



Simulation-based validation of process monitoring tasks in assembly

Clemens Gonnermann^{1,2} · Sebastian Kurscheid¹ · Benedikt Schmucker¹ · Rüdiger Daub^{1,2}

Received: 12 November 2023 / Accepted: 22 January 2024
© The Author(s) 2024

Abstract

The importance of process monitoring has increased significantly in today's manufacturing landscape due to several factors. These factors include stricter quality standards, the presence of complex and diverse processes for customized products, and a focus on sustainability in energy and material consumption. In order to seamlessly integrate process monitoring into the production planning process, it is essential to automate the monitoring process planning while also taking into account production planning. Given the increasing importance of high-quality processes and products, assembly processes have become critical, as errors in assembly can lead to expensive production costs. However, generating and validating different process monitoring alternatives in assembly planning can be challenging because different processes and resources exist for different assembly plans. This research aims to solve this problem by providing a method to validate process monitoring alternatives more efficiently, thereby reducing the amount of manual effort and expert knowledge. This goal is achieved through automated validation using two simulation approaches: Collision Testing and Ray Casting. The results of this study are presented to demonstrate the effectiveness of the proposed solution.

Keywords Ray cast · Multi-body · Collision-free · Automation · Monitoring planning

1 Introduction

In today's competitive manufacturing environment, it is critical to meet the specific requirements of individual customers and deliver high-quality customized products. To achieve this goal, a highly flexible production system must be implemented that allows for various scheduling options, increasing complexity and interactions between employees, products, and production processes [1]. Reconfigurable

production systems based on concepts of digitalization and cyber-physical systems provide detailed insights into the capabilities and functionalities of individual resources [2]. When properly described semantically, they enable their flexible use [3]. However, planning and implementing these systems requires careful attention, resulting in a significant process planning effort.

Production systems and processes are subject to constant change, which leads to a higher error rate and a greater probability of failure. This requires increased integration of monitoring processes, an aspect of diagnosability in production systems. As the flexibility of cyber-physical production systems increases, so does the complexity of planning and the associated effort and expertise required, leading to uncertainties that must be monitored and detected [4]. Unfortunately, little research has focused on the diagnosability aspect of flexible and reconfigurable production systems [2].

Process monitoring and inspection planning belong to quality control [5]. In this publication, we consider process monitoring an integral part of quality control [6], which involves comparing characteristic values with their expected values [7]. Quality control is a sub-process of assembly [8]. Within the assembly [9], processes can be classified as either value-added or non-value-added, with

✉ Clemens Gonnermann
Clemens.Gonnermann@igcv.fraunhofer.de

Sebastian Kurscheid
Sebastian.Kurscheid@iwb.tum.de

Benedikt Schmucker
Benedikt.Schmucker@iwb.tum.de

Rüdiger Daub
Ruediger.Daub@igcv.fraunhofer.de

¹ Institute for Machine Tools and Industrial Management, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Bavaria, Germany

² Fraunhofer Institute for Casting, Composite and Processing Technology IGCV, Am Technologiezentrum 10, 86159 Augsburg, Bavaria, Germany

monitoring processes falling into the latter category but executed in parallel with value-added processes.

Therefore, it is essential to integrate process monitoring into the production planning process with minimal planning and execution effort. The complexity results from considering new product variants, production processes, multiple production plans, and different quality requirements (e.g., customers and regulations). Figure 1 provides an overview of the sequential steps involved in general process planning, with a particular emphasis on the step being discussed in this publication.

This publication presents a system for reducing the manual effort and necessary expert knowledge in validating and prioritizing individual monitoring processes and plans automatically. The system uses one approach based on multi-body simulations and another approach based on ray casting. The publication is structured as follows. Section 2 provides a comprehensive review of the literature on automated process planning, resource allocation using skill-based approaches, and simulations relevant to process planning validation. In addition, the basic principles of task planning are examined.

Section 3 then briefly concludes on the concept and implementation of the system and identifies the need for further action. Section 4 outlines the overall system for process monitoring task validation, including the required inputs, outputs, and data models for both approaches. The results of these approaches are presented in Sect. 5, with a use case illustrating the differences between the two approaches. The subsequent discussion can be found in Sect. 6, followed by the conclusion in Sect. 7.

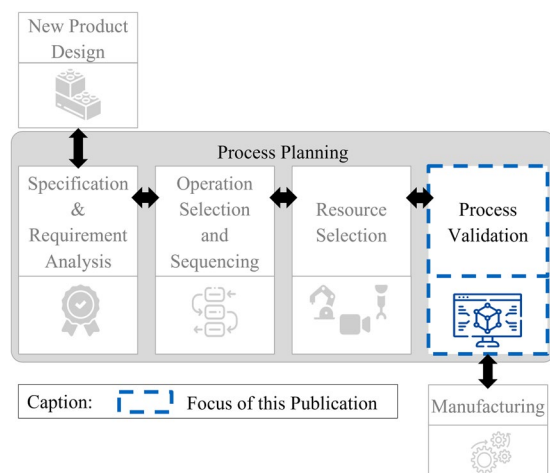


Fig. 1 Focus of this publication in relation to the process planning steps (according to ElMaraghy et al. [10])

2 Literature review

The following section gives insight into the topics essential for a comprehensive understanding of the system and the approaches proposed in this publication. Here, the need for action is identified and introduced.

Automated process planning is presented in combination with skill-based approaches to identify process resource combinations, which are a prerequisite for process validation. Current research in the context of automated process planning is presented with its advantages and disadvantages. In addition, simulation approaches are introduced, which are frequently used for validating process plans to check process-resource combinations. Finally, optimization approaches to identify suitable process plans are described.

2.1 Skill-based and automated process planning

Skills have proven their worth in process and assembly planning, as they provide a viable approach to defining the skills of individual resources independently of the specific hardware behind them. This enables transferability in a heterogeneous production landscape. The advantage is that once a skill-based approach has been created, it can be used multiple times to plan processes. Also, assigning skills as a self-description to cyber-physical systems enables the flexible planning and usage of these resources [11]. Skill-based approaches are used in various planning domains, such as automated process planning, assembly planning, and inspection planning [12].

Backhaus et al. [13] present a skill-based approach for the task-oriented programming of robots in assembly using a hardware-independent skill description. Resource skills are automatically matched with requirements to allocate production resources.

Michniewicz and Reinhart [14] show an approach where individual tasks are automatically assigned to resources to create multiple assembly plans. The individual assembly process requirements and resource skills (i.e., functionalities) are described semantically and parametrically to enable a two-step matchmaking process. Simulation is then used to validate these assembly plans.

Hammerstingl and Reinhart [15] present a taxonomy for skills in the assembly that more comprehensively describes skills semantically and according to their relevant parameters (e.g., joining force).

Järvenpää et al. [16] present a skill-based matchmaking approach to simplify system design and reconfiguration planning. In a two-step approach, the resource combinations and their skill sets are identified and then matched

with the product requirements. It is implemented using an ontology in combination with SPARQL Inferencing Notation (SPIN) rules for comparing qualification requirements.

According to Gonnermann et al. [17], the generation of process monitoring matches, which involve combinations of process resources, is facilitated through semantic and parameter-based matchmaking. Different process monitoring resource combinations are generated automatically. Further research combines the skill-based approach to automated generation of monitoring tasks with a skill-based approach to assembly processes [18]. By considering various assembly methods and monitoring alternatives simultaneously, suitable assembly plans can be further specified

A disadvantage of these approaches is often the transferability to other applications due to their narrow and domain-specific development. Often, skill-based approaches are used for specific use cases. However, once such a system is established, the time and cost of process planning can be reduced tremendously. When combined with matchmaking approaches, multiple process resource combinations (i.e., matches) and, thus, process plans (i.e., assembly and/or monitoring plans) can be generated quickly. The validation aspect of these process plans before the set-up phase is essential and can be done by simulating the individual processes.

2.2 Multi-body simulations for process plan validation

In general, a multi-body simulation can contain two elements, a description of the system's kinematics and one of its kinetics [19]. In this approach, the multi-body simulations consist exclusively of a kinematic simulation. The kinetic behavior is passed to the system via the input data. In turn, no kinetic model or simulation is required.

Multi-body simulations can detect inferences and collisions between different three-dimensional bodies according to their predefined movement. Not only the resources that execute the processes can be simulated in combination with their kinematics, but also the product and its individual assemblies and parts. In addition, the station the process is executed on can be considered, including environmental objects such as safety fences.

Michniewicz and Reinhart [14] present an approach for validating assembly plans. In a two-stage simulation, the value-adding processes (i.e., joining and bolting processes) are validated first, e.g., regarding accessibility, collision freedom, or safe gripping. Non-value-adding processes (i.e., transport and feeding processes) are tested in a global simulation. A global simulation considers the whole process at the same time, e.g., assembling a gear, while a local

simulation evaluates a single process step, such as fastening a screw.

Gonnermann et al. [18] show an approach for validating assembly plans' collision freedom and visual reachability in combination with monitoring plans. A disadvantage of this publication is the time-consuming process of executing each individual assembly plan in combination with the appropriate monitoring resources. Validating each process plan leads to redundant simulation efforts because individual processes are tested for each alternative plan in which they are contained.

Since the presented approach focuses on validating the applicability of sensors for monitoring tasks, the following two sections present the state of the art of simulation methods used. Collision detection and ray casting are applicable to any tactile or visual sensor type. In the multi-body simulations, these two methods are used to ensure collision freedom of visual and tactile sensors at the region of interest (ROI) and the field of view (FOV) of visual sensors. Visual sensors are checked for visibility and accessibility, while tactile sensors, like force-torque sensors, are only checked for accessibility.

2.2.1 Collision detection for sensor validation with bodies

In order to evaluate accessibility for sensors via collision detection within multi-body simulations, the sensors are typically modeled as three-dimensional objects [20].

In practice, simulation tools are used to detect collisions. Siemens Tecnomatix Process Simulate [21] is an example of commercial collision detection tools or tools offering a collision detection function. An open-source alternative is Unity [22], used for example by [23]. Unity is a game engine that provides collision detection functions within its physics package [24]. These tools reduce the implementation effort required for realizing collision detection, e.g., by evaluating distances between objects [25] or applying voxel-based methods [26].

2.2.2 Collision detection for sensor validation using ray casting

Ray casting is a method used in computer graphics to determine the intersection of a ray with an object (i.e., a two-dimensional or three-dimensional object) [27]. Compared to ray tracing, no reflections or refractions are detected due to the sole focus on collision detection of the rays. Because of this simplification, ray casting is often faster and more efficient [28]. Ray casting is used in various fields, such as multi-sensor tracking [29], radar simulations [30], and computational fluid dynamics [31]. It is commonly used in game engineering and other interactive applications for collision detection, determining whether a virtual object, such as a

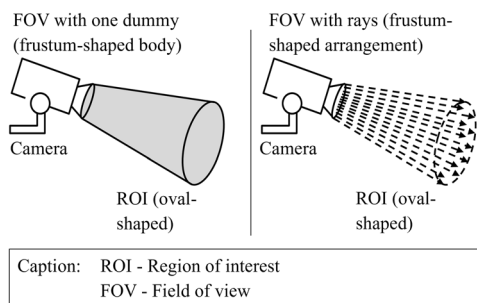


Fig. 2 Body of the field of view as a dummy (left) and ray casting field of view as a dummy (right)

character, collides with another object in a virtual world. In addition to its application in a two-dimensional scene, ray casting can also be utilized to determine the field of view (FOV) in a three-dimensional scene [28]. Ray casting operates by initially defining a field of view (FOV) within which individual rays can detect specific objects. These rays originate from a predetermined starting point and are emitted at specific angles and lengths. Intersections between the rays and other objects produce hits that can be detected and analyzed. Figure 2 compares how a FOV can be projected by a body or rays. The left side of the figure shows the FOV as a frustum-shaped body, and the right side shows the rays inside the FOV that form a frustum.

2.3 Optimization algorithms for skill-based planning

Section 2.1 introduced the skill concept to model and match resources and tasks in assembly, including monitoring in assembly. In addition, methods to match resource skills with process requirements are introduced. Section 2.2 presented approaches to validate matches regarding accessibility and visibility. Matches consist of a process quality feature with a sensor's capability to measure the magnitude of that feature. An example of this is using a torque sensor to monitor the torque needed for screw fastening. This section addresses the final aspect of automated assembly planning, namely determining an optimal combination of matches to perform or, in this case, to monitor the process (Sect. 2.1).

Neural networks (NN) are gaining popularity in assembly and production technology for their effective handling of complex optimization problems [32]. However, NNs require large sample sizes [33]. This requirement is typically difficult to meet in an assembly scenario, as data sets are not guaranteed to contain every possible scenario [32]. There are methods to deal with these circumstances, but they usually result in a sample-set-specific NN [33].

Another option is to use multi-objective optimization methods when the points of interest can not be reduced to

a single factor, e.g., costs [11]. When the key indicator can be reduced to a single factor, planning with graph search algorithms is applicable [13]. In these graphs, the nodes are the matches, and the edges represent the cost of each match (i.e., weighting). Instead of Dijkstra's algorithm [34], this approach applies the A* search algorithm [35] for process monitoring planning. Figure 3 depicts the steps of the A* search algorithm. Just like Dijkstra's Algorithm, the first step is to calculate the evaluation function f for the start node and add it to the set of open nodes to be evaluated.

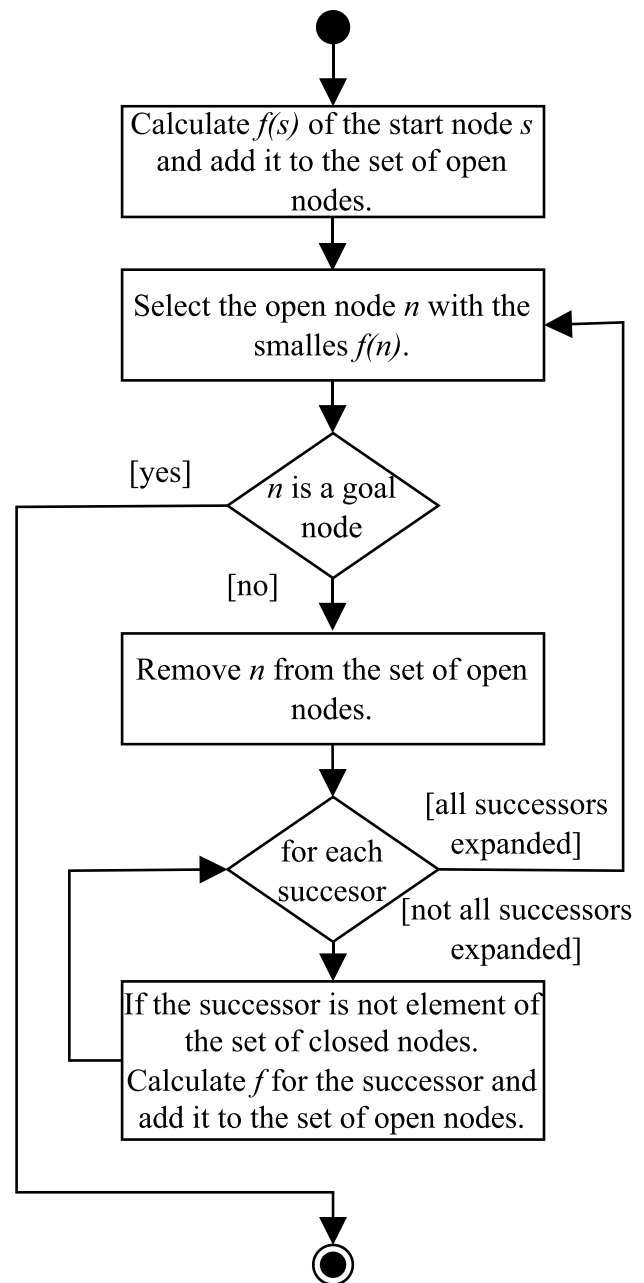


Fig. 3 Steps of the A* search algorithm as proposed by [35]

In contrast to Dijkstra's Algorithm, the A* search algorithm's evaluation function $f(n)$ is composed of the actual costs to reach a node n $g(n)$ and a prediction of the costs to reach a goal node, the heuristic $\hat{h}(n)$. The importance of the heuristic function predicting future costs will be explained later. After the set of open nodes is initialized, the node with the smallest evaluation function $f(n)$ is selected. If n is a goal node, a solution is found, and the algorithm is terminated. Otherwise, n is removed from the set of open nodes and added to the set of closed nodes. Afterward, every successor node of n is expanded. This is done by calculating the evaluation function for the successor and adding it to the set of open nodes if the successor node has not been closed already.

As pointed out, the heuristic function $\hat{h}(n)$ improves the A* search algorithm in contrast to Dijkstra's algorithm. The advantage of the heuristic is the algorithm's ability to prioritize nodes, promising a better solution. A better solution is characterized by scoring lower in the evaluation function. In turn, fewer nodes need to be expanded, leading to fewer computation steps necessary. For this purpose, the heuristic needs to comply with the condition of admissibility. A heuristic $\hat{h}(n)$ is defined to be admissible when it is constantly underestimating the actual future costs $h(n)$. By meeting this condition, it is also proven that the A* search algorithm is optimal [36]. This means that the cheapest solution concerning the specified evaluation function f is always found. In addition, the algorithm is guaranteed to be complete, so it will always find a solution if one exists [37].

3 Need for action

The importance of process monitoring has increased significantly, especially in the context of flexible or reconfigurable production systems, as highlighted in the introduction. It should be noted that process monitoring planning is exclusively concerned with generating non-value-adding processes (i.e., secondary processes). Dealing efficiently with this complexity requires the use of automated approaches to support users in the creation of production and monitoring process plans. In particular, the validation and prioritization of suitable monitoring procedures is a time-consuming and costly undertaking due to the variety of possibilities.

The goal of complexity reduction in this publication is to exclude invalid monitoring matches automatically. In assembly planning, multi-body simulation is a popular choice for match validation, for instance, applied by Backhaus & Reinhart [13] and Michniewicz & Reinhart [14]. In order to validate matches in process monitoring planning, existing approaches must be adapted to validate secondary processes. These secondary processes can use either tactile or visual sensors. Regardless of which sensor is used, it is essential to

verify that the use of the sensor allows process monitoring without interfering with existing assembly processes.

The assembly plan already specifies many requirements for automated validation of process monitoring, such as the arrangement of monitoring tasks, their corresponding characteristics, and the intended location for monitoring. All locations where assembly processes are carried out are already defined here; therefore, the monitoring processes' locations are also specified. The remaining steps for planning monitoring tasks are the identification of resources that match the monitoring tasks and the corresponding planning of process monitoring.

As the match between the skills required for the processes or their monitoring and the skills offered by the resources is extensively discussed in the literature (see section 2), the match between process and resource skills is considered as input for the approach presented in this publication. This allows the approach of this publication to focus on reducing the number of available options for planning process monitoring tasks by validating matches and thus reducing planning complexity.

4 System for accessibility and visibility checking in process monitoring

4.1 Overview

The overall process to validate matches for process monitoring planning is depicted in Fig. 4. In a preliminary step, the process steps to monitor and the matches between the monitoring features and the available equipment are imported. At this point, the matches are based on their description but are not guaranteed to comply with geometric or kinematic constraints. As all these matches can be used for process monitoring planning so far, there are a lot of possible

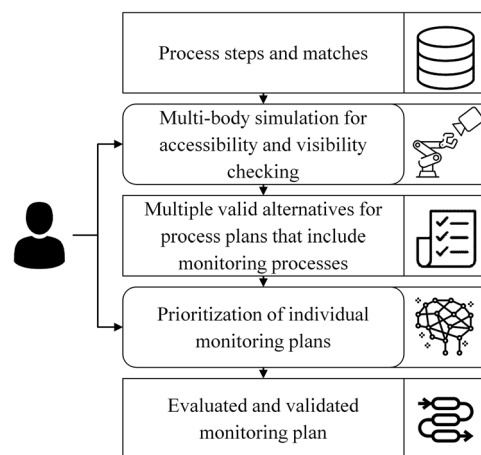


Fig. 4 Process flow describing the validation of monitoring matches

combinations. Each match is validated via a multi-body simulation to reduce the planning complexity. Based on the set of valid matches, process monitoring planning is then performed using the A* algorithm. The result of this step is the evaluated and validated monitoring plan.

The system executes the entire process from Fig. 5. The system is composed of five sub-systems, which are described in more detail below. The logic of the presented approach (Fig. 4) is modeled in the *Solution Structure* sub-system, ensuring that all steps are executed in the right order. In addition, it encapsulates the logic and the executing subsystems to enable adaptability and extendability. The interaction with the user is carried out by the *User Interface*. It serves as a communication platform handling all user inputs and returning the process monitoring plan to the user.

The user's first step is to import an assembly plan containing the monitoring tasks and the matches via the *Data Management* sub-system. This step converts the data from an external format (e.g., text-based data format - JSON file format) into an internal data structure. The data structure and the internal relations can be seen in Fig. 6 and will be discussed in detail at the end of this section. Subsequently, to the data import, the matches are passed from the *Data Management* sub-system to the *Multi-Body Simulation* via the *Solution Structure* sub-system. There, the matches are validated regarding accessibility and visibility (see Sect. 4.2).

In the third and final step, the validated matches are forwarded to the *Process Monitoring Planning* sub-system (Sect. 4.3). This sub-system builds the graph required for the A* search, customizes the evaluation function, and calculates the process monitoring plan. The user can

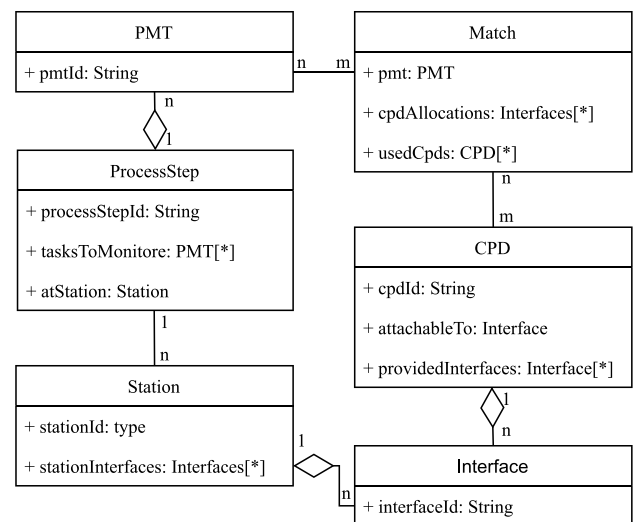
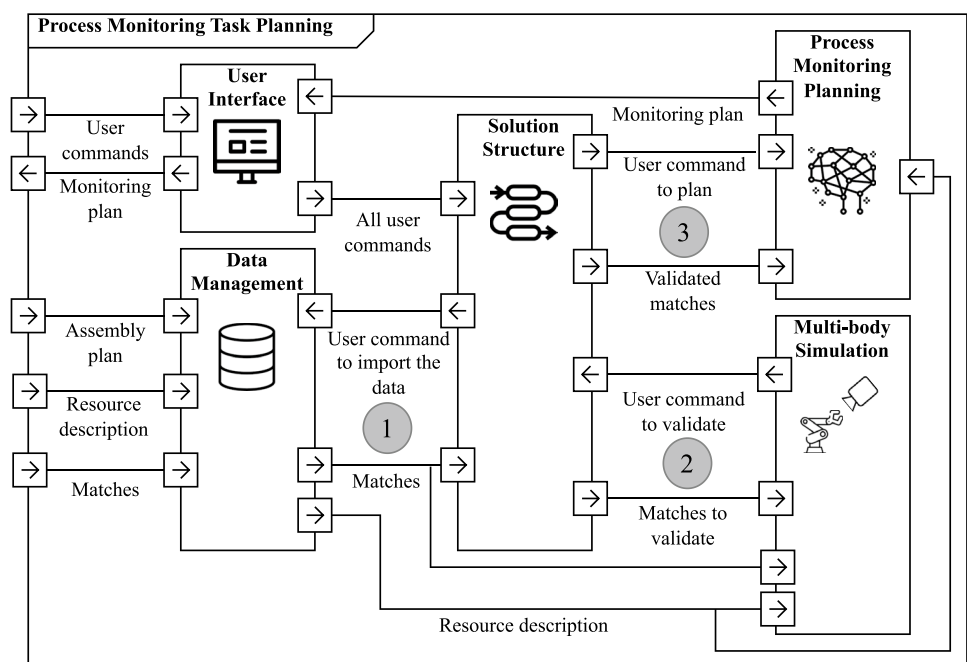


Fig. 6 Input data in the internal representation. The listed attributes are limited to the attributes that are necessary for the structure and relations

customize the evaluation function via the *User Interface*, thereby the user can set priorities in the planning process. The user receives the final monitoring plan as a text-based file (e.g., JSON file), which is optimal according to the cost model and the multi-body simulation via the *User Interface*.

As already mentioned, the assembly plan and the matches between monitoring features and resources are required for the execution of the process monitoring planning. To make the steps of validation (Sect. 4.2) and

Fig. 5 Schematic representation of the system as a SysML diagram showing the system structure. The five sub-systems are *User Interface*, *Data Management*, *Solution Structure*, *Process Monitoring Planning*, and *Multi-body Simulation*. The interactions between the sub-systems are represented and described on the connecting lines



planning (Sect. 4.3) more understandable, the input data (Fig. 6), its elements, and their relations are explained below.

One or more assembly plans are required in a text-based format and are imported into the data management system and describe the sequence of primary process steps, resources, and parts involved, as well as the movements of parts and resources. The resource descriptions are required to describe the skill sets, kinematics, and ROIs. Monitoring matches are also necessary, indicating the individual resources (i.e., cyber-physical devices—CPD) that perform a process monitoring task by being deployed at a particular station and position (see Fig. 6).

Interfaces establish the connection between stations and CPDs, defining the possible positions. Further, process steps are executed at a specific station. A process step requires all of its available monitoring matches (i.e., process monitoring tasks—PMT) that need to be validated in the simulation.

4.2 Match validation process

This section presents the match validation scheme for the monitoring task. The overall match validation simulation is presented before introducing the implementation of the two collision detection mechanisms used in this approach. In general, four types of elements are considered when simulating. These categories are distinguished by their behavior over time and whether they are a part or a CPD.

If an element remains in the same position throughout the validation simulation, it is classified as static and, therefore, referred to as a static part or static CPD. The remaining elements are movable CPDs assembled in the kinematic chain that performs the assembly step or movable parts of the product.

So far, the validation simulation can validate an assembly step with a single part being mounted simultaneously, taking one kinematic chain executing the step into account. The overall validation process can be seen in Fig. 7, including setting up the simulation (steps 1–6), executing it in step 7, and evaluating it in step 8.

When validating a match, the first step is discretizing the moving part's trajectory. This creates a limited number of static images, reducing the scenario's complexity. After the trajectory is discretized, the static parts (step 2) and CPDs (step 3) are added to the simulation environment by spawning them. Then, the moving CPDs are assembled into the kinematic chain (step 4) and spawned as well (step 5). The last step that completes the simulation setup is spawning the moving part (step 6).

The simulation is executed in step 7 by iterating over the sub-steps 7.1–7.3, depicted in Fig. 7. In each simulation step, the position of the moving part is first updated according to the definitions of the process step (step 7.1). Afterwards,

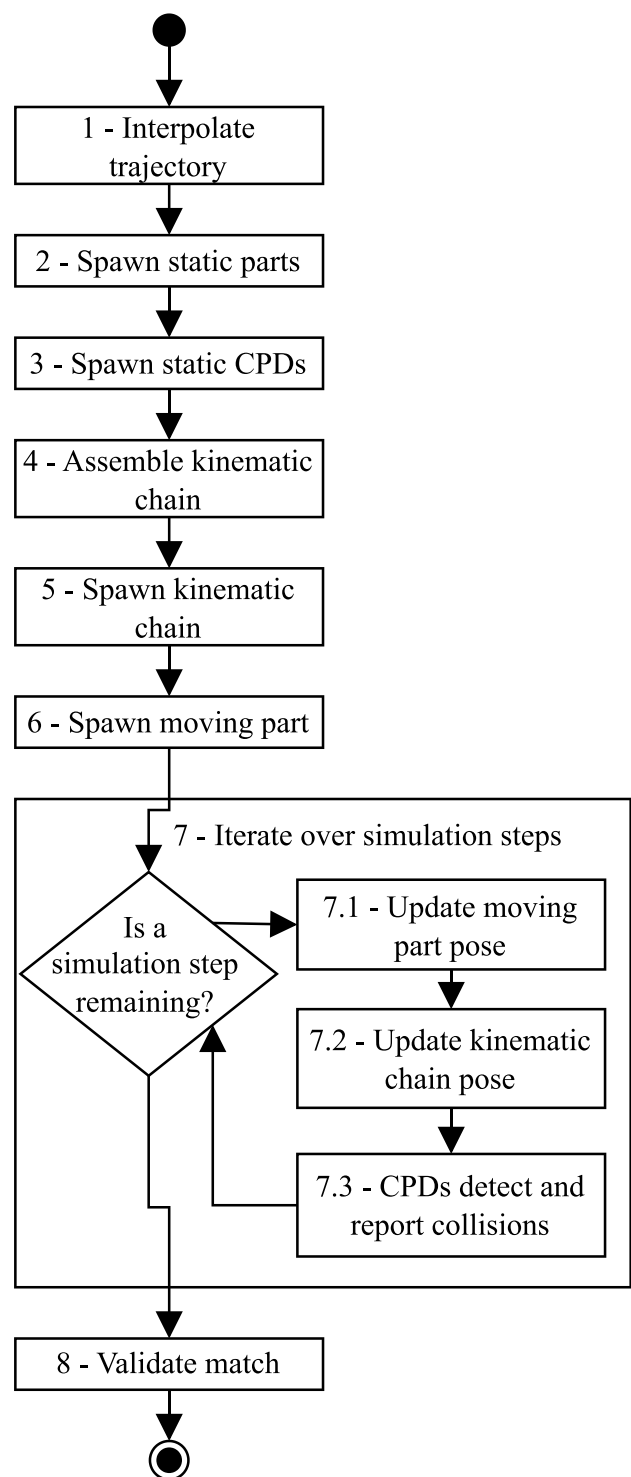


Fig. 7 Overall simulation scheme including set-up

the pose of the kinematic chain in the joint space is updated by solving the inverse kinematics (step 7.2). Then, the collision detection mechanisms are executed to validate the single simulation step (step 7.3). The visibility and accessibility checking are done by using the collision detection

mechanisms for the FOV or the device path (e.g., tactile and/or visual sensor). The collision detection mechanisms are explained in detail in Sects. 2.2.1 and 2.2.2 and will not be further elaborated at this point. Section 5 will present the mechanisms in action and the results. If a collision is detected, it is reported and stored in the collision log.

The overall match and its simulation are validated by analyzing the collision log for collisions between CPDs and parts that reduce a CPD’s visibility or accessibility of the ROI during the process monitoring task (step 8, Fig. 7). If such a collision is documented in the collision log, the match fails the validation process and is thus excluded from the set of planning options.

4.3 Process monitoring planning

The final step in process monitoring task planning is identifying an optimal plan by selecting a combination of the previously validated matches. This approach applies an A* search to find an optimal monitoring plan based on a user-specific cost model and heuristics.

In this section, the planning process is presented. In order to apply the graph-based A* search, the validated matches (Sect. 4.2) need to be transformed into a graph. The result is named the *Process Monitoring Graph*, and it is shown in Fig. 8.

The *Process Monitoring Graph* is constructed as follows. The PMTs are gathered with their valid matches. In Fig. 8, valid matches are assigned an identifier to distinguish between them; e.g., the first valid match of the second PMT (PMT2) is assigned the identifier 2.1. Then all the PMT’s valid matches are linked to all valid matches belonging to the next PMT. This results in a directed graph on which the A* search algorithm [35] can be executed.

The costs are modeled as follows. The total cost of executing a match within a process monitoring plan is defined as the sum of the following costs:

- combined operating costs of all CPDs for a planned product quantity,

- costs to install the CPDs in the desired locations,
- costs to prepare the station to install the CPDs, e.g., remove other CPDs or install hardware interfaces,
- costs to transport all CPDs from the current location to the desired one,
- costs to warehouse unused CPDs,
- costs to purchase CPDs in case they have already been used by a previously selected match for another monitoring task.

In order to define an admissible heuristic, future costs are approximated by calculating the sum of operating and procurement costs for every match in the remaining monitoring tasks. These two factors are selected, as the procurement costs are known in advance, and the operating costs can be approximated without much effort. The match with the minimal sum and all available resources is selected for each monitoring task. Then, the approximated costs are summed up to the heuristic value h . The computed plan is optimal as long as the heuristic is admissible, e.g., in the presented heuristic.

5 Results

The match validation and process monitoring tasks planning presented in the previous section are tested with two use cases. The first one applies collision detection for a force torque sensor, as well as for cameras. In the second use case, the cameras’ visibility is validated by ray casting. Both use cases are executed on the same demonstrator. In the use cases, a Lego brick scenario is assembled, offering various opportunities for monitoring task match validation.

The demonstrator and its assembly scenario are shown in Fig. 9. The station interfaces ($st_{...}$) are hardware interfaces where CPDs can be installed. An industrial robot is installed on interface st_{I_1} , and cameras can be installed on interfaces st_{I_2} and st_{I_3} . Furthermore, a force-torque sensor can be installed between the robot and the gripper. In the use cases, three sensors are available for process monitoring. A force-torque sensor (fts_I) was installed on the robot, and two cameras, cam_1 installed on st_{I_2} and cam_2 on st_{I_3} .

Figure 10 shows the validation of the cameras positioned at the two available interfaces with FOV dummies. In the left figure, a collision between the camera’s FOV dummy and the gripper can be seen where the FOV overlaps with the gripper. Therefore, a collision is reported, and the match is not feasible. In contrast, the match validated in the right figure is collision-free and, thus, valid.

Another approach investigated in this publication for determining the visibility and accessibility of sensors is by applying a ray casting approach. In Fig. 11, the yellow rays

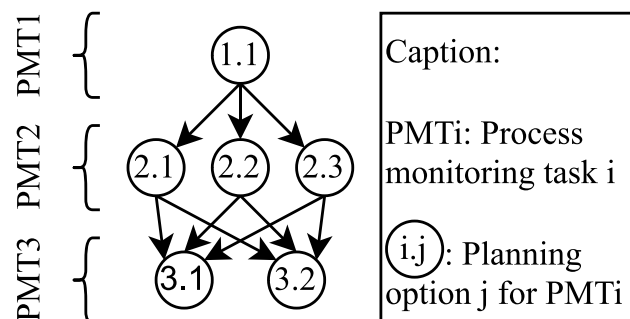


Fig. 8 Example of a *Process Monitoring Graph*

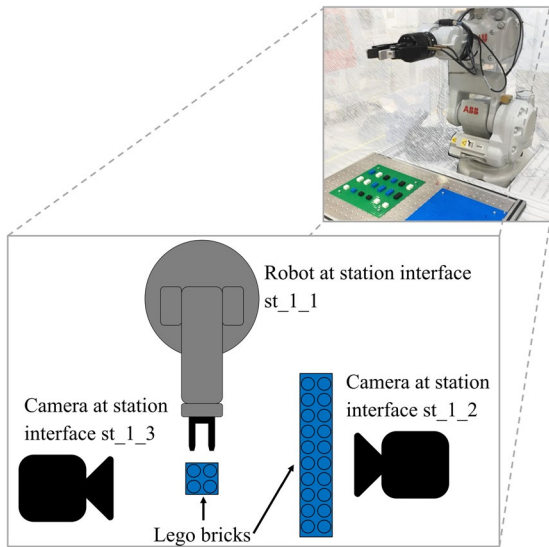


Fig. 9 Structure of the demonstrator with the assembly scenario for both use cases

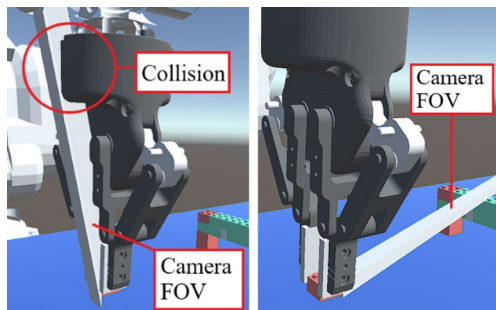


Fig. 10 Collision detection with FOV dummies

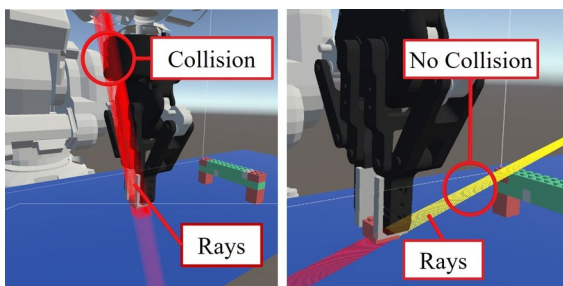


Fig. 11 A collision (red rays) with ray casting is detected between the camera and brick to be assembled (left). The other camera with a different angle has no collision (yellow rays) detected with ray casting between brick and camera (right)

detect the part that will move during the process, while the red rays show the region behind the part (i.e., not detectable by the camera). The collision in the left figure is conducted similarly to the multi-body simulation with the dummy FOV

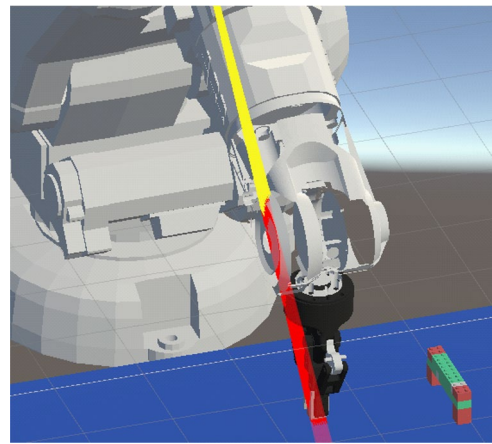


Fig. 12 Collision occurring between the sensor and one joint of the ABB robot

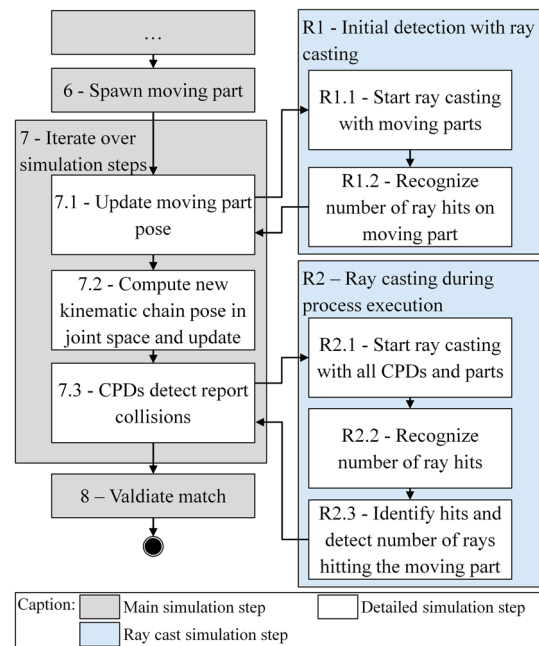


Fig. 13 Method describing the two-step ray casting approach

of the first simulation method (see Fig. 12). The visualization and accessibility checking are implemented in a two-step approach. In Fig. 13, these two steps are differentiated by R1 (Ray casting 1) and R2 (Ray casting 2), which consist of individual processes.

In the first step (R1), the part to be assembled is detected by the rays without any other resources. The result is the number of ray hits that need to be reached in the second step (R2) when the process is executed, and other resources (i.e., CPDs) are also considered. In the second step (R2), ray casting is continuously applied during the movement of the part and CPDs.

Again, the number of hits on the part to be assembled is recognized and compared to the previously detected number of hits in step one. If the number of hits is lower, it can be assumed that the visibility or accessibility of the sensor is limited by other resources or objects. The individual rays do not need to be modeled before but are directly created during the execution of the simulation, which reduces time in the preparation of the simulation.

6 Discussion

The proposed system for accessibility and visibility checking in process monitoring utilizes simulation-based validation to reduce the planning alternatives for process monitoring in assembly. This approach is suitable to validate matches in a virtual environment, saving time, cost, and expert knowledge compared to physical testing. This can lead to a more efficient process and minimize human error. The system uses a multi-body and ray casting simulation to validate the matches. The multi-body simulation requires more effort for adjustments (e.g., production system or operating equipment) than the ray cast method. Both methods are capable of identifying collision-free matches and determining the degree of collision-freeness (i.e., amount of overlap) based on the number of ray hits before and during the process on the part being assembled.

A limitation of the system is that it cannot check two simultaneous monitoring tasks for one process step, e.g., when a force-torque sensor and a camera are needed. Future research could address this limitation by incorporating multiple sensors and cameras into the simulation. A critical evaluation of the system shows that it has not been tested on complex examples, such as larger components with different free-form surfaces, multiple connection directions, larger or smaller areas of interest with other sensor types, and multiple stations connected in series. Therefore, testing the system on such examples would be important to verify its performance and suitability for more complex and industry-related scenarios. This publication demonstrates the advantages of ray casting in terms of adaptability and reusability for flexible/reconfigurable production systems. The automatic generation of rays eliminates the need for pre-modeling of bodies for the multi-body simulation.

7 Conclusion

This publication presented an approach to validate matches resulting from skill-based matchmaking in process monitoring planning. Two aspects, in particular, were considered during validation: Accessibility and Visibility. Two different methods were used to validate the latter. In turn,

the methods could be evaluated in terms of their initialization and computational effort while testing them in use cases that consider a wide range of scenarios. Furthermore, the applicability of A* search for planning an optimal monitoring task schedule was demonstrated.

The next step in this research will be to apply the simulation approach to the real demonstrator shown in Fig. 9. This allows scalable testing of the implemented approach in a semi-industrial/laboratory environment with multiple production processes and product variants. In the future, applying the presented approach to more complex and realistic use cases will be necessary. This will improve the presented approach and ensure its applicability. Moreover, adding a global simulation with the entire production surrounding (e.g., workers) is inevitable. This global simulation will confirm the validity of the selected plan and eliminate the degradation of visibility and accessibility due to environmental factors.

Acknowledgements This research was funded by the EU project ASSISTANT grant number 101000165. The authors would like to thank all the participants in the ASSISTANT project.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The datasets generated and analysed during this study are not currently publicly available.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Zhou K, Liu T, Zhou L (2015) Industry 4.0: towards future industrial opportunities and challenges. pp 2147–2152. <https://doi.org/10.1109/FSKD.2015.7382284>
2. Koren Y, Gu X, Guo W (2018) Reconfigurable manufacturing systems: principles, design, and future trends. *Front Mech Eng* 13(2):121–136. <https://doi.org/10.1007/s11465-018-0483-0>
3. Jarvenpaa E, Siltala N, Lanz M (2016) Formal resource and capability descriptions supporting rapid reconfiguration of assembly systems. <https://doi.org/10.1109/isam.2016.7750724>

4. Bortolini M, Galizia FG, Mora C (2018) Reconfigurable manufacturing systems: literature review and research trend. *J Manuf Syst* 49:93–106. <https://doi.org/10.1016/j.jmsy.2018.09.005>
5. Verein Deutscher Ingenieure (VDI) (1985) VDI 2619: 1985. Tech. rep., VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen
6. DIN-EN-ISO-9000 (2015) Quality management systems—fundamentals and vocabulary. <https://doi.org/10.31030/2325650>
7. Ahmad MI, Saif Y, Yusof Y, Daud ME, Latif K, Kadir AZA (2022) A case study: monitoring and inspection based on iot for milling process. *Int J Adv Manuf Technol* 118(3–4):1305–1315. <https://doi.org/10.1007/s00170-021-07970-y>
8. Verein Deutscher Ingenieure (VDI) (1990) VDI 2860: 1990. Tech. rep., VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen
9. DIN-8593 (2003) Fertigungsverfahren fügen_: Allgemeines; einordnung, unterteilung, begriffe. <https://doi.org/10.31030/9500684>
10. ElMaraghy HA, Nassehi A (2019) Computer-aided process planning. pp 339–345. https://doi.org/10.1007/978-3-662-53120-4_6551
11. Siltala N, Järvenpää E, Lanz M (2018) An executable capability concept in formal resource descriptions. *IFAC-PapersOnLine* 51(11):102–107. <https://doi.org/10.1016/j.ifacol.2018.08.242>
12. Järvenpää E, Siltala N, Hylli O, Lanz M (2019) The development of an ontology for describing the capabilities of manufacturing resources. *J Intell Manuf* 30(2):959–978. <https://doi.org/10.1007/s10845-018-1427-6>
13. Backhaus J, Reinhart G (2017) Digital description of products, processes and resources for task-oriented programming of assembly systems. *J Intell Manuf* 28(8):1787–1800. <https://doi.org/10.1007/s10845-015-1063-3>
14. Michniewicz J, Reinhart G (2016) Cyber-physical-robotics—modelling of modular robot cells for automated planning and execution of assembly tasks. *Mechatronics* 34:170–180. <https://doi.org/10.1016/j.mechatronics.2015.04.012>
15. Hammerstingl V, Reinhart G. Skills in assembly. <https://media.tum.ub.tum.de/1428286>. Last accessed on 06 Jan 2024
16. Järvenpää E, Siltala N, Hylli O, Lanz M (2021) Capability matching software for rapid production system design and reconfiguration planning. *Procedia CIRP* 97:435–440. <https://doi.org/10.1016/j.procir.2020.05.264>
17. Gonnermann C, Weth J, Reinhart G (2020) Skill modeling in cyber-physical production systems for process monitoring. *Procedia CIRP* 93:1376–1381. <https://doi.org/10.1016/j.procir.2020.03.095>
18. Gonnermann C, Hashemi-Petroodi SE, Thevenin S, Dolgui A, Daub R (2022) A skill- and feature-based approach to planning process monitoring in assembly planning. *Int J Adv Manuf Technol* 122(5–6):2645–2670. <https://doi.org/10.1007/s00170-022-09931-5>
19. Angeles J, Kecskemethy A (2014) Kinematics and dynamics of multi-body systems. ISBN: 9783709143629
20. Polini W, Moroni G (2014) A frame for a computer aided inspection planning system. *IJET (Int J Eng Technol)* 4(1):125. <https://doi.org/10.14419/ijet.v4i1.3937>
21. Siemens (2022) Conduct assembly simulation for virtual process verification. <https://www.plm.automation.siemens.com/global/en/products/tecnomatix/assembly-simulation.html#top>. Last accessed on 06 Jan 2024
22. Unity Technologies (2022) Unity. <https://unity.com/>. Last accessed on 06 Jan 2024
23. Sita E, Horvath CM, Thomessen T, Korondi P, Pipe AG (2017) Ros-unity3d based system for monitoring of an industrial robotic process. pp 1047–1052. <https://doi.org/10.1109/SII.2017.8279361>
24. Unity Technologies (2020) Unity physics package: collision queries. https://docs.unity3d.com/Packages/com.unity.physics@0.6/manual/collision_queries.html. Last accessed on 06 Jan 2024
25. Ciaponi LJ (1989) Digital simulation of a 6-dof multi-body dispense with body-on-body collision detection. pp 390–397. <https://doi.org/10.1109/NAECON.1989.40239>
26. Lawlor OS, Kalée LV (2002) A voxel-based parallel collision detection algorithm. pp 285–293. <https://doi.org/10.1145/514191.514231>
27. Davidovic T, Engelhardt T, Georgiev I, Slusallek P, Dachsbacher C (2012) 3d rasterization: a bridge between rasterization and ray casting. *Proc Graph Interface* 2012:201–208
28. Friston S, Steed A (2019) Real-time collision detection for deformable characters with radial fields. *IEEE Trans Visual Comput Graph* 25(8):2611–2622. <https://doi.org/10.1109/TVCG.2018.2859924>
29. Liu W, Liu Q, Chen Y, Cui H (2023) Multi-sensor tracking with partly overlapping fov using detection field of probability modeling and the glmb filter. *Eurasip J Adv Signal Process*. <https://doi.org/10.1186/s13634-022-00962-4>
30. Manivasagam S, Wang S, Wong K, Zeng W, Sazanovich M, Tan S, Yang B, Ma WC, Urtasun R (2020) Lidarsim: realistic lidar simulation by leveraging the real world. <https://doi.org/10.1109/cvpr42600.2020.01118>
31. Fernandes O, Frey S, Reina G, Ertl T (2019) Visual representation of region transitions in multi-dimensional parameter spaces. *STAG: Smart Tools and Applications in Graphics*. <https://doi.org/10.2312/stag.20191367>
32. Kusiak A (2020) Convolutional and generative adversarial neural networks in manufacturing. *Int J Prod Res* 58(5):1594–1604. <https://doi.org/10.1080/00207543.2019.1662133>
33. D’souza RN, Huang PY, Yeh FC (2020) Structural analysis and optimization of convolutional neural networks with a small sample size. *Sci Rep* 10(1):834. <https://doi.org/10.1038/s41598-020-57866-2>
34. Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numer Math* 1(1):269–271. <https://doi.org/10.1007/BF01386390>
35. Hart P, Nilsson N, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern* 4(2):100–107. <https://doi.org/10.1109/TSSC.1968.300136>
36. Du Y, Xie S, Wang Y, Ma Y, Jia B, Liu X, Rong J, Li R, Zhu X, Song CP, Tao WA, Wang P (2022) Low molecular weight protein phosphatase aph mediates tyrosine dephosphorylation and aba response in *Arabidopsis*. *Stress Biol* 2(1):23. <https://doi.org/10.1007/s44154-022-00041-6>
37. Kumar Debnath S, Omar R, Abdul Latip NB, Shely S, Nadira E, Che Ku Melor CKN, Kumar Chakraborty T, Natarajan E (2019) A review on graph search algorithms for optimal energy efficient path planning for an unmanned air vehicle. *Indones J Electr Eng Comput Sci* 15(2):743. <https://doi.org/10.11591/ijeecs.v15.i2.pp743-749>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.