratory environment. The selected in-lab task is to assemble a few LEGO blocks (parts) into a final product through the collaboration between a human and a robot following specified sequence and instructions as shown in Fig.3.We consider it as a representative flexible lightweight assembly in manufacturing that can be performed through the collaboration between a human and a robot. However, we assume that the research results derived from this assembly can be applicable directly or indirectly to all flexible assemblies in manufacturing without significant loss of generality.

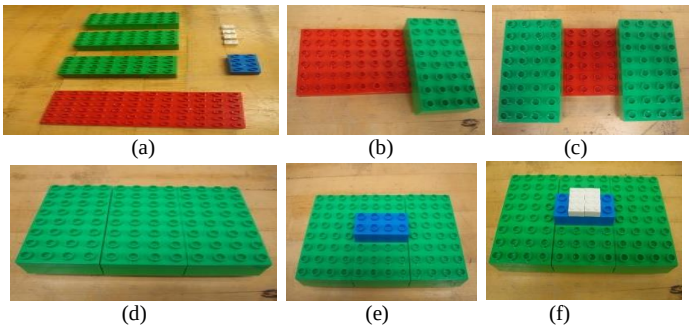


Fig.3 (a) All the parts that are to be assembled to produce the final (finished) assembled product, (b)-(f) the assembly sequence, and (f) the final assembled product.

2.3 The Human-Robot Hybrid Cell

We developed a one human-one robot hybrid cell [11] as shown in Fig.4 so that the human and the robot can collaborate to perform the selected assembly illustrated in Fig.3. We used a "Baxter robot" (research version). The robot contained an infrared (IR) rangefinder at the wrist, cameras at the head and the wrist and its head screen could be used to display images and texts [6]. As Fig.4 shows, the parts were initially input in different positions (P1, P3, P4) on table surfaces so that the human and the robot could collaborate to assemble the parts into final products at P2 and then dispatch the assembled products to P5.

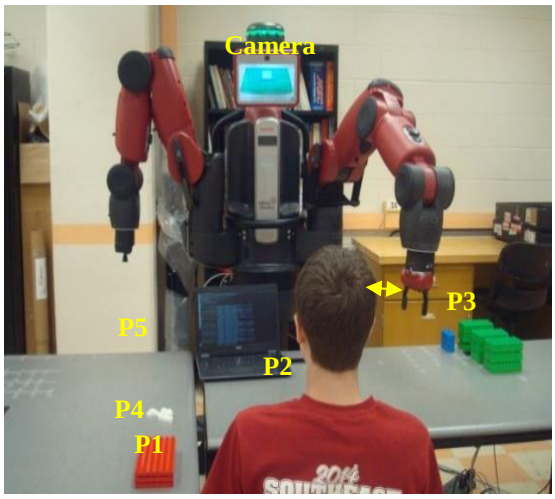


Fig.4 The human-robot hybrid cell for collaborative assembly of LEGO parts. Assembly parts required for three assembled products are seen. Assembly instructions are sequentially displayed in the robot's face screen in real-time. Arrow shows the minimum gap between human body and robot arm to avoid collision.

2.4 The Subtasks

The assembly task shown in Fig.4 was divided into several subtasks and the subtask execution sequence was determined as given in Table 1. We assumed that it was not possible to further divide the task and each subtask was independent [21]. We also assumed that the sequence in Table 1 (1-12) depends on the product requirements that are fixed by the assembly designer [14]. Optimum subtask allocation assigns the subtasks to the human and the robot in optimum way [26], as follows. Here, the sequence is the order of performance, it means that one preceding subtask must have been performed before the subsequent subtask can be performed.

Table 1: The subtasks for a complete assembly cycle with the assembly sequence

Sequence/Number	Subtask name
1	The red part is manipulated from P1 to P2
2	The first green part (green-1) is manipulated from P3 to P2
3	Green-1 is assembled (attached) with the red part (Fig.1(b))
4	The second green part (green-2) is manipulated from P3 to P2
5	Green-2 is assembled (attached) with the red part (Fig.1(c))
6	The third green part (green-3) is manipulated from P3 to P2
7	Green-3 is assembled (attached) with the red part (Fig.1(d))
8	The blue part is manipulated from P3 to P2
9	The blue part is attached with previous assembly (Fig.1(e))
10	The white parts (caps) are manipulated from P4 to P2
11	White parts are attached with the previous assembly (Fig.1(f))
12	Final assembled product is dispatched from P2 to P5

Remark 1: Here, the *collaboration* means that the human and the robot perform the subtasks assigned to them sequentially and in parallel keeping pace with each other to complete the assembly task. The subtasks preformed by each agent individually are interdependent in term of whole task performance. Each agent performs each subtask alone, but such performance is not independent. The fact is that the performance of each subtask by each agent is physically independent because each subtask is performed by each agent alone. However, one agent's performance depends on another agent's performance because the subtasks need to be performed in sequence keeping idle time zero, and one agent's output is another agent's input. It means that the agents are dependent/connected through task scheduling and sequence, which makes the overall task collaborative even though they perform each subtask alone separately in the same workspace.

Remark 2: In real industrial applications, the input parts can arrive just-in-time (JIT) through belt conveyors, and the final assembled products can be dispatched through another belt conveyor or stored in a shelf after manipulated by another arm of the robot.

Remark 3: The selected assembly task is simple, which can be more complex, e.g., the robot can hand over the parts to the human instead of just manipulating the parts, the robot itself can attach the parts, and so forth. However, our objective is not to show how complex assembly the robot and the human can perform collaboratively. Instead, the objectives are to investigate how (i) the subtasks can be allocated between the robot and the human in optimum way, (ii) mutual trust between them is modelled and measured in real-time, (iii) the assembly is performed based on mutual trust, (iv) the subtask re-allocation can be influenced by the mutual trust, and (v) all these affect the overall assembly performance. We posit that the presented HRC and subtask allocation approaches can be made applicable to all types of HRC in assembly in manufacturing directly or indirectly without significant loss of generality even in the cases where changes in assembly processes and addition of new subtasks can occur as long as the assembly task can be divided into subtasks and allocated to both the human and the robot.

3. Feedforward Optimization of Subtask Allocation

As given in Table 1, the total number of subtasks for the selected assembly (n) is 12. Hence, the total number of possible allocations of the subtasks, $N = 2^n = 4096$. We determine N possible allocations following a simple algorithm (**Algorithm 1**) in Python environment as follows:

Algorithm 1: Determination of N number of possible subtask allocations

1 Input total number of subtasks, (*n=12)*
2 Output total number of possible subtask allocations, (*N=4096)*
3 Publish each allocation separately so that each allocation contains *n* subtasks assigned to the agents (human and robot) in different combinations
4 End

From the results of Algorithm 1, an example of a possible allocation is as follows:

[1H 2R 3R 4H 5H 6H 7R 8H 9H 10R 11H 12R]

This allocation means that from Table 1, subtask no.1 is assigned to human (H), subtask no.2 is assigned to robot (R), subtask no.3 is assigned to robot (R),….., and subtask no.12 is assigned to robot (R). The optimum subtask allocation can be determined only after conducting the assembly in all *N* allocations, which is exhaustive. Furthermore, optimum subtask allocation is needed before the collaborative assembly starts. Hence, we here propose a feedforward approach to optimize the allocation, i.e. the optimum allocation is decided through minimizing a cost (objective) function derived on the information collected before the assembly starts. Literatures show various parameters that can be minimized to determine optimum subtask allocation, e.g. potential energy of agents, energy consumption, computation requirement, task execution time and costs, and so forth [17], [19], [21]. However, these parameters are used for feedback optimizations. For feedforward case as we consider here, correct information on these parameters are not available before the assembly is executed. Hence, we consider the feasibility of each allocation based on potential capabilities of the agents to perform the assigned subtasks in each allocation as the basis of the optimization of the subtask allocation [14].

We, based on our experience, identify four assessment criteria relevant to the selected assembly. Two criteria reflect potential capabilities of the robot to perform the assigned subtasks in each allocation, and two criteria reflect the requirements for the human to collaborate with the robot in the hybrid cell. These four criteria constitute the cost function (1), where D_{ia} is the level of potential incapability of the robot to generate required motion taking its available DOFs and the existing mechanical constraints into account to perform subtask *i* assigned to it in an allocation *a*, P_{ia} is the level of potential insufficiency of skills and precision of the robot to perform subtask *i* assigned to it in an allocation *a*, F_{ia} is the level of potential fatigue level (if the human continues the assigned subtask for long time during actual assembly) of the human to perform subtask *i* assigned to the human in an allocation *a*, M_{ia} is the level of requirement of potential movement of the human from his/her sitting position to perform subtask *i* assigned to the human in an allocation *a*. We assume low fatigue of the human as a requirement of the hybrid cell. We also treat it as a requirement of the hybrid cell that the human does not need to physically move from his/her sitting position while collaborating with the robot. For example, as in Fig.4, the human should not physically move from P2 to P3 (P3 is out of the human's reach from P2) to fetch a part because such movement may disrupt the assembly and may not help the human keep pace with the robot, which may reduce team fluency [40]. C_1, C_2, C_3, C_4 in (1) are positive real constants that can be decided based on the task, human and the robot.

$$J_a = C_1 \sum_{i=1}^{n=12} D_{ia} + C_2 \sum_{i=1}^{n=12} P_{ia} + C_3 \sum_{i=1}^{n=12} F_{ia} + C_4 \sum_{i=1}^{n=12} M_{ia} \quad , a = 1,2,3,\ldots\ldots N \tag{1}$$

3.1 First Level Feedforward Optimization

We see that when *N* is large it makes the optimization very difficult. We, in the first level optimization, phase out a significant number of saliently infeasible allocations (*m*) out of *N* possible allocations. The allocations with the following constraints (called the **feasibility constraints**) are identified as the saliently infeasible allocations: there is at least one subtask in an allocation where (i) the robot seems to be incapable of performing the subtask, (ii) the robot's skills and precision seem to be insufficient to perform the subtask, and (iii) the human needs to move physically from his/her sitting position to perform the subtask. The aforementioned feasibility constraints are decided based on our knowledge of the capabilities of the agents and of the assembly cell design. We develop an algorithm in Python environment (**Algorithm 2**) to execute this optimization.

Algorithm 2: First level feedforward optimization to phase out the most infeasible subtask allocations

1 Input total number of subtasks, (*n=12)*
2 Output total number of possible subtask allocations, (*N=4096)*
3 Publish each allocation separately that contains *n* subtasks assigned to the agents (human and robot) in different combinations
4 Input constraints for the human (the subtask numbers where the human has feasibility constraints, i.e. the human may not perform the subtasks)
5 Input constraints for the robot (the subtask numbers where the robot has feasibility constraints, i.e. the robot may not perform the subtasks)
6 Input constraints for both, if any (the subtask numbers where both the human and the robot have feasibility constraints, i.e. neither the human nor the robot may perform the subtasks)
7 Output total number of feasible allocations, *N-m*
8 Publish each of the feasible allocations (*N-m*) separately that contains *n* subtasks assigned to the agents (human, robot) in different combinations
9 End

3.2 Second Level Feedforward Optimization

Feasibility level of each of the feasible allocations (*N-m*) is assessed separately for the cost function criteria in (1) using a subjective rating scale (Likert scale [29]) as shown in Fig.5, where score 1 indicates extremely low level of infeasibility and score 7 indicates extremely high level of infeasibility.

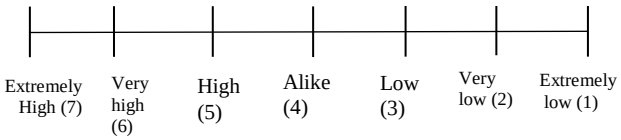


Fig.5 Likert scale to assess the cost functions criteria.

We develop an algorithm (**Algorithm 3**) in Python environment that determines the J_a for each feasible allocation that passes a set of constraints (called the **assessment constraints**) and then arranges the J_a in order. The assessment constraints are as follows: $D_{ia} < \sigma, P_{ia} < \sigma, F_{ia} < \sigma, M_{ia} < \sigma$ (σ is a threshold score and in our case $\sigma = 4$ as Fig.5 shows that score values higher than 4 correspond to higher infeasibility).The feasible subtask allocation corresponding to the lowest value of J_a returned by the optimization algorithm is identified as the optimum subtask allocation (the most feasible allocation). For this algorithm, we consider $C_1 = C_2 = C_3 = C_4 = 1$ in (1) to keep the model simple. It means that all optimization criteria are equally important. We can use some subjective scoring method to determine the exact values of these constants using system identification.

Algorithm 3: Determination of the allocation (a) with the lowest J_a value

1 Input the value of n, and the score values for $D_{ia}, P_{ia}, F_{ia}, M_{ia}$ (archive the values for each allocation in separate .csv files)
2 Input C_1, C_2, C_3, C_4 (we use all equal to 1 in our case for simplicity)
3 Input constraints: $D_{ia} < 4, P_{ia} < 4, F_{ia} < 4, M_{ia} < 4$
4 Calculate J_a for each a, $a = 1, 2, 3, \ldots \ldots \ldots N - m$
5 Return J_a values for the a that pass the constraints
6 Order the returned J_a values
7 Sort the allocation (a) with the lowest J_a value
8 End

From the results of Algorithm 3, the optimum subtask allocation for the selected assembly task was determined as follows:

[1H 2R 3H 4R 5H 6R 7H 8R 9H 10H 11H 12R]

The optimization results show that the repetitive subtasks and the subtasks requiring dexterous skills were allocated to the robot and the human respectively [18].

Remark 4: The proposed optimization techniques seem to be simple. However, the novelties and contributions are that (i) the cost function parameters are selected considering the practical needs of the targeted problems, and (ii) the parameters are assessed in feedforward. The feedforward nature of the optimization techniques focusing on feasibility and practicality is the key contribution and/or novelty of the proposed optimization techniques over the state-of-the-art techniques [18]-[20], [43]-[45]. In addition, the proposed feed forward optimization can reduce the complexity in optimization. As it is based on feasibility of subtask performance decided before the actual task is performed, this approach may be comparatively less reliable due to lack of actual assembly information, but it is not complex. However, the complexity can be further reduced by: (i) identifying more appropriate and practical assessment criteria, (ii) employing more experienced person for feedforward optimization who possesses high level knowledge and experience about the task requirements and agent capabilities to determine the feasibility of a subtask to be performed by either a human or a robot, (iii) using better and more interactive computing tool/software to compute the optimization results, etc.

Remark 5: Here, the "optimum allocation" actually means the "most feasible" subtask allocation determined in feedforward before the assembly starts, and this is the allocation that has the potential to be the "best" in terms of efficiency, quality, utilization, safety and productivity, if implemented. Equation (1) is an approximation of the objective function, and the algorithm is an approximation technique, and thus it is suboptimal. Appropriate mathematical optimization tools such as *gurobi* and *cplex* can be used to find out the optimal and scalable solution of (1).

Remark 6: Optimizations of subtask allocation can be associated with optimizations of subtask sequencing/precedence and scheduling [48]-[49]. In our case, the human and the robot perform the subtasks in sequence (Table 1) and they follow a well-defined schedule/time-line (see later). However, we here do not focus on the optimizations of subtask sequencing and scheduling. The constraints involved in *Algorithm 2* for human, robot and human-robot task execution capabilities and those in (1) are novel for the targeted assembly scenario. The allocation of subtasks may be heavily dependent on previous tasks and precedence. This is why we determined the precedence as in Table 1. Such precedence is not a constraint that limits the possibilities of parallel subtask allocations, if decided and optimized properly especially if the subtasks are allocated optimally considering necessary execution time of each subtask (see later). Let us explain what happens with subtasks 2 and 3. For subtask 2, the first green part (green-1) is manipulated from P3 to P2 by the robot, and then the green-1 is assembled (attached) with the red part (Fig.3(b)) by the human. When the human is working

(assembling), the robot does not sit idle. The robot goes back to the position from where it picked the previous part. The robot takes some time to go back to the previous position. Then, it picks a new part and again approaches to the human and places it in front of the human. In the meantime, the human needs to complete the assembly of the previous part. We believe that there is no scope to sit idle for any actor (robot or human) if such practice is repeated continuously. We can modify the precedence, sequence and scheduling and these will never be constraints if we further optimize the sequencing and scheduling [48]-[49]. The currently applied two-level feedforward optimization also leaves much room for applying tools of constraint programming and mathematical optimization as the problem size grows exponentially with the number of subtasks.

However, the subtask precedence is not constant. It depends on the nature of the task and assembly product design. It may also be customized through changes in product and process design. The optimization of scheduling of the subtasks as above can also be used to optimize the sequence/precedence so that it cannot generate any idle time in any actor. However, the subtask precedence cannot be ignored if the product design requires such precedence and if all subtasks cannot be performed at a time by the actors involved in the collaboration.

4. Computational Models for Trust and Real-Time Trust Measurement

In state-of-the-art literatures, trust usually refers to human's trust in automation, machine or robot [23]-[24]. Human's trust in robot (we denote it as T_{H2R}) can be influenced by the characteristics of the robot, the collaborative task, working environment as well as by the human's own characteristics [23]. Robot's trust in the human (we denote it as T_{R2H}) can also be influenced by similar factors. Trust is a perceptual issue and the human has actual feelings of his/her trust in the robot. However, it is not possible to produce similar feelings of trust in the robot. Recent analysis suggests that robot performance and faults are strongly correlated with T_{H2R} (r=0.34) [30]. A regression model was used in Lee and Moray's study [31] to first identify factors of human trust in automation that suggests a time-series model for T_{H2R} where robot's performance and fault factors are considered to model T_{H2R}. A general computational model of trust can be expressed in (2), where t is time step, $\emptyset_1$, $\emptyset_2$, A_1, A_2 are real constants relevant to specific collaborative system and q is random noise perturbation (if any).

$$Trust\ (t) = A_1 Performance(t) + A_1\emptyset_1\ Performance(t-1) + A_2 Fault(t) + A_2\emptyset_2 Fault(t-1) + q(t) \quad (2)$$

Based on (2), we use time-series models to capture T_{H2R} and T_{R2H} as in (3) and (4) respectively, where P_R is the reward for robot performance, F_R is the reward for robot fault status, P_H is the reward for human performance, F_H is the reward for human fault status, and $\beta_1, \beta_2, \gamma_1, \gamma_2, \varphi_1, \varphi_2, \omega_1, \omega_2$ are real constant coefficients that depend on the task, the human and the robot.

$$T_{H2R}(t) = \beta_1 P_R(t) + \beta_2 P_R(t-1) + \gamma_1 F_R(t) + \gamma_2 F_R(t-1) + q\ (t) \quad (3)$$

$$T_{R2H}(t) = \varphi_1 P_H(t) + \varphi_2 P_H(t-1) + \omega_1 F_H(t) + \omega_2 F_H(t-1) + q\ (t) \quad (4)$$

4.1 Robot Performance Modeling and Measurement

For flexible assembly, the robot needs to smoothly collaborate with the human while trying to maintain its efficiency. Thus, we consider both the robot's flexibility and efficiency for modeling its performance [32]. The robot performance model is proposed in (5), where $k_e + k_f = 1$. The coefficients k_e and k_f are gains for efficiency and flexibility respectively. In (5), V_H and V_R are human and robot's absolute manipulation speed respectively for performing the subtasks assigned to them. In (5), the first term depicts how human and robot velocity match with each other, i.e. flexibility, and the second term depicts the robot velocity, i.e. the efficiency. We use robot's manipulation speed as an indicator for its efficiency. We select human and robot's speed as a performance measure for the human and the robot respectively because achieving high assembly efficiency largely depends on human and robot's speeds during the collaborative assembly [26]. Figure 6 shows the scenario of the collaboration between the human and the robot to assemble the parts in the hybrid cell. V_R is set for the robot. V_H is measured in real-time by a phase space impulse X2 motion capture system (see Fig.6, the human wears a glove with LED markers of the motion capture system in his wrist during the assembly). We normalize the peak values of V_H and V_R between 0 and 1, thus P_R varies between 0 and 1. In the proposed model, $P_R = 0$ and $P_R = 1$ indicate the least and the best performance respectively.

$$P_R(t+1) = k_f\left[V_{R,max} - |V_H(t) - V_R(t)|\right] + k_e V_R(t) \quad (5)$$

4.2 Human Performance Modeling and Measurement

The human performance model is proposed in (6), where C_e and C_f are gains for efficiency and flexibility respectively. Here, $C_e + C_f$ =1, but $C_e = C_f$ as the human is equally efficient and flexible due to dexterous skills. The values of V_H and

V_R in (6) are same as that used in (5). The V_H can vary from zero (0) to a maximum. V_H can be the maximum when the human moves the hand to grasp a part to attach it to another part. $V_H \approx 0$ is considered when the human attaches a part to another part. The V_H can also vary between humans. We normalize the measured values of V_H between 0 and 1. Hence, P_H varies between 0 and 1, and $P_H = 0$ and $P_H = 1$ indicate the least and the best performance respectively [32]. For repetitive tasks, human's performance can be affected by muscle fatigue and recovery. However, the fatigue is expected to be low as the assembly is light. Here, the effects of fatigue are reflected through human's speed (i.e., the speed is high if the fatigue is low, and the vice versa).

$$P_H(t+1) = C_f\left[V_{H,max} - |V_H(t) - V_R(t)|\right] + C_e V_H(t) \quad (6)$$

In (5)-(6), $|V_H(t) - V_R(t)|$ indicates range of flexibility. Flexibility can be the maximum and minimum if $|V_H(t) - V_R(t)| = 0$ and $|V_H(t) - V_R(t)| = 1$ respectively. Concerned parameters of the models can be identified and the models can be verified using system identification methods [42]. However, we here validate the models experimentally as presented later. We here consider the absolute linear velocities of the robot and the human. However, angular velocities or combination of linear and angular velocities with appropriate weight factors can also be considered depending on the requirements of the assembly tasks. For the robot, we consider the absolute velocity of the robot wrist (last link) or the end effector starting when it starts picking the part and then approaches to the human and ending once it places the part in front of the human. The encoder embedded within the robot wrist measures it in real-time. We do not consider any joint closer to the human, but we consider the last link/joint, and it may be closer to the human. For the human, we consider the absolute velocity of human wrist and it is measured by the IMU worn at human wrist (Fig.6). Again, for the human velocity, V_H, to keep it simple, we only consider when the human moves the hand to grasp a part, but we do not consider when the human attaches the part. However, one can split the V_H into two components (one for part manipulation and another for part attachment) with appropriate weight factors to make the model more robust. The performance models in (5) and (6) are general. However, whether the human and robot velocities are linear or angular or combination of these two, and what methods of measurements of velocities are to be selected depend on the nature and requirements of the assembly tasks. The above measures of velocities for robot and human may not be the best measures, but as the preliminary initiatives, these seem to be very practical, realistic and also simple, and there is enough room to improve these measures.

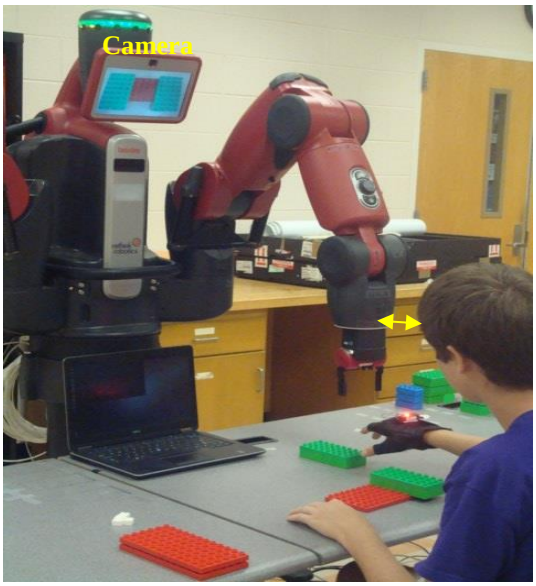


Fig. 6 The human collaborates with the robot to assemble the parts in the hybrid cell. The human is equipped with the performance measurement device (a glove with LED markers of the motion capture system worn at the wrist). Arrow shows the minimum gap between human body and robot arm to avoid collision.

4.3 Real-Time Robot's Fault Measurement

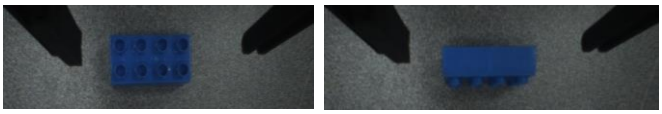
We consider it as the robot's fault if either of the following two cases occurs when the robot performs the subtasks assigned to it: (i) *case 1*: the robot end-effectors cannot grip the part for picking or the part is gripped and picked but falls from the end-effectors before it is manipulated to the human, (ii) *case 2*: the robot can grip and pick the part, then manipulates the part to the human, but it places the part with wrong orientation, e.g. tilted placement. For *case 1*, the

sensor embedded inside the robot's wrist detects if the part exists between the end-effectors when the robot reaches near the human. A message publishes 1 if the part is held between the end-effectors, otherwise the message publishes 0.

For *case 2*, the robot's vision system mounted at its wrist captures the images of the orientation of the placed part as illustrated in Fig 7. We see that for the reference orientation of a part in Fig.7 (a), the top surface of the part possesses many circles. However, at Fig.7 (b), when the part is placed incorrectly (tilted), there is no circle at its top surface. Hence, we follow the shape detection approach to detect such phenomena. To do so, the images are processed through the Circle Hough Transform algorithm [33] in real-time that outputs the number of circles on each targeted surface. Based on this information, the algorithm outputs score 1 if the reference orientation and the orientation of the placed part are the same, i.e. the numbers of circles on the top surfaces for the reference orientation and the actual orientation of the part are the same, otherwise the algorithm outputs 0.

Based on above information for *cases 1* & *2*, F_R (the reward score for the robot for its fault status) is calculated in real-time following (7). Hence, $F_R(t)$ can vary between 0 (if there are faults) and 1 (if there is no fault). In fact, the computed reward score due to fault status is equal to 1 when there is no fault.

$$F_R(t) = \begin{cases} \dfrac{case\ 1\ score + case\ 2\ score}{2} & if\ case\ 1\ score = 1 \\ 0 & if\ case\ 1\ score = 0 \end{cases} \tag{7}$$



(a) (b)

Fig. 7 Image of orientation of a part placed by the robot end-effectors after processing by OpenCV library, (a) reference placement, (b) faulty placement.

In section 4.3, fault case 2 and in Fig.7, the robot can grip and pick the part, then manipulates the part to the human, but it places the part with wrong orientation, e.g. tilted placement. If the task is known in advance, the location of placement is known and using the robot kinematics it is easy to predict whether the robot can make the proper orientation or not. We think that this is the ideal case, and in ideal case the robot can pick the fixed part from the fixed location and place in another fixed location. In ideal case, this may happen and thus no fault is expected, and thus it is not included as a part of initial optimization (i.e., feed-forward) process.

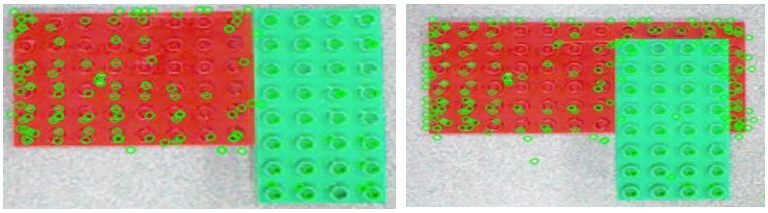
However, the fault may happen i.e., the robot may grip and pick the part, then manipulates the part to the human, but it places the part with wrong orientation, e.g. tilted placement, which is a deviation from the ideal situation. For example, due to error in robot kinematics, there may have a slight error or lack of repeatability in the targeted coordinates of the robot end-effector, the target assembly part location may be slightly deviated from the ideal location. In such cases, the robot can grip and pick the part, then manipulates the part to the human, but it places the part with wrong orientation, e.g. tilted placement because the gripping is not perfect and the part is not symmetrically hanging from the end-effector and it is about to fall. The part may narrowly reach near the human, but during placement, the lower end of the part touches the table and due to the impact forces the placement may not be perfect. Also, there is difference in surface texture and friction on the surface of the target part that can cause slipperiness that can also cause wrong orientation during placement. We can predict this and include in feedforward optimization by increasing the accuracy in robot trajectory planning, arranging vision-based guidance to the robot, taking the surface texture and friction co-efficient of the object into account when planning the gripping mechanism and end effector's task, etc.

4.4 Real-Time Human's Fault Measurement

We consider it as the human's fault if either of the following two cases occur when the human performs the assigned subtasks: (i) *case 1*: the human assembles a wrong part, and (ii) *case 2*: the human misplaces a right part when attaching it with another part. Both cases are detected by the vision system of the robot located at its head and then the images are processed by the OpenCV library. Figure 8 illustrates *case 2*. The Scale-Invariant Feature Transform (SIFT), an algorithm in computer vision, detects and describes the local features in the images in terms of key points [34]. Then, the FLANN (Fast Library for Approximate Nearest Neighbors)-based matcher algorithms determine the key point matching between the performed assembly and the reference assembly [35]. We determine $F_H\ (t)$, the reward score for the human for fault status, in real-time as in Table 2 where M is % of matching key points between reference and performed assembly at a time step. *Case 1* is a special case of *case 2* when M may be very low because in this case the assembled part is totally wrong and different from the reference part. $F_H(t)$ can vary between 0 (if there are faults) and 1 (if there is no fault).

It is to note that the measurement of F_R and F_H between 0 and 1 is a general requirement for the trust models in (3) and (4) respectively. However, the methods of measurements depend on the nature and requirements of the assembly tasks. In addition, F_R and F_H can take many intermediate values between 0 and 1. However, it depends on whether one can

classify/segment the entire fault into different severity levels, assign a score value to each severity level/segment, and measure the fault segment in real-time.



(a) (b)

Fig. 8 Demonstration of the processing of the key points by the OpenCV library, (a) an example for the reference assembly, (b) an example for a faulty assembly.

Table 2: Measurement of reward scores for fault status for the human

M value (%)	$F_H\ (t)$ score
M ≥70	1
70>M≥50	0.75
50>M≥30	0.5
30>M≥10	0.25
M<10	0

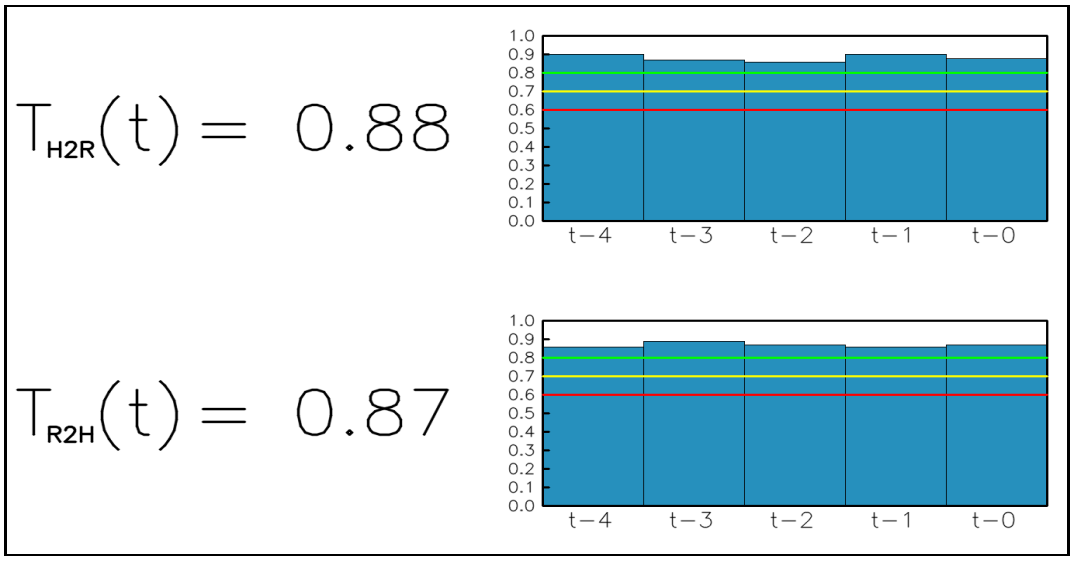


Fig.9 Typical real-time display of the computed trust values in the computer screen (human-machine visual interface). The green, yellow and red lines indicate different warning levels to the agents (human and robot) based on the concerned trust values.

Reward scores for fault status for the robot and the human are updated at every time step, *t*. However, if multiple fault incidents occur in a *t*, then the score for each incident is computed and the average score is used to compute the trust for that time step. T_{H2R} and T_{R2H} can be computed in real-time following (3) and (4) respectively once the performance and faults for the human and the robot are measured in real-time. The computed T_{H2R} and T_{R2H} are displayed in the computer screen placed in front of the human subject as in Fig.6 and updated at every time step, t . Fig.9 shows how the trust values are displayed in the computer screen. The trust values of five recent time steps are displayed in the graph as shown in Fig.9 so that the human worker can easily understand the trend of the trusts.

The fault cases 1 and 2 for human can probably occur due to the inexperience of the human, and that after some attempts the human can probably become expert at it. Note that even though the human can practice it many times and become an expert, the human can make mistakes and cause faults through many ways. For example, human fatigue, lack of attention, absent mind, excessive work pressure, overseeing other associated works, etc. So, such faults can happen even when experienced and expert workers work. In industry, where tasks may be changing often and pre-learning may cause losses in production, this may not be feasible. This is the reason why even an expert human worker can make fault

that needs to be modeled and evaluated. This is exactly reflected in what we propose here. Thus, the proposed model seems to be necessary and practical.

If the human and robot work together on a particular common subtask, then the proposed approach can still work. Similarly, it is also not a problem whether the human and the robot are physically coupled or decoupled, and whether the task is complex such as where three hands (two from human and one from the robot) are required to simultaneously interact with the object. In such cases, the individual performance and fault status of both the robot and the human need to be modeled and determined separately in real-time to determine mutual trust. We posit that the proposed model is general and it can be applied to any collaborative task between human and robot, but the measurement of performance and fault status of robot and human need to be decided based on the particular scenario. However, the entire task should be such that it can be divided into several subtasks and be assigned to both human and robot depending on their capabilities and constraints.

The human and the robot work collaboratively on the same task though they are individually working on separate subtasks keeping pace with each other. As they are collaborating, one's trust in another can affect the overall assembly performance, and thus transparency of trust between them is necessary, which we consider. Here, the human and the robot do the same task, not the same subtasks. Thus, the human monitoring the robot's trust in real-time makes sense if they work together and if one's work is impacted by and depended on other's work whether the interdependency or impact is based on performing on a single task (but separate subtasks as in our case) or on a single subtask.

5. Integrated Feedforward and Feedback Optimum Subtask Allocation Strategies Based on Trust

The overall trust-based integrated (feedforward + feedback) subtask allocation scheme for HRC in assembly in the hybrid cell is shown in Fig.10. As the figure shows, at first, the optimum feedforward subtask allocation is determined through two-level feedforward optimization procedures (Section 3). Then, motions are planned to enable the robot to perform each subtask allocated to it sequentially in parallel with the human. Function (Applications Programming Interface, API) for predefined motion trajectory for each subtask is separately stored in the robot control server. The server and client communicate via Robot Operating System (ROS). The Remote Procedural Call (RPC) [36] is used to generate motion in the robot in real-time to perform each subtask assigned to the robot sequentially. Fig. 11 illustrates the RPC for a subtask.

The assembly starts in the hybrid cell based on the feedforward optimum subtask allocation (robot motions are executed for the allocated subtasks and the human performs the subtasks allocated to the human in parallel with the robot). T_{H2R} and T_{R2H} are estimated in real-time, updated at every time step t and displayed in the screen (see Fig.9). The assembly continues as long as the computed T_{H2R} and T_{R2H} are above the thresholds δ_{11} and δ_{12} respectively and the difference between the trusts is below a threshold δ_3. This satisfactory trust levels are shown in Fig.9 as the green line. However, if any of the thresholds is not satisfied, a second set of thresholds (δ_{21} and δ_{22}) related to trusts are checked. If the thresholds are satisfied, then the collaborative assembly continues as usual, but a mild warning message (e.g., 'human should improve trust' or 'robot should improve trust') is displayed in robot face screen to make the concerned agent aware of the trust level. This trust level is shown in Fig.9 as the yellow line. If the awareness message applies to the human, the human as an attempt to respond the message should speed up the assembly and/or become more aware of any faults based on what is warned in the message. The same applies to the robot. In our case, the robot itself does not speed up or become more aware of its faults. Instead, based on the warning messages to the robot, the experimenter sends a command to the robot to increase its speed and the experimenter also checks the robot's gripping status of the input parts to avoid any fault, which may enhance T_{H2R}. However, an autonomous controller may be used to enable the robot to automatically adjust its speed to achieve the desired T_{H2R} [26].

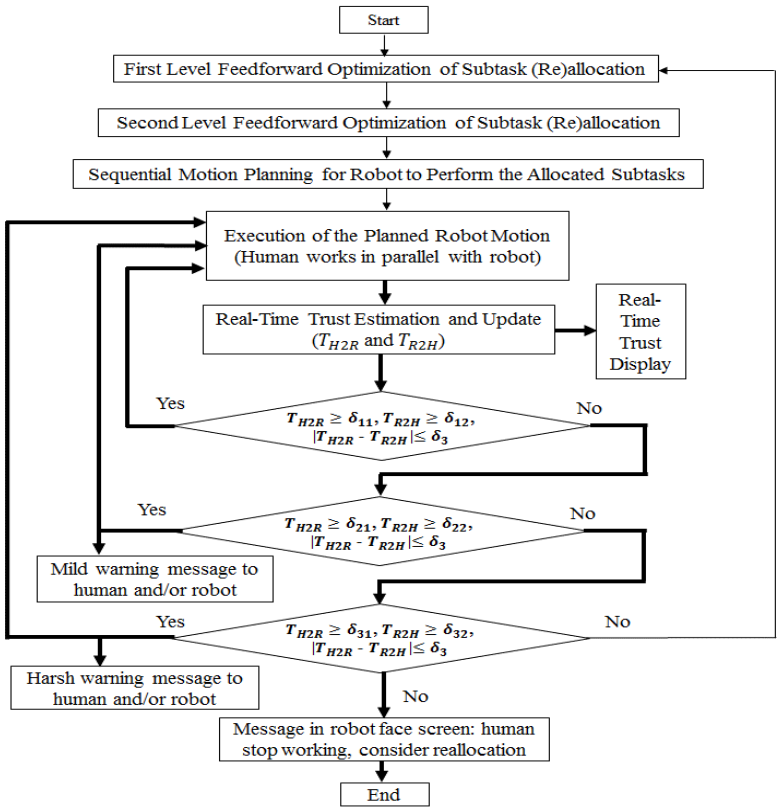


Fig.10 Execution plan of the trust-based integrated subtask allocation for HRC in assembly in the hybrid cell. The process is updated at every time step *t*. The thick and the thin arrows indicate real-time and off-line execution respectively.

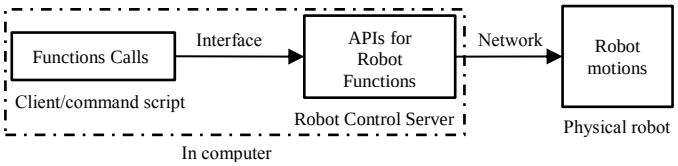


Fig.11 The RPC that enables the robot to perform the allocated subtasks sequentially keeping pace with the human.

However, if any of the second set of thresholds is not satisfied, a third set of thresholds (δ_{31} and δ_{32}) are checked. If this set of thresholds are satisfied, the assembly continues, but a harsh warning message is displayed in robot face screen (e.g. 'human must improve trust', 'robot must improve trust'). This trust level is shown in Fig.9 as the red line. The human and/or the robot respond the messages following the same procedures as stated above. However, if any of the third set of thresholds is not satisfied, the assembly is suspended, the robot stops working, a message is displayed in robot face screen to ask the human to stop and consider subtask reallocation. We adopt the following hypothesis in this regard:

Hypothesis 1: In human-robot collaborative assembly, the human and the robot should continue the assembly as long as they trust each other (i.e., trust values up to certain levels are maintained), and their mutual trust may be enhanced or regained through subtask reallocation between them.

Here, the subtask reallocation means redistributing the subtasks (Table 1) between the human and the robot (Fig.4) based on new optimization results [22]. For subtask reallocation, the two-level feedforward optimization procedures (Sections 3.1 and 3.2) are repeated. However, feasibility constraints (Section 3.1) and feasibility assessment (Section 3.2) may be influenced by the feedbacks (updates in knowledge and experience about the feasibility of each subtask) achieved through the actual assembly that takes place until the decision of reallocation is made. Thus, the feedbacks may also influence the new optimization results. Hence, the newly determined optimum subtask allocation (subtask reallocation) based on feedbacks and the new optimization results can be different from and more accurate than the previous optimum subtask allocation determined through feedforward optimization method. Feedforward optimization of subtask allocation integrated with subtask reallocation through re-optimization of subtask allocation taking actual feedbacks into account is

treated here as the "integrated optimum subtask allocation scheme" (Fig.10). However, subtask reallocation is considered if and only if the feedforward allocation fails to maintain appropriate trust levels.

Re-allocating the subtasks to the robot may prevent human becoming an expert. It is always not true that a human expert can do the subtasks much better than the robot. Human may do better in term of quality, but may face fatigue and may not be suitable for long time repeated assembly task. However, re-allocation is necessary sometime depending on overall situations as reflected in Fig.10, and such reallocation is also determined through an optimization procedure. Hence, we need to make a compromise between creating expert human worker and reallocation of subtasks to maintain desired mutual trust, performance and quality levels.

According to the proposed policies in Fig.10, if performance of both the human and the robot is bad and trust falls below the third level threshold, then the subtask re-allocation is proposed. Note that the re-allocation is also determined through the feedforward optimization procedure. Hence, after re-allocation, it is expected that the mutual trust will be high due to high performance and low fault of both human and robot. However, if it happens that first the human does a particular subtask with bad trust before the re-allocation and then in the re-optimization/re-allocation that subtask is allocated to the robot and then the robot's trust is also below the third threshold, then according to Fig.10, re-allocation will be conducted again. Even though re-allocation may improve overall assembly performance, re-allocation wastes some time and thus too many repeated re-allocation practices may not be acceptable. In such cases, the capabilities of the actors (robot, human) can be enhanced. For example, human worker may be provided additional trainings. New accessories and intelligence can be added to the robot or even the robot can be replaced by a better and more suitable robot. The assembly task design and subtask division can be revisited and some complicated subtasks can be outsourced from or performed by external facilities. We here do not specify how many times the re-allocation can be considered for a particular assembly task. This decision is open to the assembly management team.

Risk assessment is a fundamental topic in HRC application, and it is highlighted in the standards ISO 10218-1/2 and ISO/TS 15066 [52]. We successfully managed potential harmful movements of the robot due to task reallocation, and in all movements that the robot made. Evaluation of potential risks and the related counteractions were not included in the proposed trust model directly, but we addressed this issue indirectly, and we ensured that the human-robot collaboration was free of risk. We confirmed this in the following ways:

(i) As Figs.4 and 6 shows, Baxter has a camera near its head screen. We developed and implemented a vision-based algorithm that worked in such a way that if the distance between human body part (head, hand) and the left arm of Baxter is less than or equal to a specified threshold distance, then the robot stops movement of the arm. The robot holds the movement of its arm if the human does not remove his/her body part from the proximity of the robot arm and maintain a safe distance. We call it "*vision-based proximity sensing*", which was effective to avoid any accidents and injuries as well as damages of materials while the human was working with the robot. Risk was not directly assessed. However, potential harmful movements of the robot due to task reallocation and in all other cases were avoided using the proposed *vision-based proximity sensing* method. Thus, the presented method was aware of the safety issue and tried to avoid collision and risk in parallel with increasing performance and reducing fault. Hence, we posit that the collaborative system was guided by the standards ISO 10218-1/2 and ISO/TS 15066 [52].

(ii) Agent performance and fault avoiding abilities that constitute trust do not include safety issues directly. However, performance and fault avoidance and hence high trust cannot occur properly in the cases of collisions. Hence, in the high mutual trust cases, we indirectly guarantee that there is no collision and thus there is no or very little risk. In the future, it can be considered to include a risk factor in the mutual trust model to directly comply with the standards ISO 10218-1/2 and ISO/TS 15066 [52].

6. Evaluation of the Mutual Trust-Based Integrated Optimum Subtask Allocation Scheme

6.1 Recruitment of Human Subjects

Twenty (20) human subjects (mechanical engineering graduate and undergraduate students, 17 males and 3 females, mean age 25.46, standard deviation 3.71) were selected to participate in the experiment. The subjects gave informed consent. The experiment involving human subjects was approved by the Institutional Review Board (IRB).

6.2 Methods and Procedures

The subjects were instructed about the experiment procedures and evaluation methods. In the first round practice session, each subject was separately asked to practice the subtasks assigned to the human (the robot also performed the subtasks assigned to it). The practice trials might remove the learning effects of the subjects. Completion time of each subtask

performed by the human was recorded. Then, the standard time for each subtask performed by the human was determined. The robot speed was adjusted with that of the human so that the idle time could be kept zero. Fig. 12 shows the standard timeline (schedule) for the subtasks. Note that the human and the robot perform the assigned subtasks as depicted in Fig.12 in sequence (see Table 1 as well) following the timeline/schedule. However, we here do not focus on optimizations of subtask sequencing and scheduling [48]-[49]. We determined utilization ratio (U_r) following (8) based on Fig.12, where T_h and T_r are the total time utilized for a single assembly by the human and the robot respectively. We found that $U_r \approx 1$, which indicates no idle time in ideal case.

Based on practice trials, the parameter values necessary to estimate trust in real-time and execute the subtask allocation scheme (Fig.10) were determined (Table 3).The information on agent performance and faults in the practice trials was used to compute the constant coefficients of the computational trust models in (3) and (4) following Autoregressive Moving Average Model (ARMAV) [31], [32]. The gains k_e, k_f, C_e and C_f were decided based on our experiences, but those could also be determined following system identification method [42]. We depended on our experiences to keep these simple at the preliminary initiatives. For human performance, we used $C_e = C_f = 0.5$ as the human is equally efficient and flexible due to dexterous skills. For robot performance, the coefficients k_e and k_f are gains for efficiency and flexibility respectively. We used $k_e = 0.25$ and $k_f = 0.75$ based on our experiences as we expect that the robot needs to be more flexible than efficient for the targeted application. However, this is our assumption and this can be adjusted. Thresholds in Table 3 used in Fig.10 and the time step *t* were also decided based on our experiences. However, in actual industrial applications, these parameters can be adjusted for specific requirements. As *t*=30s, the trust measurement (Section 4) and the integrated optimum subtask allocation scheme (Fig.10) may be treated as "near real-time".

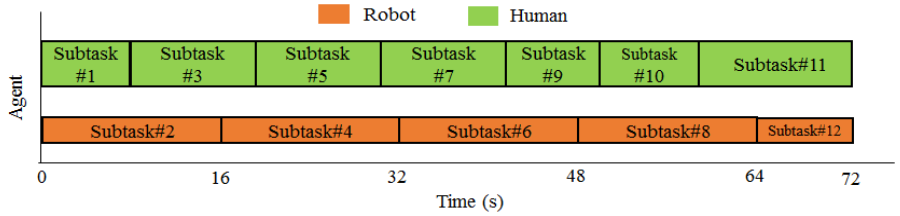


Fig. 12 Targeted standard timeline (amount of time)/schedule to perform the subtasks for a single assembly by the agents.

$$U_r = \frac{T_r}{T_h} \tag{8}$$

Table 3: Parameters with values to estimate trusts and execute the optimum subtask allocation scheme

Parameter	Value	Parameter	Value
β_1	0.459	C_e	0.5
β_2	0.063	C_f	0.5
γ_1	0.431	δ_{11}	0.8
γ_2	0.047	δ_{12}	0.8
φ_1	0.463	δ_{21}	0.7
φ_2	0.069	δ_{22}	0.7
ω_1	0.426	δ_{31}	0.6
ω_2	0.042	δ_{32}	0.6
k_e	0.25	δ_3	0.3
k_f	0.75	t	30s

In the second round experiment session, we conducted formal experiments based on optimum subtask allocation scheme in three separate conditions (protocols):

i. Considering both T_{H2R} and T_{R2H} ("*two-way trust*")
ii. Considering only T_{H2R} ("*one-way trust*")
iii. Without considering trust ("*no trust*")

We adopted the following hypothesis:
Hypothesis 2: In human-robot collaborative assembly tasks, two-way trust condition may be more beneficial than one-way or no trust conditions.

For *two-way trust protocol*, we implemented the trust-based optimum subtask allocation scheme of Fig.10. Fig. 6 illustrates the experiment procedures. Each subject was separately asked to continue the assembly with the robot for 30 minutes (due to limited number of LEGO parts, we quickly disassembled the finished assembled products and input them again to the agents to keep the assembly continuing). Total times taken by the human and the robot to perform the subtasks assigned to them for each finished assembly were recorded by stop watches. We measured the team fluency between the

human and the robot for the HRC. Team fluency is the coordinated meshing of joint efforts and synchronization between the human and the robot during the collaboration [40]. Four criteria as follows were used to measure the team fluency objectively:

- *Robot and human's idle time*: It was the time that the robot and the human waited for sensory input, information processing, computing and decision-making, etc.
- *Non-concurrent activity time*: It was the amount of time during a trial that was not concurrent between the human and the robot when it was supposed to be concurrent.
- *Functional delay*: It was the time between the end of one agent's action and the start of another agent's action.

Each team fluency criterion was expressed as a % of the total trial time. The criteria might be interrelated. The selected 20 subjects participated in the assembly separately following the above procedures. After the experiment, each subject evaluated his/her cognitive workload following NASA TLX [37]. We assessed each subject's situational awareness using the "Situational Awareness Global Assessment Tool-SAGAT" based on a subjective rating scale (score 1 for the least, score 2 for the most) [38]. The subject's likeability was expressed through reliability, naturalness and engagement, and was assessed by each subject using a 5-point Likert scale (1 for the lowest, 5 for the highest) [12], [29]. Here, reliability is human's intention to collaborate with the robot for similar tasks in the future. Naturalness is normalcy, comfort, ease of use and non-complexity experienced by the human for the collaborative assembly. Engagement is human's physical and mental involvement, rapport in interactions and connectedness [39].

For *one-way trust protocol*, the same experiment procedures and evaluation methods as used for the *two-way trust* were followed, but T_{R2H} and $|T_{H2R} - T_{R2H}|$ were not considered when implementing the subtask allocation scheme of Fig.10. Only T_{H2R} was computed and displayed to subjects, warning messages applied to the robot only and the decision of subtask reallocation was made based on the thresholds related to T_{H2R} only.

For *no trust protocol*, the same experiment procedures and evaluation methods as used for the *two-way trust* were followed. The human and the robot collaborated for the assembly based on the optimum subtask allocation and the robot's motion was planned for the subtasks allocated to the robot (Fig.10). However, trust was not estimated and displayed for the human and the robot. Trust-based warning messages were not displayed and trust-triggered reallocation was not considered.

Each of the 20 subjects participated in each experiment condition (protocol) separately, but we assigned the experiment protocols to the subjects randomly that might help counterbalance the learning effects.

7. The Evaluation Results

Fig.13 shows, as example, the typical evaluation (reward) scores for performance and fault status for a human subject and the robot for a few consecutive time steps in a trial under the *two-way trust protocol*. Note that we are not presenting the evaluation scores for performance and fault status for all time steps for a trial (30 minutes) because it requires huge space, and the results for the selected time steps can sufficiently represent the trends of the entire results. Fig.14 shows the corresponding T_{H2R} and T_{R2H} values. The figures show that the reward scores for the human and the robot performance and fault status varied randomly, and the variations in reward scores caused variations in the corresponding trust levels. As Fig.13 shows, the reward scores for robot fault status were low that caused low T_{H2R} in Fig.14, and the reward score for robot fault status was zero at t=150s that caused T_{H2R} in Fig.14 to cross the red line (Fig.9), which suggested subtask reallocation (Fig.10). Investigation shows that the fault was related to the difficulty of the robot in performing subtask#12 allocated to it (Table 1) based on feedforward optimization scheme of subtask allocation. During feedforward optimization (Section 3), it seemed to be feasible that the robot would be able to pick the final assembled product (Fig.3 (f)) at P2 (Fig.4) and dispatch the product to P5 (Fig.4) as the robot grippers and the reach of the robot arm seemed to be suitable for such pick and place operations. The robot was also able to perform similar operations during the practice trials. However, the practice trials were discrete in nature, i.e. one assembly was performed, then there was a pause and then another assembly was practiced. During actual continuous assembly, the human was under time pressure and thus could not perfectly synchronize with the robot motion. The human also placed the finished product at a position near P2 that slightly deviated from the predefined position. Thus, position coordinates for the finished product did not match with the commanded position coordinates of the robot for pick and place operation of the product. Thus, the robot could not pick the product, or it could pick the product but could not place it properly at P5 as the product was about to fall from the grippers, which reduced the reward score for the robot fault status as of (7). Based on execution plan in Fig.10, reallocation of subtask was required. We determined new optimum subtask allocation following Sections 3.1 and 3.2 as follows:

[1H 2R 3H 4R 5H 6R 7H 8R 9H 10H 11H 12H]

We see in the new allocation that only the difference was that subtask#12 was allocated to the human instead of the robot. We implemented the new allocation following the same procedures as above. However, a new timeline (similar as Fig.12) was necessary to maintain $U_r \approx 1$ through adjusting the robot speed. Typical reward scores for performance and

fault status for a subject and the robot for a few consecutive time steps in a trial under *two-way trust* for the reallocated optimum subtask allocation are shown in Fig.15, and the corresponding T_{H2R} and T_{R2H} values are shown in Fig.16. Fig.16 shows that both T_{H2R} and T_{R2H} are above the green line. Similarly, trust values for the entire assembly trial for the subject were found above the green line. Similar trends of trust values were also observed for other subjects under *two-way trust*. Similar trends of T_{H2R} values were also observed for *one-way trust* for each trial of each subject. Hence, further reallocation of subtasks was not necessary. The results thus justified hypothesis 1.

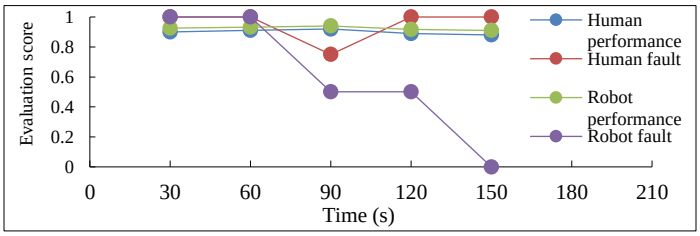


Fig.13 Representative evaluation (reward) scores for performance and fault status for a human subject and the robot for a few consecutive time steps in a trial under *two-way trust* protocol for the feedforward optimization of subtask allocation.

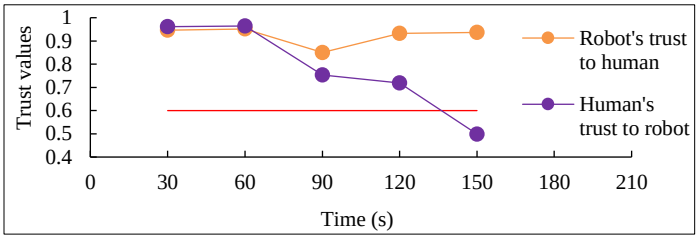


Fig.14 Representative trust values for a human subject and the robot for a few consecutive time steps in a trial under *two-way trust* protocol for the feedforward optimization of subtask allocation. The red line is the threshold for subtask reallocation decision.

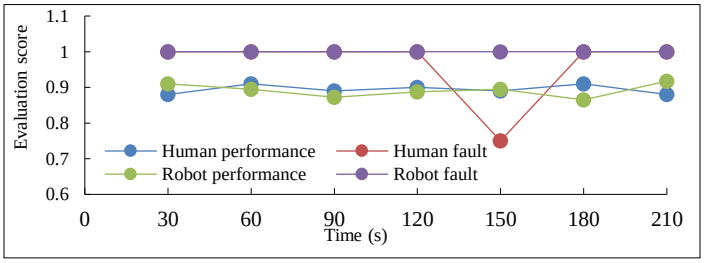


Fig.15 Representative evaluation (reward) scores for performance and fault status for a human subject and the robot for a few consecutive time steps in a trial under *two-way trust* protocol for the reallocated (integrated=feedforward +feedback) optimum subtask allocation.

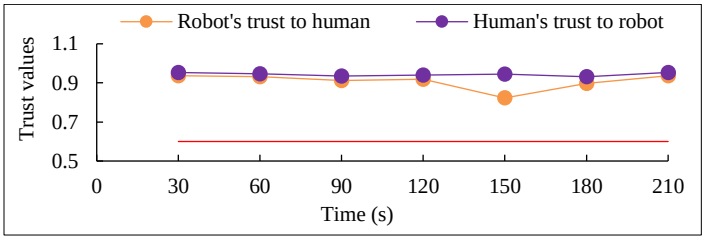


Fig.16 Representative trust values (between 0 and 1) for a human subject and the robot for a few consecutive time steps in a trial under *two-way trust* protocol for the reallocated optimum subtask allocation. The red line is the threshold for subtask reallocation decision.

The overall results indicate the effectiveness of the trust-based integrated optimization of subtask allocation. The effectiveness of the feedforward optimization was also proven as the allocation of only one subtask (#12) resulted from feedforward allocation was found troublesome, which was then corrected through new optimization in the reallocation.

Fig. 17 compares the situational awareness among *no trust*, one-*way* and *two-way trust* protocols for the integrated optimum subtask allocation. The results show that situational awareness was significantly higher for *two-way trust* than that for *one-way trust*. We posit that the higher bilateral transparency about performance and faults of the agents and the real-time communication between the agents through display of trusts helped the human perceive situations correctly, comprehend the situations properly and predict future situations confidently, which enhanced the overall situational awareness of the human [38]. We see that the situational awareness was high for *one-way trust* though it was

comparatively lower than that for *two-way trust*. On the other hand, the situational awareness especially situation comprehension and prediction were poor for *no trust* due to no transparency and communication for agent actions, performance and faults. Results of Analyses of Variances (ANOVAs) showed that variations in situation perception, comprehension and prediction among *no trust*, one-*way* and *two-way trust* protocols were statistically significant (p<0.05 for each case), which statistically proved the differential effects of consideration of trust on human's situational awareness. However, variations in situation perception, comprehension and prediction for *no trust*, one-*way* and *two-way trust* protocols were not statistically significant (p>0.05 for each case) between the subjects, which indicated the generality of the results.

Fig.18 compares the cognitive workload among *no trust*, *one-way trust* and *two-way trust* protocols for the integrated optimum subtask allocation [14]. The results show that the workload was significantly lower for *two-way trust* than that for *one-way trust*. We posit that displays of trusts improved transparency of performance and fault of the human and the robot and enhanced bilateral real-time communication and predictability between the agents that reduced the workload. Better situation awareness for *two-way trust* (Fig.17) might also contribute lower workload to the *two-way trust*. We see that the workload for physical demand was not so different between *one-way* and *two-way trust*. It happened because the human invested almost the same amount of physical labor for both conditions as the physical characteristics of the subtasks (e.g., complexity, part weight) were the same. For *one-way trust*, the above advantages were unilateral and partial, i.e. robot's status was displayed to the human, but the human's status was not displayed to the robot, which made the workload slightly higher. Nevertheless, the workloads for both conditions were not so high though it was higher for *one-way trust* than that for *two-way trust*. On the contrary, the workload for *no trust* was higher than that for *one-way* and *two-way trust*. We believe that absence of transparency and communication of performance and fault status of the agents reduced predictability of one agent to another, which increased human's cognitive workload. Results of ANOVAs showed that variations in cognitive workload among *no trust*, one-*way* and *two-way trust* protocols were statistically significant (p<0.05 for each case), which statistically proved the differential effects of consideration of trust in collaborative assembly on human's cognitive workload. However, variations in cognitive workload were not statistically significant (p>0.05 for each case) between the subjects, which indicated the generality of the results.

In *no trust* condition, the human has no or little information about the status of the collaborative task performance. The human cannot assess how he/she is performing and maintaining the quality. Similarly, the human also cannot clearly understand how the robot is performing and maintaining the quality. In such unclear situation where there is no or little transparency, human's mental and temporal demand and frustration increase, mental pressure to maintain performance increases, and the human may need more efforts. For these reasons, the overall cognitive workload can increase. Workload for physical demand also increases slightly. The opposite happens for *two-way* trust. The human has very high level transparency about human's own and the robot's performance and fault status, which significantly lowers human's cognitive workload. It is true that the human needs to focus on the monitor to know how the mutual trust is displayed in real-time. However, such monitoring may slightly put pressure on human's attention, and it should not impact human cognition too much as the human just observes the displayed trust and does not need to compute it. In this way, we believe that the reduction in cognitive workload due to transparency through trust display is more impactful than the cognitive workload needed to understand the displayed trust in the monitor. As a result, the resultant cognitive workload decreased for *two-way trust* condition over *no trust* condition.

Fig.19 compares the human subject's likeability of the collaborative assembly for *no trust*, *one-way* and *two-way trust* conditions for the integrated optimum subtask allocation. The results show that the likeability was significantly better for *two-way trust* than that for *one-way trust* probably due to the higher situation awareness and lower cognitive workload. Nevertheless, the likeability was high for both *one-way* and *two-way trust* protocols. However, the likeability was low for *no trust* condition probably due to high workload, low situation awareness and poor communication and transparency between the agents. Results of ANOVAs showed that variations in likeability among *no trust*, one-*way* and *two-way trust* protocols were statistically significant (p<0.05 for each case), which statistically proved the differential effects of consideration of trust in collaborative assembly on human's likeability of the collaboration. However, variations in likeability were not statistically significant (p>0.05 for each case) between the subjects, which indicated the generality of the results.

Table 4 compares the team fluency among the three trust protocols, where less idle time, less non-activity time and less functional delay indicate higher team fluency. The results show that the *two-way trust* generated highest team fluency during the HRC. We believe that the highest level engagement and situation awareness and transparent communication

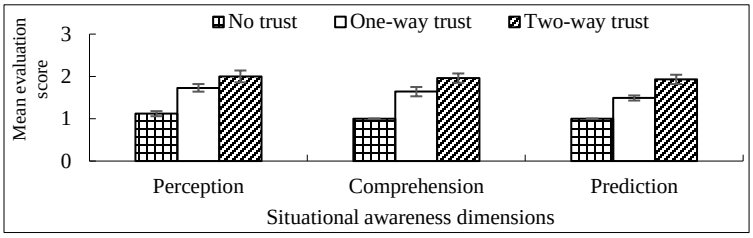


Fig.17 Mean (*n*=20) and standard deviations of situational awareness of subjects for *no trust*, *one-way trust* and *two-way trust* conditions for the integrated optimum subtask allocation.

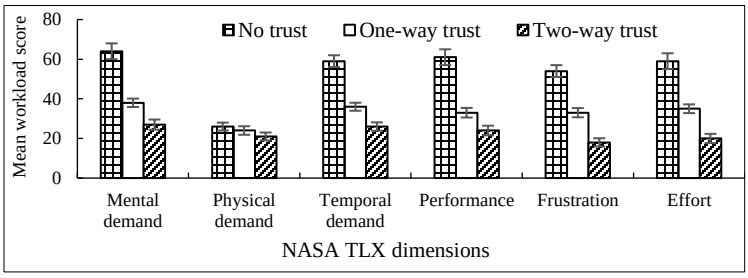


Fig.18 Mean (*n*=20) and standard deviations of cognitive workload for *no trust*, *one-way trust* and *two-way trust* conditions for the integrated optimum subtask allocation.

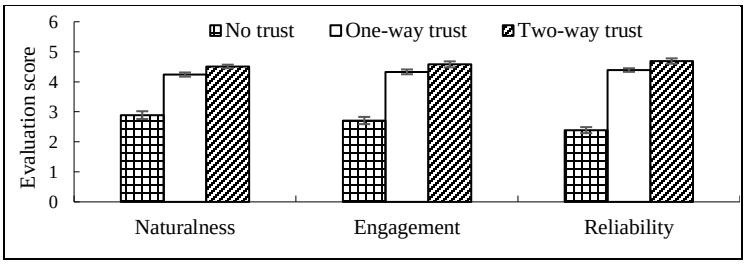


Fig.19 Mean (*n*=20) and standard deviations of likeability of humans in terms of naturalness, engagement and reliability for *no trust*, o*ne-way trust* and *two-way trust* conditions for the integrated (reallocated) optimum subtask allocation.

Table 4: Comparison of team fluency among the *no trust*, *one-way trust* and *two-way trust* protocols

No trust				*One-way trust*				*Two-way trust*			
Mean of % of total trial time (standard deviation)				Mean of % of total trial time (standard deviation)				Mean of % of total trial time (standard deviation)			
Human idle time	Robot idle time	Non-concurrent activity time	Functional delay	Human idle time	Robot idle time	Non-concurrent activity time	Functional delay	Human idle time	Robot idle time	Non-concurrent activity time	Functional delay
9.14 (0.22)	8.32 (0.29)	4.98 (0.11)	4.06 (0.12)	4.67 (0.17)	4.26 (0.14)	1.92 (0.06)	1.84 (0.09)	1.28 (0.07)	0.97 (0.05)	0.54 (0.002)	0.28 (0.002)

through bilateral trust and trust-based warning displays for the *two-way trust* resulted in the highest level synchronization and team fluency between the human and the robot for the *two-way trust* condition over *one-way trust* or *no trust* condition. Results of ANOVAs showed that variations in team fluency among *no trust*, one-*way* and *two-way trust* protocols were statistically significant (p<0.05 for each case), which proved the differential effects of consideration of trust in collaborative assembly on team fluency. However, variations in team fluency were not statistically significant (p>0.05 for each case) between the subjects, which indicated the generality of the results.

We expressed assembly productivity in term of assembly efficiency as (9), where T_t is the targeted time per complete assembly and T_m is the actual time per complete assembly.

$$Efficiency = \left(\frac{T_t}{T_m}\right) \times 100\% \tag{9}$$

We expressed assembly quality as (10), where n_{wo} is the mean total number of wrong orientation found in finished assembly products and n_a is the mean total finished assembly products in an experiment condition (e.g. *no trust*, *one-way*, *two-way trust*).

$$Quality = \left(1 - \frac{n_{wo}}{n_a}\right) \times 100\% \tag{10}$$

Table 5 compares the productivity and quality among no trust, one-way trust and two-way trust. T_t for the integrated allocation was 78s. For *two-way trust*, the mean T_m was 79.51s, but for *one-way* and *no trust*, the mean T_m was 82.23s and 91.96s respectively, which resulted in 98.10%, 94.86% and 84.82% efficiency for *two-way*, *one-way* and *no trust* respectively based on (9).The n_a (in round figure) was 19, 21 and 22 for *no trust*, *one-way* and *two-way trust* respectively. The corresponding n_{wo} (in round figure) was 4, 2 and 0 for *no trust*, *one-way* and *two-way trust* respectively, which resulted in 78.95%, 90.48% and 100% quality for *no trust*, *one-way* and *two-way trust* respectively based on (10). For *two-way trust*, human's wrong orientations of assembly parts (faults) were monitored through trust updates, and warning messages were issued based on trust status that made the human aware of wrong orientation. We believe that such awareness reduced the number of wrong orientations in the finished assembly, which increased quality in *two-way trust*. Then, reduction in wrong orientations reduced the amount of reworks that also increased the efficiency. However, such advantages were partial for *one-way trust* that produced lower productivity and quality than that produced for *two-way trust*. Again, these advantages were absent for *no trust* that reduced productivity and quality. We see that productivity and quality for *one-way trust* was also high though it was comparatively lower than that for *two-way trust*. Results of ANOVAs showed that variations in productivity and quality among *no trust*, one-*way* and *two-way trust* protocols were statistically significant (p<0.05 for each case), which statistically proved the differential effects of consideration of trust in collaborative assembly on productivity and quality. However, variations in productivity and quality were not statistically significant (p>0.05 for each case) between the subjects, which indicated the generality of the results.

Mean utilization ratios among *no trust*, *one-way* and *two-way trust* based on (8) are given in Table 6. The results show that the utilization ratio for *two-way trust* was better than that for *one-way* or *no trust*. We posit that human's higher performance and better synchronization with the robot through bilateral trust display in *two-way trust* might result in better utilization ratio [16], [21].

Table 5: Assembly productivity and quality for *no trust*, *one-way trust* and *two-way trust* conditions

Condition	Productivity (%)	Quality (%)
No trust	84.82	78.95
One-way trust	94.86	90.48
Two-way trust	98.10	100

Table 6: Comparison of mean utilization ratios among *no trust*, *one-way trust* and *two-way trust* conditions

Condition	Utilization ratio
No trust	0.842
One-way trust	0.936
Two-way trust	0.983

In general, the above results show that the *two-way trust* resulted in higher team fluency, situation awareness, productivity, quality, utilization ratio and likeability, and lower cognitive workload than that resulted in *one-way* and *no trust* conditions, which justify hypothesis 2. Nevertheless, the evaluation results seemed to be satisfactory for both *two-way* and *one-way trust* as the assemblies in both cases were conducted with optimum subtask allocation. The evaluation results in *no trust* were not satisfactory, but it could be worse if optimum allocation would not be used.

We report above results of feedback allocation or reallocation for only one case because it is a dynamic process based on Fig.10 and there may have unlimited opportunities depending on situations in term of mutual trust. The subtask reallocation (feedback allocation) presented in this paper shows an example of how and when subtasks need to be reallocated, and the subsequent reallocations that become necessary while the assembly continues for long period can follow similar procedures. In fact, the presented approaches and results just illustrate the main concepts, and these are not exhaustive.

8. Discussion

8.1 Feasibility, Effectiveness and Generality of the Proposed Approaches

Feasibility and potential effectiveness of the proposed strategy of mutual trust-based subtask allocation and reallocation (Fig.10) can be discussed in several ways. Let us discuss the feasibility of the proposed mutual trust-based method in terms of same task/subtask execution by both agents/actors at the same time. In such a case, the following limitations may be expected:

1. It may be difficult to separately identify and assess/measure the performance and fault criteria for both the human and the robot in real-time.
2. One agent's action can affect the performance and fault of another agent, and thus the measurements of performance and fault status may be contaminated.

3. Even if the performance and fault criteria for both the human and the robot can be identified and assessed separately, the measured values of both the human and the robot performance and fault status may be very similar, which may result in very similar computed trust in each other. Such similarity in mutual trust may apparently impact the execution and effectiveness of the proposed strategies and approaches in Fig.10.

However, these potential limitations can be overcome and the proposed approaches can be proved effective and feasible through the following efforts and understandings:

1. Even though the human and the robot collaboratively perform on the same task/subtask at the same time and their performance and fault criteria can be similar, the computed performance and fault status values may not be literally the same or similar. The trust of an agent in a trial or time step does not depend only on fault and performance status of that time step. It also depends on fault and performance status of the previous time step and on different constant values, which may be different between the human and the robot that may create differences in the mutual trust values. This may also happen due to differences in autonomy and capabilities between human and robot. For example, for a long work period, a human may experience fatigue that may impact human's performance and fault status. However, robot's performance and fault status can remain ideally unchanged irrespective of work duration. In contrast, robot's performance and fault status may change due to variations in task situations during the collaborative task due to the reason that the robot has limited adaptability. On the other hand, human can adjust sudden changes in task situations during the collaborative task between the robot and the human. Again, human and robot's actions may also affect each other. For example, human's shaking hand can affect both human and robot's precision and performance. Robot end-effector's force can affect human's precision and performance. These may also impact the measurement results of performance and fault status of both the human and the robot. All these can create differences between the computed trust values in each other.
2. In some cases, both the human and the robot may have same or similar performance and fault status, and thus the mutual trust can also be the same or very similar. This also does not make the strategies and approaches in Fig.10 ineffective if the thresholds and decision rules are adjusted.
3. In the collaborative task where same task/subtask is executed by both agents/actors at the same time, it may happen that the human and the robot have same or similar performance and fault status. However, in order to execute the strategy in Fig.10, it is not an issue whether or not the human and the robot have same or similar performance and fault status. The issue is whether or not we can assess/measure the human and the robot's performance and fault status in real-time. We believe that the collaborative nature of a task/subtask does not impede the measurement of performance and fault status (measurement results may be the same, but measurements are still possible). Instead, the feasibility of measurement of performance and fault status depends on the task/subtask itself irrespective of whether it is collaborative or independent. Hence, same subtask performed by both the human and the robot collaboratively at the same time cannot affect the feasibility of the proposed strategy of mutual trust-based task/subtask allocation and reallocation.

The proposed mutual trust model is general in the sense that, in all types of tasks performed by the human and the robot either separately or collaboratively, their individual performance and fault should be measurable separately. It does not mean that their performance and fault need to be measured through the same criteria. It also does not mean that their performance and fault status, and hence the computed trust values of both agents cannot be the same or similar. We believe that in collaborative task where both the human and the robot perform the same task, there is a chance that they can have same performance and fault status, but there is no chance that their performance and fault status cannot be determined. In fact, determination of performance and fault status of an agent does not depend on whether they work standalone or together. It is more likely that it depends on the task itself. In extreme case, if the performance and fault status of an agent cannot be measured for a particular collaborative task, then it can be an objective of further research. Hence, we posit that the presented approaches should be applicable to all types of HRC in assembly in manufacturing directly or indirectly without significant loss of generality. Here, we do not mean that the proposed measurement methods of performance and fault are applicable to all types of HRC in assembly in manufacturing. Instead, we mean that if we can measure the performance and fault status of an agent either performing separately or collaboratively with another agent for an assembly task in any way, then we can compute the trust. Here, generalization is made with the objective of performance and fault measurement, not with what criteria and method are used for such measurement.

8.2 Generality and Scalability of the Obtained Results

It is already proved that variations in the results among subjects were not statistically significant, which indicate generality of the results. However, generality can be further achieved in the following ways: (i) the hybrid cell should be developed in real industry environment using industry facilities, and different types of practical assembly tasks observed in factory floors should be considered for the proposed HRC, (ii) the proposed strategies should be verified with different robot platforms in different industries, and (iii) more human subjects including industry workers should be used in the experimental evaluation, etc. By incorporating more parameters in trust modeling, advancing real-time trust measurement methods and adjusting the robot hardware with assembly requirements, we can scale up the proposed concepts to large assembly operations with different product-mix. Even though some subjective approaches were used to evaluate the effectiveness of the proposed strategies, the evaluation results should be reliable because we used standard subjective evaluation methods and metrics such as Likert scale [29], NASA TLX [37], SAGAT [38], etc. However, there is scope to pursue stochastic modeling of mutual trust that can further enhance the reliability of the presented results.

8.3 Extrapolation of the Results to Real Industrial Applications

In real practices in industries, new hybrid cells need to be developed following the procedures illustrated in this article to shift the existing manual assemblies to HRC assemblies, and the trust measurement and display techniques need to be incorporated with the hybrid cells. Existing hybrid cells, if any, can be upgraded through incorporating trust measurement and display. Any type of suitable robots or multi-robots can be used in the hybrid cells. Additional external sensors (position sensors, vision systems) can be incorporated with the hybrid cells to facilitate trust measurements if the sensors of the robots seem to be unsuitable/inadequate. In all cases, subtask allocation should be optimized and the decision of subtask reallocation should be associated with trust status. The practitioners should have good knowledge about capabilities of the robot and requirements of the hybrid cells to assess the agents (human, robot) for feedforward optimization of subtask allocation. In cases of new types of assemblies, subtasks need to be determined newly and the assessment criteria and constraints for optimization of subtask allocation also need to be adjusted. The hybrid cells should be integrated with the entire manufacturing systems. For example, the assembly parts input to the robot can be manipulated to the robot just-in-time through belt conveyors and the finished assemblies can be dispatched through another conveyor. Robot and human speeds should be adjusted considering targeted production rate and human capability. Human co-workers can be trained before adding into hybrid cells. One supervisor can supervise multiple hybrid cells assembling different products. The trust thresholds can be decided based on specific requirements. The hybrid cell infrastructures can be modular and the cells can be mobile for quick transfer to assembly units for flexible usages.

9. Conclusions and Future works

We developed a human-robot hybrid cell for collaborative assembly of a few lightweight parts into a final product. We derived computational models of human's trust in robot and robot's trust in human and illustrated real-time (or near real-time) measurement of trusts. We proposed an integrated (feedforward + feedback) optimum subtask allocation scheme for the assembly triggered by trusts and the evaluation results showed the efficacy of the integrated scheme. The evaluation results showed that the *two-way trust* was more effective than the *one-way trust* or *no trust* in terms of HRI and assembly performance. Application of the proposed results can significantly improve productivity and quality in flexible assembly operations in manufacturing industries such as automotive, aerospace, electronics, rubber and plastics, etc.

In the future, we will investigate stochastic phenomena and non-linearity in trust models. We will associate the optimization of subtask allocation with that of subtask sequencing/precedence and scheduling. We will search for scalable optimal solution for the subtask allocation with optimal sequencing and scheduling. We will investigate optimization of subtask sequence and scheduling through changes in assembly product and assembly process design. We will further analyze the concepts of 'independent' and 'interdependent/collaborative' task/subtask execution for the proposed human-robot collaborative task. We will investigate how to reduce complexity in the proposed integrated especially in the feed-forward optimization method. We will present experimental evidence of effectiveness and applicability of the proposed mutual trust-based subtask allocation and reallocation approach for the scenario where both the human and the robot perform simultaneously on the same piece of task/subtask. We will also develop novel automatic controls to adjust the robot's speed based on mutual trust status to keep pace with the human.

Acknowledgement

This work was supported in part by the National Science Foundation under grant No. CMMI-1454139. The authors thank BMW Manufacturing Co., Spartanburg, SC, USA for supporting with the robot used in the study and the case study. The authors also thank the subjects who participated in the experiments.

References

[1] M. Baily, B. Bosworth, "US manufacturing: understanding its past and its potential future," *Journal of Economic Perspectives*, vol. 28, no. 1, pp. 3–26, 2015.

[2] A. Rane, D. Sudhakar, V. Sunnapwar, S. Rane, "Improving the performance of assembly line: Review with case study," *in Proc. of 2015 Int. Conf. on Nascent Technologies in the Engineering Field*, pp.1-14.

[3] R. Aziz, M. Rani, J. Rohani, A. Adeyemi, N. Omar,"Relationship between working postures and MSD in different body regions among electronics assembly workers in Malaysia," *in Proc. of IEEE Int. Conf. on Industrial Engineering and Engineering Management*, pp.512-516, 2013.

[4] R. Patel, M. Hedelind, P. Lozan, "Enabling robots in small-part assembly lines: the ROSETTA approach - an industrial perspective," *in Proc. of the 7th German Conference on Robotics*, pp.1-5, 2012.

[5] J. Fryman, B. Matthias, "Safety of industrial robots: from conventional to collaborative applications," *in Proc. of the 7th German Conference on Robotics*, pp.1-5, 2012.

[6] http://www.rethinkrobotics.com/products/baxter/ accessed on May 18, 2017.

[7] http://kinovarobotics.com/products/mico-robotics/ accessed on May 18, 2017

[8] KUKA Robot, www.kuka-robotics.com/ accessed on May 18, 2017.

[9] J. Tan, F. Duan, Y. Zhang, K. Watanabe, R. Kato, T. Arai, "Human-robot collaboration in cellular manufacturing: design and development," *in 2009 IEEE/RSJ* Int. Conf. on Intelligent Robots and Systems, pp. 29–34.

[10] R. Wilcox, S. Nikolaidis, J. Shah, "Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing," *in Proceedings of Robotics: Science and Systems conference (RSS)*, 2012.

[11] K. N. Kaipa, C. Morato, J. Liu, and S. K. Gupta, "Human-robot collaboration for bin-picking tasks to support low-volume assemblies," *in Proc. of RSS Workshop on Human-Robot Collaboration for Industrial Manufacturing* , July 2014.

[12] A. Sauppe, B. Mutlu, "Effective task training strategies for instructional robots," *in Proc. of Robotics: Science and Systems Conference*, July 2014.

[13] B. Gleeson, K. Maclean, A. Haddadi, E. Croft, J. Alcazar, "Gestures for industry intuitive human-robot communication from human observation," *in 2013 ACM/IEEE Int. Conf. on Human-Robot Interaction*, pp.349-356.

[14] A. M. Howard, "Role allocation in human-robot interaction schemes for mission scenario execution," *in Proc. of IEEE International Conference on Robotics and Automation*, pp.3588-3594, 15-19 May 2006.

[15] A. Clair, M. Mataric, "Role-based coordinating communication for effective human-robot task collaborations," *in Proc. of Int. Conf. on Collaboration Technologies and Systems*, pp.241-244, 20-24 May 2013.

[16] M. Majji, R. Rai, "Autonomous task assignment of multiple operators for human robot interaction," *in Proc. of American Control Conference*, pp.6454-6459, 2013.

[17] T. Zheng, X. Zhao, "Research on optimized multiple robots path planning and task allocation approach," *in Proc. of IEEE Int. Conf. on Robotics and Biomimetics*, pp.1408-1413, 17-20 Dec. 2006.

[18] F. Chen, K. Sekiyama, F. Cannella, T. Fukuda, "Optimal subtask allocation for human and robot collaboration within hybrid assembly system," *IEEE Transactions on Automation Science and Engineering*, vol.11, no.4, pp.1065-1075, Oct. 2014.

[19] B. M. Ghosh, M. G. Helander, "A systems approach to task allocation of human-robot interaction in manufacturing," *Journal of Manufacturing Systems*, vol. 5, no. 1, pp.41-49, 1986.

[20] H. Ding, M. Schipper, B. Matthias, "Optimized task distribution for industrial assembly in mixed human-robot environments - Case study on IO module assembly," *in Proc. of 2014 IEEE Int. Conf. on Automation Science and Engineering*, pp.19-24, 18-22 Aug. 2014.

[21] T. Fang, L. Parker, "A Complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation," *in Proc. of 2007 IEEE Int. Conf. on Robotics and Automation*, pp.3351-3358.

[22] G. Li, Y. Tamura, H. Asama, "Dynamical task allocation and reallocation based on body expansion behavior for multi-robot coordination system," *in Proc. of IEEE Int. Conf. on Mechatronics and Automation*,pp.537-542, 2011.

[23] K. A. Hoff and M. Bashir, "Trust in automation: integrating empirical evidence on factors that influence trust," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Sept 2014.

[24] J. Lee, N. Moray, "Trust, self-confidence, and operators' adaptation to automation," *Int. Journal of Human-Computer Studies*, vol. 40, pp. 153-184, 1994.

[25] P. Kaniarasu, A. Steinfeld, "Effects of blame on trust in human robot interaction," *in Proc. of IEEE International Symposium on Robot and Human Interactive Communication*, pp.850-855, 25-29 Aug. 2014.

[26] S. Rahman, B. Sadrfaridpour, Y. Wang, "Trust-based optimal subtask allocation and model predictive control for human-robot collaborative assembly in manufacturing," *in Proc. of ASME Dynamic Systems and Control Conference*, Oct. 28-30 2015, Columbus, Ohio, USA.

[27] S. Lewandowsky, and M. Mundy, "The dynamics of trust: Comparing humans to automation," *Journal of Experimental Psychology: Applied*, vol. 6, pp. 104-123, 2000.

[28] S. D. Ramchurn, C. Mezzetti, A. Giovannucci, J. A. Rodriguez-Aguilar, R. K. Dash and N. R. Jennings, "Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty," *Journal of Artificial Intelligence Research*, vol. 35, pp.119-159, 2009.

[29] J. Carifio, J. P. Rocco, "Ten common misunderstandings, misconceptions, persistent myths and urban legends about Likert scales and Likert response formats and their antidotes," *Journal of Social Sciences*, vol. 3, no.3, pp. 106-116, 2007.

[30] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Chen, E. Visser, R. Parasuraman, "A meta-analysis of factors affecting trust in human-robot interaction," *Human Factors*, vol. 53, no. 5, pp. 517–527, 2011.

[31] J. Lee, N. Moray, "Trust, control strategies and allocation of function in human-machine systems," *Ergonomics*, vol. 35, pp. 1243-1270, 1992.

[32] B. Sadrfaridpour, H. Saeidi, J. Burke, K. Madatahil and Y. Wang, "Modeling and control of trust in human-robot collaborative manufacturing," *Robust Intelligence (RI) and Trust in Autonomous Systems*, Book Chapter, pp 115-141, Springer, 2016.

[33] F. Xu, S. Wang, B. Li, "Industrial robot base assembly based on improved Hough transform of circle detection algorithm," *in Proc. of 2014 World Congress on Intelligent Control and Automation*, pp.2446-2450.

[34] D. Lowe, "Object recognition from local scale-invariant features," *in Proc. of IEEE Int. Conf. on Computer Vision*, vol.2, pp.1150-1157, 1999.

[35] M. Muja and D. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, 2014.

[36] S. Rahman, "Evaluating and benchmarking the interactions between a humanoid robot and a virtual human for a real-world social task," *Communications in Computer and Information Science*, Springer, vol. 409, pp. 184–197, 2013.

[37] S. Hart and L.Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," *Advances in Psychology*, vol. 52, pp. 139-183, 1988.
[38] S. Cooper, J. Porter, L. Peach, "Measuring situation awareness in emergency settings: a systematic review of tools and outcomes," *Open Access Emergency Medicine*, vol. 6, pp. 1–7, 2014.
[39] C. Sidner, C. Lee, C. Kidd, N. Lesh, "Explorations in engagement for humans and robots," *Artificial Intelligence*, vol. 166, pp. 140–164, 2005.
[40] G. Hoffman, "Evaluating fluency in human-robot collaboration," *in Proc. of Robotics: Science and Systems Workshop on Human-Robot Collaboration*, 2013.
[41] P. Basili, M. Huber, T. Brandt, S. Hirche, and S. Glasauer, "Investigating human-human approach and handover," *Human Centered Robot Systems-Cognition, Interaction, Technology*, vol. 6, pp 151-160, Springer, 2009.
[42] Y. Li, Y. Liu, X. Liu, Z. Peng, "Parameters identification and vibration control for modular manipulators," *in Proc. of IEEE Int. Conference on Robotics and Automation*, vol. 3, pp.3254-3259, 2003.
[43] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41-48, Jan. 2017.
[44] D. Bertsimas, A. Takeda, "Optimizing over coherent risk measures and non-convexities: a robust mixed integer optimization approach," *Computational Optimization and Applications*, vol. pp.62, 613–639, 2015.
[45] D. Bergman, A. A. Ciré, W.-J. van Hoeve and J. N. Hooker, "Discrete optimization with decision diagrams," *INFORMS Journal on Computing*, vol. 28, 47-66, 2016.
[46] S. Martello, "Jeno Egerváry: from the origins of the Hungarian algorithm to satellite communication," *Central European Journal of Operations Research*, vol. 18, pp.47–58, 2010.
[47] L. Liu, D. Shell, "Optimal market-based multi-robot task allocation via strategic pricing," *in Proc. of Robotics: Science and Systems*, Berlin, Germany, June 2013.
[48] M. Gombolay, R. J. Wilcox, and J. A. Shah, "Fast scheduling of multi-robot teams with temporospatial constraints," *in Proc.* of *Robotics: Science and Systems*, Berlin, Germany, June 2013.
[49] E. Nunes and M. Gini, "Multi-robot task allocation with spatial, temporal, and ordering constraints: a review," *in New Research Frontiers for Intelligent Autonomous Systems at IAS*, 2014.
[50] D. McKnight, L. Cummings, N. Chervany, "Initial trust formation in new organizational relationships," *Academy of Management Review*, vol. 23, no.3, pp.473-490, 1998.
[51] L. Robert, A. Dennis, and Y. Hung, "Individual swift trust and knowledge-based trust in face-to-face and virtual team members," *Journal of Management Information Systems*, vol. 26, no.2, pp.241-279, 2009.
[52] https://www.iso.org/standard/62996.html Accessed on March 13, 2018.