A.H.C. Ng et al. (Eds.)

doi:10.3233/ATDE220131

Robotic Window Assembly – A Simulation Study and a Proposed Self-Adaptive Software Architecture

Osama A Q Ziada [a,1], Tobias Schauerte [a], Jetro Kenneth Pocorni [a], Hatem Algabroun [a], Gunnar Bolmsjö [a] and Lars Håkansson [a]
[a] *Linnaeus University, Department of Mechanical Engineering, Växjö, Sweden*

Abstract. The key driving factors in using humans and robots in collaborative applications for assembly processes are to reduce assembly time, cost and to improve the human working environment from an ergonomic viewpoint. Currently, there are limited automated procedures in assembly operations in house construction because the traditional type of assembly process depends entirely on manpower. This is common in the assembly process in different industries since assembly is one of the most demanding and intense manufacturing processes, and it is difficult to automate. This paper presents a case study on the implementation of human-robot collaboration for window assembly by way of an offline robot programming simulation. A self-adaptive software architecture that runs on a real-time target machine is also proposed for robotic window assembly. The window assembly method that will be used in this study is called "Click-In" and is manufactured by Fixture System Sweden AB. Apart from robot simulations, detailed suggestions are given for building a pilot cell for robot window assembly. The case study presented in this paper has both economical and ergonomic goals. The economic goal is to reduce the assembly time which will lead to an increase in window production. By introducing human-robot collaboration, operators do not need to perform uncomfortable assembly operations—rather the robot will perform these un-ergonomic operations. The feasibility of both goals is verified with offline robot programming simulation.

Keywords. Robot assembly, Self-adaptive software architecture, Human-robot collaboration, robot safety, real-time simulation

1. Introduction

In the Swedish industry for prefabricated wooden single-family houses (WSFH), mounting windows has been found to be a time-intensive activity that creates bottlenecks [1]. There is no common way to mount windows either. Some companies mount the windows upright directly into the wall, as one of the last steps in the production process (see Figure 1(a)), and some have a separate working station for windows parallel to the main production line. In the latter case, window modules are pre-manufactured (see Figure 1(b)) and transported to the first station in the production line, the so called framing station, where the window module is placed in the wall together with the remaining studs the wall's frame (see Figure 1(c)).

[1] Corresponding Author. [E-Mail address: osama.ziada@lnu.se]

(a) (b) (c)

Figure 1: Different window mounting methods: (a) upright assembly method, (b) working station for pre-assembly of window modules, (c) window module is placed in the wall.

As companies in the WSFH industry have different production processes and layouts, a general solution on where to mount windows could not be proposed; however, [1] suggested that product development activities, the application of computer integrated manufacturing tools and computer controlled machineries, i.e. automation, could be a way towards a more balanced production flow that might reduce bottlenecks.

In different countries, different ways of attaching windows to the adjacent studs exist. A new method, which currently was introduced to the Swedish market is the so called "Click-In" method. It can be used for mounting windows both on-site and off-site; yet, as the task at hand consists of labour-intensive and heavy operations [2], problems might occur in finding fully automated solutions. Additionally, the industry is still known for lagging behind other industries in terms of digitalization and production equipment [3], which might add an extra layer to the challenge of investing in automation. Even though these problems eventually could be overcome, a first step towards a more automated mounting of windows might include humans-in-the-loop [4, 5] and thus, the human-robot collaboration (HRC) perspective is applied in the current study.

There is a growing trend regarding publications and industrial use of HRC in industry and assembly applications [6, 7]. Multiple studies emphasize that in HRC should improve the performance with respect to different aspects [7–9]. These aspects are explained as follows. According to Hentout [7], collaborative robots can optimize the quality of task execution, ensure safety and be ergonomic for the human user. Other relevant work focused on the categorization criteria for the HRC, for example, Kumareet al. [9] categorized evaluation criteria found in the literature into 1) objective criteria, which can be subcategorized under productivity, safety and performance, 2) subjective criteria contains sets of interviews and observation and questionnaires. Castro, Silva and Santos [8] surveyed metrics related to different aspects that can be used to measure the task performance for HRC. The empirical results of different studies in industrial HRC showed improvements in industrial performance from different aspects [6], such as reducing completion time and minimizing error, better understanding of the operator space and higher precision of workpiece manipulation. Nevertheless, it is apparent from studies that the aspect of operational efficiency, i.e., higher production rate and low unit production cost, is the main motivation of having HRC in industrial production [10, 11].

Several researchers developed different approaches for the automation of assembly processes using HRC. For instance, Malik and Bilberg [10] developed a structured methodology to determine the automation potential of every task in a manual assembly operation based on physical properties of the component, task characterization, and collaborative work space. Then assignment of the tasks to the robot or the human is through a set of assembly attributes to balance the workload. The method is then implemented in a case study of HRC assembly process to validate the developed framework. Ranz et al., [12] proposed a morphological analysis approach to come out with relevant aspects, current technology, and conceptual options to serve as a description model for supporting design and a tool to be used in the comparison of applications. The proposed analysis approach has been validated through its application on various existing industrial HRC applications, research demonstrators and interviews of experts from academia. Scimmi et al., [13] described an HRC layout for assembly task and tested it in a case where the robot and operator shared the one workspace at the same time at real time. The robot performs pick-and-place tasks to provide an operator the parts to be assembled. A real time collision avoidance algorithm is implemented to avoid collision with the human operator. The layout setup was tested, and the performance was evaluated in terms of safety and productivity. The collision avoidance algorithm could avoid collisions and the productivity was improved by saving up to 18% of cycle time in the case of highest form of collaboration. More detailed surveys in this domain can be found in Matheson et al., [6] and Hentout et al., [7] and the more recent study Castro, Silva and Santos [8].

There is, however, a lack of studies that examine the automated HRC mounting of windows. Therefore, this study aims to contribute to the field by proposing an HRC mounting of windows, using the "Click-In" method. This is done by: 1) offline programming using Yaskawa's Motosim simulation software and 2) proposing a setup using a software architecture for "Click-In" based HRC.

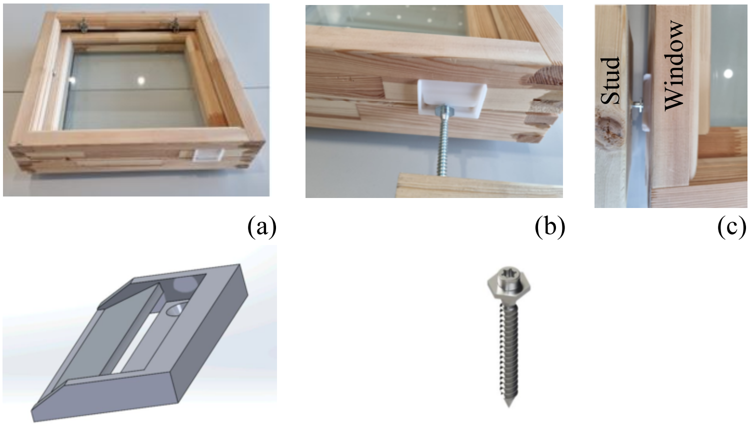
Next section presents the "Click-In" method, section 3 introduce in which way the offline robot simulation is presented. section 4, the setup for the robotic window assembly, as well as a self-adaptive software architecture are proposed will be discussed and conclusion in section 5.

2. Case Study

There is limited research and literature available on collaborative robot automation for house fabrication, specifically for window assembly. Human collaboration in automation is relevant to Industry 5.0, which is a human-centric vision of industry that complements the existing Industry 4.0 approach [4]. The proposed automation solution allows for an improvement of the assembly process, without modifying the window design or factory production layout. The case study is based on a mock-up of a small size window with dimensions of 480 mm x 480 mm, (see Figure 2). Robot window assembly is relevant because it aids in automation of similar fabrication steps, such as door assembly, and eventually full automation of house fabrication.

2.1. Components of the "Click-In" system

The components of the "Click-In" system consist of a plastic fixture and a screw—all shown in Figure 2. The window installation with the "Click-In" method requires tools, such as an electrical drill to fasten the screw, a measuring tool to measure the window size, and adjusting tools to adjust the screw for correct the screws alignment.



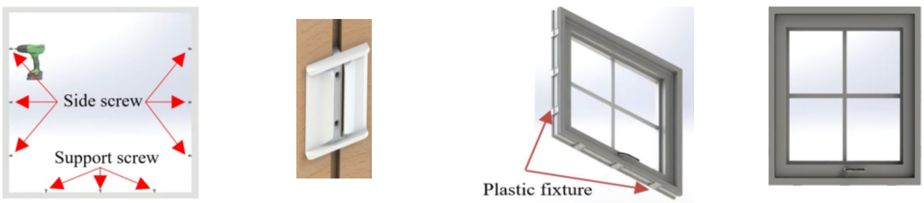
(a) (b) (c)

(d) (e)

Figure 2: (a) Mock-up window, (b) “Click-In” system with plastic fixture – before assembly, (c) “Click-In” system with plastic fixture – after assembly, (d) drawing of plastic fixture, (e) screw.

2.2. Steps in the window assembly process

The “Click-In” method can be divided into three major steps, i.e., 1) driving screws into the studs, (see Figure 3(a)). The challenge here is that the screws should be driven up to a certain depth into the frame at specific locations; 2) adding plastic fixtures to the window itself (see Figure 3(b) - (c)). The number of plastic fixtures must be equal the number of screws. The plastic fixtures are fastened at corresponding positions as the screws 3) Placing the window with plastic fixtures inside the window opening with fastened screws (see Figure 3(d)). The plastic fixtures are placed on the screws which are fastened in the window -opening – this action is accompanied by a clicking sound, hence the name “Click-In”. Table 1 shows a detailed list with steps in the window assembly process for the existing “Click-In” methods (the steps in Table 1 are preformed solely by humans).



(a) (b) (c) (d)

Figure 3:Steps in the window assembly process; (a) placement of screws in the studs, (b)-(c) plastic fixtures, (d) window inserted in its frame.

Table 1: Detailed list with steps in the window assembly process with the "Click-In" method.

#	Operation details
1	Install the support screws (3 screws in the bottom of the window opening)
2	Level the support screws
3	Install the side screws (6 screws on the right/left sides in the window frame)
4	Level the screws on left side of the opening
5	Measure the window with the measuring tape
6	Adjust the screws on the right side of the opening, according to the measurement of the window
7	Attached the "Click-In" plastic fixture to the window (3 pieces on the bottom + 6 pieces at the sides)
8	Pick up the window
9	Place the window in the window opening

3. Offline robot simulation

3.1. Simulation setup

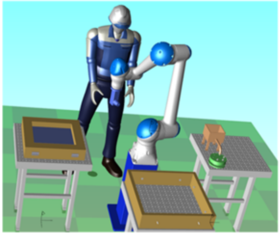
In this section, the steps are described which are needed to carry out the robot simulations in the study. The following is a brief explanation of the simulation of the mock-up assembly process.

The window assembly process consists of several steps with several parts, such as operations for driving screws into the window frame and the window itself, and assembly of these parts. The window assembly process consists of steps shown Figure 4. Offline robot programming is done in the software Yaskawa MotoSim. In the simulation, four steps are distinguished, i.e., robot picks the screws, robot fastens the screw, tool change by an operator (human-in-the-loop), and robot pick-and-place of the window.

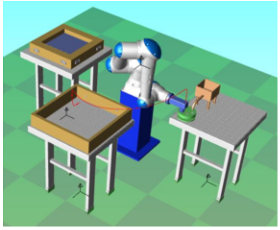
A video of the simulation can be seen in the following link: robotSimulation.

3.2. HRC window asssembly Time

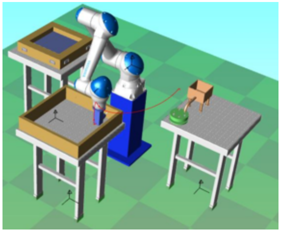
To determine the feasibility of robotic automation in window assembly, offline robot programming is used [14]. With MotoSim's monitoring function called "Trace Manager" (see Figure 5(a)), the time duration from start to end of the simulation is recorded. For tool changing and screw fastening, time measurement experiments were done. The time duration needed for tool changing was found to be approximately 30 seconds (see Figure 5(b)) and the time duration for screw fastening was measured to be 10 seconds. The tool change and screw fastening time durations was added to the simulation in the form of "Timer" functions. The total time of the window assembly process was found to be approximately 140 seconds which equals 2.3 minutes. Figure 5(a) shows the time of the window assembly process and Figure 5(b) part of the INFORM programming code (INFORM is the programming language used in Yaskawa robots).



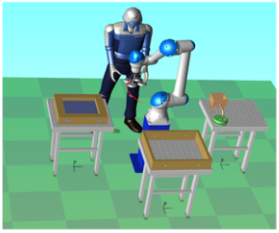
Step1 – Human attaches drilling tools



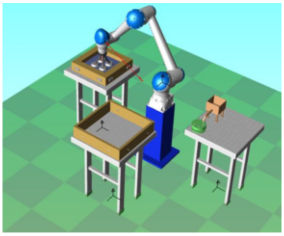
Step 2 - Picking up screws



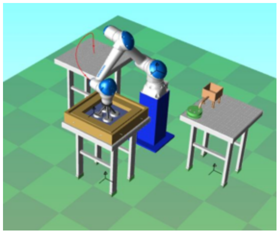
Step 3 - Fastening the screw



Step 4 – Changing tools to vacuum



Step 5 - Pick up the window



Step 6 - Place the window on the frame

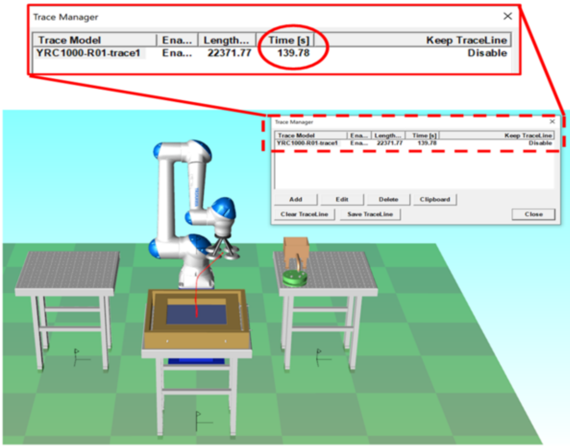
Figure 4:MotoSim simulation steps

4. Proposed implementation setup of a robotic window assembly cell

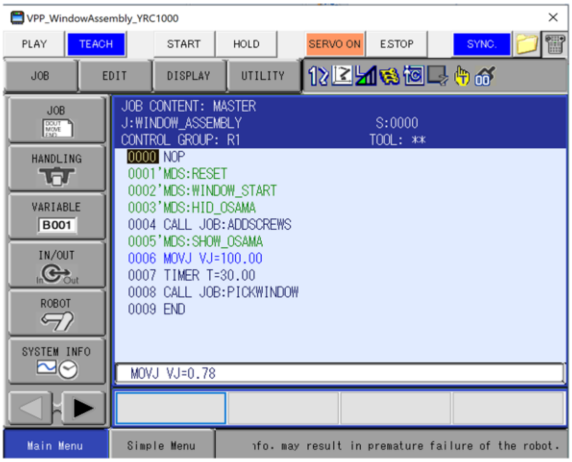
In this section detailed suggestions are given for the hardware needed to implement a pilot setup of a robotic window assembly cell as well as for the system architecture for such setup.

4.1. Setup of a robotic window assembly cell

Ergonomics is defined by [15] as a design approach for equipment, processes and corresponding environments. Ergonomics ensures that the working activities of the operator are performed in a safe environment and within acceptable limitations, enabling the operator to perform his tasks to the best of capabilities. The introduction of robots will improve ergonomics issues that may happen to the operators because operators do not need to lift heavy materials (preventing back injuries) or need to apply high torque for fastening screws (preventing hand and wrist injuries) [16,17]. In order to apply automation of the assembly process, and to make it highly effective to achieve the required goals, different devices and sensors related to the assembly process should be taken into consideration (see Figure 6).



(a)



(b)

Figure 5: Assembly process time and coding; (a) Time duration of the simulation, (b) Yaskawa robot programming language, called INFORM, within the Motosim interface.

The Yaskawa Motoman HC10 robot will be used for the window assembly cell, (see Figure 6(a)). This six-axis collaborative robot enables operation with human operators. Precision hand-guiding facilitates easy programming for fast implementation of the robot system. The operator's safety is assured by power and force limit technology that stops the robot in case of contact with an operator. This collaborative robot can operate without additional protective measures like a safety fence, depending on the risk assessment. Instead of fences around the robot cell, safety sensors will be used in conjunction with the Yaskawa HC10 collaborative robot. Safety sensors are used to detect human motions, i.e., when a human approaches the area of the robot, the robot will reduce its speed. The robot operation can even be paused if the operator comes too close to the robot itself – all to ensure the safety of the operator. As the operator withdraws from the vicinity of the robot, the robot increases its operation speed again. For the window assembly operation, a SICK safety sensor will be used, (see Figure 6(b)).

(a) (b) (c)

(d) (e) (f)

Figure 6: Hardware and peripheral equipment for the robotic window assembly cell: (a) Yaskawa Motoman collaborative robot, (b) SICK sensor, (c) Stöger automatic screwdriver RSX, (d) Vacuum gripper, (e) Photoneo vision sensor (PhoXi 3D Scanner), (f) Specim hyperspectral camera

SICK safety sensors can be programmed such that different safety zones are defined, with each zone corresponding to a certain speed reduction.

The automatic screwdriver for Human-Robot-Interaction is from the manufacturer Stöger RSX (see Figure 6(c)). A vacuum gripper and an automatic screwdriver will be used as robot tools. The vacuum gripper is self-designed and consists out of 6 vacuum suction cups, (see Figure 6(d)). In the pilot cell, a clamping system for holding the frame in its place is not considered. Instead, it is assumed that the window frame is fastened to the table (see Figure 4) with F-clamps. Operators are used for the tool exchanger operation, (see Figure 4(a)) 3D vision camera (manufacturer: Photoneo, model: PhoXi 3D Scanner, (see Figure 6(e)) is used to determine the orientation of the window opening on the table in order for the screwing operation to be done correctly. If the window is tilted on the table, the programmed robot positions for the screwing operations will also be adapted relative to the window opening orientation. Image processing of the camera data is done on a real-time target machine (manufacturer: SpeedGoat). The target machine has MATLAB Simulink as a programming language and can run different MATLAB toolboxes, such as image processing, and is also able to perform real-time simulations.

The torque needed for the screwing operations performed by the robot is influenced by knots in the wooden frame (see Figure 7). Knots are considered as defects and irregularities according to Briggert et al.[16] and Fathi et al.[17] Qu et al.[18] explain that knots can make wood processing more difficult due to their increased hardness. To detect knots on the wooden frame surface, a hyperspectral camera (see Figure 7) is used. The hyperspectral camera (manufacturer: Specim, model: FX10e) measures the light intensity as a function of wavelength. With hyperspectral imaging knots are detected and distinguished by their characteristic reflectance spectrum from the wooden surface. Whenever a knot is detected, the robot will either avoid the region for the screwing operation or the robot will use a higher torque.

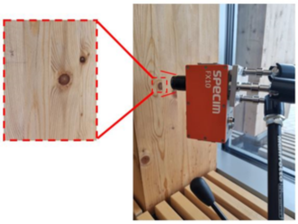


Figure 7: Hyperspectral camera detecting the knot

4.2. Self-adaptive software architecture

In order to develop such a setup, several elements should be taken into account including – and not limited to – robot system, human, sensors, software and applications. Research has pointed out that a software architectural approach endows a level of abstraction that facilitates the development of such a system [19]. As such, in this paper, we propose IBM's software architecture MAPE-K (monitor, analyze, plan, execute, knowledge i.e., data repository) [20] to be employed as a reference framework for this setup. MAPE-K architecture is considered among other alternativs such as [21,22]as it has the necessary generic components (i.e., monitor, analyze, plan, execute-knowledge) to properly operate this setup. The real-time target machine (SpeedGoat) will be the computational platform that hosts the controlling elements of this setup.

Typically, the self-adaptive approach can be represented as a managed system that is controlled by the managing system using a feedback loop [19, 23](See Figure 8). The feedback loop follows the MAPE-K reference model which consists of four main components: monitor, analyze, plan, execute-knowledge. Monitor monitors the managed system and updates the knowledge which is a data repository that stores models, parameters and other information of the managed system. Analyse analyses the updates from Monitor and determine if an adaptation action is required by the managed system. If so, Plan is triggered to select a plan that allows the managed system to fulfill its goals. The plan then is executed by the Execute component. This control process is to be deployed in a real-time computational platform (i.e., SpeedGoat) that fulfils the computational requirements.

5. Conclusions

In this paper the feasibility of introducing human-robot collaboration in window assembly is discussed by using offline robot simulation. In this study both economical and ergonomic goals have been considered. The following conclusions are drawn in the study:

- Offline robot programming simulations show that the robot window assembly time equals approximately 2 minutes
- Automation of window assembly still requires some form of humans-in-the-loop
- Offline robot programming simulation has been used as a visualization tool (For economic and ergonomic aspects) before investments in hardware are made)

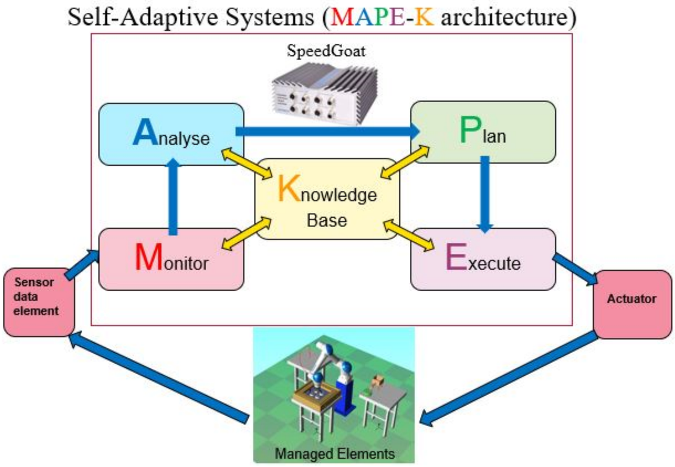


Figure 8: Self-adaptive software architecture

- MAPE-K has been proposed as a software architecture for managing the robot window assembly process
- Real-time target machines are proposed as platforms for running self-adaptive software and image processing algorithms in assembly operations

Acknowledgement

The support of this work by Tillväxtverket (The Swedish Agency for Economic and Regional Growth), Region Kronoberg and Linnaeus University within the Smart-IAT project is gratefully acknowledged. The authors thank Fixture Systems AB for the provided Click In system and window equipment, Yaskawa Nordic AB for fruitful discussions and their valuable contributions, and master student Martin Håkansson Burelius for help with offline robot programming simulations.

References

[1] I. Convention *et al.*, "Balancing the production flow in prefabrication of wooden houses In : Tools for sustainability Forest Products Society N . B . When citing this work , cite the original published paper . Permanent link to this version : Balancing the production flow in pr," 2018.

[2] Daniel Pooyan &Yousif Yousif, "Effektivisering av produktionsflöde hos ett trähustillverkande företag Streamlining of the production flow at a wooden house manufacturing company."

[3] T. Vestin, "Productivity of Firms in the Swedish Industry for Wooden Single-Family Houses," pp. 359–367, 2019.

[4] L. De Nul, M. Breque, and A. Petridis, *Industry 5.0 - Towards a sustainable, human-centric and resilient European industry*. 2021.

[5] C. J. Turner, R. Ma, J. Chen, and J. Oyekan, "Human in the Loop: Industry 4.0 Technologies and Scenarios for Worker Mediation of Automated Manufacturing," *IEEE Access*, vol. 9, pp. 103950–103966, 2021, doi: 10.1109/ACCESS.2021.3099311.

[6] E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio, and G. Rosati, "Human–Robot Collaboration in Manufacturing Applications: A Review," *Robot. 2019, Vol. 8, Page 100*, vol. 8, no. 4, p. 100, Dec. 2019, doi: 10.3390/ROBOTICS8040100.

[7] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017," *Adv. Robot.*, vol. 33, no. 15–16, pp. 764–799, Aug. 2019, doi: 10.1080/01691864.2019.1636714.

[8] A. Castro, F. Silva, and V. Santos, "Trends of Human-Robot Collaboration in Industry Contexts: Handover, Learning, and Metrics," *Sensors*, vol. 21, no. 12, p. 4113, Jun. 2021, doi: 10.3390/s21124113.

[9] S. Kumar, C. Savur, and F. Sahin, "Survey of Human-Robot Collaboration in Industrial Settings: Awareness, Intelligence, and Compliance," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 1, pp. 280–297, 2021, doi: 10.1109/TSMC.2020.3041231.

[10] A. A. Malik and A. Bilberg, "Collaborative robots in assembly: A practical approach for tasks distribution," *Procedia CIRP*, vol. 81, pp. 665–670, 2019, doi: 10.1016/j.procir.2019.03.173.

[11] F. Ranz, T. Komenda, G. Reisinger, P. Hold, V. Hummel, and S. Wilfried, "A Morphology of Human Robot Collaboration Systems for Industrial Assembl," in *51st CIRP Conference on Manufacturing Systems*, 2018, pp. 99–104, doi: 10.1016/j.procir.2018.03.011.

[12] F. Ranz, T. Komenda, G. Reisinger, P. Hold, V. Hummel, and W. Sihn, "A Morphology of Human Robot Collaboration Systems for Industrial Assembly," *Procedia CIRP*, vol. 72, pp. 99–104, 2018, doi: 10.1016/j.procir.2018.03.011.

[13] L. S. Scimmi, M. Melchiorre, M. Troise, S. Mauro, and S. Pastorelli, "A practical and effective layout for a safe human-robot collaborative assembly task," *Appl. Sci.*, vol. 11, no. 4, pp. 1–14, 2021, doi: 10.3390/app11041763.

[14] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robot. Comput. Integr. Manuf.*, vol. 28, no. 2, pp. 87–94, Apr. 2012, doi: 10.1016/j.rcim.2011.08.004.

[15] B. McPhee, "Ergonomics in mining," *Occup. Med. (Chic. Ill).*, vol. 54, no. 5, pp. 297–303, Aug. 2004, doi: 10.1093/occmed/kqh071.

[16] A. Briggert, A. Olsson, and J. Oscarsson, "Prediction of tensile strength of sawn timber: definitions and performance of indicating properties based on surface laser scanning and dynamic excitation," *Mater. Struct.*, vol. 53, no. 3, p. 54, Jun. 2020, doi: 10.1617/s11527-020-01460-5.

[17] H. Fathi, V. Nasir, and S. Kazemirad, "Prediction of the mechanical properties of wood using guided wave propagation and machine learning," *Constr. Build. Mater.*, vol. 262, p. 120848, Nov. 2020, doi: 10.1016/j.conbuildmat.2020.120848.

[18] H. Qu, M. Chen, Y. Hu, and J. Lyu, "Effect of Trees Knot Defects on Wood Quality: A Review," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 738, no. 1, p. 012027, Jan. 2020, doi: 10.1088/1757-899X/738/1/012027.

[19] H. Algabroun, M. U. Iftikhar, B. Al-najjar, and D. Weyns, "Maintenance 4 . 0 Framework Using Self- Adaptive Software Architecture," *Proc. 2nd Int. Conf. Maint. Eng. IncoME-II 2017 Univ. Manchester, UK*, no. November 2018, pp. 1–11, 2017, [Online]. Available: https://www.researchgate.net/publication/328841192.

[20] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer (Long. Beach. Calif).*, vol. 36, no. 1, pp. 41–50, 2003, doi: 10.1109/MC.2003.1160055.

[21] D. Garlan, S. Cheng, A. Huang, B. Schmerl, and P. Steenkiste, "Rainbow : Architecture- Based Self-Adaptation with Reusable," pp. 46–54, 2004.

[22] P. Oreizy, N. Medvidovic, and R. N. Taylor, "Architecture-based runtime software evolution," *Proc. - Int. Conf. Softw. Eng.*, pp. 177–186, 1998, doi: 10.1109/icse.1998.671114.

[23] M. U. Iftikhar and D. Weyns, "ActivFORMS: Active Formal Models for Self-adaptation," *Proc. 9th Int. Symp. Softw. Eng. Adapt. Self-Managing Syst.*, pp. 125–134, 2014, doi: 10.1145/2593929.2593944.