UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Visión Electrónica
Más que un estado sólido
https://doi.org/10.14483/issn.2248-4728

VISIÓN ELECTRONICA
A RESEARCH VISION

LIAISON graph generation for assembly tasks based on data extraction: case plates assemblies

Generación **DELGRAFO** *de contactos para procesos de ensamble basado en extracción de datos: caso montajes de placas*

Miguel Angel Villanueva-Portela[1], R*icardo Emiro Ramírez-Heredia*[2]

INFORMACIÓN DEL ARTICULO

Historia del articulo
Enviado: 23/09/2018
Recibido:12/04/2019
Aceptado: 13/05/2019

Keywords
Liaison Graph,
Step-CAD,
Assembly Sequence,
Data Extraction.

Palabras clave:
Grafo de contacto,
Step-CAD,
Secuencia de ensamble,
Extracción de datos.

ABSTRACT:

Welding is one of the most fundamental manufacturing processes, is natural that companies and researchers develop new methods and tools to improve its productivity and flexibility (e.g., Robotic welding). Other manner to do it is the automated generation of the assembly plan for the product. This is a complex task mainly because the size of the configuration space of assembly states [1] and the high dimensionality of the motion planning involved [2]. Researches like [1] and [3] worked to solve the configuration space problem through soft computing techniques, others proposed new methods base on the liaison graph, like the AND/OR graph [4] and a rule based assembly sequence generation system [5]. In the other hand [6] worked in the motion planning problem. Others start to use CAD files to obtain the assembly information as [7-9].
This work focuses in the developing of a method to generate the liaison graph LG in assemblies composed by prismatic plates which will be welded. The assembly geometric information will be gathered from a CAD file, the format ISO 10303 known as STEP [10] was selected. The scope is limited to bodies with parallel positioning and rotation with a step angle of 90 degrees, this scope is sufficient to probe the approach, later extension for step angles between zero and 90 degrees is an implementation issue.

RESUMEN

La soldadura es un proceso de fabricación fundamental, las empresas e investigadores desarrollan nuevos métodos y herramientas para mejorar su productividad y flexibilidad (por ejemplo, soldadura con robots). Una forma de hacerlo es la generación automática del proceso de ensamble de producto. La tarea es compleja, por el gran tamaño del espacio de configuración de estados de ensamble [1] y por la alta dimensionalidad de la planeación de movimiento [2]. Investigadores como [1] y [3] trabajaron para resolver el problema del espacio de configuración con técnicas de computación flexible, otros propusieron nuevos métodos base en el gráfico de enlace (LG), como el gráfico AND / OR [4] y un sistema de generación de secuencias de ensamblaje basado en reglas [5]. Por otro lado [6] trabajaron en el problema de planeación de movimiento. Otros han comenzado a usar archivos CAD para obtener la información de ensamble–[79].
Este trabajo muestra el desarrollo de un método para generar el gráfico de enlace LG en ensamblajes compuestos por placas prismáticas que serán soldadas. La información geométrica del ensamblaje se recopila de un archivo CAD, se seleccionó el formato ISO 10303 STEP [10]. El alcance está limitado a cuerpos con posicionamiento paralelo y giros en ángulos de 90° suficiente para probar el enfoque, la extensión para ángulos de entre cero y 90° será el siguiente problema de implementación.

1 BSc. In Mechanic Engineering, Universidad de Ibagué, Colombia. PhD(c) in Mechanic Engineering and Mechatronic, Universidad Nacional de Colombia, Colombia. E-mail: mavillanuevap@unal.edu.co
2 BSc. In Mechanic Engineering, Universidad Nacional de Colombia, Colombia, and BSc. In Electronic Engineering, Universidad Distrital Francisco José de Caldas, Colombia. MSc In Industrial Automation Engineering, Universidad Nacional de Colombia, Colombia. PhD. In Mechanic Engineering Sciences, Universidade Federal do Rio de Janeiro, Brasil. Current position: Professor Asociated Department of Mechanic Engineering and Mechatronic, Universidad Nacional de Colombia, Colombia. E-mail: reramirezh@unal.edu.co.
Cite this article as: M. A. Villanueva-Portela and R. E. Ramirez-Heredia, "LIAISON graph generation for assembly tasks based on data extraction: case plates assemblies", *Visión Electrónica*, vol. 2, no. 1, Special edition, January-June 2019.

1. Methods

The proposed method to generate the liaison graph consists in three main stages: Information collection, Information transformation and Information analysis. The first stage begins with the STEP-file processing with the aim of take the geometric information of the assembly's components. The second stage is where all the geometric data is transformed in order to get the correct position and orientation of the different elements of the assembly and the last stage is in charge of the data analysis to determine which assembly elements are in contact.

1.1. Information Collection

This stage go ahead with the processing of the STEP-file. The structure inside of this CAD file consists of a sequence of lines of characters which represent "*shells*, *planes*, *edges*" and "*vertices*". In the Figure 1 an example of that structure can be seen. Through reading and processing of each line of the STEP-file related to geometric data the information is collected. The topology of the processed lines with geometric information is displayed in Figure 2. The implementation code was made with the language C++ and the libraries STL, Boost and Eigen. To verify the collected information and its subsequent transformation a viewer window was implemented and embedded in the GUI developed in (Villanueva Portela and Ramirez 2015). The PCL library was used to build of the aforementioned viewer.

```
#641=CLOSED_SHELL('',(#646,#647,#648,#649,#650,#651));
#646=ADVANCED_FACE('',(#714),#680,.F.);
#680=PLANE('',#1191);
#714=FACE_OUTER_BOUND('',#748,.T.);
#748=EDGE_LOOP('',(#782,#783,#784,#785));
#782=ORIENTED_EDGE('',*,*,#974,.T.);
#926=VERTEX_POINT('',#1384);
#974=EDGE_CURVE('',#926,#927,#1046,.T.);
#1046=LINE('',#1383,#1118);
#1118=VECTOR('',#1232,1.);
#1190=AXIS2_PLACEMENT_3D('',#1382,#1230,#1231);
#1230=DIRECTION('',(0.,0.,1.));
#1382=CARTESIAN_POINT('',(0.,0.,0.));
```

Figure 1. Sequence of string lines that conform the geometric structure inside the ISO 10303 file format. Source: own.

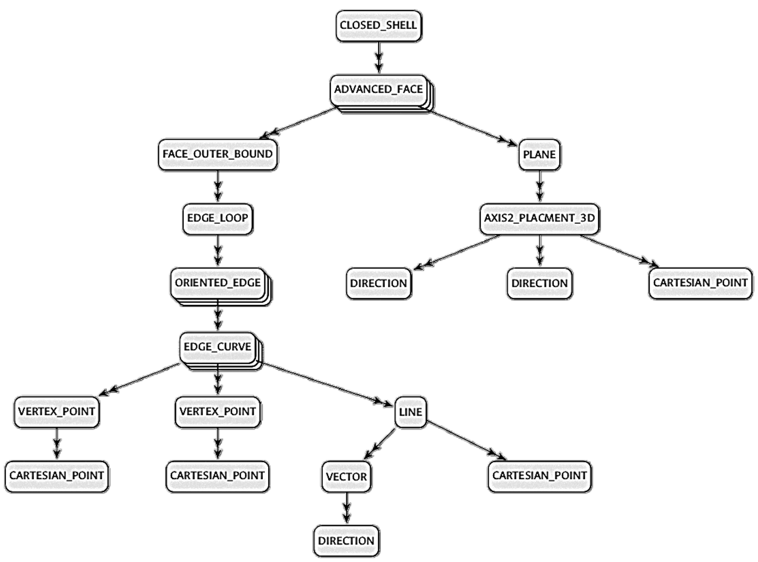


Figure 2. Sequence of string lines that conform the geometric structure inside the STEP-file. Source: own.

The first one stage is composed by the following steps:

1. The geometric information gathering process starts with the analysis of each line of the STEP file. The aim of this initial analysis is to check if every one of the instruction lines fill up more than one line in the STEP file or not. If instruction line takes more than one line, it is reorganized in one line.
2. The information included in the lines with the label *"CLOSE_SHELL"* is obtained. It is identified which is the line number for every plane belonging to the external surface of the plate's bodies in the lines with the label *"ADVANCED_FACE"*.
3. It is obtained the information included in the lines with the label *"ADVANCED_FACE"*, those are the line number for the lines under the label *"PLANE"* and *"FACE_OUTER_BOUND"*.

4. From the lines with the *"PLANE"* label, the line number of *"AXIS2_PLACEMENT_3D"* lines is gathered.
5. From the lines with *"FACE_OUTER_BOUND"* label the line number of *"EDGE_LOOP"* lines is gathered.
6. From the lines with *"AXIS2_PLACEMENT_3D"* label, the line numbers of the lines containing the position of one Cartesian point and two unit vectors one for the directions X and Z in each plane is obtained.
7. From the lines with *"EDGE_LOOP"* label, the information of the points that conforms the perimeter in every plane is collected. This goes through the lines with labels: *"ORIENTED_EDGE"* and *"EDGE_CURVE"*. Also the direction of each edge is obtained.

1.2. Information Transformation

Until this point, the geometric information of the elements of the assembly is related to the local coordinate system of every part. In this stage the correct position and orientation of each element in the assembly is applied to them.

The following steps make up the stage two:

1. With the information obtained in the **Step 6** of the stage one, the transformation matrices which relate each component of the assembly with the local coordinate system in the STEP file are built.

2. All the geometric information (e.g., vertices, edges and planes) collected is transformed implementing the transformation matrices calculated as the case may be. The general procedure is shown in

$$V_{vertices}^{org\_assembly} = T_{plate_k}^{org\_assembly} V_{vertices}^{plate_k} \quad (2)$$

1.3. Information Analysis

Now the liaison graph can be built. The process starts classifying the perpendicular planes of every assembly component to the axes $\{X, Y, Z\}$ These axes belong to the global coordinate system within the assembly. Two main groups can be an organizer base on the number of perpendicular planes $N_{PP_{x,y,z}}$ to each axis. It is shown in the Table 1.

Group	Description
1	$N_{PP_{(x,y,z)}}^{m} = \sum_{m=1}^{PP} p_{m_{(x,y,z)}} = 2$, where: $\{p_{m_{yz}} \perp X\} \vee \{p_{m_{xz}} \perp Y\} \vee \{p_{m_{xy}} \perp Z\}$
2	$N_{PP_{(x,y,z)}}^{m} = \sum_{m=1}^{PP} p_{m_{(x,y,z)}} > 2$, where: $\{p_{m_{yz}} \perp X\} \vee \{p_{m_{xz}} \perp Y\} \vee \{p_{m_{xy}} \perp Z\}$

Table 1. Classification of general assembly components. Source: own.

For assemblies in the Group 1, the following analysis was used: to each couple (A, B) of components to be analyzed, the highest value of the plane$B$ and vice versa for each axis$\{X, Y, Z\}$ as is shown in (1).Figure 3, shows an example of one assembly from the first group.

$$P_{A_{max}} == P_{B_{min}} \; or \; P_{A_{min}} == P_{B_{max}} \quad (1)$$

The comparison process continues until all the components are checked. As an outcome, the neighbor's components of the component are listed.

The following step is the evaluation of the founded matches. This is performed by doing a Boolean intersection operation between the planes. With this evaluation, false contacts can be eliminated.

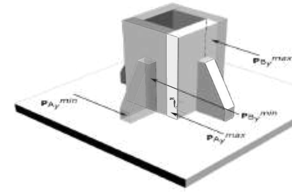


Figure 3. False contact example in assemblies from the Group 1. Source: own.

In the Figure 3 an example of this false contact is displayed. As can be observed, applying the relation of the equation 1, there should be a contact between the analyzed components (A, B), respect the red $\boldsymbol{P}_{B_y}^{min}$ and $\boldsymbol{P}_{A_y}^{max}$ planes. But, there is not. Through the contact evaluation this false contact can be eliminated because the intersection between those two areas is zero.

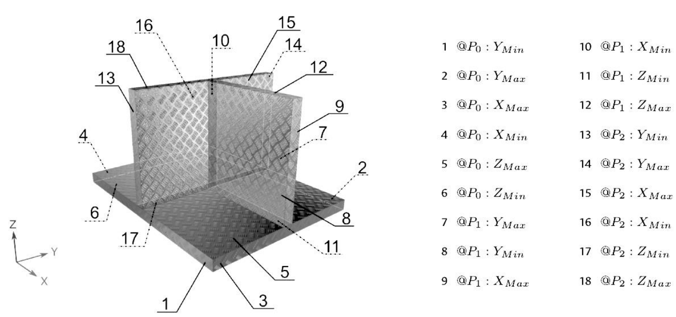


Figure 4. Plane designation for an assembly model in the Group 2. Source: own.

For assemblies in the **Group 2**, Figure 4, the next analysis was proposed: for every couple (A, B) of components being analyzed, each one of the perpendicular planes to the axes $\{X, Y, Z\}$, in *"A"* are compared with all the parallel planes to them in *"B"*.

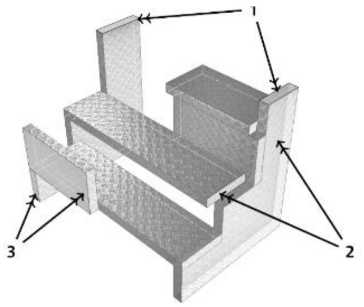


Figure 5. False contact cases could be present in assemblies of the Group 2. Case 1: There is a co-planar plane. But, the plates do not have a real contact between them. Case 2: There is a co-planar plane. But, the contact between the plates is through an edge. Case 3: There is a co-planar plane. But, the contact between the plates is made by normal planes to the co-planar ones. Source: own.

If a co-planar plane is found, it is saved into a preliminary contact matrix $M_{\_x}$,$M_{\_y}$ or $M_{\_z}$.according to the case, this operation last until all the components are checked. Afterwards, all the contact relation are evaluated with a Boolean intersection operation between the planes in contact. Thus, the false contacts are eliminated. In the Figure 5 more false contact cases are outlined.

The evaluation of the contact relation has the following general steps:

1. Each set of points are organized from de lowest to the maximum value,
2. The repeated elements are eliminated in both sets,
3. With the updated sets two polygons are created,
4. One Boolean operation of intersection is performed between the polygons,
5. The contact matrices are created.

The Algorithm 1 show, Figure 6, in further detail the process to obtain the matrices matrix $M_{\_x}$, $M_{\_y}$ and $M_{\_z}$ and Then, the liaison graph is derived from them.

Algorithm 1 Contact finder Algorithm

```
procedure CONTACTFINDER(va1, var2, var3, var4)
    for all Assembly_Components do
        PlaneN(i)^x ← Normal planes to X
        PlaneN(i)^y ← Normal planes to Y
        PlaneN(i)^z ← Normal planes to Z

    if PlaneN(i)^X ∨ PlaneN(i)^Y ∨ PlaneN(i)^Z = 2, then
                                        ▷ For assemblies in the Group 1
        for i ← 1 to Number of components do
            for j ← 1 to Number of components do
                if P_i^{X_Min} == P_j^{X_Max} || P_j^{X_Min} == P_i^{X_Max} then
                    Part[j] ← j
                if P_i^{Y_Min} == P_j^{Y_Max} || P_j^{Y_Min} == P_i^{Y_Max} then
                    Part[j] ← j
                if P_i^{Z_Min} == P_j^{Z_Max} || P_j^{Z_Min} == P_i^{Z_Max} then
                    Part[j] ← j
                Evaluate the contact
            Component_Contacts[i] ← Part
        return Component_Contacts

    else
            ▷ For assemblies in the Group 2. Build of the preliminary contact Matrix Pre_M_Axes
        for all Axes do
            Axes = {X, Y, Z}
            for r ← 1 to Number of components do
                for w ← 1 to length of PlaneN[r]^Axes do

                    for q ← 1 to Number of components do
                        for s ← 1 to length of PlaneN[q]^Axes do
                            if r! = q ∧ PlaneN[r][w]^Axes == PlaneN[q][s]^Axes then
                                Pre_M_Axes[r][q] ← PlaneN[q][s]^Axes
                                Pre_M_Axes[q][r] ← PlaneN[q][s]^Axes
            return Pre_M_Axes
                                        ▷ Analysis of the contact in Pre_M_Axes
            for fi ← 1 to All rows in Pre_M_Axes do
                for co ← 1 to All columns in Pre_M_Axes do
                    Get the plane points of the components fi and co
                    Organize the points in ascend order for each component
                    Get rid of repeated points in each set
                    if The number of the remaining points is == 4 then
                        Organize the points in counterclockwise order
                        Create one polygon per set
                    else
                        Create one polygon per set
                        Use the Boost "correct" function on the polygon
                    Perform the intersection Boolean operation between the polygons
                    if Polygon_1 ∩ Polygon_2 == True then
                        M_Axes[fi][co] ← co
        return M_x, M_y and M_z
```

Figure 6. Contact finder algorithm. Source: own.

2. Results

In order to evaluate the proposed algorithm five assemblies was implemented. In the Figure 7 the tested assemblies are shown as well their correspondent component number.

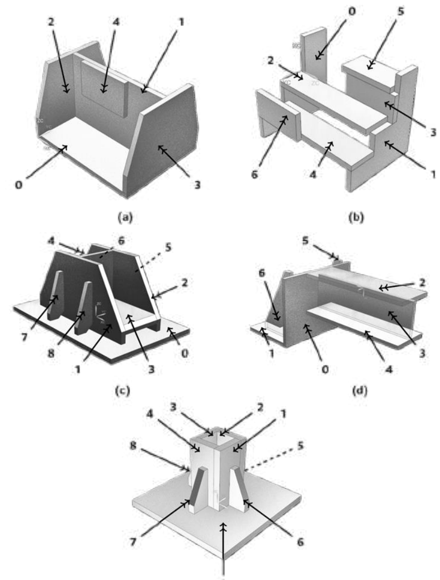


Figure 7 . Test plate Assemblies. The dashed lines in (c) and (e) point to the hidden components. Source: own.

The Figures 8 to 12 show the geometric information gathered and transformed from every STEP file as is viewed in the GUI developed. Also, the liaison graph obtained is plotted.

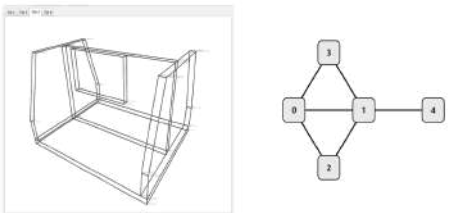


Figure 9. Test assembly (b). Source: own.

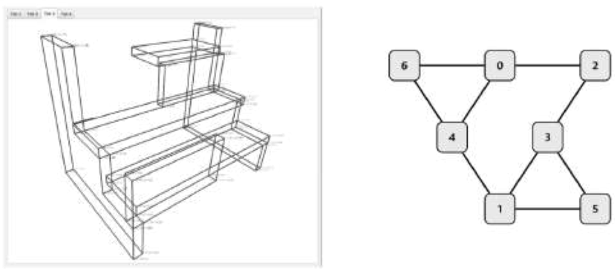


Figure 10. Test assembly ©. Source: own.

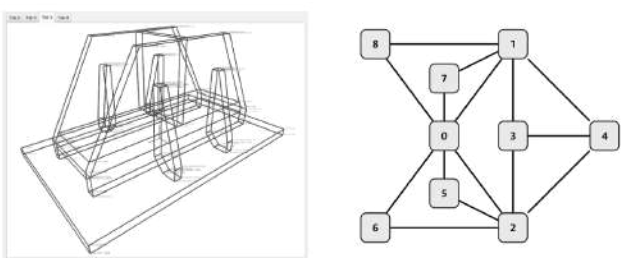
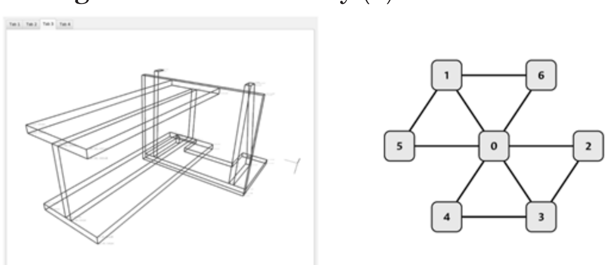


Figure 11. Test assembly (d). Source: own.



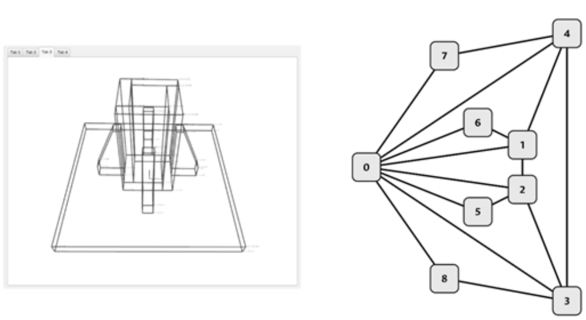


Figure 12. Test assembly (e). Source: own.

3. Conclusions

Despite of the simplicity of the proposed method it is capable of generate the liaison graph avoiding the false contact in certain coplanar planes and streamline the search of the assembly sequence by search algorithms.
An important step is the building and application of the transformation matrices of each assembly's component, as is explained in the section **1.2. Information transformation**, these matrices are built with the information which traceability have been checked through the labels along the right branch of the graph in the Figure 2 and must correctly applied to vertices, flat edges and each component of the assembly that is collected along the left branch of the same graph illustrated in Figure 2. Therefore, a proper traceability control must be carried out for all collected information to avoid fake contact results in the Information analysis stage, due a wrong assembly representation.
The next step is the widening of the range of orientation for the planes to analyze and the addition of curved surfaces. Aiming to work with more complex assemblies and geometries, for example the type of assemblies analyzed in researches as –[9, 1115]. Nevertheless, this stage is beyond the scope of the project.

Acknowledgements

The first author expresses thanks to Colciencias for the Ph.D. course scholarship and both authors express thanks the Curricular Area of Mechanical and Mechatronic Engineering of the Universidad Nacional de Colombia for the support in the present research.

References

[1] M. F. F. Rashid, W. Hutabarat, and A. Tiwari, "A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches", *Int J Adv Manuf Technol*, vol. 59, no. 1-4, pp. 335–349, 2012. https://doi.org/10.1007/s00170-011-3499-8

[2] F. W. Heger, IEEE International Conference on Robotics and Automation, 2008: ICRA 2008; 19 - 23 May 2008, [Pasadena, CA]. Piscataway, NJ: IEEE, 2008.

[3] C. Hui, L. Yuan, and Z. Kai-fu, "Efficient method of assembly sequence planning based on GAAA and optimizing by assembly path feedback for complex product", *Int J Adv Manuf Technol*, vol. 42, no. 11-12, pp. 1187–1204, 2009.

https://doi.org/10.1007/s00170-008-1661-8

[4] L. S. Homem de Mello and A. C. Sanderson, "AND/OR graph representation of assembly plans", *IEEE Trans. Robot. Automat.*, vol. 6, no. 2, pp. 188–199, 1990. https://doi.org/10.1109/70.54734

[5] M.-C. Lin, Y.-Y. Tai, M.-S. Chen, and C. Alec Chang, "A Rule Based Assembly Sequence Generation Method for Product Design", *Concurrent Engineering*, vol. 15, no. 3, pp. 291–308, 2016. https://doi.org/10.1177/1063293X07083084

[6] A. I. Kokkinaki and K. P. Valavanis, "On the comparison of AI and DAI based planning techniques for automated manufacturing systems", *J Intell Robot Syst*, vol. 13, no. 3, pp. 201–245, 1995. https://doi.org/10.1007/BF01424008

[7] Chunxia Pan, "Integrating CAD files and automatic assembly sequence planning," Iowa State University, 2005.

[8] M. P. Bhandarkar and R. Nagi, "STEP-based feature extraction from STEP geometry for Agile Manufacturing", *Computers in Industry*, vol. 41, no. 1, pp. 3–24, 2000. https://doi.org/10.1016/S0166-3615(99)00040-8

[9] Z. Han, R. Mo, H. Yang, and L. Hao, "CAD assembly model retrieval based on multi-source semantics information and weighted bipartite graph", *Computers in Industry*, vol. 96, pp. 54–65, 2018. https://doi.org/10.1016/j.compind.2018.01.003

[10] SCRA, STEP APPLICATION HANDBOOK ISO 10303 VERSION 3, 2006.

[11] H. Cao, R. Mo, N. Wan, and Q. Deng, "An intelligent method to generate liaison graphs for truss structures", *Journal of Engineering Manufacture*, vol. 232, no. 5, pp. 889–898, 2016. https://doi.org/10.1177/0954405416654187

[12] L.-M. Ou and X. Xu, "Relationship matrix based automatic assembly sequence generation from a CAD model", *Computer-Aided Design*, vol. 45, no. 7, pp. 1053–1067, 2013. https://doi.org/10.1016/j.cad.2013.04.002

[13] Y. Xing and Y. Wang, "Design and optimisation of assembly technique for auto-body components", *International Journal of Production Research*, vol. 51, no. 22, pp. 6515–6533, 2013. https://doi.org/10.1080/00207543.2013.776189

[14] R. B. Hadj, I. Belhadj, M. Trigui, and N. Aifaoui, "Assembly sequences plan generation using features simplification", *Advances in Engineering Software*, vol. 119, pp. 1–11, 2018. https://doi.org/10.1016/j.advengsoft.2018.01.008

[15] S. Sierla, V. Kyrki, P. Aarnio, and V. Vyatkin, "Automatic assembly planning based on digital product descriptions", *Computers in Industry*, vol. 97, pp. 34–46, 2018. https://doi.org/10.1016/j.compind.2018.01.013