INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.56

Automatic evaluation of the misplacement risk during manual assembly based on a CAD design

Alexander De Cock, Ncamisile Khanyile and Bieke Decraemer ✉

Flanders Make, Belgium

✉ bieke.decraemer@flandersmake.be

Abstract

In HMLV manufacturing, assembly mistakes by operators are common due to the ever increasing product variability and complexity. If mistakes can be detected early-on in the design process, product designers can reduce the possibility for mistakes. We present an algorithm to automatically detect and evaluate potential misplacements of parts that need to be fastened. Evaluation starts from a product CAD and returns the risk of misplacement as well as visual feedback on possible misplacements. An implementation with FreeCAD of our algorithm is illustrated on different use cases.

Keywords: design for x (DfX), computer-aided design (CAD), design analysis, design for assembly, poka yoke

1. Introduction

High mix low volume (HMLV) manufacturing companies produce a diverse range of products with different specifications, configurations, or customization options. This inherent variation requires a high degree of flexibility in the manufacturing and assembly process which often prohibits complete automation of the assembly process (Hu et al., 2008). Therefore, HMLV companies rely on skilled human operators to perform these complex assembly operations as cost efficiently as possible (Hollnagel, 2012). Given the high product variability and complexity, operator doubt and mistakes are common issues during the assembly process. This inevitably leads to slower or wrong execution of the assembly tasks which in turn leads to increased assembly time and cost. Besides the direct impact, operator mistakes also create additional costs through investments in error detection and mitigation processes. Additionally, assembly mistakes that go undetected and are shipped with the product may also affect customer satisfaction and trust. Hence it is important to reduce operator mistakes as much as possible.

Different types of mistakes can be made such as wrongly positioning parts, forgetting parts, connecting parts in an incorrect order, using incorrect tools or connection techniques. In this paper we will limit ourselves to misplacements of parts/subassemblies that are connected by fasteners. The reason we focus on this type of mistakes is that fastener connections are the most common connection method in mechanical products. Additionally, the reuse of the same parts in different product variants leads to parts with surfaces containing excess holes (of which only some are used in each product variant) increasing the chance of mistakes when the operator tries to connect parts by aligning the wrong holes.

In this paper we present the "misplacement rule", an algorithm that automatically identifies potential misplacements of parts/subassemblies connected by fasteners. Evaluation of the misplacement rule starts from a CAD design and a minimal set of user inputs and returns a single value indicating the risk of misplacement. Additionally, potential misplacements can be visualized to give further insight into

how the situation can be remediated. Identification of which assembly operations are most susceptible to mistakes can be used for different risk mitigation strategies. Proactively, the rule output and visualization can help designers to change their product design to reduce the risk of mistakes. Since the output of the rule is qualitative, different designs can be compared against each other. Reactively, the algorithm can be used to identify the assembly operation for which operator support is more desirable or how the work environment can be reorganized to reduce the risk of mistakes.
The remainder of this paper is organized as follows. In Section 2 we position our work with respect to the current state of the art and state of practice. In Section 3, we present an overview of the misplacement identification and evaluation algorithm. In Section 4, we will discuss applications on a set of use cases. Finally in Section 5, we summarize our findings and elaborate on potential extensions to further improve and extend the algorithm.

2. Related work

Different strategies are already employed in industry to reduce the risk of operator mistakes. On the one hand, there are reactive approaches that try to better support operators during their tasks. Examples are (digital) work instructions (Leder et al., 2022; Pimminger et al., 2021; Zogopoulos et al., 2022), visual aids and operator tracking software (Peruzzini et al., 2020). These approaches require additional equipment and resources for each operator and may result in a substantial investment cost. Moreover, studies show that enforcing strict guidelines or control on operators is perceived as demotivating (Eiriksdottir and Catrambone, 2011), therefore undermining the effectiveness of the reactive approach.
Alternatively, proactive approaches can be employed that try to reduce the risk of mistakes by changing the assembly process or product design before the product is placed in production. Examples of proactive approaches are design methodologies such as Design for Assembly (Boothroyd et al., 2011; Eskilander, 2001), Poka-Yoke (Lazarevic et al., 2019; Widjajanto et al., 2020) and Design for Assembly Meaning (Parmentier et al., 2020). While not all methodologies focus directly on operator mistakes, they do contain principles that reduce the risk of mistakes. Proactive approaches have the advantage that additional effort spent during the design stage avoids the need of increased operator support. This makes proactive approaches often more cost effective than reactively solving the issue in production.
Design for Assembly (DfA) is a set of principles and techniques that focuses on reducing the assembly time and cost. While it does not explicitly focus on operator mistakes, it does advocate complexity reducing principles and promotes the use of standardized parts and tools. Thus, it indirectly contributes to reducing the risk of mistakes. Poka-Yoke directly focuses on avoiding assembly mistakes by error-proofing principles. It contains guidelines for both prevention and detection of mistakes. The preventive principles focus on guaranteeing a singular way of assembly by exploiting part symmetry and anti-symmetry. Design for Assembly Meaning is a more recent design methodology that advocates design principles that lead to products that are intuitive to assemble by taking the human psychology and intuition into account. The goal of this methodology is to design products that no longer require any work instructions to be assembled. None of these methodologies address the problem of operator mistakes related to misplacements explicitly.
Applying the aforementioned methodologies involves manual computation, visual inspection and substantial design experience, hence increasing the cognitive load of product designers. Software tools to support the designers to apply these methodologies automatically are therefore critical for the adoption and integration in the current design processes. In academia multiple algorithms have been presented to automate the evaluation of design rules in CAD software, especially in the context of automatic assembly sequence planning (Melckenbeeck et al., 2020; Tariki et al., 2021), hierarchical exploded view generation (Li et al., 2008; Yu and Zhang, 2017) and DWI generation (Gors et al., 2021). To the best of our knowledge, the academic literature does not present algorithms that automatically quantify the risk of misplacement from a CAD file.
Commercial software tools supporting designers with design methodologies are also limited. DFMA® (Boothroyd Dewhurst, Inc., 2023) and AVIX DFX (Solme AB, 2015) provide support in evaluating DfA rules semi-automatically through digital forms. However, the connection with the CAD model is largely absent. Therefore, the amount of manual input these tools require is high and relies on human judgement. These tools are also not tailored to assess the risk of assembly mistakes. DFMPro (HCL

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

Technologies, 2023) is a CAD based tool that evaluates design rules through geometric reasoning, requiring less human input. It focuses on DfM but supports some DfA rules that evaluate the feasibility of connections involving fasteners and holes. Assessing risk for misplacement during an assembly operation is not part of the tool. DFA-Tools (DFA-Tools Ltd, 2017) is a software tool that combines the digital form approach with CAD based reasoning specifically for micro scale assemblies and PCB designs. No support is included to evaluate the risk of operator mistakes.

3. Methodology for the misplacement rule

In this paper we focus on one type of operator mistakes, namely misplacement of parts connected by fasteners. A misplacement is any unique position in which a part/subassembly can be placed and connected during an assembly operation other than the intended position. We assume it is more likely for an operator to make a mistake when there are more possibilities to make a mistake.

The goal of our algorithm is to identify all possible misplacements based on a description of the assembly operation and the product CAD design. The list of identified misplacements will be used to estimate the risk of an operator mistake. To minimize the amount of manual inputs, information is extracted from the CAD design through geometric reasoning. The algorithm relies on standard functionality provided by a CAD kernel such as basis algebraic operations, inspection of geometric shapes, application of transformation matrices and Boolean set operators between geometries.

The approach of the misplacement rule is based on the following assumptions:

- The risk of operator mistakes is proportional to the number of misplacements and the number of misplacements is finite.
- An assembly operation is considered to be the connection between exactly two distinct sets of product parts (one or more per set), also called subassemblies.
- All parts in the product are considered to be rigid bodies, no scaling or deformation of parts is considered.
- The geometries of parts do not intersect in the CAD model, such that collisions can be evaluated in the CAD based on the intersection between geometries.

The first input of the misplacement rule is a CAD file containing the geometries of all the product parts in their fully assembled position. To make the methodology as generally applicable as possible, we only rely on information available through the neutral STEP CAD file format. Secondly, we need a formal description of the operations to evaluate. Each operation is defined by the two subassemblies involved. The subassembly that is assumed to stay in place during the operation is the static subassembly. Its coordinate axes will serve as the reference axes. The subassembly that is assumed to be moved and connected to the static subassembly is called the dynamic subassembly. The user can manually indicate the dynamic and static subassembly by selecting the corresponding geometries in the CAD software.

Given these inputs, the algorithm will generate a list of transformation matrices corresponding to the possible misplacements for the dynamic subassembly. Based on the list of misplacements an estimation of the risk of an operator mistake will then be computed as the output value. The algorithm of the misplacement rule can be broken down in 4 mains steps:

1. ***Interface preprocessing:*** identify the connection and interfaces between the subassemblies from their CAD geometries.
2. ***Position generation:*** based on the open connection interfaces in the operation, all possible positions of the dynamic subassembly are generated by trying to align the connection interfaces of the static and dynamic subassembly.
3. ***Position evaluation:*** the generated positions for the dynamic subassembly are evaluated to see if the position is a valid misplacement.
4. ***Estimating risk:*** Estimate the risk of an operator mistake based on the list of misplacements.

Notice that step 1 and 2 are strongly dependent on the type of connections that are present in the product and will require their own implementation for each type. In contrast step 3 and 4 are more independent of the connection types. In this paper we will focus only on connections involving fasteners since these type of connections are the most common in mechanical products. However the methodology can be extended to other connection types such as welding, brazing, or gluing.

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

Another consideration is whether steps 2, 3 and 4 are performed in a single iteration loop (where each step is performed for each position) or the steps are performed sequentially in batch (where first all positions are generated, next all positions are evaluated, and finally the risk is estimated). Depending on the generation, evaluation and estimation details one implementation may be more efficient than the other. We added a setting such that the user can decide to use a sequential or batch evaluation.

3.1. Interface preprocessing

To generate the positions for the dynamic subassembly, we have to identify the connections and interfaces between the static and dynamic subassembly. For fastener based connections, the interface preprocessing is reduced to identifying which hole-ends and fasteners are involved in the operation based on geometric and logical reasoning on the design. The preprocessing itself consists of 4 steps:

1. ***Identifying and grouping holes:*** for each assembly part the open hole-ends are identified. For this we implemented a rule based hole detection algorithm that can find cylindrical straight holes with multiple diameters. Holes can be open on one or both sides. We assumed that the open ends of the hole are always positioned in a planar surface of a part. For more advanced hole detection algorithms we refer to (Ibrahim et al., 2021; Slyadnev et al., 2020). Identified holes are then grouped by the plane in which their hole-ends lie.
2. ***Identifying fasteners:*** Fasteners are usually standard parts and hence interchangeable. Therefore we do not consider misplacement of fastener parts. We still want to identify fasteners to exclude them from the connection check and to identify which holes are used and open. In our algorithm the fasteners are differentiated from assembly parts based on naming convention in the CAD (e.g., ISO standard naming). Only the fasteners inserted in the hole will play a further role. By checking if the centre points of the hole are within a fastener we determine which fasteners are associated to which holes.
3. ***Identifying connections:*** Connections are not explicitly modelled in the CAD file. We assume that if parts are closer together then a predefined minimal distance, that they can be considered connected. To identify connections we evaluate all assembly part pairs where the first part belongs to the static subassembly and the second part belongs to the dynamic subassembly. For each pair we first check if the bounding boxes of the parts overlap. If this is the case we perform the (computationally more expensive) minimal distance check.
4. ***Identify fastener connections:*** For all identified connections we check which fasteners (and holes) are involved in the connection. Fasteners are associated based on the distance between the fasteners and the two connected parts. If not at least two fasteners and holes can be assigned to a connection, the connection is not considered for further analysis.

After the interface preprocessing we know which holes are present in the subassemblies, which holes are located in the same planes (see Figure 1). Each such plane will be called a hole plane. We also know which fasteners are inserted in which holes, which connections have to be realized between the two subassemblies and which holes and fasteners play a role in these connections. From each hole the diameters and hole-ends are identified. Hole-ends are the outer most open wires of the hole that lie on an outer surface of the part. Finally the 'state' is also known for each hole: Holes without a fastener and holes that are used in the connections in the assembly operation are considered open. The other holes will be considered closed.

3.2. Position generation

Based on the interfaces between parts, different positions for the dynamic subassembly can be generated that still align the connection interfaces correctly but result in a different position than the intended one. New positions are defined by combining translation and rotational transformations to align matching interfaces between the subassemblies. Since we focus on fastener based connections, we propose an algorithm to generate positions for the dynamic subassembly that match at least two open holes between the static and dynamic subassemblies. Two holes are considered matched if:

1. the centre points of one of their hole-ends coincide,
2. the normals of hole planes containing the hole-ends are parallel and opposite in direction,
3. the hole diameters are in such order that the intended fastener can be placed in them.

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

Our position generation algorithm will consider every pair of hole planes where the first plane is a plane from the static assembly and the second plane is a plane from the dynamic assembly. For each plane pair we check if each plane contains a pair of open holes that have matching diameters (condition 3) and have the same distance between the pair of hole-ends in the plane. If this is the case, a position can be generated to make the hole pairs match. This position corresponds to a transformation matrix that represents one translation and two rotations. The translation is chosen such that the two centres of matching hole-ends coincide. Here, the algorithm takes into account a user specified tolerance bound for what will be considered a successful hole match. The first rotation will ensure that the normals of the centre points of these hole-ends are parallel and opposite in direction (condition 2). The second rotation will rotate the dynamic subassembly around the normal hole plane in order to align the hole-end of the second hole.

Notice that the position generation algorithm assumes that there are at least two matching holes between the subassemblies. If a single hole is used the number of misplacements becomes infinite since the second rotation is no longer well defined. However connecting parts with only one fastener is often considered bad design and not common in industrial design.

The described position algorithm will generate all positions for which at least two holes of the dynamic subassembly are matched with two holes of the static subassembly. However, it will potentially generate the same position multiple times. These duplicate positions will be filtered out in the next step. Additionally, the number of generated position may quickly increase in case of many matching hole pairs (see Figure 1). To reduce the number of generated positions some additional constraints can be imposed on the position generation.

We implemented the following constraints on the transformation matrix:

- Only allow translations in coinciding hole planes, no rotation. Corresponds to the case where the operator can only slide the dynamic subassembly in the plane of its connection interface.
- Only allow rotations, no translation. Corresponds to the case where the operator knows the position of the dynamic subassembly but may mistakenly orient it.
- Only consider translations that move the dynamic subassembly less than a predefined distance threshold. Corresponds to the case where it is unlikely that the operator will consider positions far from the initial position (for example due to limited mobility of the operator at the line).

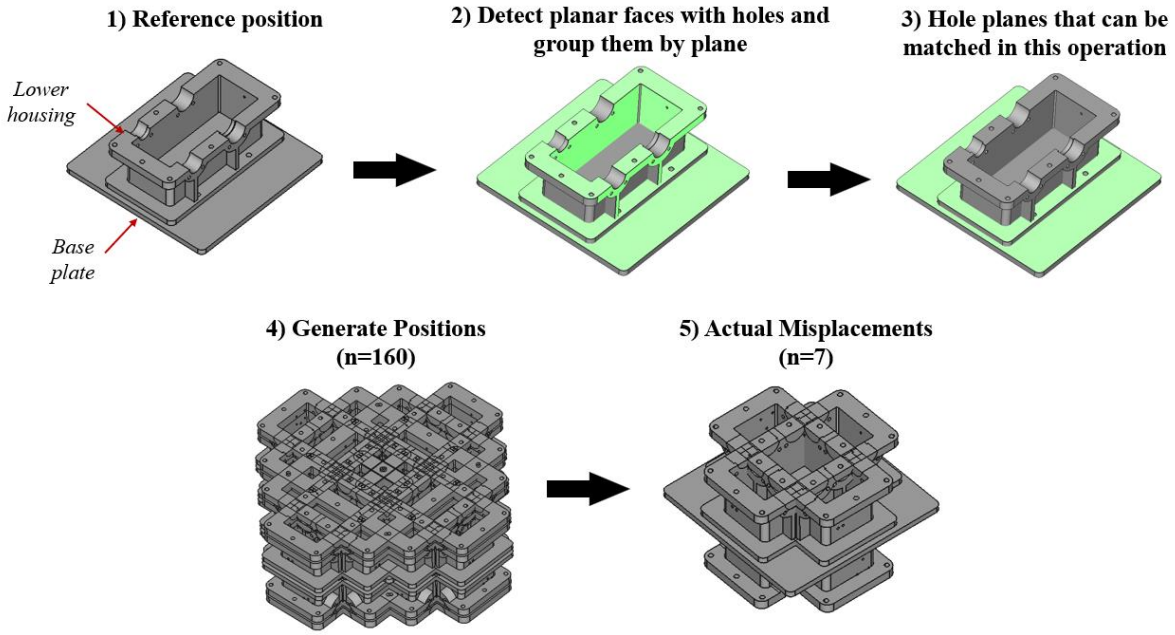

Figure 1. Overview of the different steps of the misplacement rule

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

3.3. Position evaluation

In the position evaluation step we want to determine if a position corresponds to a new misplacement. Distinction between position generation and evaluation can be somewhat blurry since the generation algorithm may already excludes certain operations (by restricting the transformation matrix as discussed in the previous section). However it remains important to keep a clear view on which conditions are imposed on positions in order to be valid misplacements.
We identified the following conditions to decide if a position is a valid misplacement:

1. ***Uniqueness check:*** Is the position different from previously found misplacements? This check is needed since our position generation algorithm does not guarantee to return unique positions.
2. ***Connectivity check***: Does the position allow the dynamic subassembly to be sufficiently connected with the static subassembly? For this we need to check how may hole planes are aligned between static and dynamic subassembly and how many holes are matched within these planes. We added a setting called the hole match ratio that prescribes how large the fraction of matching holes (relative to all holes involved in the connections) needs to be to consider a position correctly connected.
3. ***Collision check:*** Does the dynamic subassembly not collide with the static subassembly or the environmental objects? This involves an intersection operation between the geometry of the dynamic subassembly and the other geometries in the scene. If there is a non-zero common volume the dynamic subassembly is colliding and the position cannot physically be realized.
4. ***Symmetry check:*** Is the resulting geometric configuration different from configurations obtained for already identified misplacements? This step is important if the subassemblies are symmetrical, since for such subassemblies different positions may result in exactly the same configuration. Without prior knowledge on symmetry this check requires an intersection operation between the dynamic subassembly in the current position and a copy of the misplaced dynamic subassembly.

Since a position is only a valid misplacement if it passes all checks (see Figure 1), it is best to evaluate the checks in order of increasing computation time. The sooner a check fails the less time is needed to evaluate the position. The checks were presented in order of expected increasing computation time, however actual timings may be case specific.

3.4. Estimating risk

In step 4 the risk of a mistake is estimated based on the identified misplacements. Such estimation is based on a heuristic since the exact probability depends on variables outside of the CAD. Three different risk heuristics are implemented:

- ***Binary risk:*** the risk is zero when there are no misplacements and one when there is at least one misplacement (Equation 1):

  $$risk = \min(\#\mathrm{M}, 1), \tag{1}$$

  where M is the set of misplacements. This is a very strict assessment and forces designs that can only be assembled in one specific way.
- ***Uniform risk:*** each misplacement is considered equally likely to be chosen by the operator. In this case the risk can be expressed as follows (Equation 2),

  $$risk = 1 - \left(\frac{1}{1+\#M}\right). \tag{2}$$

  Notice that the risk is 0 in case of no misplacements and increases gradually towards one for infinite number of misplacement.

4. Examples on industrial use cases

We implemented the misplacement rule with the FreeCAD (The FreeCAD Team, 2023) API, an open source CAD software. This implementation was applied on an academic and multiple industrial use cases. The academic use case (Figure 2 (i)) consists of a small gearbox which is used as a physical

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

demo case. This product has intentional design issues and no specific functional requirements. The second use case is a crusher for iced fruit, which is a subassembly within the Albert's smoothie machine (Figure 2 (ii)). The third use case is an automatic bag dispenser designed by Voxdale (Figure 2 (iii)). The findings during the misplacement rule evaluation of these 3 cases are discussed in the following sections. Note all the evaluations were conducted using a Dell Precision 7560, 11th Gen Intel Core i7 machine.

4.1. The generic use case: a small gearbox

The gearbox is a an academic use case made of 12 assembly parts (base plate, lower and upper housing, 2 drive axles, 4 housing caps, 2 sensors and a single connection box) and 34 fasteners. We manually defined an assembly sequence consisting of 11 assembly operations and evaluated each of them with the misplacement rule. In Table 1 you can find a summary of these evaluations. Total evaluation time took 6 minutes and 56 seconds with 30 seconds for the interface preprocessing of the whole product (on a laptop with Intel core i7 processor). Based on the risk values in Table 1, we can see that 5 operations have 0% risk and don't pose any problems, while 6 operations have a risk of 50% or more. In Figure 3 the visualizations of the misplacements are given. For the connection box the operator can turn the box 180 degrees or place it on the inside. For Housing_Cap_1 and housing_Cap_2 we see that the caps can be placed up-side down. For Housing_Cap_3 the cap can be placed up-side down or placed at an opposite interface.

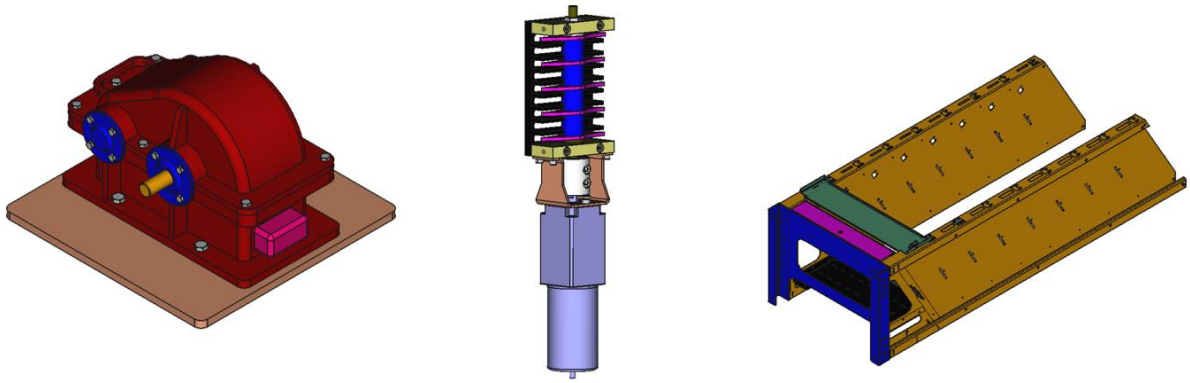

Figure 2. The 3 use cases, from left to right: i) the generic use case of a small gearbox, ii) Albert's smoothie machine crusher, and iii) Voxdale's automatic bag dispenser

Table 1. Results for applying the misplacement rule for the small gearbox

Operation	*Risk*	*m*	*p*	*ctime (s)*	*avg ctime (ms)*
Lower_Housing	0,88	7	160	6,536	41
Connection_Box	0,67	2	4	0,292	73
Lower_Sensor_Mount	0,00	0	0	0,008	8
Upper_Sensor_Mount	0,00	0	8	3,347	418
Small_Axle	0,00	0	0	0,008	8
Large_Axle	0,00	0	0	0,009	9
Upper_Housing	0,67	2	408	8,941	22
Housing_Cap_4	0,00	0	160	102,957	643
Housing_Cap_3	0,75	3	160	123,395	771
Housing_Cap_2	0,50	1	80	48,492	606
Housing_Cap_1	0,50	1	80	81,814	1023

Columns from left to right contain: part added, estimated risk, number of misplacements, number of generated positions, computation time in seconds and average computation time per position in milliseconds.

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

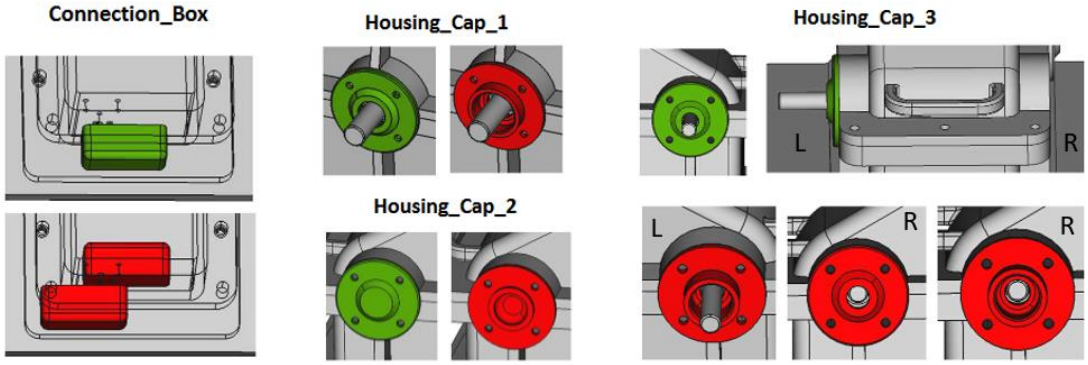

Figure 3. Visualization of the detected misplacements for the small gearbox. Green is the intendent position, red are the misplacements

4.2. Alberts's crusher unit

The second use case is the Alberts crusher unit, a subassembly of their smoothie machine. This subassembly consists of 18 parts and 12 fasteners. For this example the rule was evaluated on a manually defined set of operations that involve fasteners (but not a full sequence). In the first operation only the motor and connector are present (as depicted in Figure 4). The connector is fastened by 4 screws and has 4 double sided holes (2 hole faces) and the motor has 4 matching single sided holes (1 hole face). As a result the connector has 3 potential misplacements for which all holes are aligned. While two misplacements are unlikely to be made by the operator, the third misplacement is more subtle and will lead to issue later on in the assembly sequence. The second operation is the shaft holder assembly operation for which the entire subassembly is present. In this operation the shaft holder has 2 matching faces that match with 1 face of the grill, and 3 misplacements were found. The first misplacement entails the countersunk hole facing the wrong direction (not shown in Figure 4), the second and third misplacements involve the holes at the top ending up facing inside. These misplacements do not only lead to functional requirements issues, but for this particular machine they may lead to a food risk problem as the fruit pieces may get stuck in the holes if the shaft holder is mounted upside down.
In terms of the computational effort for these 2 evaluations: the connector-motor evaluation takes 39.5 seconds to evaluate (a total of 80 positions are generated and processed) and the preprocessing took 6 seconds, the shaft holder evaluation takes 8.324 seconds (only 8 positions are generated and processed) and the preprocessing of the entire subassembly takes 16 seconds.

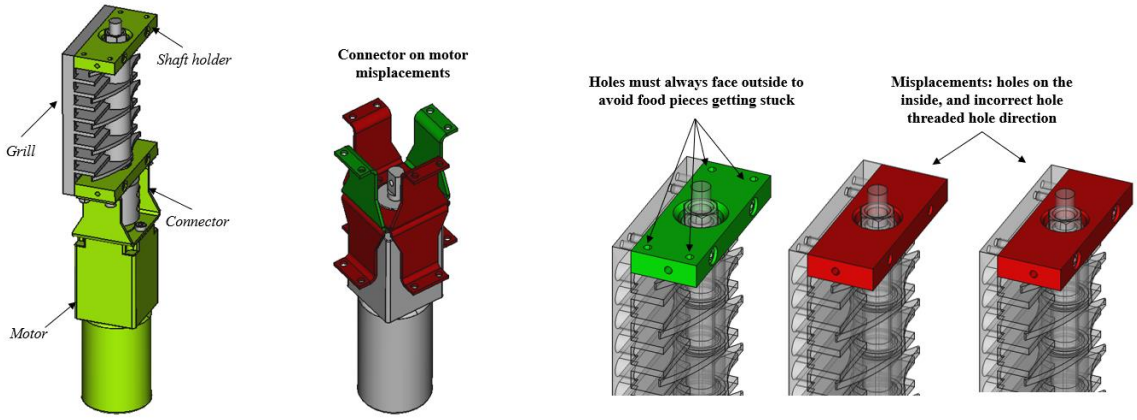

Figure 4. Misplacements on the connector and the shaft holder of the crusher unit subassembly

4.3. Voxdale automatic bag dispenser

With the Voxdale case (Figure 5 - left) we demonstrate how the misplacement rule can be used as both a proactive and reactive approach. Evaluation of the first operation in a typical assembly sequence - the back and right side panel - indicates that the back plate can be misplaced and mounted on the outside

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

part of the side panel (Figure 5 - middle). However, we can see that if both side panels are present in the sequence, this misplacement would be impossible as the requirement on the connectivity check would not be met. By performing this evaluation in the design phase, it can proactively inform the design engineer on the best and most optimal sequence that minimises mistakes. We also looked at the assembly operation involving the base plate and both side panels (Figure 5 - right). Here we found a single true misplacement. In contrast with the first assembly operation, a reconfiguration of the assembly sequence would not necessarily mitigate this problem. However, with this misplacement being detected in the early stages of design, it allows the designer an opportunity to make provisions for the operators on how this wrong assembly may be prevented.

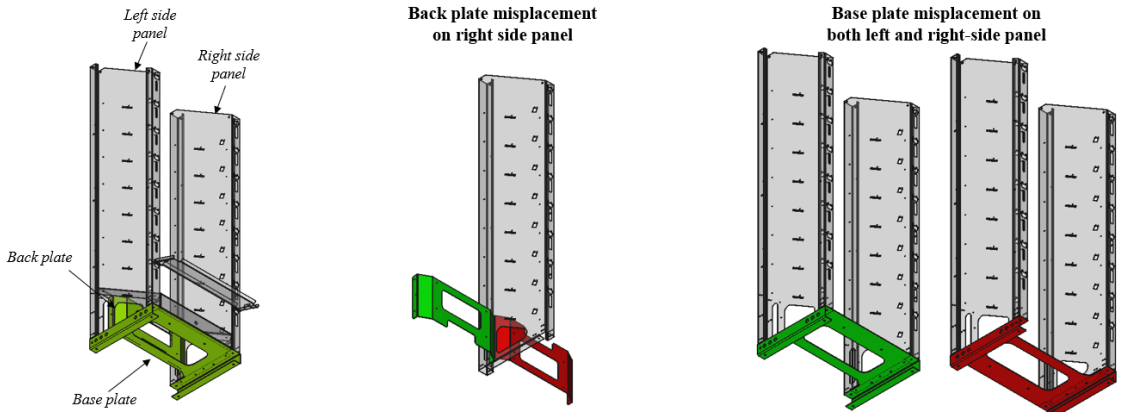

Figure 5. The misplacement found in the evaluation of the Voxdale automatic bag dispenser

5. Conclusion & future work

Given the high product variability and complexity in HMLV manufacturing companies, operator mistakes are common issues during the assembly process. We presented the misplacement rule, an algorithm to automatically detect and evaluate potential misplacements of parts that need to be fastened. We presented an overview of the steps of the algorithm, including the preprocessing of the CAD file, the generation and evaluation of positions to identify misplacements, and the estimation of the risk of mistakes based on the identified misplacements. The functionality of the algorithm was illustrated by applying it on different use cases. The algorithm was able to identify valid misplacements and provide useful input for the designers of these products. The presented algorithm shows that there is still untapped potential for software tools that (semi-)automatically evaluate or apply design rules.
While the algorithm is functional in its current form, we still see possibilities to further extend and refine the algorithm. Some extensions of functionality are:

- Consider other types of operator mistakes such as wrong order, tools or connections.
- Support other connection types such as welding, gluing, interlocking parts, …
- Refine the risk estimation with a data driven weighting function
- Experimentally validate the risk of mistakes in real life.
- Consider accumulation of tolerances in the case of multiple connection interfaces between parts.
- Improve the computational performance by exploiting part symmetry and parallelization

Acknowledgements

This research was partially supported by Flanders Make, the strategic centre for the manufacturing industry in Flanders and the Flemish government. The misplacement rule algorithm was developed in the PACo_SBO project and further refined and validated in the AssistedDFA_ICON project. We thank all collaborators from the Flanders Make CodesignS and ProductionS corelabs that contributed to these developments.

References

Boothroyd, G., Dewhurst, P. and Knight, W.A. (2011), Product Design for Manufacture and Assembly, Third Edition, CRC Press, Boca Raton. https://doi.org/10.1201/9781420089288

https://doi.org/10.1017/pds.2024.56 Published online by Cambridge University Press

Boothroyd Dewhurst, Inc. (2023) DFMA (Version 2023A) [Computer program]. Boothroyd Dewhurst, Inc., East Greenwich, USA.
DFA-Tools Ltd (2017) DFA-Tool (Version 7.1) [Computer program]. DFA-Tools Ltd, Vantaa, Finland.
Eiriksdottir, E. and Catrambone, R. (2011), "Procedural Instructions, Principles, and Examples: How to Structure Instructions for Procedural Tasks to Enhance Performance, Learning, and Transfer", Human Factors, Vol. 53 No. 6, pp. 749–770. https://doi.org/10.1177/0018720811419154
Eskilander, S. (2001), Design for Automatic Assembly – A Method for Product Design: DFA2 [PhD Thesis], Royal Institute of Technology, Stockholm.
Gors, D., Put, J., Vanherle, B., Witters, M. and Luyten, K. (2021), "Semi-automatic extraction of digital work instructions from CAD models", Procedia CIRP, Vol. 97, pp. 39–44. https://doi.org/10.1016/j.procir.2020.05.202
HCL Technologies Ltd (2023) DFMPro [Computer program]. HCL Technologies Ltd, Mumbai, India.
Hollnagel, E. (2012), "Coping with complexity: Past, present and future", Cognition, Technology and Work, Vol. 14 No. 3, pp. 199–205. https://doi.org/10.1007/s10111-011-0202-7
Hu, S.J., Zhu, X., Wang, H. and Koren, Y. (2008), "Product variety and manufacturing complexity in assembly systems and supply chains", CIRP Annals, Vol. 57 No. 1, pp. 45–48. https://doi.org/10.1016/j.cirp.2008.03.138
Ibrahim, A.D., Hussein, H.M.A. and Abdelwahab, S.A. (2021), "Automatic Feature Recognition of Cross Holes in Hollow Cylinders", J. Inst. Eng. India Ser. C, Vol. 102 No. 2, pp. 257–274. https://doi.org/10.1007/s40032-020-00649-5
Lazarevic, M., Mandic, J., Sremcev, N., Vukelic, D. and Debevec, M. (2019), "A systematic literature review of poka-yoke and novel approach to theoretical aspects", Strojniski Vestnik - Journal of Mechanical Engineering, Vol. 65 No. 7–8, pp. 454–467. https://doi.org/10.5545/sv-jme.2019.6056
Leder, R., Stern, H. and Freitag, M. (2022), "Towards design guidance for the digitalisation of work instructions by focusing on technological possibilities and industrial requirements", Procedia CIRP, Vol. 109, pp. 466–471. https://doi.org/10.1016/j.procir.2022.05.279
Li, W., Agrawala, M., Curless, B. and Salesin, D. (2008), "Automated generation of interactive 3D exploded view diagrams", ACM Transactions on Graphics, Vol. 27 No. 3, pp. 1-7. https://doi.org/10.1145/1360612.1360700
Melckenbeeck, I., Burggraeve, S., Van Doninck, B., Vancraen, J. and Rosich, A. (2020), "Optimal assembly sequence based on design for assembly (DFA) rules", Procedia CIRP, Vol. 91, pp. 646–652. https://doi.org/10.1016/j.procir.2020.02.223
Parmentier, D.D., Van Acker, B.B., Detand, J. and Saldien, J. (2020), "Design for assembly meaning: a framework for designers to design products that support operator cognition during the assembly process", Cognition, Technology and Work, Vol. 22 No. 3, pp. 615–632. https://doi.org/10.1007/s10111-019-00588-x
Peruzzini, M., Grandi, F. and Pellicciari, M. (2020), "Exploring the potential of Operator 4.0 interface and monitoring", Computers & Industrial Engineering, Vol. 139, p. 105600. https://doi.org/10.1016/j.cie.2018.12.047
Pimminger, S., Kurschl, W., Panholzer, L. and Schönböck, J. (2021), "Exploring the Learnability of Assembly Tasks Using Digital Work Instructions in a Smart Factory", Procedia CIRP, Vol. 104, pp. 696–701. https://doi.org/10.1016/j.procir.2021.11.117
Slyadnev, S., Malyshev, A., Voevodin, A. and Turlapov, V. (2020), "On the Role of Graph Theory Apparatus in a CAD Modeling Kernel", Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020). https://doi.org/10.51130/graphicon-2020-2-3-70.
Solme AB (2015) AVIX DFX (Version 4.6.5) [Computer program]. Solme AB, Göteburg, Sweden.
Tariki, K., Kiyokawa, T., Nagatani, T., Takamatsu, J. and Ogasawara, T. (2021), "Generating complex assembly sequences from 3D CAD models considering insertion relations", Advanced Robotics, Vol. 35 No. 6, pp. 337–348. https://doi.org/10.1080/01691864.2020.1863258
The FreeCAD Team (2023) FreeCAD (Version 0.21.1) [Computer program].
Widjajanto, S., Purba, H.H. and Jaqin, S.C. (2020), "Novel poka-yoke approaching toward industry-4.0: A literature review", Operational Research in Engineering Sciences: Theory and Applications, Vol. 3 No. 3, pp. 65–83. https://doi.org/10.31181/oresta20303065w
Yu, J. and Zhang, J. (2017), "Hierarchical exploded view generation based on recursive assembly sequence planning", International Journal of Advanced Manufacturing Technology, Vol. 93 No. 1–4, pp. 1207–1228. https://doi.org/10.1007/s00170-017-0414-y
Zogopoulos, V., Geurts, E., Gors, D. and Kauffmann, S. (2022), "Authoring Tool for Automatic Generation of Augmented Reality Instruction Sequence for Manual Operations", Procedia CIRP, Vol. 106, pp. 84–89. https://doi.org/10.1016/j.procir.2022.02.159