UNIVERSITÁ DEGLI STUDI DI GENOVA

Department of Mechanical, Energy, Management and Transportation Engineering

Curriculum Mechanics, Measurements and Materials (XXXV cycle)
a.y. 2021-2022

Ph.D. Thesis

Extracting, managing, and exploiting the semantics of mechanical CAD models in assembly tasks

Supervisors:
Dr. Franca Giannini
Prof. Roberto Raffaeli
Prof. Giovanni Berselli
Co-supervisor:
Dr. Marina Monti

Ph.D. Candidate:
Brigida Bonino

Abstract

The manufacturing of mechanical products is increasingly assisted by technologies that exploit the CAD model of the final assembly to address complex tasks in an automated and simplified way, to reduce development time and costs. However, it is proven that industrial CAD models are heterogeneous objects, involving different design conventions, providing geometric data on parts but often lacking explicit semantic information on their functionalities. As a consequence, existing approaches are mainly mathematics-based or need expert intervention to interpret assembly components, and this is limiting.

The work presented in the thesis is placed in this context and aims at automatically extracting and leveraging in industrial applications high-level semantic information from B-rep models of mechanical products in standard format (e.g. STEP). This makes possible the development of promising knowledge intensive processes that take into account the engineering meaning of the parts and their relationships.

The guiding idea is to define a rule-based approach that matches the shape features, the dimensional relations, and the mounting schemes strictly governing real mechanical assemblies with the geometric and topological properties that can be retrieved in CAD models of assemblies. More in practice, a standalone system is implemented which carries out two distinct operations, namely the data extraction and the data exploitation. The first involves all the steps necessary to process and analyze the geometric objects representing the parts of the assembly to infer their engineering meaning. It returns an enriched product model representation based on a new data structure, denoted as liaison, containing all the extracted information. The new product model representation, then, stands at the basis of the data exploitation phase, where assembly tasks, such as subassembly identification, assembly planning, and design for assembly, are addressed in a more effective way.

Contents

List of Figures

List of Tables

List of Acronyms

AI Artificial Intelligence

API Application Programming Interface

AP Assembly Planning

AR Augmented Reality

ASME American Society of Mechanical Engineers

ASP Assembly Sequence Planning

B-rep Boundary Representation

BOM Bill Of Materials

CAD Computer Aided Design

CAE Computer Aided Engineering

CAM Computer Aided Manufacturing

CAx Computer Aided Technologies

DFA Design for Assembly

DFMA Design for Manufacturing and Assembly

DIN Deutsches Institut für Normung

EN European Standards

FEA Finite Element Analysis

IGES Initial Graphics Exchange Specification

ISO International Organization for Standardization

NURBS Non-Uniform Rational Basis Spline

PR Precision-Recall

SI Subassembly Identification

STEP Standard for the Exchange of Product model data

STL Standard Tessellation Language

UNI Italian National Unification

VR Virtual Reality

Introduction

The advent of Industry 4.0 and the increasing potential of new technologies and machines largely affected industrial manufacturing. They brought to the concept of smart factory, where people, machines, and products are meant to communicate with each other throughout the entire life cycle of a product allowing improved storage, exchange, and leveraging of the associated information [41, 74].

Mechanical engineering is one of the sectors that nowadays broadly takes advantage of digitalization to make the production processes and supply chains more efficient, achieving progress in productivity and huge savings in material and energy, as well as facilitating the human-robot collaboration [51, 57, 117]. A wide range of mechanical production phases is impacted by this revolution, such as design, machining, and assembling, along with educational, simulation and reuse processes. The key technologies and the systems involved are various: from Computer Aided Technologies (CAx) to Virtual/Augmented Reality (VR/AR), from additive manufacturing to autonomous robots and artificial intelligence systems. Also, research is very active in the study and development of approaches to algorithmically address the most onerous and error-prone tasks, from machining and assembly sequence planning (ASP) to human-robot collaboration, from product retrieval for model and related knowledge reuse to production and life cycle costs estimation and optimization.
In light of these premises, the availability of a consistent and expressive digital model of the final product is the necessary condition that enables both a beneficial use and integration of the different technologies and the achievement of optimal and feasible results from the production planning.

Already patented methods to digitally represent the 3D models of real products are commonly used and provided through commercial Computer Aided Design

(CAD) software, as well as common file formats for the storage are established (e.g. STEP, IGES, STL, etc.). However, these mainly ensure the shape product description. Conversely, the inclusion and availability in CAD models of semantic information, i.e. all the non-geometric information, such as category membership, technological data, kinetic and kinematic properties, and functionality [53], is still an open issue, along with its managing and exploitation in the different production tasks.

Even if first a CAD model processing phase, where the features interesting for the specific process/analysis and their relations are recognized, is usually performed, the main weakness is that the intrinsic engineering meaning of the assembly's components is neglected [98], or else human intervention is often required for their inferencing. This is because usually details associated with parts' functionalities are implicit. They may be included as annotations in the CAD models, but these attributes are not rigorous and unique since they depend on the designer's choice, thus it results difficult and time-consuming to interpret them [96]. Moreover, this information is often subjected to being discarded due to importing and exporting of parts and assemblies between software and to standard exchange formats [32, 92] and thus a manual extraction is needed.

As a consequence, the development of tools for the automatic extraction from CAD models of engineering usable information deserves to be investigated and can be exploited to improve the different assembly tasks.

This thesis tackles the problem of the semantic interpretation of industrial CAD models of mechanical products under various aspects, from the definition of an enriched model which involves shape and assembly process data automatically extracted, to their possible leveraging and application in some of the widespread assembly tasks. In the following, the main challenges encountered and the proposed contributions for facing them are briefly pointed out.

Challenges and proposals

The work presented in this thesis is part of an industrial Ph.D. project carried out in partnership with the Italian engineering software development company Hyperlean[1], which provides innovative software platforms integrated into 3D CAD and management systems to support product configuration, design and cost estimation. Consequently, the main objective is to address research problems in mechanical assembly tasks, such as meaningful subassembly identification (SI) and ASP, and

[1] https://hyperlean.eu

provide compelling results from the scientific standpoint. At the same time, the outcomes should actually cover needs at the industrial level and be of benefit to the company.
The algorithms and the approaches that will be discussed are, therefore, developed as prototype modules of the company's software to exploit their tools and the already existing functionalities, allowing easy integration of the work carried out. Also, most of the data and the test cases used are supplied by the company.

According to these premises, the first challenge emerges, which is the handling of industrial CAD models. In fact, approaches for addressing assembly tasks existing in the literature typically assume too abstract hypotheses and almost always use ideal CAD models, i.e. thorough and generally made of few parts as test cases. Taking as input CAD models supplied by industrial companies actually used for manufacturing scopes outlines different issues right from the model processing and data extraction phase. Thus, the first part of the work focuses on deeply analyzing the characteristics of industrial CAD models. Inconsistencies, common design strategies, and major issues are detected and methodologies to automatically address and overcome them, rather than ignoring or manually fixing them, are provided. This led to the need of examining every component of the assembly and their relations in order to assign them unambiguous meaning and a functional interpretation employing extended engineering-driven analysis. It consists of different steps, some implemented using and adapting existing techniques (e.g. feature recognition and contacts detection), and some others implemented from scratch by the author. For the latter case, a considerable contribution of the thesis is the coupling of shape and context-based recognition of standard mechanical parts.

The second challenge arises at this point, namely the definition of a suitable representation that can contain and intuitively provide the semantic data extracted. For this purpose, the *liaison* data structure is defined and implemented to totally express the relation between two mating parts of the assembly, providing high-level semantic information concerning multiple aspects, from the geometry of the contact to the assembly process features.

Finally, the third challenge stands in taking into account that the input model truly reflects a real mechanical assembly when solving complex assembly tasks starting from CAD models, thus improving current methodologies concerning the engineering feasibility of the results. Approaches are studied and developed to address subassembly identification, assembly sequence planning, and design for assembly and manufacturing tasks in a pioneering and promising way, i.e. taking as a starting point the semantic enriched model and exploiting the liaisons properties.

The guiding principle of the thesis, according to which the different challenges are faced, is the strict integration of engineering and design knowledge with geometric and topological analysis of the CAD model. That is, much of the work exploits a novel rule-based approach that converts engineering features and schemes into geometric requirements that have to be met by the CAD models of the assembly's parts. In this way, high-level and more realistic operations can be performed only on the base of low-level data.

Thesis structure

Six chapters covering the thesis's subject matter are presented in the following order:

- **Chapter 1** is an introductory chapter providing background information on mechanical products characteristics and habits employed in their digital representations. General key concepts are fixed and definitions are given that are fundamental to ensure the understanding of the manuscript. The problems arising when using CAD models in industrial applications are introduced, and the need for a semantic interpretation and a CAD model enrichment is pointed out as one of the purposes of the thesis.

- **Chapter 2** aims to contextualize and enforce the validity of the proposed system through a comprehensive literature review in the areas of research covered by the thesis. In particular, works that support the need for the semantic enrichment of CAD models and the definition of new product representation structures are reported. Successively, the analysis specializes first in part recognition methodologies and then in assembly planning techniques, focusing on the common lack of automation and the rare extraction and use of engineering meaningful data.

- **Chapter 3** outlines the proposed approach, which is based on the promising idea of computationally analyzing CAD industrial models and solving the assembly tasks by replicating the reasoning followed by experts in reality. Two implementation phases are distinguished, namely the *data extraction* and the *data exploitation*, with the *liaison* data structure serving as their key connecting element. The organization of each of the two phases in separate modules is briefly reported. Finally, the theoretical definition of liaison is given.

- **Chapter 4** and **Chapter 5** deepen the description of the frameworks developed respectively for the execution of the *data extraction* and *data exploitation* phases. Each module is singularly illustrated. The main goal is stated along with the input used and the output attained. Results and application examples are presented after the description of the implementation specifics.

- A final discussion on the weak aspects of the modules that have to be improved, possible extensions of the approaches, and future perspectives are provided in **Chapter 6**.

Publications

Journal

- **Shape and context-based recognition of standard mechanical parts in CAD models**
  Brigida Bonino, Franca Giannini, Marina Monti, and Roberto Raffaeli
  Computer-Aided Design, Vol. 155, Pages 103438, 2023

- **Automatic Assembly Precedence Detection in Axisymmetric Products**
  Brigida Bonino, Franca Giannini, Marina Monti, and Roberto Raffaeli
  Computer-Aided Design and Applications, 2023
  In printing

- **Enhancing product semantics understanding through automatic part type recognition in CAD assembly models**
  Brigida Bonino, Franca Giannini, Marina Monti, and Roberto Raffaeli
  Computer-Aided Design and Applications, Vol. 19(5), Pages 896-912, 2022

- **Review on the leveraging of design information in 3D CAD models for subassemblies identification**
  Brigida Bonino, Franca Giannini, Marina Monti, and Roberto Raffaeli
  Computer-Aided Design and Applications, Vol. 18(6), Pages 1247-1264, 2021

Conference

- **Automatic Assembly Sequence Planning for Axisymmetric Products**
  Brigida Bonino, Franca Giannini, Marina Monti, and Roberto Raffaeli

Proceeding of CAD'22 conference, Beijing, China, July 11-13, 2022, Pages 334-338

- **Geometric analysis of product CAD models to support design for assembly**
  Brigida Bonino, Franca Giannini, Marina Monti, Roberto Raffaeli, and Giovanni Berselli
  Advances on Mechanics, Design Engineering and Manufacturing IV, JCM 2022, Lecture Notes in Mechanical Engineering, Pages 698-710

- **Automatic parts classification to enhance the semantics of the CAD assembly models**
  Brigida Bonino, Franca Giannini, Marina Monti, and Roberto Raffaeli
  Proceeding of CAD'21 conference, Barcelona, Spain, July 5-7, 2021, Pages 139-143

- **A heuristic approach to detect CAD assembly clusters**
  Brigida Bonino, Roberto Raffaeli, Marina Monti, and Franca Giannini
  Procedia CIRP 2021, Vol. 100(6), Pages 463-468

- **Identification of subassemblies by leveraging design information in 3D models**
  Brigida Bonino, Franca Giannini, Marina Monti, and Roberto Raffaeli
  Proceeding of CAD'20 conference, Barcelona, Spain, July 6-8, 2020, Pages 338-342

1 General concepts on mechanical assemblies

This chapter presents background information on mechanical products and the fundamental engineering knowledge at the basis of the addressed research issues and proposed solutions. The widespread approach that uses CAD models to design, inspect, and evaluate the products before their actual fabrication is addressed. The common practices and conventions for digitally representing mechanical parts are identified, along with the issues that arise from them when developing automatic model analysis processes. Some hints to overcome the problems are suggested, underlying the need for a comprehensive semantic analysis of an input CAD model before tackling increasingly complex tasks.

1.1 Generalities on mechanical assemblies

Mechanical products are assemblies given by the combination and aggregation of parts whose number can vary from a few units to hundreds depending on the product's complexity. The choice of the parts, the order in which they are assembled, and how they are positioned are not arbitrary, rather all these factors are carefully analyzed, studied, and established by experts. These types of products, in fact, are in general moving objects that can include several kinds of mechanisms. The mechanisms perform specific functions, that can be both basic or more cumbersome, such as motion transmission, reducing friction, carrying loads, or moving and lifting other components. In any case, reciprocal movement between the parts of the assembly is required. They, thus, have to be precisely manufactured and installed to avoid problematic and dangerous stuck or missing contacts. Also, mechanical parts can be made from several different types of materials from high-grade steel to various forms of plastic. In general, the material used depends on the final function of the product and its setting, the temperatures that can be reached when in action, along with the role of the specific part in the assembly and its importance.
Therefore, to allow correct working and optimal performance, the principles of physics and material science, as well as core concepts including mechanics and kinematics, must be taken into account to properly design and fabricate mechanical assemblies.

Before moving on, to make clearer the discussion and avoid wrong interpretations, the definitions adopted in this thesis of the elements constituting a mechanical assembly are provided:

Part

The term part refers to the most elementary object that stands at the basis of the structure of a mechanical product. Namely, all the elements that can not be further disassembled into simpler ones are parts, and they are the building blocks of mechanical assemblies.

Subassembly

The term subassembly refers to another element included in the mechanical assembly's structure which is more complex than the single parts defined above. In particular, a subassembly is an assembled mechanical unit made up of a combination of different parts, which can be independently mounted and can have its own meaning and function. Moreover, in the case all the parts of a subassembly move steadily with each other, i.e. by translating and rotating the subassembly the parts maintain their reciprocal positions without being disconnected, the subassembly is also defined as stable. In summary, it can be stated that a subassembly reflects the properties of an assembly, but it actually is a sub-part of an assembly in the sense that it is designed to fit with other subassemblies and parts in a finished manufactured product.

Component

The concept of component is more comprehensive than that of part and subassembly and covers different situations. It can be referred to either a single-part or a multi-part object.
In the first case, component is synonymous with part. In the latter case, a component is a group of parts mounted together and arranged in a predetermined manner to constitute a stable object having a specific engineering meaning. That is, a component can be regarded as a subassembly, but its distinctive feature is that it is usually considered within the product as a single part due to its function and structure. Typical examples of multi-part components are bearings, seals, actuators, and pulleys.

According to the above definitions, it has to be specified that some parts have a semantic meaning expressing their usage by themselves. This is the case of connecting elements and spacers, along with gears, belts, or springs. Some other parts acquire their importance and fulfill their functionality if combined with other parts, hence when considered within a multi-part component, e.g. the bearings. Others, instead, are less meaningful from the functional point of view, but at the same time crucial since they constitute the supporting structure of the assembly, thus they express their role when seen in the whole of a subassembly.
Consequently, to create a feasible and optimized mounting sequence, it is essential to identify the engineering meaningful components in the assembly and to know its organization in subassemblies that can be mounted independently and in what order they can then be positioned in the final product. This implies the attempt to reduce the number of operations and to allow easy placement of the various components, even planning some parallel production lines on the basis of which independent subassemblies are first mounted and then easily joined together. In this process, a further determining factor is the choice of the most advantageous and

reliable assembly technique according to the characteristics of the components and their accessibility.

1.1.1 Types of assembly techniques

With regard to assembly techniques, two main macro classes can be identified based on whether additional parts are employed to connect two components or not. These will be referred as mounting by connection elements and absence of connection elements, namely:

- **Mounting by connection elements** is one of the most common assembly techniques used in mechanical engineering. It consists in the application of specific parts, i.e. the fasteners, as physical connections to join two or more parts. The class of fasteners includes a wide variety of different categories and subcategories. The first main distinction can be done between threaded and not threaded fasteners. Screws, bolts, and studs can be listed among the threaded fasteners, while rivets are among the not threaded (Fig. 1.1).

(a) Screw.

(b) Bolt.

(c) Stud.

(d) Rivet.

Figure 1.1: Examples of threaded (a, b, c) and not threaded (d) fasteners.

The two types of fasteners work similarly to each other, in the sense that they both bound two or more parts of the assembly being inserted in preformed holes or creating holes on the parts during the installation. In general, the use of threaded fasteners embodies a stable but non-permanent joint between the parts. Thus, this type of assembly is primarily used when there is a need for or significant benefit to being able to dismantle the assembly or take individual parts out of the assembly throughout the life cycle of the product. On the contrary, not threaded fasteners are generally permanent connections, that are deformed during the placement and thus their removal is not expected since it would imply breaking or damaging the parts involved. A weakness associated with fasteners is that they are prone to failure under vibration because of the loosening of joining elements. Also the strength of joint is not very high and thus these processes are not suitable for heavy-load applications.

(a) Key. **(b)** Circlip. **(c)** O-ring.

Figure 1.2: Examples of locating and spacing elements.

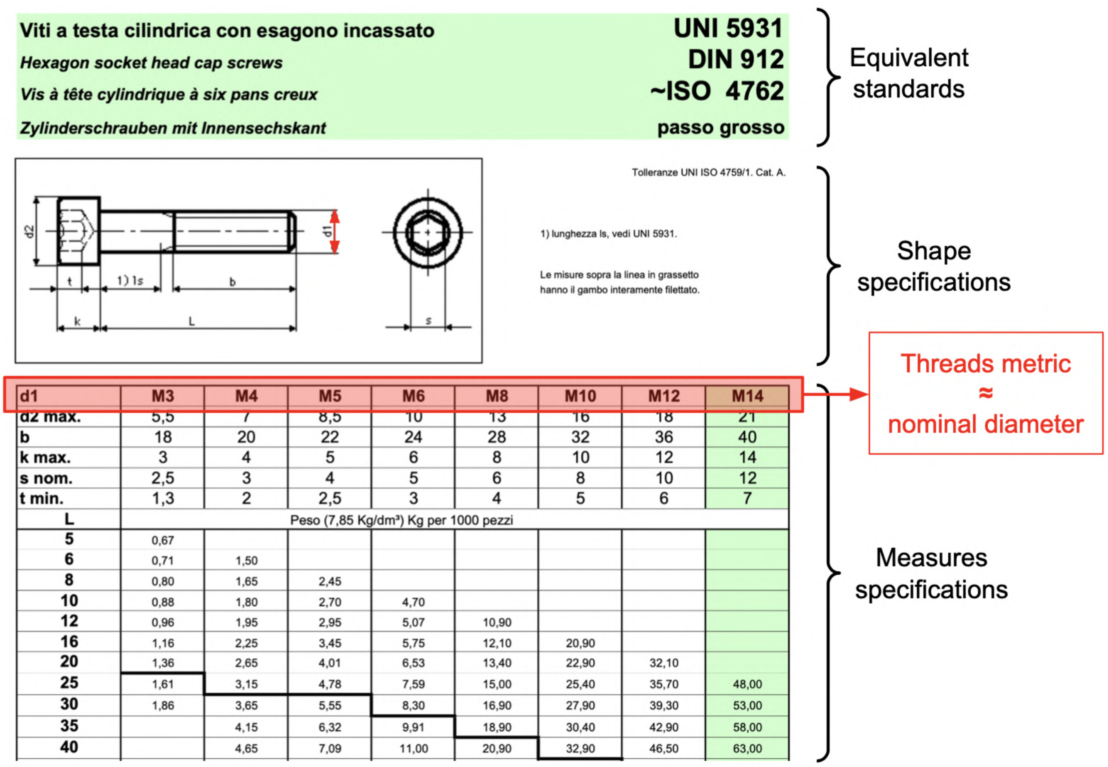
Always in the class of assembly techniques using connection elements, locating, spacing, or sealing elements have to be included, such as keys, circlips, and O-rings (Fig. 1.2). These parts fit in specific seats and are used to limit the degrees of freedom of the parts or to make a fluid seal. On the one hand, keys and circlips prevent respectively relative rotation and translation between two parts, on the other hand, O-rings ensure the tightness to the fluids or avoid the entrance of dirt.

- **Absence of connection elements** refers to all those connections between the components of an assembly that do not require the use of further parts, rather they provide for the deformation or alteration of the parts or their properties. Welding, gluing and interference fit are common examples of this type of assembly techniques. The first two are crucial since provide permanent joints. They in fact generate a solid and sufficiently strong joint and thus can be safely used for heavy load applications where, instead, the use of fasteners is not reliable. One process rather than another is used according to the materials of the parts to be connected and the type of product. For example, welding is usually carried out to bond materials such as metals or thermoplastics. By welding the parts are heated to high temperature to be melted and then joined. In some cases there can be an addition of material, i.e. welding beads, to strengthen the connection. Different from welding, gluing is a joining technique suitable for connecting almost all materials that, moreover, do not need to be heated and altered during the process. Interference fit, finally, is a type of permanent or semi-permanent connection used to join parts by interlocking them with each other. That is, two parts can be assembled only because wedged together due to their dimensions, namely, in the case of permanent connection, the solid part is slightly larger than the hollow one, and thus this discrepancy creates a friction that holds the components together. To be fixed, the parts are pressed by a force of varying intensity according to the discrepancy between the size and the elasticity and deformability of the materials, as well as thermal expansion can be exploited.

1.1.2 Types of parts in mechanical assemblies

From the industrial point of view, a company can fabricate a mechanical product from scratch, internally designing and producing every single component, but this results a very onerous operation in terms of time and costs. On the contrary, it is a common strategy to reuse existing parts and purchase components from external suppliers. In particular, the elements that are usually imported are well-known and already patented components that do not need to be modeled every time.
At this purpose, standards have also been introduced over the years to uniform and fix the aspect of some components, and then easily identify, demand and manage them for their reuse. On the one hand, there are standards less rigorous and concerning certain common features of the parts (e.g. the number of teeth in a gear) or the structure of multi-part components (e.g. the number of spheres and rows of spheres in a bearing and their arrangement). On the other hand, there are more restrictive standards, actually recommended rules, that completely regulate the shape and the properties of some categories of parts.
As for the latter case, from the early 1900s, different institutes have worked to provide a formal standardization at national and then international level of some widespread categories of parts in different industrial fields. The aim is to make easier communication between different companies, improve manufacturing efficiency and reduce trade barriers. Due to the large variety of industrial fields covered by standards, these are defined according to the specific product domain area. As for the mechanical field, standards mainly concern fasteners and locating elements, among

which, but not only, screws, nuts, washers, O-rings, clips, keys, and rivets. These are grouped by product classes so that similar parts but with different usages can be correctly regulated. The DIN standards from the Deutsches Institut für Normung, the UNI standards from the Italian National Standards Institute, the European Standards (EN), and the ISO international standards given by the International Organization for Standardization can be listed among the standards nowadays widely exploited. By means of a set of specifications, each standard defines the requirements an object must meet in terms of measures, shape and structural properties in order to be adequately defined and identifiable. They are designated through a code with a prefix associated with the standardization organization in charge and a number; for instance, DIN 1 is the set of specifications for taper pins according to German regulations, UNI 7435 defines the norms for circlips for shaft according to the Italian norms, ISO 4762 defines the specifications for hexagon socket head cap screws at international level. Then, as shown in Figure 1.3, the equivalent standards representing the same parts are then usually indicated on the specifications tab, as to avoid misleading classifications.



d1	M3	M4	M5	M6	M8	M10	M12	M14
d2 max.	5,5	7	8,5	10	13	16	18	21
b	18	20	22	24	28	32	36	40
k max.	3	4	5	6	8	10	12	14
s nom.	2,5	3	4	5	6	8	10	12
t min.	1,3	2	2,5	3	4	5	6	7
L	Peso (7,85 Kg/dm³) Kg per 1000 pezzi							
5	0,67							
6	0,71	1,50						
8	0,80	1,65	2,45					
10	0,88	1,80	2,70	4,70				
12	0,96	1,95	2,95	5,07	10,90			
16	1,16	2,25	3,45	5,75	12,10	20,90		
20	1,36	2,65	4,01	6,53	13,40	22,90	32,10	
25	1,61	3,15	4,78	7,59	15,00	25,40	35,70	48,00
30	1,86	3,65	5,55	8,30	16,90	27,90	39,30	53,00
35		4,15	6,32	9,91	18,90	30,40	42,90	58,00
40		4,65	7,09	11,00	20,90	32,90	46,50	63,00

Figure 1.3: Example of tab that defines the specifications for a standard screw (taken from [5]).

Thus, when dealing with mechanical assemblies it is therefore possible to distinguish between parts cataloged by standardization institutes and parts that instead are ad-hoc designed for a specific product. In the rest of the manuscript the following definitions are adopted:

Standard parts

Standard parts are the set of all the parts that conform to an established industry, governmental agency, or consensus standards organization specification. The specifications must include all information necessary to produce the parts and must be published so that any person or organization may manufacture the parts. The parts are designated with a unique code (e.g. UNI, DIN, EN, ISO, ASME, etc.).

Custom designed parts

Custom designed parts are the set of parts that do not fulfill published and uniquely identified standard specifications. These, in fact, are modeled and sized depending on the structure and purpose of the assembly.

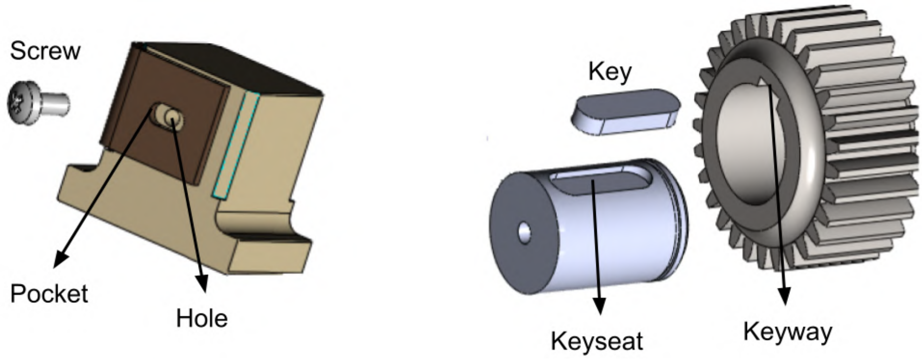
The above definitions are of particular interest for the comprehension of the approach presented in this manuscript. In fact, they do not only outline a distinction based on the shape properties of the parts, rather it can be assessed that they also definitely provide a sort of function-based grouping of the parts. According to their role within the assembly, indeed, on the one hand, there are parts that are mainly structural, i.e. the custom designed parts, on the other hand, there are components with a more explicit function of creating a blocking, connection, or spacing relation between other parts, i.e. the standard parts.
It is necessary to clarify that the concept of standard parts should not be confused with that of fasteners. As already said, only some standard parts categories are used to fix components with each others and thus are related to the concept of fasteners. Other categories, instead, just create e relation with some constraint so to reduce the degree of freedom between the assembly components and therefore are not included in fasteners. In the most general cases, the fasteners set can also intersect the custom designed parts set, namely when a fastener does not meet the international norms and it is specially modeled by designers, but it is a rare situation at the industrial level.

1.1.3 Typical arrangements in mechanical assemblies

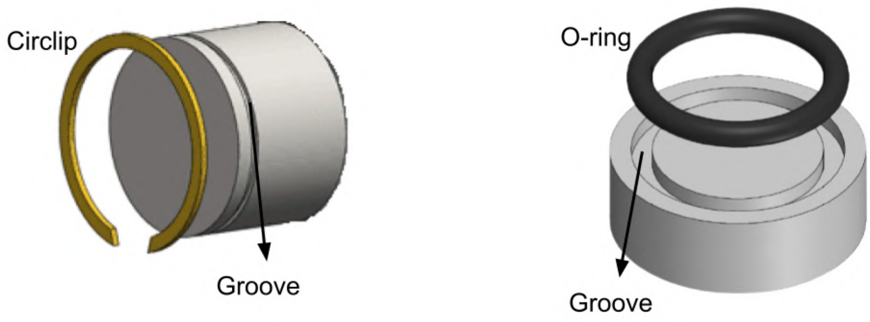
A further distinctive feature of mechanical assemblies is the presence of recurrent components' arrangements, especially when applying the mounting by connection elements. In particular, during product design and manufacturing, some rules and typical mounting schemes concerning standard parts, and more generally fasteners and locating elements, are followed to correctly and reliably position and assembly parts. The guidelines are basically of two types since they can be about the relations between fasteners or the positioning of fasteners and locating elements in defined seats. The two types are now pointed out and briefly described.

- **Positioning of fasteners and locating elements.** In mechanics, it is known that almost all connection elements have to be inserted in defined seats to fulfill their function.
  With regard to fasteners, they generally have to be inserted into aligned holes of two or more mating parts to fix them with each other. More specifically, threaded fasteners, such as screws and studs, can be inserted both in threaded holes, not threaded holes and through pockets (Fig. 1.4a). Moreover, at most one hole can be blind, while the others must be pass-through, to allow the employed fastener to move across all the parts it has to join. Unthreaded fasteners, instead, such as pins and rivets, are designed to be inserted only through preformed holes. Since they are mounted by interference fit and deformation, the diameters of holes are usually less than or equal to that of the used fastener.

  As far as locating elements, the most relevant cases of keys, circlips, gasket, and O-rings are reported. Keys are components used to connect a rotating element (e.g. a gear) to a shaft, the mounting of which follows a standard scheme (Fig. 1.4b). For a key to function, in fact, the shaft and the rotating element must have a keyway and a keyseat, which can be circular-end pockets, rectangular pockets or slots (through or blind) in which the key is installed by interference fit. Similarly, circlips, gaskets, and O-rings always fit into specific seats, namely circular grooves. Depending on the type of part, the groove can be both radial, if carved into a bore, inside a tube, or over the circumference of a cylindrical-like component (Fig. 1.4c), and frontal, if cut in a planar surface (Fig. 1.4d).
  The above configurations refer to the positioning of convex shaped fasteners and locating components into defined and standard seats described as concave features carved into assembly's components (obtained by turning, drilling or



(a) Screw in through pocket and hole. **(b)** Key in keyway and keyseat.



(c) Circlip in radial groove. **(d)** O-ring in frontal groove.

Figure 1.4: Examples of standard parts inserted in the associated seats.

milling operations). Nonetheless, also fasteners and locating components typically mounted on convex portions of mechanical parts exist. This is the case, for instance, of nuts and washers always mounted on the stem of a screw or on a stud. However, these cases are closely associated with the relations between fasteners rather than their positioning and are thus addressed in the next item.

- **Relations between fasteners.** Some categories of fasteners, besides being positioned in the specific seats, need to be tightened together to perform their function of fixing two or more components, otherwise the connection is not stable or durable. Consequently, recurring groups of connected fasteners can be identified. In particular, these are distinctive of threaded fasteners (i.e. screws, nuts, and studs), but also not threaded components have to be taken into account (i.e. washers). A meaningful case is that of screws. As a matter of fact, screws may be used alone, or in conjunction with nuts, to avoid axial movements and secure the connection. More than one nut can be tightened against a screw, thus two nuts can be in contact, or they can be positioned at the two opposite ends of the screw stem (e.g. one at the end to fix the screw and the other near the head). Also washers can be mounted on screws, in general, with the function of spacer, or to distribute the load of the fastener. One or more washers can be arranged between the head of the screw and a nut, between the head of the screw and the part to be fastened, as well as between a nut at the end of the stem and the other part to be fastened. Similar to screws, studs are used in combination with nuts and washers to fit at least two components together. Depending on the category of the stud, that can be partially threaded, totally threaded or threaded on the two ends, the number and the arrangement of nuts and washers vary, but at least a nut should be mounted to block the joint and avoid axial movement.

1.2 Digital representation of mechanical assemblies

In the previous section, the main features and properties characterizing mechanical assemblies are pointed out.
According to the several types of parts available, their variable arrangements, and possible assembly techniques applicable, it is evident that the manufacturing and managing of mechanical products rely on a deep engineering knowledge to optimally choose parts, make them fulfill their functions, and allow correct smooth movements. The main difficulty is in defining a product from scratch since many different options should be evaluated by the companies to release optimal products and be competitive on the market. In order to plan, define and test a product before it is actually manufactured, and thus avoid waste of material or damage to parts, the production is increasingly assisted by the introduction and exploitation of digital representations of the assembly, defined virtual prototypes or digital twins in the Industry 4.0 context. It is the crucial element on which technologies are based to allow the simulation and the analysis of different behaviors and scenarios, covering almost all the manufacturing phases, such as design, assembly, maintenance, and reuse.

CAD systems and their add-ins are used to create detailed geometry of the 3D models of each part of the product and then arrange them in a unique space to generate the final assembly and simulate its behavior. SolidWorks® [4], Creo® [3], Catia® [2], and Autodesk Inventor® [1] are some of the commercial CAD software widely used to design mechanical products, to name a few.

As concerns the techniques used to digitally reproduce solid objects' appearance, the boundary representation (B-rep) is the most common representation adopted in CAD systems. It allows to totally define a 3D shape by means of the limits of its

volumes, i.e. a collection of bounding faces. More precisely, the elements comprised in a B-rep are of two types, specifically *topological* and *geometric*.
Topological entities include faces, edges, and vertices. These are then referenced to geometric entities, namely the equations of surfaces and curves that underlie faces and edges respectively, along with the coordinates of the points corresponding to vertices.
For sake of clarity, throughout the manuscript, when referring to the CAD model of a part, if not specified, it is assumed that it is a CAD model in B-rep format and, according to the definition given above, it includes geometric and topological data. In addition, descriptions and attributes are frequently attached to the models of the parts to provide technological data such as type of parts, category of standard parts, dimensions, and material. Although the use of standard formats guarantees the transmission of the B-rep information from one system to another and thus facilitates collaboration and exchange of data, on the contrary, the semantic information may be lost during the import/export of models, and it should not be taken for granted. It should be noted that, for a similar reason, the feature-based representation of the models has not be taken into account. On the one hand, the modeling features can not be included in any neutral file format, rather they can be read only through a specific CAD system. On the other hand, different strategies can be adopted to get the same final shape and thus managing and analyzing the design history of a model would become an extremely complex operation.

A CAD model of an assembly is, then, generated by grouping the models of all the involved parts and arranging them in the 3D space as to provide the final product structure. Notice that, in the following, assembly will be always referred to as an object consisting of several parts all with a single body, and it has not to be mistaken with a multibody part. During the positioning, CAD systems allow to impose mate and assembly constraints between the parts to specify the conditions that their design must satisfy (e.g. coincidence between vertices, edges or faces, concentricity, parallelism, or perpendicularity between surfaces, etc.) and to mimic their reciprocal movements and kinematic relationship (e.g. imposing relative movement between two parts, such as axial rotation, translation, etc.). However, this information is intrinsic of the system, and they are not supported by the B-rep structures.

As a consequence, it can be stated that the most general CAD model of an assembly is given by the combination of parts precisely located in a common space. The data definitely available associated with it are the boundary representation of each single part, together with their absolute position in the space.
Moreover, parts can be organized in a simple list or according to some hierarchical

or logical structures. The structures envisage the aggregation of parts into subgroups. Among the criteria adopted to define subgroups, it can be mentioned the grouping by functionality, which basically reflects the organization in subassemblies of real mechanical products, the grouping of parts having same materials, or else the grouping of parts that constitute a welded object. Even if the product structure is transmitted as information of the CAD model of an assembly, it is clear that it can be misleading and not reliable, since it is not unique and depends on the purpose of the designer or the application task.

Until now, it has been illustrated the general method by which the 3D model of a real object is created, the structures used and the data that are always available regardless of the system used. Nonetheless, when dealing with industrial mechanical assemblies, it is to be taken into account that real complex parts are represented through simple geometric entities. Therefore it is not always feasible, or convenient, to provide realistic renderings. At this purpose, a current practice in industry is to take advantage of some common strategies and conventions in the representation of the mechanical components and their relations. These are primarily associated with standard parts appearance, but also custom designed parts are involved, as well as assembly techniques. This topic is of particular interest for the work presented here and, hence, it is further addressed in the next two subsections.

1.2.1 Part representation

Even if the objective of the computer-assisted design activity is the detailed specification of the product to be manufactured, it can happen that the obtained CAD representation of the product doesn't fully reflect the shape of the final product. Two different reasons can be given to explain this situation. On the one hand, the practice of idealizing the shape of the components of an assembly to lighten the CAD model is widely employed. On the other hand, the existence of deformable parts implies that these can be modeled in different states.

Idealization

As far as simplification, in some models and at some stages of the design phase, mechanical components can be streamlined because only their overall dimensions are of interest. This is generally the case of components that are designed or acquired by third parties. They include standard parts, that are almost always simplified since their appearance is well known and does not need to be fully modeled.

Other shape differences concern detail features. That is, real mechanical parts are trimmed with fillets and chamfers to remove sharp edges or add final refinements. On the contrary, these details can be reported totally, partially, or not at all in the associated 3D digital models. It depends on the precision with which the parts need to be modeled, as well as on the designer choice. For example, small finishes are not essential for evaluating a part within an assembly, but they are crucial to define and produce the single part. As a consequence, real mechanical components and their 3D representations can differ from each other due to the variability of the level of details that are rendered. Furthermore, for the same reason, different representations of the same part can also exist.

More specific considerations can be done in connection with some types of mechanical components. Both single-part and multi-part components can be idealized in the CAD models according to common design strategies. For the first case, gears can be mentioned. Gears are mechanical parts similar to wheels with a particular silhouette characterized by the presence of teeth on the inner or outer rim. However, teeth are often not defined in their digital models as an idealized representation (Fig. 1.5). Namely, when the aim is just to represent the volume occupied by a gear in an assembly, its CAD model can appear as simple holed cylinder where, depending on the presence of the teeth on the inner or outer rim, the outer diameter corresponds to the maximum diameter given by teeth on the outer rim or the hole's diameter corresponds to the minimum diameter given by teeth on the inner rim.

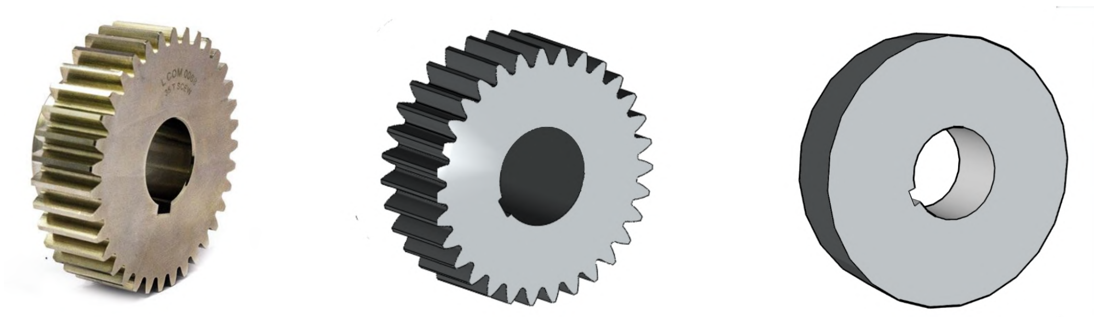
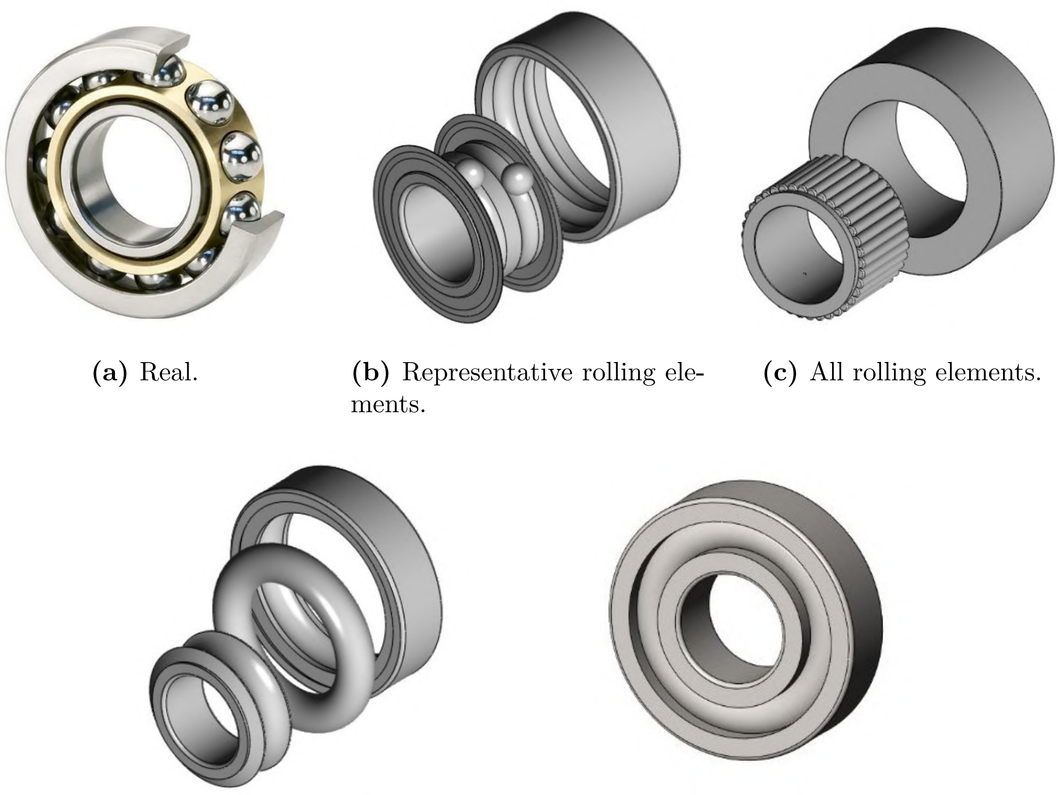


Figure 1.5: Example of real gear compared with a detailed model and a simplified model.

Bearing is instead a typical example of multi-part component usually differing from its digital version. In particular, the simplest physical bearing includes two rings, inner and outer, a cage and rolling elements (e.g. balls, rollers, cones, spheres or needles) arranged on a circular pattern, besides some gasket or cover (Fig. 1.6a). In CAD modeling, it is not always necessary to precisely design each single part of a bearing, rather for convenience some details can be omitted (e.g. the cage is rarely modeled) or simplified but maintaining available the principal properties of the component [83]. Several modeling conventions can be adopted, depending on the company or designer choices, thus different 3D models of the same bearing can exist. For instance, only a representative copy of the rolling elements can be included in the model instead of the whole series (Fig. 1.6b,1.6c). Or, the series of rolling elements can be collapsed in a single part that leaves the section view unchanged (Fig. 1.6d), e.g. ball series can be collapsed in a single torus with diameter equal to that of the spheres that will always produce a circle representing the sphere as prescribed by the drafting standards. In an even more idealized way, a bearing can also be totally collapsed into a single part (Fig. 1.6e).



(a) Real.

(b) Representative rolling elements.

(c) All rolling elements.

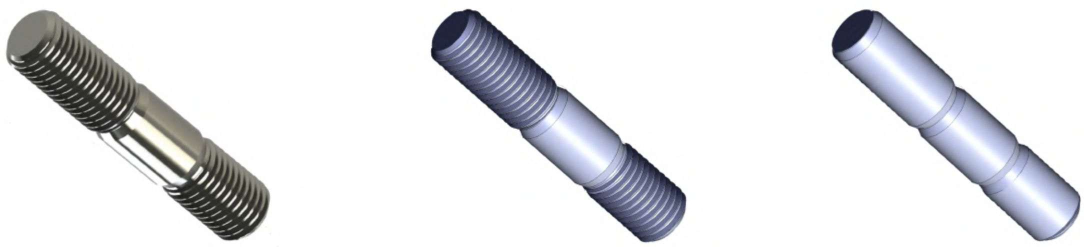
(d) Rolling elements collapsed in a torus.

(e) Single part.

Figure 1.6: Exploded views of bearings modeled according to different conventions.

Threaded parts deserve to be described separately, as there are common guidelines that regulate the design of their digital models and that are almost always applied in industry. In particular, threads are typical of fasteners such as screws, studs, and nuts. Namely, a thread is a ridge wrapped around a cylinder or cone in the form of a helix. It is an outer thread when it covers a convex surface (e.g. the stem of a screw or a portion or all a stud), while it is an inner thread when it is on a concave surface (e.g. in the hole of a nut). It is characterized by some dimensional properties, namely nominal (or major) diameter, minor diameter and pitch (i.e. the distance from a point on the thread to a corresponding point on the next thread measured parallel to the axis) that then become representative property of the threaded component itself. Just think to the classification of standard screws by means of their metric, which actually refers to dimensions of threads (e.g., also by referencing Figure 1.3, in screw designation ISO 4762-M4-5, M4-5 means that the nominal diameter is 4 cm and the thread length is 5 cm). From the modeling point of view, the helical structure would result in a highly detailed zone the representation of which is demanding and weighs down the 3D model. Hence, a widespread strategy is to idealize the threaded area of real part into a cylindrical or conical area. Conventionally, the diameter of the simplified surface is equal to the nominal diameter of threads. In Figure 1.7 the CAD models representing a screw and a stud with and without threads are shown as examples.

(a) Screw.



(b) Stud.

Figure 1.7: Examples of real threaded parts compared with a model with threads and a model with no threads.

Deformable components

Finally, deformable components are a wide class of parts that are subject to shape variations in the modeling phase. This class includes both fasteners (e.g. rivets), locating components (e.g. retaining rings, clips, and washers) and any other part of elastic or semi-elastic material (e.g. belts, springs, etc.). The distinctive feature of deformable parts is that they can assume different shapes whether they are in resting or pulled position. In particular, two kinds of deformation have to be distinguished: temporary deformation and permanent deformation. The first type occurs when the component gets deformed just during the mounting operation, after which it returns to its original form.

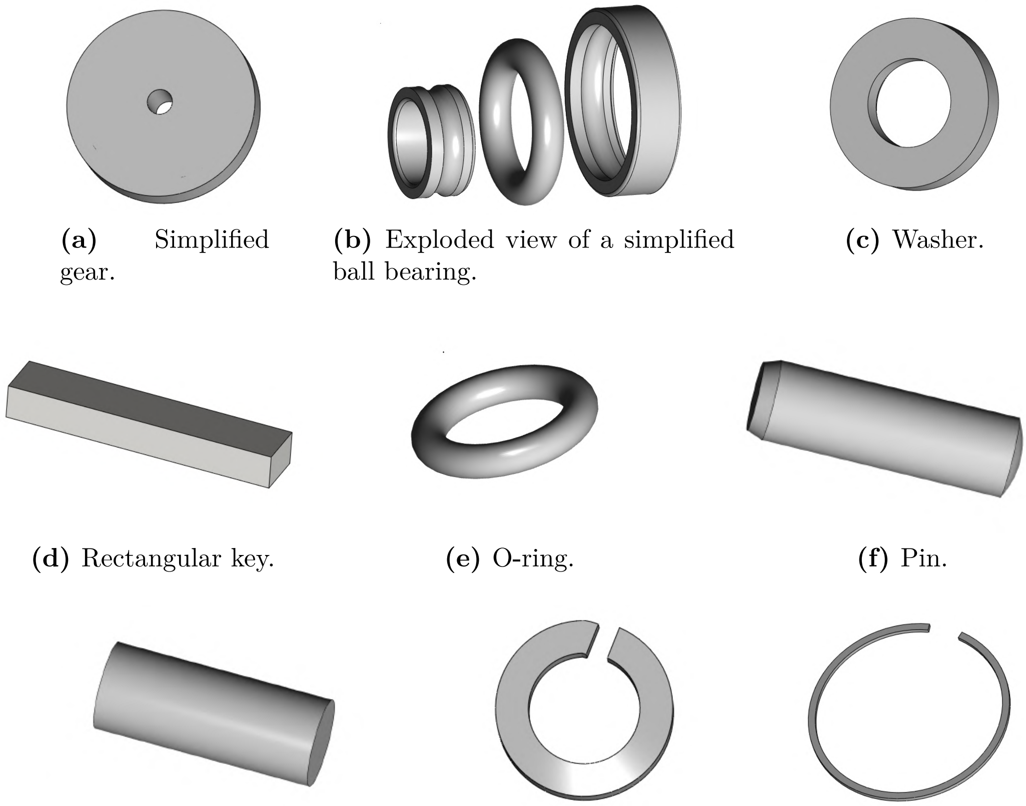
(a) Washer.

(b) O-ring.

Figure 1.8: Example of real deformable components compared with a model in released state and a model in constrained state.

Clips, seegers, gaskets, and O-rings are some examples of temporary deformable components. The latter type of deformation is when a component, like rivet, changes its form definitely and this assures the joint of two distinct parts. From the modeling point of view, nor specific rules exist on how to model elastic components within an assembly, rather it is strongly user-dependent and depends on the designer choice or company conventions. Deformable parts can be interchangeably represented as rested or stretched, as if they are in the extraction/insertion phase, or in an assembled setting [53]. Thus, it often occurs that for the reasons just mentioned and for the possible idealizations/simplifications listed above the CAD model of a deformable part differs from the real part and can assume variable shapes (Fig. 1.8).

It becomes apparent that due to idealization and possible representations of deformable parts some mechanical components result quasi-identical from a modeling standpoint, besides the already mentioned possible existence of different represen-



(a) Simplified gear. (b) Exploded view of a simplified ball bearing. (c) Washer.

(d) Rectangular key. (e) O-ring. (f) Pin.

(g) Stud. (h) Spring washer. (i) Snap ring.

Figure 1.9: Examples of CAD models of mechanical parts with shape similar to other parts.

tations of the same part. Shape similarities between the models of standard parts and also between that of standard and custom designed parts can occur. As for the first case, for example, pins and studs can both be modeled as simple cylinders (Fig. 1.9f, 1.9g), spring washers in constrained state and snap rings have both the shape of an open ring (Fig. 1.9h, 1.9i). Concerning the second situation, just think of simplified gears and washers that can both be modeled as simple holed cylinders (Fig. 1.9a, 1.9c), or else simple parallelepiped shaped components that can be mismatched with rectangular keys (Fig. 1.9d). Moreover, the balls of a bearing collapsed in a torus take on the same appearance as an O-ring (Fig. 1.9e).

1.2.2 Assembly techniques

Even more challenging is a realistic transposition in a digital model of the different assembly techniques. In fact, a precise engineering relation between two or more components has to be represented. For this purpose, several conventions were introduced and exploited in the creation of CAD models of mechanical assemblies.

Threaded connections

Mountings by threaded components are characterized by the placing of fasteners into coaxial holes, with at least one of them being threaded. In reality, the nominal diameter of the outer threads of the convex part (e.g. the stem of the screw) and that of the inner threads of the hole correspond, so that the parts can be screwed with each other by means of a roto-translation movement along the axis and a stable relation is thus provided. However, this relation is not explicit in CAD models, since, as already mentioned, threads are generally not represented in fasteners, and similarly they are idealized in holes too. Indeed, conventions state that the inner and outer threads have different diameter values. Respectively, the cylindrical surface of the hole has the radius equal to the minor diameter of the internal thread, while the cylindrical surface representing the shank has diameter equal to the external thread. It follows that threaded links are most frequently geometrically represented as cylindrical interference volumes (Fig. 1.10). Moreover, depending on designer choice and purpose, threaded fasteners, especially screws, can be partially substituted by a placeholder, or even not included in the model. As a consequence, these strategies entail that most of the knowledge associated with the product is implicit and generally its interpretation and inference is demanded to experts who are aware of engineering know-how, company conventions, and design techniques.

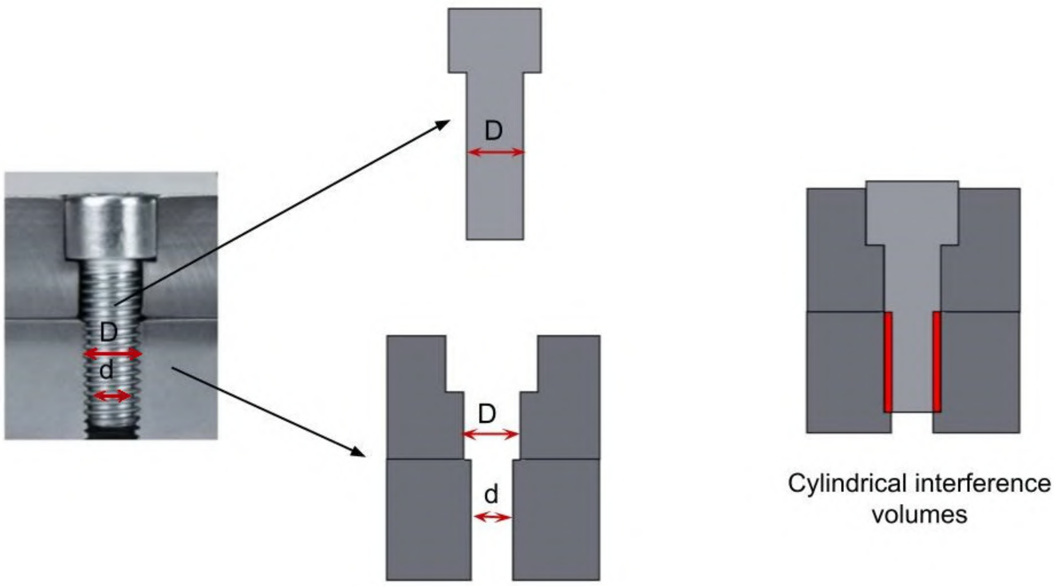
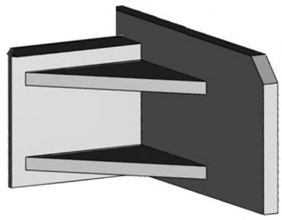


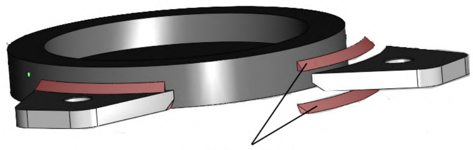
Figure 1.10: Section view of the 3D representation of a threaded connection. D and d are respectively the major and the minor diameters of the thread. The modeling of the threaded cylindrical portions of the shank and the hole generates volumetric intersections.

Welding and gluing

As for the assembly techniques characterized by the absence of connection elements, in most cases these are consistently represented without adding further elements in the 3D models of the joined parts. That is, the widespread strategy is not to explicitly model welding and gluing, but at most to indicate them as an attribute of the parts, or by grouping the merged set in a subassembly in the CAD model (Fig. 1.11a).



(a) Weldings not explicitly represented.



(b) Weldings with solid beads.

Figure 1.11: Examples of weldings in CAD models.

Moreover, another less common practice that may be found in industrial CAD models to represent weldings is by means of solid welding beads. That is welding beads are explicitly modeled as a 3D object positioned in the corner formed by the parts to be joined and in contact with both of them (Fig. 1.11b). Usually the name of these extra parts refer to the fact that they are weldings, so to clarify that they are indicative of an assembly process, rather being actual mechanical components.

Interference fit

Finally, the representation of interference fit in CAD models deserves to be mentioned. In reality, two parts mounted by interference fit are characterized by slightly different sizes in their contact areas, but once assembled the sizes would match because of deformation. Since the variations in shape caused by deformation can not be simulated in parts CAD models, because parts are represented in only one state (i.e. deformed or not deformed), the interference fit is a very challenging process to reproduce. Whether to model components with the proportions they assume before or after the fitting relies on the designer's preference or company conventions. The two parts are, in general, modeled with congruent sizes, but some meaningful information on the type of assembly technique can be implicitly available as dimensional tolerances.

1.3 Challenges in using industrial CAD models of mechanical assemblies

From the previous section, it can be concluded that the practice of using computer-assisted tools for the design of new products is now consolidated. Then, to support products life cycle, the computational analysis of the 3D models through different tools allows to simulate and validate product functionality and characteristics together with the concerned complex processes before the product is fabricated. However, as a consequence of habits and conventions adopted in the design phase, as described in the previous section, problems may arise when dealing with industrial CAD models both at the design stage and at the processing and evaluation stages. These are briefly summarized in the following.

1.3.1 The interpretation issues

With the primary goal of automating as much as possible every process while minimizing human interaction, interpretation issues have to be faced when considering and evaluating CAD models of mechanical products in B-rep format. This is because of the widespread use of implicit knowledge and design conventions, as described in Section 1.2. The issues can be categorized according to the following typologies: lack of semantic information, lack of uniformity in the models, missing components, and the presence of modeling errors.

Lack of semantic information

Since this expression will be used extensively in the following, it is first required to clarify what *semantic information* is here regarded when referring to CAD models of mechanical assemblies. Namely, semantic information indicates all that non-geometric information, such as category membership, technological data, kinetic and kinematic properties, and functionality [53].

The difficulty of describing a mechanical part by means of the only geometric and topological data included in the B-rep is evident, as well as creating a realistic and consistent CAD model of a mechanical assembly. In particular, it is difficult to ensure that a geometric part incorporates and expresses a precise engineering meaning as well as that a simple contact between surfaces implies physical constraints. It was mentioned in Section 1.2 that, to overcome these problems, CAD systems enable the association of attributes to the models of the parts, as well as the naming of components with descriptive names (e.g. a standard part is identified by its code, a component is designated by its class, etc.). This strategy is typically used to enrich the mere geometric representation with high-level information or to complement missed details. However, this further information is simply reported by means of characters' strings and no norms exist about how they should be written and included in the model. As a consequence, several ambiguous descriptions can be found and this leads to unclear interpretations and difficult understanding [96]. Additionally, because all this information is expressed as annotations rather than formal features applied to the components model, they are commonly subject to being discarded. In fact, complex CAD assemblies are created in collaboration, importing and exporting models of parts using standard formats for data exchange. Importing and exporting CAD models are delicate processes that can lead to the loss of non-geometric data due to incompatibility between various CAD systems, and sometimes even when exporting and importing in the same CAD system [92]. It follows that, in the most general scenario, the CAD model includes only the geometric representation of the product while semantic information both relative to parts and their relations is not explicitly provided. For instance, the distinction between standard parts and custom designed parts, the identification of fasteners, the material and the flexibility of a part, or else the type of connections and the degrees of freedom for a part to be moved can not be taken for granted.

Lack of uniformity

The capability of representing the same real mechanical assembly and its components in different 3D models is referred to as a lack of uniformity. It takes into account both the variation in the semantic data description, that was just mentioned, and the heterogeneity in the representation of shapes, see sections 1.2.1. The existence of categories of standardized parts and general conventions in the modeling of mechanical parts and assembly techniques have been discussed, that may prevent shape inconsistencies, at least within the same model of an assembly. However, it is to underline that the use of standard parts and conventions is not mandatory at the industrial level. Rather, these are suggested guidelines the application of which depends on designer choice and can vary depending on the CAD modeler used and on the company practices. Moreover, the CAD model of a complex assembly can be developed in a collaborative way and in several separate stages, by exploiting existing elements and then combining them with each other. That is, it can include standard parts imported from multiple catalogs, components provided by separate repositories and supplied by external companies.

As a consequence, components belonging to same classes but modeled according to different criteria and with different level of details can be merged and thus be found in the same model. As a result, absence of homogeneity is a common issue within the same model as well.

Missing components

CAD models can often have missing parts and this may refer to different situations. On the one hand, a common practice is not to physically include connectors and fasteners with the aim of making the model leaner and lighter. As mentioned in Section 1.2.2, their membership in the group of standard parts allows to avoid their representation and inclusion in the product model, since their shape and their specifications are known. On the other hand, it could be a choice of the designer to omit some parts. Among these, insignificant parts or parts external to the specific product portion under design, but interfacing with it (chassis, support base component, etc.). This can cause incomplete models that may lead to misleading interpretation of the parts and their functions. For instance, there can be found fasteners inserted in a single part and not connecting any other component, or even floating parts that are totally disconnected from the rest of the model.

Modeling errors

It must be taken into account that industrial CAD models are not always accurate, rather they can include errors. Some components can in fact be badly modeled or wrongly positioned, due to format translation numerical errors, as well as inattention of the designer. It follows, for example, holes misalignment and also the generation of false features, like intersections (volumetric interference) or, vice versa, empty spaces (clearance) among parts, cause of misleading interpretation of the assembly or missing detection of contacts.

1.3.2 The need for semantic interpretation

Summing up the issues discussed above, when dealing with an industrial CAD model of a complex mechanical assembly little information is certain, that is the list of parts composing the model and the geometric and topological entities according to which parts are represented. Semantic data and components designation are not obvious, as well as consistency in parts' representation and positioning is not guaranteed. From these considerations, four hypotheses can be formulated, which will be the basis of the provided work. Namely:

Hypotheses

1. CAD models of the same component can differ in the shape representation.
2. CAD models with same shape can represent different components.
3. CAD models do not contain semantic data.
4. CAD models are not perfect.

The hypotheses underline the impossibility of automating tasks such as subassembly identification, assembly sequence planning, costs estimation and component retrieval only having as input the CAD model of a mechanical assembly. The risk is to get unfeasible or not reliable results since semantic data can be lost from sight and elements subjected to physics and having precise engineering characteristics would be treated as simple geometric shapes. In this regard, experts intervention is often required, on the one hand to correctly interpret the CAD model, detect parts functionality and not explicit features, on the other hand to check and validate the outputs. However these are time consuming and error prone operations for which excessive human workload is needed.
As a consequence, it can be assessed that, despite the continuous development of new technologies and enhancement of systems, the comprehensive and automatic semantic interpretation of an industrial CAD model is still an open issue, that certainly deserves to be addressed and improved.

A solution lies in developing a system that algorithmically processes and inspects the CAD model of a mechanical product assuming the worst hypotheses and returns an enriched model including engineering meaningful information.
As it will be better discussed in Section 3.1, the challenge lies in defining computerized tools that can simulate in an automatic manner the process carried out by an expert when interpreting the geometry of the parts using its own experience and knowledge. At this purpose, the above described conventions and criteria generally adopted for representing mechanical parts in 3D models can be leveraged in a reverse process to detect technological data from the geometric and topological features of components and their relations.

Table 1.1 provides a first general overview of the missing data and key inconsistencies that can be found in industrial CAD models and must be resolved for returning a reliable interpretation. The first three columns report the fundamental

Table 1.1: Overview of the needed information to provide a semantic enrichment of a CAD model and possible techniques to use.

		Assembly techniques			Issues in CAD models				
		Threaded connections	Welding, gluing	Interference fit	Deformable components	Missing components	Imported components	Model simplification	Modeling errors
Techniques for semantic analysis	Contact analysis	✓	✓	✓	✓	✓		✓	
	Volumetric interference analysis	✓		✓	✓			✓	✓
	Clearance analysis				✓				✓
	Cylindrical surfaces analysis	✓				✓			
	Dimensions matching			✓	✓	✓			
	Feature recognition	✓	✓	✓	✓	✓	✓	✓	
	Standard parts knowledge	✓			✓		✓	✓	
	Dimensional tolerance			✓	✓				✓

assembly techniques that are implicitly included in the CAD models, but whose information has to be extracted to understand the assembly. The remaining columns, then, list the typical issues of mechanical CAD models that have to be detected and solved to avoid inconsistencies. The rows suggest the possible techniques and the strategies that can be applied to perform the semantic analysis. The checks in a column state that the engineering concept or the CAD model issue associated with the column can be detected or overcame by exploiting one or the combination of the indicated techniques.
It is evident that some techniques exploit only the analysis of the geometric entities and their relative positions. This is the case of contact analysis, cylindrical surfaces analysis, volumetric intersection and clearance analysis. Some other techniques, such as feature recognition, refer to more complex operations, geometric based and driven by engineering purposes. Finally, also standard parts knowledge is reported for sake of completeness as instrument to apply. However, it is a more elaborate data, that can be obtained by means of a specific process, i.e. parts recognition, as it will be discussed in Section 4.2.

In order to better visualize the issues mentioned above and demonstrate that they can be tackled in an automatic way only relying on B-rep data, some examples are now discussed. It should be considered that the strategies pointed out are rough hints on how to extract data and overcome issues. The formalization of the process and the detailed description of the tools developed are then provided in Chapters 3, 4, 5.

Example 1. The example of a CAD model of a portion of a gripper mechanism presenting a set of welded parts is shown Figure 1.12. The weldings are not explicitly represented, but it is immediate for an expert to understand that the parts are assembled with this technique. According to Table 1.1, the presence of weldings can be deduced by evaluating parts' contact and features. In particular, two main conditions have to be satisfied:

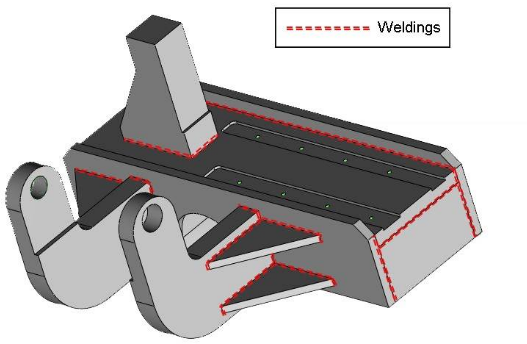
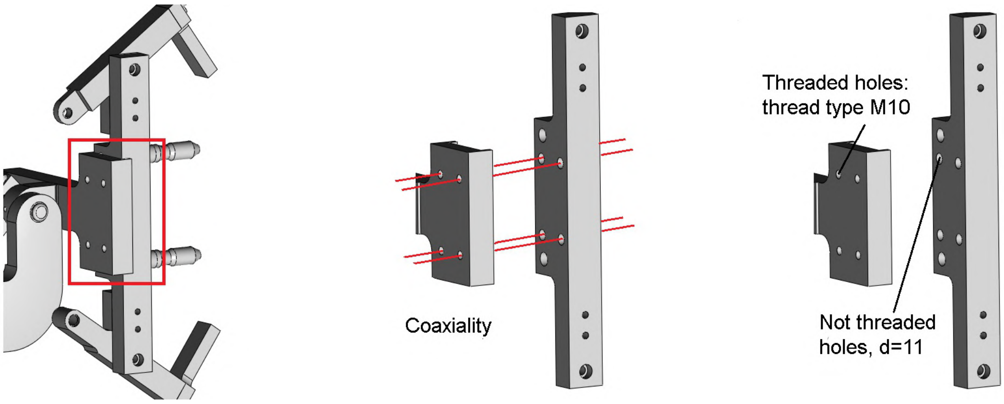


Figure 1.12: Example of weldings in a gripper mechanism.

1. The parts must have two planar surfaces in contact, these can be either an entire face of the part or only a portion of it (i.e. contact analysis).
2. There must not be coaxial holes among the two parts (i.e. feature recognition) or a third component with cylindrical contact with both the parts must not be found (i.e. contact analysis). These two situations, in fact, would imply the presence of fasteners, such as screws.

Example 2. A second example, taken from the same gripper mechanism, is provided in Figure 1.13 to visualize the issue of missing components and outline a possible detection strategy. Most fasteners are actually not modeled in the presented CAD assembly. This design choice is perceived from the presence of empty coaxial holes, visible in the figure. As reported in Table 1.1, missing components can be inferred through contact analysis, cylindrical surfaces analysis, dimensions matching, and feature recognition techniques. More in detail, to address the issue, these techniques can be applied according to the following schema:

1. Detection of two parts with a planar face in contact (i.e. contact analysis) on which coaxial holes exist (i.e. feature recognition and cylindrical surfaces analysis).



(a) Planar faces in contact. **(b)** Coaxial and empty holes. **(c)** Diameters analysis.

Figure 1.13: Example of the issue of missing components in a gripper mechanism.

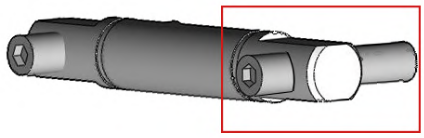
2. The holes are empty, in the sense that no components are found with a cylindrical not hollow face in contact with any of the holes' walls (i.e. contact analysis).

3. The diameters of the coaxial holes are equal or else the diameter of one hole is slightly bigger than the diameter of the other (i.e. dimensions matching). The first case refers to the scenario in which the holes are both not threaded, while the latter is verified when an hole (the smallest) is threaded.

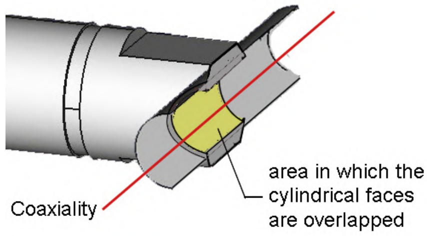
Example 3. An example of threaded fastener mounted on a part is shown in (Fig. 1.14) to support the description of a possible schema that can be applied to infer the presence of threaded connections in CAD models. The techniques that can be combined to extract this information are contact analysis, volumetric interference analysis, cylindrical surfaces analysis, and feature recognition, as reported in Table 1.1, along with standard parts knowledge. In particular, a possible strategy is to check the following points:

1. An hole is detected (i.e. feature recognition) such that its cylindrical hollow face is coaxial with a cylindrical not hollow face and they are overlapped for a certain portion of area (i.e. cylindrical surfaces analysis).

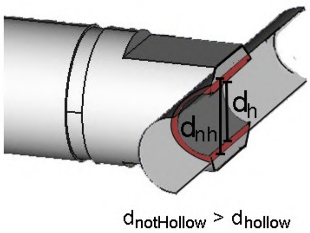
2. The hollow face has a smaller diameter than the not hollow one (i.e. dimensions matching and volumetric interference analysis). Namely, in the shown case (Fig. 1.14c), the diameter of the hollow face is $d_h = 3.3$ while the one of the not hollow is $d_{nh} = 4$.



(a) Pin in shaft's hole.



(b) Coaxial and overlapped cylindrical faces.



(c) Diameters analysis.

Figure 1.14: Example of threaded component mounted on a shaft.

1.4 Conclusions

This chapter points out the challenges in designing, processing and managing industrial CAD models of mechanical products with the final goal of automatically addressing complex assembly tasks and assist the manufacturing processes.

As for design, conventions and common practice have been fixed over the years and are commonly used by engineering companies and designers to represent mechanical parts and assemblies by means of structures that mainly contains geometric and topological data (e.g. B-rep). This, however, affects the appearance of the represented object, since it is very hard to achieve a perfect digital copy of a real product. In particular, the main difficulty is in providing engineering meaningful information, such as category of parts, functionality of components, and typical dimensions. These data are generally implicitly included in the CAD models in the form of geometric features, but they are not intuitive to be algorithmically understood unless experts provide them. Especially when dealing with industrial CAD models generated by gathering up parts and subassemblies imported from different sources or supplied by different companies, several conventions can be combined in

the same 3D model, further generating inconsistencies and misleading representations.
Therefore, assuming to have an industrial CAD model of a complex assembly, and have to inspect and manage it to simulate or plan assembly tasks, the creation of an enriched model obtained from an automatic semantic analysis of the original CAD model is crucial. An overview has been provided on which analysis techniques and data can be applied to derive the necessary information for such enriched model highlighting which conventions and criteria generally adopted for representing mechanical parts in 3D models have impact of the listed techniques (see Table 1.1).

The need for such an enriched model and the issues to be faced stand at the basis of this thesis and suggest the definition of a comprehensive approach combining geometry-based tools and engineering design knowledge to enhance the semantic meaning of a CAD model assembly and its parts and to address assembly tasks, such as subassembly identification, sequence planning and design for assembly. The leveraging of that new data will be crucial to simplify and improve existing methodologies addressing manufacturing tasks.

The next chapter explores the literature behind this thesis in order to establish the background in which it is placed. The milestones and the more recent works associated with the main topics addressed, such as ASP, SI, and DFA, are highlighted with the aim of drawing attention to their weakness and limitations, particularly those related to the lack of high-level information or their current manual extraction.

2

Related works

This chapter aims to contextualize the work done and place it in the current state of the art. The main purpose is to strengthen the validity and novelty of the proposed system and its single modules by highlighting open issues and challenges not yet addressed through a broad literature review. First, a general discussion on the need for the semantic interpretation of mechanical CAD models and the enhancement of product representation is reported. In this regard, the absence of automation in data interpretation and extraction is underlined, which also affects the handling of more complex tasks. Then, a more targeted examination of the literary works that treat the two areas of research in which the scientific contribution made is greater is provided. In particular, the most promising approaches and algorithms in addressing part recognition and assembly planning problems respectively are detailed focusing on their advantages and pioneering choices, as well as emphasizing the issues that deserve to be overcome.

2.1 Semantic interpretation of CAD models

In the industrial manufacturing field, the use of a digital representation of the mechanical assemblies through CAD models is a long-established practice that underlies the production process [40]. CAD models and their analysis are increasingly employed to assist the product life cycle in all its phases. Technologies have been developed to improve, simplify and possibly automatize complex and onerous tasks such as design and simulation [33, 95], assembly and disassembly [75, 93], together with maintenance and reuse [37, 54, 69]. However, issues related to the availability and management of the data in the CAD models are claimed. Among them, the lack or the loss of high-level information, i.e. that not related to the product shape, but rather associated with engineering concepts, as well as the need for massive experts intervention to interpret the geometric data.

In this context, the automatic enrichment of CAD models with semantic information gained much interest in the last years. A great effort has been done in the enhancement of product modeling process to represent product knowledge and technology information [84]. It results a crucial step to intelligently and algorithmically address the manufacturing tasks, reduce the human workload, and improve product knowledge exchanging and sharing. Several works can be found in literature that are focused on that topic and address it under different aspects. Some aim to mitigate the semantic gap by providing functional semantic annotation methods for CAD models based on ontologies [15, 24, 58, 59, 89]. Some others, instead, are more focused on the recognition and extraction of specific engineering knowledge implicitly contained in the geometry of CAD models [48, 52, 103].

However, the semantic interpretation of CAD models and the leverage of high-level information are still open issues that deserve to be further investigated. In fact, in many applications the only geometric and topological data are still used, or semantic information, such as parts types or assembly features are manually provided by experts and this strongly affect the production process, in terms of goodness and reliability of results, amount of resources required, time and costs.

According to the above considerations, it results innovative and very useful the development of a comprehensive and stand alone system capable of combining both the needs, i.e. the automatic extraction of high level semantic information from a CAD model and their usage in assembly tasks. All the steps to achieve this goal have been developed, addressing several different operations, from the simplest to the more elaborate. However, referring to the literature, the main effort and the greatest contribution were principally made in two fields. That is, the part recognition field, as far as the data extraction is concerned, and the assembly sequence planning field, in the matter of data exploitation.

As a consequence, the state of the art reported here is meant to review the most remarkable works in these two separate fields. The purpose is to identify the most interesting approaches provided over the years, underlining their pioneering choices as well as pointing out their weakness, especially according to the semantic enrichment point of view and the use of engineering meaningful data.

2.2 Part recognition

The recognition of the membership of a 3D part to a specific category of objects can be addressed either as a similarity problem or as a recognition problem. In the first case, a part is considered to belong to a specific class if it is similar to the other parts in the class according to a suitable distance function [35]. In the second case, a part is considered belonging to a given class if it holds the characteristics peculiar to the class elements.

In particular, similarity and retrieval of mechanical components are research topics widely addressed in recent years [28, 60, 64, 66, 78, 83]. Due to the growing size and variety of databases of 3D models of components and assemblies, as well as the fact that CAD models rarely include parts' specifications, such as category, dimensions and functionality, it is time consuming to design, reuse and manage mechanical product models from the early stages. The study and development of tools for the retrieval and cataloging of specific parts can therefore largely improve designers' activities in several manufacturing phases by allowing the reuse of previously designed parts and of the associated information (e.g. [77]).

In literature, several works exist which provide part classification methods. Ip et al. [63] tackle the problem of solid model classification providing a shape learning algorithm. A feature space is defined, where a decision tree learning is applied. Wheels, sockets and housing models are classified. Pernot et al. [91] propose the categorization of products based on shape descriptors and classify them in terms of characteristics that might affect the simplification of parts for the Finite Element Analysis (FEA). The parts distinguished are thin parts and parts with thin portions. In [62] an approach for the automatic classification of mechanical CAD models according to the manufacturing process is provided. It exploits the surface curvature as shape descriptor. In [39] a shape-based recognition approach is described that uses rules on face adjacency relations and attributes. Manda et al. [85] present a deep learning approach for the classification of engineering CAD models. The aim is to catalog models according to functional classes. To do this, a convolutional neural network is built, which takes in input different 2D views generated from the CAD models. In [112] a shape-based retrieval method of mechanical parts is developed. It uses voxels to represent the 3D models, and divides it into several subspaces. The entropies of all subspaces are then calculated and constitute the feature vector exploited to classify the parts. Rucco et al. [94] present a supervised artificial neural network system which can classify 15 subcategories of mechanical parts. The recognition is based on different shape descriptors, such as spherical harmonics, geometric statistics, inner distances and shape distribution.

It is evident that both procedural and artificial intelligence classification approaches have been implemented over the years. Both machine learning and rule-inferencing can be used for semantic enrichment. The approach should be chosen according to the nature of the problem context. While machine learning is suitable for objects with less distinct or undefined characteristics, objects with well defined and predictable features can be identified using rule-inferencing [29]. However, in general the above works can reliably distinguish models with different geometries, but can fail when models with similar shape and completely different functionalities have to be identified. In the engineering domain, in fact, parts with same shape may perform multiple functions (see Section 1.3.1). Thus, the membership to one class rather than the other mainly depends on the context of use of the parts in the assembly [65].
The functional and semantic classification of mechanical components considering the context is still an open issue that has been investigated in recent years. Jian et al. [67] try to overcome the limitation of traditional shape-based classification methods by exploiting non-geometric semantic information, such as tolerances, surface roughness, material and function, which are product manufacturing data, not always available in the assembly B-rep CAD model. Foucault et al. [52] propose a method that, only relying on geometric data and engineering knowledge formalization, automatically infers functional and mechanical information. In particular, conventional simplified representations and conventional interfaces between parts, typically employed by designers, are used to characterize threaded linkages and, hence, screws. The properties can be easily translated in geometric conventions, and are thus exploited as criteria for the identification of these assembly features and standard components in raw geometric CAD models. Lupinetti et al. [80] focus their attention on the identification of a single specific mechanism within assemblies, in order to enhance the mechanical function knowledge. Even if the work is

targeted at the recognition of only rolling bearings, its strength is in the capability of detecting bearings independently on their design level of details. In [82] a shape-and-context-based classification is provided. The work is limited at the identification of some specific functional sets in speed reducers, but it underlines the importance of a multi-step approach. In particular, it is evident that the context of use of a part in the assembly is crucial to overcome misleading situations and not correct classification deriving from recognition methods solely based on shapes.

The work presented here, and in particular the part recognition module (see Section 4.2), is placed in this line of research and aims to overcome some of the limitations and improve existing approaches. It provides a feature based approach focused on the recognition and classification of a set of standard parts considering both their shape characteristics and the context in which they are assembled. The identification of these specific and engineering meaningful components is a strength of the work, since it facilitates the subsequent interpretation and processing of the CAD model and enhances its semantic value. Moreover, a further distinctive feature, that is missing at all in the already published approaches, is that the algorithm not only assigns the assembly's parts to standard macro categories, but it distinguishes among more precise subcategories and returns a characterization through dimensions relevant in the mechanical engineering field.

2.3 Assembly planning problem

In industrial manufacturing the production of mechanical assemblies is demanding in all its stages. Due to the increasing complexity of products, design, management, and end of life phases are becoming very onerous and time consuming tasks [61]. In order to remain competitive, manufacturers need to speed up the development time, minimize the manufacturing cost, as well as to find solutions to reuse products or parts of products. Among the most expensive and time-consuming operations that constitute the manufacturing process, the Assembly Planning (AP) stands out. It is proved that assembly operations consume 40-60% of the total production time and 20-30% of the overall production cost [20].

As a consequence, in the last decades, technologies have been studied to facilitate and automate the different manufacturing operations, and in particular the assembly and the disassembly planning processes have drawn the interest of many researchers. In fact, the ability to automatically derive and plan the assembly sequence before the product is effectively mounted/dismantled would be beneficial. It can significantly contribute, for example, to optimize the assembly sequence according to specific aims, to ensure parallel production lines and promote human robot collaboration, and finally to improve the overall product design [73, 93]. A summary outline of the major assembly-related issues and associated scopes in product development stages is reported in Figure 2.1.

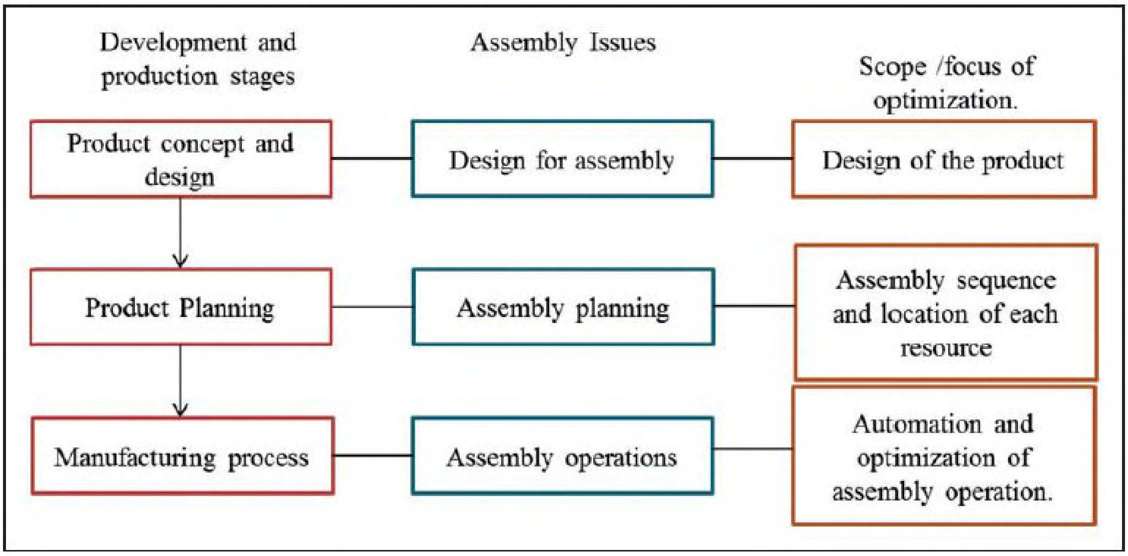


Figure 2.1: Assembly-related issues in the production stages (Figure 1 in [46]).

In this regard, Assembly Sequence Planning (ASP) is considered one of the most challenging topics in the industrial manufacturing field, and often it is also treated in conjunction with Disassembly Sequence Planning (DSP) and Subassembly Identification (SI).

Many works have been presented in this direction from the 80's, and it is evident that different methodologies have been proposed over the years. Comprehensive surveys can be found in [20, 68, 73, 109].

This section focuses on two specific facets of assembly sequence planning approaches, that is, the strong connection with the subassembly identification problem and the collision detection approaches. In the following, for sake of completeness, a brief overview of the existing ASP methodologies, their common features and the main strategies adopted is first reported. Then, the definition of subassembly as it can be found in literature is presented. Finally, some interesting works dealing with the treated aspects are discussed more in details.

2.3.1 Overview of the assembly sequence planning in literature

The assembly sequence planning process aims at algorithmically identifying the order in which components have to be assembled to obtain the final complete and functional product. Since, basically, all possible combinations in which parts can be mounted should be evaluated, ASP is known to be a NP-hard combinatorial problem, especially when the assembly parts numbers become important [93]. Moreover, multiple solutions can exist on how mounting components with each others, but the selection of one sequence rather than another has great effect on assembly feasibility, complexity, and accuracy.

As a consequence, to reduce the complexity of the problem, a largely employed strategy is that of the *assembly-by-disassembly* [56]. In practice, assuming that the assembly sequence can be derived exploiting the reverse of the disassembly sequence, it is possible to tackle the ASP problem starting with the 3D model of a complete product. This implicitly includes the constraints on its components, both relative to parts' contacts and assembly operations, and this would reduce the range of assembly motion a planner has to consider.

More in details, given a CAD model of an assembled product, the preliminary steps expect to extract the contacts and the precedences relations between parts and to include those data in matrices or graphs. At this purpose, the earlier provided techniques, which can not be supported by advanced technologies and hence are more theoretical, are based on the manual definition of liaisons graphs and of the precedences relations. These latter, in particular, are identified by asking the user on the feasibility of the assembly operations and then ordering the (dis-)assembly of components based on the yes/no answers [42, 43, 45]. Successively, the computation of the parts relationships, and especially of the precedences, has been more automated. Mainly thanks to the improvement of computer-aided algorithms and the availability of the APIs of commercial CAD systems, semi-automatic collision detection analyses are implemented, that simulate the movement of parts and evaluate their intersections [88, 102, 106, 107, 110]. Once the necessary data structures are obtained, the sequencing process is performed. It consists in the actual definition of at least a feasible sequence according to which (dis-)mount components and the establishment of the paths to follow. Due to the fact that the extracted information are provided as graphs and associated matrices, first approaches are based on graph-search theory and cut-set algorithms [23, 26, 44]. Then, in recent years, significant work has been done in the application of artificial intelligence (AI) and soft computing techniques to ASP problem [46], among which neural networks

[36, 99], simulated annealing [22, 97], and genetic algorithms [19, 38, 86, 100] are included. The peculiarity of these methodologies is that they are overall powerful optimization paradigms able to treat assemblies more complex than those considered in the graph-search and knowledge-based methods. Moreover, they allow not only to find feasible sequences, rather to identify optimal sequences according to one or multiple specific objectives (e.g. minimization of directional changes, reduction of tool changes, reduction of time, etc.).

Despite the exploitation of the simplification strategy above discussed, the ASP process remains a challenging and computationally expensive task. Especially with the increase of the number of parts the size of graphs and matrices increase exponentially and become more heavy and complicated objects to compute and then manage. To further reduce the complexity of the task, assembly sequence planning based on *subassemblies identification* has demonstrated its suitability to limit the amount of parts to be considered at the same time [34, 105, 109]. The idea is to break down the assembly into groups of connected parts which can be treated independently of one another, avoiding to work with all assembly's parts simultaneously. It is engineering reasonable, in fact, to plan the mounting of a complex product by functional modules. Operationally, each of the components can be produced separately, and then all of them are joined to make the final product. This means that the ASP algorithms can be applied to the single subassemblies instead of on the whole assembly, visibly reducing the computational cost and the data structures dimensions.

In the following, before reporting more in-depth some of the most interesting works in the ASP field, a brief digression is made on subassemblies and the different definitions that can be found in literature.

2.3.2 The subassembly identification problem

Subassembly is a widespread concept which has been studied and leveraged in many works in the last decades.
It is to clarify that subassembly identification can be leveraged in several phases of a product life cycle. For instance, in the design phase, for the identification of reusable components [115], or else in maintenance and recycling operations, to reduce waste of resources [108]. However, the section will mainly focus on subassemblies for the (dis-)assembly sequence planning task.

From the historical point of view, one of the pioneering works in which the subassemblies detection in CAD models of mechanical assemblies is addressed, and it is also seen as a crucial task in the assembly sequence planning, is that of Dini and Santochi [49]. There the authors give a precise definition of subassembly, which is also an operative definition in the sense that the essential features that must be found in a subassembly, and thus in its CAD model, are listed. Namely:

Given an assembly of n *components, subassembly means the set of* m *components* $s_1, \ldots, s_m$ *(with* $2 \leq m \leq n-1$*) that can be separately assembled before the final assembly of the product. [...] The essential features of a subassembly are:*

1. *each element of a subassembly must be in contact almost with another element of the same subassembly;*

2. *the set of elements representing a subassembly must constitute a stable structure however it is oriented in the space;*

3. *after the assembly, a subassembly must not prevent the other elements to be assembled.*

In summary, the three conditions imply that a subassembly is a connected set of parts, it can not interfere with the other parts of the assembly in the assembly process, and it is stable, that is, if it is manipulated, it won't dismount and parts will maintain their mutual positions.

It results that the given definition is realistic, but at the same time it is highly theoretical and quite difficult to validate from the computational standpoint. The practical evaluation of all the requirements is based on a massive human intervention, since the parts of a CAD model have to be analyzed both in respect to each other and in respect to the entire assembly, and the tools adopted in that period were very coarse.

With the advent of more powerful technologies and the desire to automate as much as possible the manufacturing process many works deal with subassemblies and their identification starting from CAD models. However, depending on the specific focus and purpose of the single paper weaker conditions are assumed in the characterization of a subassembly. For instance, in most cases, condition 1) is the only one employed. This is the case of works in which the definition of subassembly is not explicitly formulated. It is assumed that a subassembly is a generic connected subset of parts grouped mainly considering contacts and constraints information [25]. In other manuscripts the definition of subassembly somehow takes into account the assembly/disassembly process. For instance, Ko et al. [70] assume that a subassembly is a collection of components and all the assembly tasks among them.

In [110] subassemblies are referred as *assembly units* made of less parts than the whole assembly, extracted considering the assembly process constraints. Watson et al. [111], instead, speak about *logical subassemblies* as sets of at least two parts gathered together addressing part access and assembly precedences. The notion of functionality is then included in [72], where a subassembly is established as a set of directly connected parts which achieves a particular function. However, the conditions of stability and independence from the rest of the models are not referred in the above definitions, or, if they are mentioned, they do not have to be necessarily fulfilled by a subset. Rather, they are than considered a discriminant factors for the evaluation of the resulting subassemblies and their feasibility.

On the contrary, other works adopt more complete and specific definitions, more similar to that given by Dini and Santochi. Among them, Cao et al. [34] consider subassemblies as groups with stability and independence. They are made of more than two parts with steady mutual positions and having at least an assembly direction between them. Zhang et al. [115] refer to *significant subassembly* and give a definition including five characteristics, two of which are distinctive of subassembly, i.e. connection of parts and independence of the subassembly from the rest of the product, while the remaining three measure the level of significance (in the reuse context), i.e. reusability, contained information, and complexity. Kou et al. [71] define a subassembly as a collection of components with specific relationships, and independence from the rest of the assembly. In addition the presence of a base part is mentioned, as well as the stability for some parts.

Finally, in some cases the concept of stability can assume a different meaning. In [21], for instance, the authors assess that a subassembly is indistinctly every group of connected parts and it can be partially stable, if at least one component is not totally blocked, or permanent stable, if components maintain their positions irrespective of orientation. In [90] the subassemblies are stable when the freedom of each part is zero. Other papers evaluate stability based on the type of contact among parts, that is to say that a subassembly is stable if parts are fixed with connectors. In this perspective, Agrawal et al. [16] assume stable subassemblies to be sets of components which have fastened contacts and can not be easily removed individually; they adopt stability as the criterion for choosing some subassemblies among those identified rather than others. Smith et al. [101] state that a subassembly is stable if all the parts are supported by connections to adjacent parts or are gravitationally stable. Finally, stability can be seen as a quantitative measure. For example, Dong et al. [50] define stability through an index calculated on how parts deviate from their correct position while removing connectors.

It is evident that subassembly and stable subassembly concepts are indistinctly employed in literature with various meanings, and this is confusing. In this manuscript the definition of subassembly proposed by Dini and Santochi is assumed to be the most characterizing and complete. Given the difficulty in treating it computationally and automatically identify these objects starting from CAD models, an additional concept is introduced, i.e. that of *cluster*, whose detection is addressed, as it will be better explained in Section 5.1.

2.3.3 ASP and SI main works

From a general literature review it appears that the assembly sequence planning and the subassembly identification are closely intertwined topics. As a matter of fact, they are usually addressed one according to the other. That is, subassemblies are generated through the assembly order computation, or vice versa the sequence is computed according to the subassemblies first identified.

In the following, some of the most relevant works in ASP and SI are reported. The aim is to underline their key points, the input data used, along with the strategies adopted to address the two tasks, and, if both of them are addressed in the same work, how they are combined. For sake of completeness, also some works more centered on the collision detection analysis and the leverage of fasteners as particular parts in the ASP are listed. Finally, a summary of the main features of the discussed works is provided in Table 2.1.
It is to clarify that, because the assembly-by-disassembly methodology prevails throughout the literature that covers the assembly planning problem, both assembly ASP and DSP works will be indistinctly considered.

A work that emphasize the strong connection between SI and ASP is that of Kou at al. [71]. In fact, the authors state that one of the main challenges in the assembly sequence planning is actually the identification of subassemblies. As a consequence, they develop a subassembly identification algorithm in order to improve and simplify the assembly sequence planning, but this second task is not at all addressed in the paper. The method starts computing the weighted undirected connected graph to describe the relations between the parts of the assembly and assign different weights depending on the type of coupling (i.e. simple contact or fastening connection). Taking into account this information structure, then, the subassembly recognition is based on a mathematical model. It is mainly a clustering procedure that iteratively minimizes an objective function involving the distances of each part to the subassemblies centers. At each iteration the centers and the subassemblies are updated until the stop requirements are met. The number of centers, and thus the number of subassemblies, is manually established before the algorithm starts, and this limits the automation of the process. Moreover, such a mathematical approach is not entirely suitable for handling real mechanical assemblies, since many semantic information and engineering situations would be overlooked.

Always focused on the existence of base parts on which each subassembly is built, it deserves to be mentioned the approach described in [110]. There an assembly sequence planning approach is presented which is based on a subassembly identification strategy denoted as assembly unit partitioning. Assembly unit partitioning aims to decompose the whole product into smallest groups of parts taking into account significant assembly design constraints and assembly process constraints, such as the connection strength, the degree of freedom, or assembly directions. The units are generated starting from a weighted decision graph, involving the evaluation of all the constraints, the establishment of a given number of base parts and the subsequent aggregating to each base part of the components connected with a minimum acceptable weight.

Under the same hypotheses, a method is presented in [111] that generates successive subassembly decompositions defining a tree structure that makes the assembly sequence apparent, along with the possibility of carrying out parallel operations. The peculiarity of the method is that it prioritizes both assembly constraints and part access introducing the blocking fraction as an heuristic for estimating the obstructions between two parts based on the measure of the volumetric space where a part is free to be moved. In particular, after selecting the base part for the complete assembly, then subassemblies are recursively generated identifying a base part among the remaining components and growing the candidate subassembly by adding to the base part the adjacent parts that do not increase the blockage of the group.

Similarly based on the identification of the base parts, but making the management of the SI and ASP tasks independent, Trigui et al. [105] provide an approach that first computes the subassemblies of the given CAD model and then considers them as single units whose disassembly sequence has to be planned. In particular, the authors combine two separate researches, i.e. the one focused on ASP and the other linked to SI, to simplify the sequence planning by reducing the number of components to deal with simultaneously. In details, independently from the introduction of concept of subassemblies, the assembly sequence planning task are first addressed in an innovative way. In [27, 106] the algorithm for carrying out the interference analysis and evaluating the collisions between parts is described. A relevant feature of it is that the (dis-)assembly directions are established according to the parts contacts properties, and this is more realistic compared to the common use of the only x, y, and z axes. However, the collision analysis is performed in a simulation environment by displacing parts in the chosen directions, which is not a totally automatic operation. Furthermore, in [55] a significant simplification of the ASP approach is reported, which takes into account some semantic information contained in the CAD model, i.e. the presence of standard parts. These are first removed form the list of parts whose collision analysis has to be performed, and then inserted in the final assembly sequence according to some specific rules. The weakness of the method is that the standard parts are manually detected, and this is an onerous and error prone operation. As for subassembly identification, instead, in [25] an approach that builds upon geometric and topological information of the CAD model parts is presented. Namely, an enriched contact matrix evaluating the contacts according to the x, y, and z directions is generated. The number of contacts, along with the relative volume and relative surface of a part with respect to the whole assembly are considered in an objective function computation that classifies the assembly's components and determine the base parts that underlie the subassemblies. The base parts are then removed from the liaison graph of the CAD model to disconnect it into smallest units, i.e. the searched subassemblies. Finally, combining the works in the two different fields, in [105] the obtained subassemblies are considered as single units and their interference matrices are computed to evaluate if a subassembly intersects with the others during its removal along the three orthogonal directions. This allow to provide the disassembly sequence for the subassemblies and thus notably reduce the interference analysis to a limited number of components.

A different methodology, that combine the subassembly detection and the assembly sequence planning issues and solve them as a single task, is presented in [21]. In particular, subassemblies are generated through the concatenation of parts according to their contacts and, then, evaluating their level of stability when mounted together. It is evident that, in this work, the assembly sequence planning and the subassembly identification tasks are simultaneously addressed. However, the concept of subassembly reported here is very general, and no engineering and realistic considerations are made on parts unless that of stability. Also, the ASP analysis just returns an order in which parts can be mounted, but it is mainly a static analysis. In fact, no movement directions or parts collisions are mentioned.

Another method that proceeds by concatenation of parts to generate subassemblies and compute the disassembly sequence for an assembly is that of Zhang et al. [116]. In the specific, the approach involves the definition of the precedence graph, where the precedence constraints between parts according to the x, y, z directions are collected. At each iteration a disassembly subset is generated by aggregating parts which result free to be removed according to the precedence relations. In addition a swarm algorithm is then employed to select the optimal sequence, that should minimize the directions and tools changes. Compared to the previous work, more engineering information here is considered (e.g. the use of different tools), but this is manually provided, as well as precedence constraints computation is not specified. Furthermore, strict hypotheses are pointed out, such as the assumption that all the components are rigid and contacts are perfect.

As for the development of collision analysis and a more targeted exploitation of fasteners and their functions, the work of Agrawal et al. [16] is meaningful. The authors presented an approach that is focused on computing the assembly sequence planning for a product, but it simultaneously takes into account a particular type of subassemblies. Starting from a CAD model in standard format, the undirected graph representing contacts, i.e. the liaison graph, as well as the directed graphs representing parts obstructions during their movement along the $\pm x, y, z$ directions, i.e. the blocking graphs, are generated. To this second aim a simple collision analysis is performed by projecting rays from the boundary points of the faces of each part and evaluating if they intersect any of the faces of the other parts. In the affirmative case an arc is added going from the first part to the one intersected in the graph associated with the evaluated direction. The blocking graphs are the only tools used in the assembly sequence generation. If a part is not connected to the other in one of the graphs, it can be removed. If instead a set of parts block each other in all the graphs, they will be removed as a unit. The units are actually the subassemblies the authors claim to identify. In details, these are computed by exploiting fastener information as input and grouping into a unique set (an thus in a unique node) all that components fastened together. It is evident that this strategy is useful to reduce the size of liaison and blocking graphs and simplify computations. Moreover, it suggests the importance of fasteners in mechanical assemblies. However the hypotheses assumed to define both the subassemblies and the sequence are too simplified and not reliable at all, since they can be applied in few real scenarios.

Neb et al. [88], instead, underline the importance of specific mechanical parts in the assembly sequence generation. The authors take into account the existence of categories of parts, e.g. screws, nuts and circlips, that must be treated in a different manner in the collision analysis, because they have a particular well-known function in the assembly, and thus a precise relation with other parts. However it is assumed that the type of the parts is known. Similarly, in [76] the authors present a method to generate feasible disassembly sequences exploiting the knowledge of connectors. In particular, the behavior of threaded fasteners and keys is analyzed within the assembly and corresponding precedence rules are extracted. The discussion is although limited to these two types of fasteners, while, for example, circlips and elastic parts are not considered, and their identification is given for granted.

Finally, intending to generate more realistic details, especially those regarding the possible directions of movement for components during the (dis-)assembly, Tao et al. [104] introduce the use of Gaussian spheres to define the set of all feasible moving directions of each part. Each contact implies a sphere with a constraint. The intersection of all the constrained spheres associated with a pair of parts' contacts returns the possible moving directions for that parts, that are then not limited to the orthogonal axes.

Table 2.1: Summary of some of the most relevant (dis-)assembly sequence planning and subassembly identification approaches.

Ref.	Field	Input	Use of subassembly	Base parts (BP)	Fasteners	Contacts computation	Contact graphs	Contact matrices	Subassembly computation	Collision computation	Precedence graphs	Precedence matrices	Sequence computation	Semantic information	Directions	Assembly by disassembly	Parallel assembly
[71]	SI	CAD model	Yes	N BP (N in input)	In input	In input	Weighted undirected graph	Adjacency matrices	Min distance from base parts	No	No	No	No	No	x,y,z	No	No
[110]	SI ASP	CAD model	Yes	N BP (N in input)	In input	Manually (contact and constraints)	Decision weighted graph	No	Append parts to BP by graph search algorithm	Manually	Decision weighted graph for subass.	Interference matrix	Assembly sequence merging from subass. seq.	Fasteners + design and process constraints	x,y,z	Yes	Yes
[111]	SI ASP	Mesh model	Yes	1 BP (wirh max Vol) + nuclei for subass.	No	Method in [114]	Disassembly influence graph	No	Append parts to BP according blockage value	Evaluation of the volumetric removal space	Disassembly influence graph	No	Precedence layer based on subassembly computation	No	Free removal space	Yes	Yes
[105]	SI DSP	CAD model	Yes	N BP (N in input)	From the assembly tree	Extracted with API	Liaison graph	Simplified adjacency matrix	Cut liaison graph starting from BP	No	No	Interference matrices	For subass.: zero row/col in matrix	Fasteners	x,y,z	No	No
[27] [55] [106]	ASP	CAD model	No	N BP (N in input)	From the CAD system	No	No	No	No	Displace part in given directions with API	No	No	Simplif. seq. + rules for including fasteners	Type of fastener + info on their mounting	Depending on mating faces	No	No
[25]	SI	CAD model	Yes	N BP (N in input)	From the assembly tree	Extracted with API	Liaison graph	Simplified adjacency matrix	Cut liaison graph starting from BP	No	No	No	No	Fasteners	x,y,z	No	No
[21]	SI ASP	CAD model	Yes	No	In input	Surface contact	No	Liaison and stability matrices	Append parts to existing subset	No	No	No	Given by the order of subass. generation	Fasteners	No	No	Yes
[116]	SI DSP	CAD model	Yes	No	No	No	No	No	Append parts to existing subset	No	Precedence graph	No	Sequence of parts with no predecessor + swarm algorithm	No	x,y,z	No	No
[16]	SI ASP	STP file	Yes	1 BP (with max connections)	In input	Faces in contact	Liaison graph	No	Connected by fasteners + cycles in graph	Projection of rays	Blocking directed graph	No	By free nodes in blocking graph	Fasteners	x,y,z	Yes	Yes
[88]	ASP	CAD model	No	No	In input	Extracted with API	No	No	No	Extracted with API	No	Restriction matrix	Evaluate the restriction matrix + fasteners knowledge	Type of fasteners + info on their mounitng + gravity	Depending on contacts	Yes	No
[76]	DSP	CAD model	No	No	In input	From the CAD system	No	Contact matrix	No	Geometric reasoning	Disassembly constraint graph	No	Graph search algorithm	Type of fasteners + precedence rules	No	No	No
[104]	ASP	Mesh model	No	No	No	Detection of plane/line/ point contacts	Contact relation graph	Contact relation matrix	No	Manually	No	No	Remove from graph the parts with no interferences	No	Intersection of the Gauss spheres given by contacts	Yes	Yes

2.4 Conclusions

Through an extensive analysis of the research works dealing with the digitalization and automation of industrial manufacturing phases, in particular those addressing the assembly planning, some general needs to bridge has been pointed out.

The first general weaknesses that unite most works regard the only use of numerical structures and the development of mathematics-based algorithms to represent and resolve complex assembly tasks.
This is limiting because both engineering meaningful information are omitted and most of the considered hypothesis are rarely applicable to real scenarios, thus affecting the applicability of the results in industrial settings. The issue can be seen, for example, in considering fasteners and locating components as any other part, ignoring their function, which instead would be beneficial both in SI and ASP. Also, the use of base parts as reference components for the assembling on which subassemblies should be mounted and their mathematical identification according to some decision values are not totally realistic. That is, the presence of base parts, their number, and their characteristics can not be generalized, rather they depend on the type of product evaluated. Furthermore, the collision detection is often too limited by considering only the orthogonal directions x, y, z for the movement, while the definition of sequences involves all the assembly parts simultaneously, regardless of the existence of functional sets, and does not handle challenging components, such as the deformable ones.

On the contrary, when instead high-level information is considered and leveraged within the approaches, such as the different types and the functionalities of parts, its knowledge is taken for granted and there is no reference to the difficulty in obtaining it. That is to say, the existing works taking into account semantic data assume that those data are already included in the input CAD models or they are manually provided by experts. This can be seen as an issue to overcome. In fact, on the one hand, high-level information are rarely included in CAD models. On the other hand, when this is present it is provided as attributes in the form of strings, that may depend on company or designer choices not following standardized rules thus requiring to be interpreted, and consequently the necessary human intervention limits the automation of the approaches. In addition, the only awareness of screws and, sometimes, keys are usually the enriching information involved, but more details should be considered to provide reliable solutions and tackle realistic scenarios. Examples are the possible presence of deformable parts, weldings, as well as the several type of fasteners beside screws.

The system presented here, that will be deeply discussed in next chapters, aims to address the challenges listed above by tackling them in an innovative way, trying to overcome the problems and optimize some aspects of existing approaches.

3

System Overview

This chapter provides a general overview of the approach developed in this thesis aimed at automatically enriching product CAD description to enhance product development activities, with a particular focus on assembling tasks. The main ingredients that constitute the foundation of the approach proposed in the thesis are outlined in the next chapter. These are engineering knowledge, design expertise, and geometric processing techniques. After presenting the way these are meant to be gathered up, the framework structure and its organization in different modules are pointed out. The discussion, then, focuses on motivating the choice of the liaison as new data structure that stands at the basis of the enriched representation of product models, and theoretical definitions of the key concepts are provided. Thanks to the introduction of liaison, the developed system can automatically extract, store, and leverage semantic information from CAD models starting from purely geometric data.

3.1 The proposed approach

The proposed research aims at developing a tool suite to automatically process CAD models of mechanical products, extract high-level semantic information, and then exploit it to ease several manufacturing phases such as the assembly planning or the design for assembly on which this thesis focuses.

Existing approaches mainly treat a mechanical assembly as a collection of geometric objects respecting proximity relations and address complex engineering tasks through mathematics-based methodologies. This way of working is restrictive, since information associated with parts is lost from sight and results can't explicitly reflect engineering experience and assembly knowledge, factors that instead should strongly affect the outcomes and simplify the processes.

To overcome these limitations, the peculiarity of the proposed tools stands in exploiting the geometric and topological analysis in combination with engineering knowledge to reproduce, by means of an automatic process, the human capabilities to understand and interpret the parts of the assembly and their relations, and then to address complex assembly tasks.

When dealing with a mechanical product, in fact, an expert can recognize the membership of certain parts to specific categories, understand the functionality of components according to their positioning, distinguish between different mounting techniques, and also detect groups of parts that share some characteristics (e.g. all the parts fastened together by screws, a welded set, etc.). All these data are at the base of the manufacturing operations planning and execution. Moreover, they are instinctively and naturally inferred by engineers just looking at the product, because they refer to basic notions that are learned with study and experience in the field.

On the contrary, as described in Section 1.3, when loading a 3D model in a CAD system, none of the above data should be taken for granted, since most of them are not available unless a bill of material (BOM) is provided, parts' names refer to parts' functionality, or some descriptions are added as attribute. Even in these cases, the information can depend on designer or company conventions, thus not fully trustful. In addition, human intervention is in general required to manually attach or read these attributes and, furthermore, it should be borne in mind that information may be lost due to file format conversion or switching from one system to another. As a consequence,with the final aim to computationally address assembly tasks in an automatic way starting from a 3D model of a mechanical product, the parts interpretation and assembly understanding become a crucial phase, and this thesis proposes to tackle it algorithmically simulating engineers reasoning.

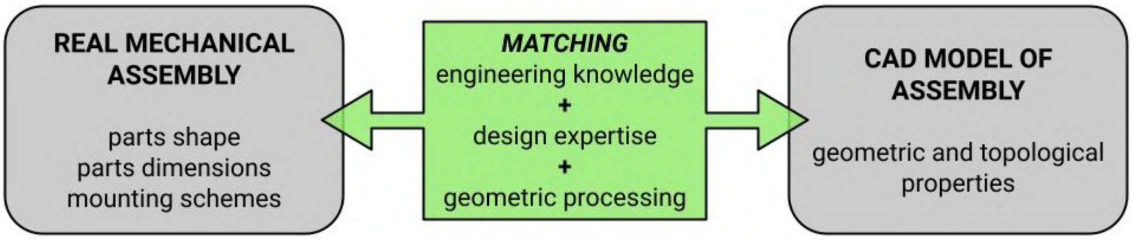


Figure 3.1: Diagram of the matching process between real mechanical assembly and its CAD model.

The guiding idea is to create a matching between the shape features, the dimensional relations and the mounting schemes strictly governing real mechanical assemblies and the geometric and topological properties that can be retrieved in CAD models of assemblies (Fig. 3.1). For this purpose, engineering knowledge is applied to assess and point out those primary and consolidated laws typical of mechanical components; design expertise allows to convert the rules into geometric and topological requirements, geometric processing allows to extract the necessary information from the components' B-rep models.

Gathering up these three ingredients it is possible to define an automatic rule-based approach able to both understand and interpret CAD assembly components, and create engineering meaningful relations between parts as an expert would intuitively do in reality. The semantic interpretation of a geometric model, allows then to optimize existing methodologies and obtain more realistic and feasible results in operations such as sequence planning and subassembly identification, or else DFA analysis.

3.2 Structure of the system

From the implementation point of view, two distinct phases can be distinguished that characterize the approach, depending on whether the CAD model taken in input has been already processed and interpreted or not. Namely, the phases will be referred to as the *data extraction* and the *data exploitation*. The two are strictly connected since, in the provided flow, the data extraction is mandatory for the data exploitation. As it will be better explained, both the phases are then made up of different modules. Each module can fully perform a specific operation and it is also preparatory for later ones. Figure 3.2 summarizes the topics addressed in the two phases through the different modules. Respectively, the orange colored boxes refer to the data extraction phase, while those colored blue refers to the data exploitation. In practice, the link between the two phases lies in a new representation of the

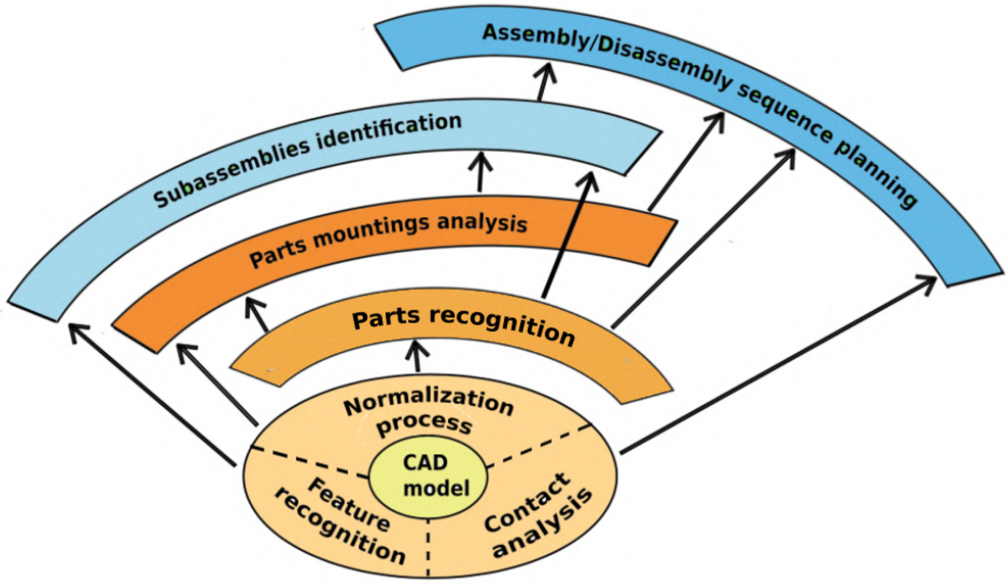


Figure 3.2: Overview of the topics addressed in authors' research. In orange those included in the data extraction phase; in blue those included in the data exploitation phase.

assembly according to which the semantic data extracted in the first stage can be stored in a meaningful manner and, then, can be read in an intuitive way in the second stage. Hence, a crucial point is in the definition of a data structure called *liaison*, which stands at the base of the new semantic enriched assembly representation.

In the following, a general overview of the data extraction and the data exploitation phases, i.e. their structure, the input and the resulting outputs, is briefly discussed. The concept of liaison is then explained and defined in a separate section, since it deserves to be deeply described.

3.2.1 Data extraction phase

The data extraction can be seen as one of the strong points of the presented system. In fact, in existing approaches this activity is only partially addressed and the inclusion of part of the information crucial for the automation of production processes is manually carried out exploiting commercial CAD software's tools. However, it is evident that to provide a standalone and comprehensive system to be applied in multiple assembly contexts, regardless the availability of a tool rather than another, without human intervention, a built-in data extraction phase is necessary.

This crucial stage aims at automatically inferring high-level semantic information from the CAD model of a mechanical product. This would aid the comprehension of the geometric object and allow its understanding from the engineering point of view.

In details, the data extraction phase starts taking in input the boundary representations of parts, that combined make up an assembly. To guarantee a wider applicability, the input is given in standard formats for file exchange, such as STEP files. In this manner the shape of the mechanical product is totally defined by the bounding elements (i.e. faces, edges and vertices) of the occupied volumes of its parts that are placed in the space according to the absolute reference system, and no details about parts' constraints are explicitly available (as discussed in Section 1.2). As a result, some processing of the component models is required, which includes both the most common tasks to extract essential information and more focused tasks to semantically interpret the whole assembly. The necessary operations are carried out step by step through the development of three different modules, namely:

- **Geometric processing**. It deals with the inspection of the geometric and topological entities that constitute the parts of the assembly. These are analyzed both singularly and in relation with each other. The main focus is the geometric classification of components according to the faces' arrangement and properties, the identification of characteristics related to single portions of the geometry of each part (i.e. geometric features), and the detection of faces belonging to different components but in close proximity with each other.

- **Part classification**. In this step a semantic meaning is associated with the single parts of the assembly. The previous extracted geometric information is exploited to classify each component according to engineering criteria. Namely, in accordance with the definitions in Section 1.1.2, the parts are categorized into two macro classes, custom designed parts and standard parts. The latter, in turn, are organized into different categories based on the type and functionality.

- **Mounting analysis**. This module thoroughly analyzes the pairs of parts in contact with the aim of enriching the semantics of contacts outlining information that can indicate the type of mounting. This is possible thanks to the knowledge of the different classes of parts and their geometrical characteristics.

3.2.2 Data exploitation phase

The data exploitation phase leverages the information previously extracted and organized in the newly defined assembly representation to derive knowledge directly usable in production processes. In particular, it exploits the semantically enriched assembly representation to address some of the widespread studied assembly operations in a groundbreaking way. That is, some of the most relevant engineering characteristics of the mechanical product, that are usually overlooked or manually provided, are taken into account and used to provide more realistic results.

As for the covered tasks, the tools that have been developed in this thesis are mainly placed in the assembly sequence planning field. In particular, the computation of a feasible assembly sequence based on the identification and use of meaningful subassemblies is the addressed challenge. Furthermore, such information together with the semantic enrichment of the product components obtained in the previous phase, can be seen as a first step in improving the efficiency of the DFA (Design for Assembly) analysis, as described in Section 5.3. In summary, in this work, the data exploitation modules considered are:

- **Subassembly identification**. This module aims to group the assembly's components into engineering meaningful subassemblies. They are denoted as *clusters*, and are identified relying on some heuristics that evaluate the type of mounting and exploit the extracted semantic information.

- **Sequence planning**. This module seeks to define a mounting sequence for assembling the parts to obtain the final product. To carry out this task all the information retrieved in the previous stages are needed. The idea is to first compute the sequence for each cluster, taking advantage of their specific characteristics, and then define a mounting order between the subassemblies themselves and the remaining not clustered parts.

- **Design for assembly**. This is a more theoretical module that aims to investigate a potential automation of DFA analysis by taking advantage of all the information extracted from the original CAD model through the described modules. To this aim the correspondence between the results obtained and the DFA parameters is verified.

Before going deeper into individual modules, some preliminary concepts are given and the *liaison* structure is now introduced and defined.

3.3 The Liaison structure

As previously discussed in Section 1.2, when importing a model of an assembly in standard format in an ordinary CAD system, usually the only information that is definitely available is the list of the parts, represented by means of their topological and geometrical entities. The assembly's components can be typically presented according to a hierarchical tree, where parts are grouped in subassemblies respecting some parent-child relationships or in groups that follows some logical criteria (e.g. same material, same function, same mounting technique, etc.), or else they can be all at the same level, like in a list. However, the existence of the grouping and its characterization depend on the designer choice and on the importing/exporting operations. It is, thus, not necessarily reliable and meaningful in the engineering sense, e.g. from the assembly sequence standpoint. In addition, this representation does not explicitly describe the contacts between pairs of parts and their properties. These might be implicitly contained in the tree, but they must be computed by means of surfaces or volumes proximity evaluation to be available. Also, parts' type and the functionality are unknown, unless some names may refer to them or else codes are added as descriptions, but it is not mandatory and human intervention is needed to insert and interpret them.

3.3.1 The need for a new assembly representation

On the basis of the above premises, a novel representation of the assembly deserves to be identified, that, once the data extraction is carried out, takes into account the categorization of the assembly's parts and makes explicit all the meaningful information of their contacts.

The new representation is meant to integrate and provide product model and assembly process information in a unique object that meets the requirements of completeness and ease of use, along with intuitiveness in the visualization. Respectively:

- **Completeness** concerns the capability of combining in the same object both geometric data (e.g. surfaces' orientation, surfaces' area, volumes' extension, distances between surfaces, etc.) and high-level information (e.g. category of the parts, typical dimensions of engineering, assembly techniques, etc.).
- **Ease of use** refers to the data retrievability facility at the computational level. It implies the availability of the semantic data stored in the new representation avoiding further computations and in the possibility to leverage them in the data exploitation phase through simple queries.
- **Intuitiveness** concerns the user's ease of reading the data. In particular, it means that the new representation is structured in such a way that allows the

visualization of semantic data and the understanding of the relations of the parts in an intuitive way, trying to include all the information that an engineer would deduce from the observation of a real mechanical product.

A common strategy adopted in literature to represent the relationships and the constraints between the parts of an assembly is the creation of matrices or graphs [109]. As for the first case, indicated as M_n the set of square matrices of dimension n, where n is the number of components in the assembly, the mostly adopted matrices are of three types, and can be found with different names:

- Adjacency/Contact/Liaison Matrix: $A \in M_n$.
  A is a symmetric matrix, where the element a_{ij} represents the existence of the contact between parts i and j.
  Element a_{ij} usually assumes values 1 or 0, in some cases it can be an integer equal to the number of relationships between the two parts.
  The matrix can be transformed by considering the contacts according to the three axes x, y, z separately and making the element a_{ij} a 3-digital array.

- Constraint/Collision Matrix: $C \in M_n$.
  Element c_{ij} can be a 3-digital or 6-digital array representing constraints between parts i and j along the directions $d \in (\pm x, \pm y, \pm z)$ of the coordinate system of the assembly. $c_{ij}^d = 1$ means component j stops component i if moved in direction d, whereas $c_{ij}^d = 0$ means component j does not stop component i along d.

- Stability Matrix: $S \in M_n$.
  The element s_{ij} represents the stability or the type of fastening between any pair of components. Usually are distinguished permanent stability due to external connectors, permanent stability due to mating features, partial stability or unstable pairs, with $s_{ij} \in \{0, 1, 2, 3\}$.

The content of the above matrices can be equivalently stored in graph structures. Each assembly part is a node of the graph and the information extracted from the CAD model are included in the edges and in their attributes. The standard graphs employed are the following:

- Liaison Graph: it is the representation of contact information between any pair of parts, corresponding to the Adjacency Matrix. Two nodes are connected by an edge if the respective two parts have contacting faces.

- Blocking/Precedence Graph: similarly to the Constraint Matrix, it provides information about the blocking relationships within a component for a given direction (mainly the x, y, z axes) of assembly.
  The blocking graph is a direct graph where the predecessors of a node are blocked by the corresponding node along that direction.

These graphs can be enhanced, for example, making them weighted graphs. In the simplest case, weights are given by the type of contact (i.e. $w = 2$ if the parts are fastened by connectors, $w = 1$ if parts have only contact coupling relationship, $w = 0$ if parts are not in contact), and represent the same data expressed by the Stability Matrix. In more specialized cases, weights are calculated based on the evaluation of different factors, such as the combination of functional, structural and process constraints.

These structures have aroused great interest over the years because they can be managed as computational objects and then given as input data to well-known algorithms to address different assembly tasks. However the weaknesses in that assembly representations are several. First, when dealing with assemblies made of hundreds of parts, matrices and graphs have big dimensions and the increase of computational time and costs is the immediate consequence. Secondly, matrices are too abstract structures that can not comprehensively describe the contact between two parts, both from the geometric and the engineering point of view. Moreover, the data stored are not at all intuitive to read, since even high-level information is associated with a numerical value.

At this purpose, to overcome the limitations and semantically enhance the representation of on assembly, the key idea is to represent it as a list of elements defined as *liaisons*, each of which identifies a couple of mating components. It is to underline that in general the term liaison is referred to the simple contact between the components, but in this work, the liaison concept is intended in an extended way, more similar to [103]. Namely, a liaison is defined as a new data structure that totally express the relation between two mating parts of the assembly. That is, a liaison provides high-level semantic information concerning multiple aspects, from the geometry of the contact (e.g. type of contact faces, common axes, percentage of covered surfaces, etc.) to the assembly process features (e.g. mounting features, presence of connection elements, etc.).In this way, liaisons incorporate the information on the presence of connecting elements, such as bolt and screws, that in the case of adjacency graphs are normally represented as standalone elements.

3.3.2 Preliminary concepts for liaison definition

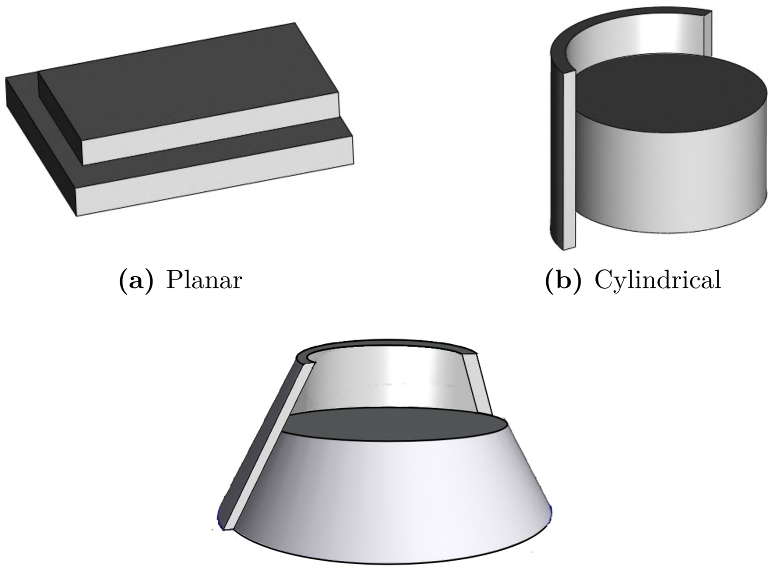
To better understand the liaison data structure and discuss its attributes, the principal concepts on which the definition of liaison is based have to be introduced. First, the concepts of coupling and mounting are defined. In particular:

Coupling

Coupling is the elementary concept underlying a liaison. It identifies the existence of a contact between two parts of an assembly. More precisely:

> **Coupling**
>
> Given two parts P_1 and P_2 of an assembly, a surface contact between a face f_1 of P_1 and a face f_2 of P_2 is called *coupling* and it is defined as $c(f_1, f_2)$.

A coupling can be planar (Fig. 3.3a), cylindrical (Fig. 3.3b) or conical (Fig. 3.3c) according to the geometric type of the faces in contact.



(a) Planar **(b)** Cylindrical

(c) Conical

Figure 3.3: Examples of coupling.

All the information associated with the two mating faces are included in the coupling and are readable properties. Among these the parts P_1 and P_2 to which they belong, their orientations and relative positioning in the space, the surface contact type, and the surface contact area.

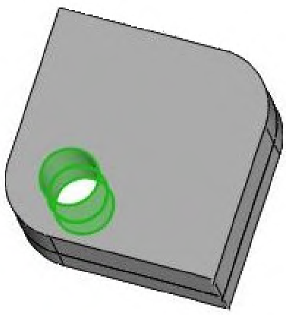
Mounting

Mounting is seen as an attribute of coupling. In particular, it identifies the existence of two coaxial features each having a border lying on one of the two mating faces that define the coupling. That is:

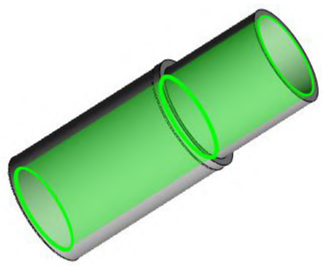
Mounting

Given two parts P_1 and P_2 of an assembly, and assuming the existence of a coupling $c(f_1, f_2)$ between the planar faces f_1 of P_1 and f_2 of P_2, the existence of two coaxial concave features, each having a loop of edges lying respectively on the faces f_1 and f_2, is called *mounting*. It is defined as $m(f_1, f_2, F_1, F_2)$, with F_1 and F_2 the list of faces of the features of respectively P_1 and P_2.

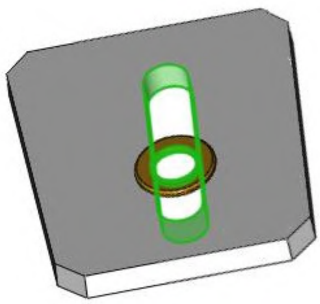
The concave features considered are cylindrical holes, straight or curve pockets and polygonal pockets. Some possible combinations of the features and their alignment characterizing a mounting are shown in Figure 3.4.



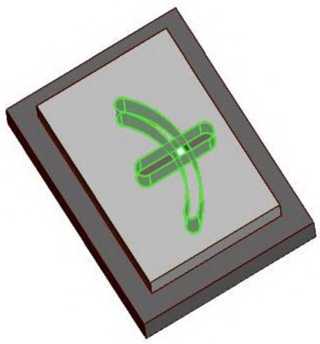
(a) Hole-hole



(b) Hole-hole



(c) Pocket-hole



(d) Pocket-pocket

Figure 3.4: Examples of mountings.

All significant and useful information associated with the faces involved in a mounting are stored as properties and are accessible. Besides the parts P_1 and P_2 to which they belong and their orientation and relative positioning in the space, the general dimensions of the mounting features (e.g. depth and minimal width) are reported, as well as a flag that indicates if the features are aligned with each other (e.g. the centers of two holes coincide, the center of a hole is on the central curve connecting the ends of a slot) or, in the case they are not aligned, the value of their axial misalignment.

A further key concept to be taken into account in the liaison definition is the distinction between custom designed and standard parts (see Section 1.1.2). In fact, liaisons are only meant to be established between custom designed parts, while standard parts are added as contact attributes.

3.3.3 Liaison Definition

Based on the above definitions and concepts, a liaison is then defined as follows:

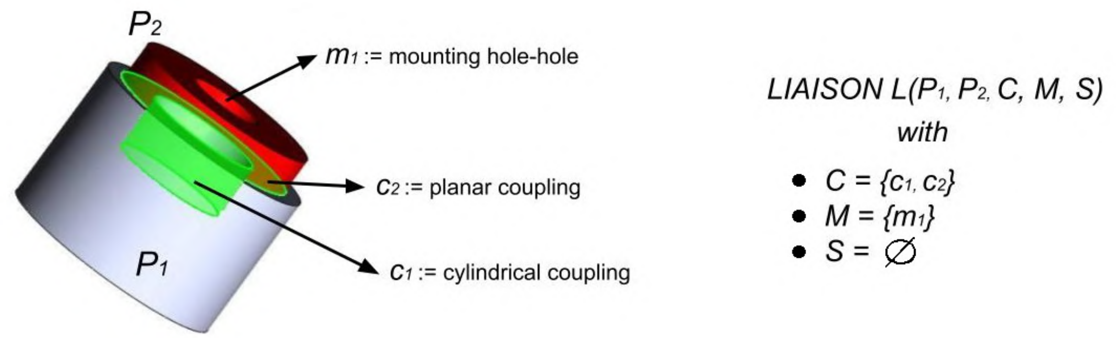
Liaison

Given two parts P_1 and P_2 of an assembly, such that P_1 and P_2 are custom designed parts and at least a coupling exists between P_1 and P_2, the liaison between P_1 and P_2 is defined as $l(P_1, P_2, C, M, S)$, where :

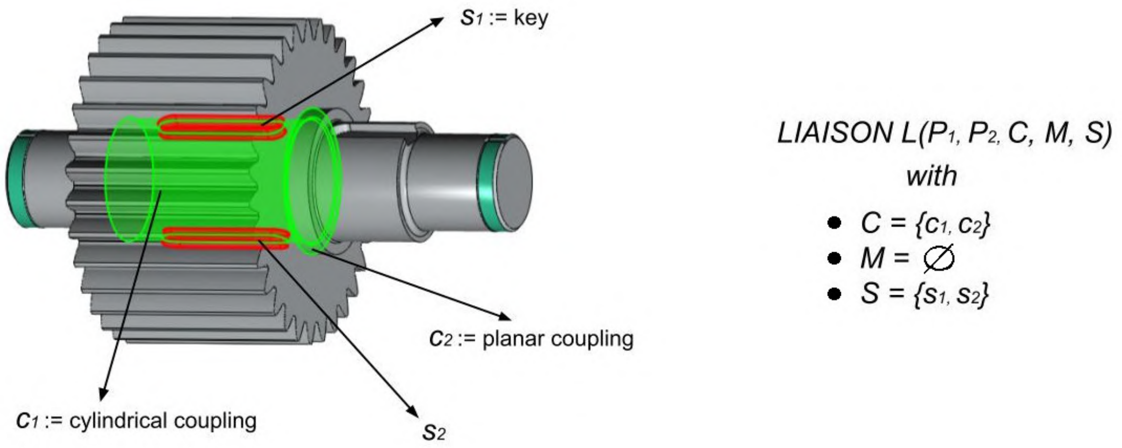
- $C = \{c_1, \ldots, c_r\}$ with $r > 0$ is the list of couplings between P_1 and P_2;
- $M = \{m_1, \ldots, m_s\}$ with $s \geq 0$ is the list of mountings between P_1 and P_2;
- $S = \{s_1, \ldots, s_t\}$ with $t \geq 0$ is the list of standard parts connecting P_1 and P_2.

The presence of a list of couplings, and not a single coupling, is justified by the fact that multiple faces in contact can clearly exist between a pair of parts, one for each couple of colliding faces (e.g. Fig. 3.5a, Fig. 3.5b, and Fig. 3.5c). Thus, in the liaison object a list of all the identified couplings is stored. As already said, each coupling c_i contains the references to a couple of mating faces and thus, to all their geometric characteristics.

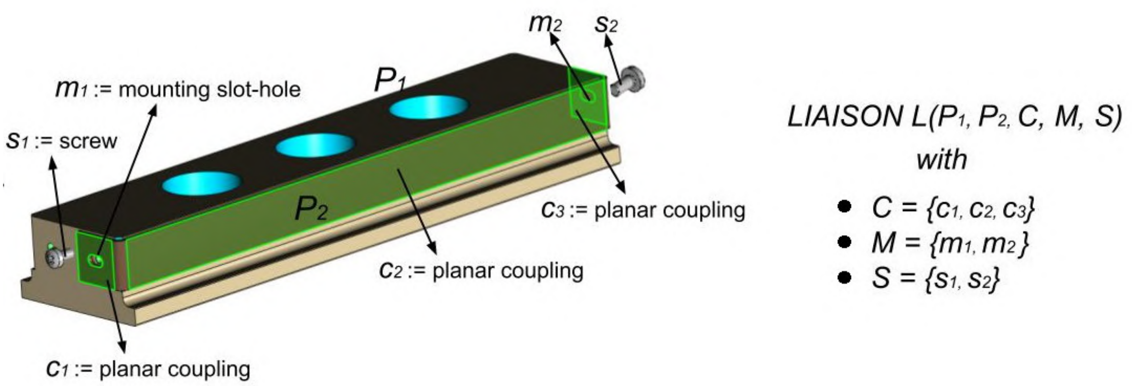
This can be mentioned as one of the features that distinguish liaisons and encourage their use to describe an assembly and its parts relations, instead of the conventional matrices or graphs. In fact, matrices simply report the existence of a contact between two parts, or at most can estimate the existence of the contacts according to the three orthogonal directions (i.e. along the x, y, and z axes). Graphs,



(a)



(b)



(c)

Figure 3.5: Examples of liaisons.

instead, can contain information as attributes of nodes and arcs, but in a less intuitive manner. Moreover, in general, information regarding the number of faces in contact and their type is rarely given, which is extremely restrictive and unrealistic from the perspective of how the contact should be interpreted. On the contrary, the availability of all couplings associated with the same pair of parts and accessibility to their data ensure a more in depth description of the contact. The knowledge of the number of mating faces (i.e. the number of couplings), their geometry (i.e. planar, cylindrical, conical) and orientations (i.e. common axes for cylindrical and planar faces, normal vectors for planar faces), as well as the overlapping area extension, allow to infer meaningful information on the level of relative clamping between the parts, the degrees of freedom and possible movement directions.

Similarly, a liaison also includes a list of mountings, since more than one mounting can be identified between two contact faces and, moreover, for each liaison mountings lying on different couplings can be found (e.g. Fig. 3.5c).

Also in this case, the data relative to each mounting are contained in the m_i. The list M is another key element of a liaison, that considerably improves the description of a CAD model by giving engineering sense to the contacts between parts and that is generally overlooked. As a matter of fact, it is important to underline that a mounting is not only a topological attribute of a contact, rather it conveys a deeper semantic meaning. From the engineering point of view, the existence of coaxial holes, in fact, is a typical situation of parts mounted by threaded fasteners or pins (see Section 1.1.3). Thus, the presence of mountings results in a crucial feature to understand components' relations and to enforce their connection properties, as well as deducing the assembly process. Usually, the mountings analysis could be considered a redundant operation to deduce the assembly by threaded fasteners, since the presence of the fasteners is enough explanatory. When dealing with industrial CAD models, which can be affected by the issue of missing components (see Section 1.3.1), the knowledge of mountings nevertheless results fundamental because the unique way according to which infer the presence of the not modeled fasteners. The accessibility to mountings' properties, such as their number on each pair of faces, their relative positioning, as well as holes diameters specifications further makes reliable the assumption.

Standard parts as well can be multiple in a liaison (e.g. Fig. 3.5b and Fig. 3.5c), and this is why the list S is provided. Each item s_i of the list can be both a single standard part or a functional set (e.g. screw-nut-washer), in order to enhance its meaning and its role within the liaison.

However, this topic will be discussed in details in Section 4.3.1, while what is relevant to know at this point is that in both cases all the information (e.g. category and subcategory) and dimensional values of the standard parts, along with their orientation and positioning, are accessible form each s_i. Moreover it is also readable if the standard parts included in S are all instances of the same subcategory (e.g. a collection of screws belonging to the same subcategory and having same dimensions) or belong to different categories. It is evident that this knowledge stands at the basis of a comprehensive high-level interpretation of the relations between two custom designed parts. For instance, the presence of a specific collection of fasteners joining two components along a given pattern not only suggests the assembly operation carried out, but also the tools needed, and the direction of extraction/insertion, as well as the path according to which they are mounted. Or else, the presence of a key allows to deduce that the two custom designed parts are respectively a shaft and a rotating element.

In summary, considering standard parts as an attribute of contacts rather than treating them like any other component results an innovative idea. It definitely distinguishes the liaison structure defined in this work from that presented in [103], which does not address this meaningful aspect, and does not actually mention the possible presence of standard parts. In addition, the choice of differently manage standard parts is beneficial in a number of ways. First, it has a strong semantic value, but it also promotes the use of the liaison structure for CAD models reorganization in relation to the computation and the intuitiveness. In fact, thinking of the liaison list as the collection of all pairs of contact parts, it would indeed contain too much elements when dealing with assemblies made of hundreds of parts, and thus its usage would not be advantageous in respect with matrices or graphs. However, as standard parts are in general a substantial portion of the total parts of an assembly (taking into account the set of models analyzed in the thesis, standard parts are on average 40% of the total number of parts), omitting them from the parts that can underlay a liaison consistently reduces the final number of liaisons in the list, making it easy to handle. It is known that starting from the similar aim of reducing the dimensions, there are methodologies that remove the fasteners from the contact matrices and then consider them later (e.g. [25, 55]). Nevertheless this type of approach is targeted at fasteners only, usually manually detected, while locating components are overlooked. Also, this does not allow to fully exploit standard parts semantic meaning and intuitively visualize their function within the mating relation, as is instead done when using liaisons.

3.4 Conclusions and remarks

This chapter outlines an innovative approach which aims at automatically dealing with industrial CAD models of mechanical products replicating the reasoning an experts would do in reality.

It is assumed that the only data available are that geometric and topological, certainly contained in the boundary representations of the parts, along with the awareness of their positioning in the space. Given these premises, a way to extract semantic information, both relative to single components and to assembly processes, store and present it in a complete and intuitive structure, and then leverage it to address complex assembly tasks is presented.

Two can be identified as the main efforts. On the one hand, the creation of a mapping that makes the correspondence between the properties of the shape and the engineering characteristics, which are patented in the real product and that make recognizable some parts or assembly techniques, with geometric and topological requirements that CAD models must meet to represent that specific part or concept, regardless of the design conventions and companies strategies that may have been adopted during modeling. On the other hand, the definition of a new data structure, denoted as *liaison*, that allows to represent the original CAD model in an enriched manner.

A standalone system is implemented, which is constituted by several modules each addressing an operation, both related to the processing of the mere geometric CAD model and the leveraging of the enriched CAD model to face assembly tasks.

The modules developed as prototypes of the software of the company Hyperlean are investigated more in depth in the next two chapters. For sake of clarity, Chapter 4 includes the data extraction modules, i.e. those necessary steps crucial to rigorously define and compute liaisons, while Chapter 5 suggests some promising applications of the liaisons' data to support the manufacturing process in an improved way. Both chapters also present and discuss results obtained using industrial CAD models of mechanical product as test cases.

4

Phase 1: Data Extraction

Challenges and key concepts underlying the approach were presented in the previous chapters, along with the general overview of the system implemented and the distinction between the data extraction and the data exploitation phases. The purpose of this chapter is to extensively address the data extraction phase. It includes all the operations and tasks necessary to correctly and reliably deal with the industrial CAD model of mechanical assembly. In particular, through the data extraction process, the transition from a bare geometric description of a CAD model to a semantic enriched and engineering meaningful representation is accomplished. Each module is singularly illustrated by detailing the choices that characterize it, the methodologies and the algorithms implemented, and the results obtained.

4.1 Geometric processing module

This module includes all the functionalities for preparing the B-rep and extracting the geometric and topological information useful for the successive processing steps.

The operations carried out in this phase are respectively normalization of the B-rep of the model, feature recognition, and contacts detection. The arrangement of the vertices, edges, and faces of the parts is evaluated in order to first obtain a B-rep representation independent of the construction operations and CAD system conventions and then to identify local features and general geometric and topological properties for each component and for pairs of components. At this stage the computed data are not yet associated with engineering meanings. But this phase is crucial to move from the only knowledge of bare geometric entities to higher level geometric and topological information. Moreover, in this way, onerous calculations are avoided successively, since all the data necessary for the actual semantic analysis are already collected, associated with the respective parts, and easily retrievable.

The performed tasks are well known and widely addressed both in literature and in the industrial field, so that some are also implemented as functionalities of commercial CAD software. However, to ensure a standalone system with the maximum generality and usability, all the operations have been developed, some taking advantage of the features already implemented by the partner company in their software (e.g. LeanCOST), some others are implemented from scratch.
The following subsections will focus more on the aspects of interest to the next modules rather than on a comprehensive description of the three topics.

4.1.1 Normalization process

As already mentioned in Chapter 3, much of the work is built on a rule-based approach which converts engineering features and schemes in geometric requirements. The geometric requirements have to be met by the CAD models of the assembly's parts, and thus they must be found in their solid representations, namely in their B-rep.
More in details, the geometric requirements mainly concern the presence of faces arranged in a certain way (e.g. forming a concave cluster, concentric, parallel, etc.) and having a certain shape (e.g. opened cylindrical faces with a minimum/maximum opening angle). All these conditions are evident to the human eye, but when computationally dealing with CAD models in B-rep format the detection and validation of the requirements is very demanding. In particular, ambiguous situations can occur relating to the non-uniqueness of the boundary representation. Indeed,

the same component can be described through several B-rep instances [48], as shown in Figure 4.1.

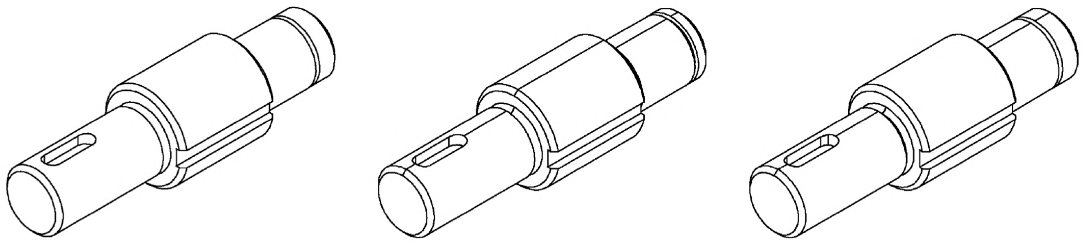


Figure 4.1: Different B-rep instances of the same component (Figure 2 in [48]).

The differences between two representations of the same model stand both in the topological decomposition in terms of faces and edges and in the adopted geometric description of the underlying surfaces and curves. A face can be divided into smaller ones that share the same surfaces and are topologically connected (e.g. a cylindrical surface can be represented either with two half cylinders or a single cylindrical face). Consequently, the same applies to edges: a set of topologically connected smaller edges laying on the same curve, rather than a single edge, can be present in the B-rep model. In addition, canonical surfaces can be described either in terms of their analytical form or in terms of NURBS patches.

From these premises, it is clear that working directly on a CAD model as it is given in input requires an excessive effort. In fact, given the variability of the representation of even one face, there would be to evaluate too many combinations of cases and many others could be left out. In order to allow a consistent definition and application of the geometric requirements underlying the provided approach, it is therefore essential to standardize the geometrical entities used in the assembly's parts models, and restrict the space of possible representations for a certain geometry. To this aim, all the parts of the assembly considered are processed and their representation is normalized.

The normalization phase returns a copy of the input CAD assembly model, where the B-rep of each part has been modified according to the criteria of single geometric formulation and maximal topological entities.

Single geometric formulation

Single geometric formulation is the criterion according to which the geometry of faces and edges is restricted to only some meaningful formulations, but still covering all the possible cases. In fact, in a CAD model, the geometric formulation of each face and edge can vary according to the designers' choice, the originating modeler kernel and undergone file format conversions. Consequently, to optimize the approach, it is necessary to standardized it to provide and use unambiguous formulations. To this aim two steps are carried out:

- *Face conversion*. The geometry of faces is restricted to the following forms: planar, cylindrical, conical, toroidal, spherical, revolved, ruled and NURBS. For the sake of clarity, a ruled surface is obtained from a curve extruded along a linear path, while NURBS includes any level of degree, spans number and knots arrangement.
  In particular, a face of revolution is checked whether it is reducible to a cylinder, a cone, a torus or a sphere respectively. Similarly, a NURBS face is verified whether it can fall respectively into a plane, a cylinder, a cone, a torus, a sphere, a surface of revolution or a ruled surface. To this purpose, algorithms provided in OpenNURBS library [6] have been adopted and the conversion is subjected to a fixed level of tolerance bounding the deviation between the original and the new form.

- *Edge conversion*. The geometry of edges is limited to linear, circular, elliptical and NURBS forms. With a similar approach, edge geometries are reduced to the meaningful canonical form according to the fixed tolerance.

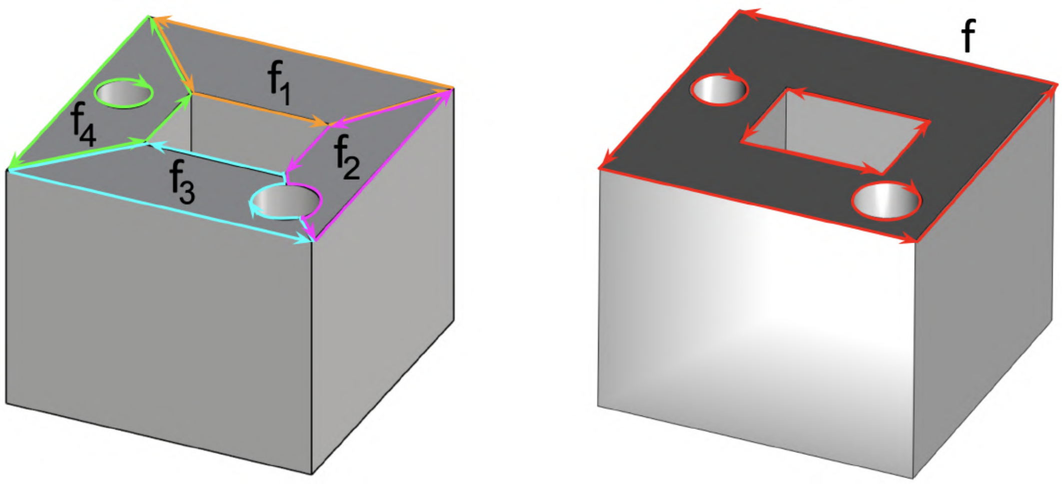
These choices relating to the geometric formulation adopted derive from a practice in the most popular geometric kernels and in the neutral STEP interchange format.

Maximal topological entities

The maximal topological entities criterion is intended to avoid the presence of a redundant number of faces and edges in a part's representation by combining, when some precise conditions are met, two or more faces/edges in a single one. Specifically it provides for:

- *Faces merging*. Two faces in a body become candidates to be merged to a maximal face if they share a common tangent edge, i.e. an edge whose solid angle is 180°. Then, the underlying canonical geometry is checked to be the same, as well as the surface orientation. It applies to planar, cylindrical, conical, toroidal, spherical and revolved faces, by verifying straightforward conditions according to the specific analytical form (i.e. radii equality, points coincidence or alignment, etc.). Appropriate classification criteria are adopted to distinguish among internal and external loops after merging faces in closed ones.

- *Edges merging.* For each loop in a face, contiguous edges (i.e. sharing the same vertex) are analyzed and merged to form maximal ones if they share the same geometrical definition. The merge operation is accomplished by removing the connection vertex, after verifying that the vertex joins only the two edges. It applies to linear edges which lie on the same line, and arc edges which need to share the same arc center and radius.



(a) Original B-rep model with custom topological entities.

(b) Normalized B-rep model with maximal topological entities.

Figure 4.2: Example of faces and edges merging.

An example of maximization of the topological entities, and thus of faces and edges merging, with also the loops classification, is shown in Figure 4.2. In particular, the first cube is modeled in the Rhinoceros software with custom topological entities, namely the top face is made by four adjacent faces. Among them f_1, f_2, and f_3 have a single external loop of edges, while f_4 has an external loop and an internal circular loop. When the normalization process is applied to the cube, the four faces are merged due to their properties, and a single maximal face f is returned. This includes an inner square loop derived from the combination of edges of external loops, an external square loop derived from the combination of edges of an external loop, and two inner circular loops, one already existing, the other given by the combination and merge of edges of external loops.

The adjustment of the B-rep to maximal topological elements and to restricted geometry types allows stronger assumptions in the following algorithms and limits the variability of the geometric and topological conditions to be recognized.

4.1.2 Shape type classification

Another operation carried out in the geometric processing phase is that of classifying the assembly's components into major categories, referred as *shape types*. The shape types are defined according to the overall geometric properties of the parts, such as symmetries, general shape, thickness, and proportions between the volume sizes. In addition they are meant to integrate the geometry with engineering concepts, namely the shape types can be associated with fundamental forms used in mechanical products obtained by specific industrial manufacturing processes. The list of the shape types categories includes:

- *Axisymmetric*. The type covers all the parts whose shape can be traced back to a solid mainly characterized as a profile rotated around an axis.
- *Sheetmetal*. The parts included in this category are obtained by folding and stamping metal foils. The distinctive geometric characteristic is that their thickness is constant and much lower than the other two sizes.
- *Blocks*. This category refers to the parts whose geometry resembles or is drawn from a milled parallelepiped as for plates.
- *Beams*. This type refers to the components obtained by cutting, bending and carving standardized profiles such as rods, tubes, plates, and all other shapes such as IPE, HEA, HEB, L, etc.
- *Other*. This category is aimed at grouping all the shapes which do not fall in the previous categories and which are usually obtained by foundry, stamping, injection molding, advanced milling or additive manufacturing.

Afterwards, a shape type is assigned to each individual part of the assembly and is stored as information associated with it, so that at any time it is always possible to read the shape type of a part.
The knowledge of this kind of property can in general enhance the semantic interpretation of the assembly and its components. However, among the different shape types, the axisymmetric is of particular interest and will be widely exploited in the following modules (e.g. Section 5.1 and Section 5.2). In this regard, some additional clarifications on this category deserve to be pointed out, in order to demonstrate the consistency of the method with respect to mechanical parts. In fact, most mechanical parts are not strictly axisymmetric in a geometric sense, because some of their features or small details are present only in a given portion of the part and/or they can not be derived from the simple rotation of a profile. The detection of

axisymmetric parts takes into account this scenario and is thus robust with respect to the presence of such characteristics. That is to say, components among which shafts with keyseat, plates or blocks with screwseat, screws with different types of head and drive, or toothed gears will be classified as axisymmetric parts.

4.1.3 Feature recognition

Feature recognition is a well known and widespread research topic that has been deeply investigated since the last decades of the '90 and finds application in several domains, among which the industrial manufacturing (e.g. [18, 95]).
It can be affirmed that features are those more articulated geometric entities that form the first link between a CAD model of a mechanical product and its semantic and engineering meaningful properties [87]. This consideration stands at the base of the proposed approach.

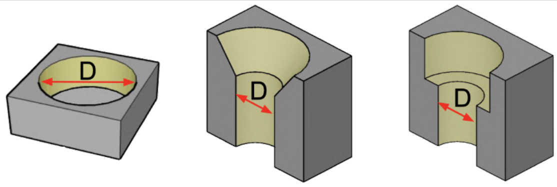
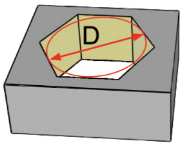
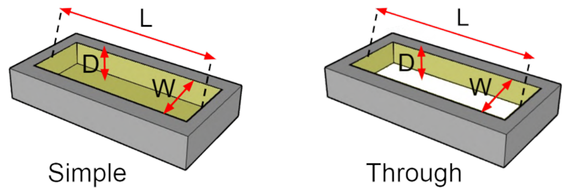
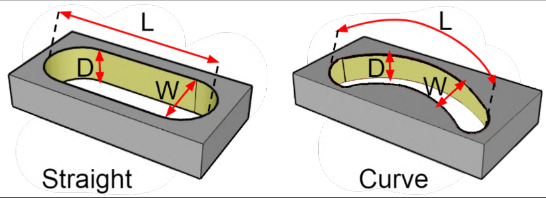
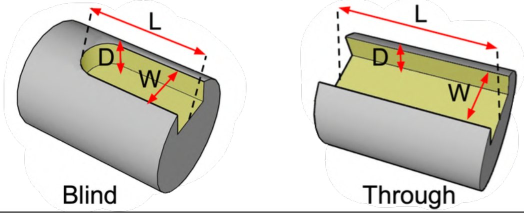
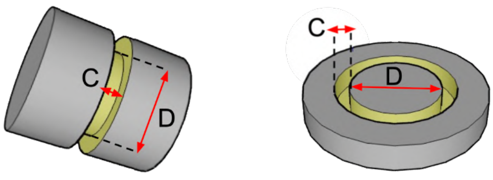
For sake of completeness, the presented system includes its own feature recognition module, relying on functionalities already existing in the industrial software LeanCOST of the company partially supporting this PhD research.
This section is thus not targeted at a detailed description of the recognition algorithm, rather it focuses on providing the features of interest in the work and their key properties. This would be helpful for comprehending the next modules.

As it will be better discussed in Section 4.2, features can be leveraged to identify the seats where fasteners and locating components are positioned (e.g. holes, keyways, keyseats, grooves, etc.), as well as distinctive characteristics of some categories of parts suggesting, for example, the need of specific fastening tools (e.g. the drive of a screw).
At this purpose, the features recognized and stored are mainly concave portions of parts forming a specific shape. These are described in the following and listed in Table 4.1 for a better visualization, where their typical dimensions and how they will be denoted in the rest of the thesis are also highlighted.

- **Holes**: holes are detected in the presence of axis-aligned hollow faces of cylindrical, conical and toroidal type, as well as disk- and ring-shaped planar faces. According to the shape of the adjacent faces, plain, counterbore, countersunk and tapped holes are distinguished, and further differentiated in blind on one side or pass-through. The characterizing dimensions for holes are diameters and depth.

- **Pockets**: pockets are in general concave clusters of faces. In particular, polyg-

Table 4.1: Table of the main features recognized.

Feature	Examples of the feature	Dimensions
Hole	 Blind, Countersunk, Counterbore	Diameter (D)
Polygonal pocket	 Hexagonal	Diameter (D)
Rectangular pocket	 Simple, Through	Length (L) Width (W) Depth (D)
Circular-end pocket	 Straight, Curve	Length (L) Width (W) Depth (D)
Slot	 Blind, Through	Length (L) Width (W) Depth (D)
Groove	 Radial, Frontal	Diameter (D) Chord (C)

onal, rectangular and circular-end pocket are distinguished. Polygonal pockets are concave clusters with the faces arranged as to form a regular polygon, e.g. an hexagon. Rectangular pockets are concave cluster with four sided faces and possibly a bottom face adjacent to them. The sided faces must include two pairs of opposite parallel planar faces with same rectangular shape, orthogonal with each other, as to generate a rectangle. Moreover, through pockets are pockets with the bottom face missing. Length, width and depth are the typical dimensions for pockets. Circular-end pockets also are concave cluster with four sided faces and possibly a bottom face adjacent to them. The difference from rectangular pockets is that at least the smallest sided faces are opposite cylindrical faces, and it is denoted as straight. If, instead, even the largest sided faces are a pair of equal cylindrical faces (respectively one concave and the other convex) the circular-end pocket is defined as curve.

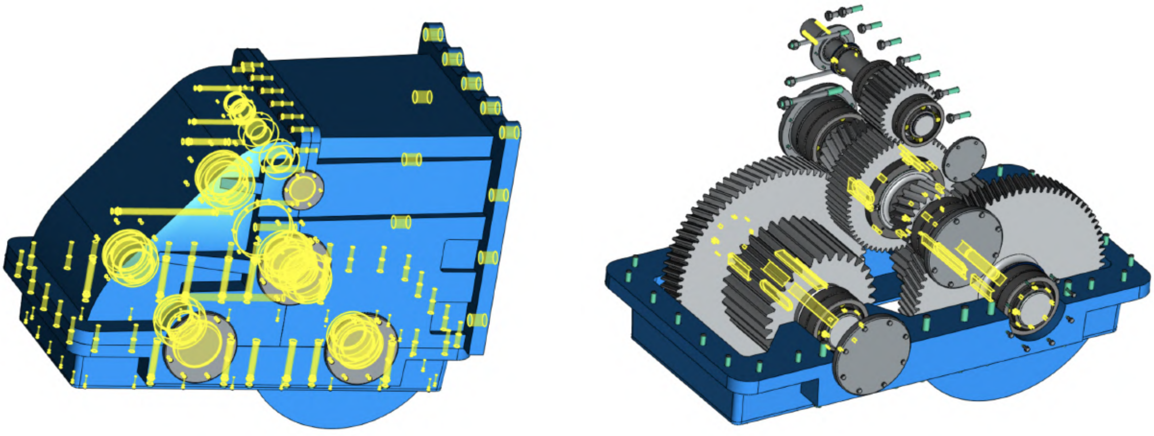
- **Slots**: slots are identified as concave clusters of faces with two or three sided faces connected by a bottom face. Slot are through when they are two sided and the faces are a pair of opposite parallel planar faces. Slots are, instead, blind if there is a third sided face connected with the other two. Also for slots, the characteristic dimensions are length, width and depth.

- **Grooves**: grooves are circular cutouts. In particular, they can be radial, when they include two equal planar rings connected by a cylindrical face, or frontal, when they include two coaxial cylindrical faces connected by a planar ring. The diameter and chord extensions are the characteristics dimensions for grooves.

It is important to note that the number and type of faces mentioned in the description of the features refer to the main necessary faces characterizing the feature. Although, taking into account the variability of the features and to ensure maximum generality, the existence of additional faces representing fillets and chamfers is allowed in clusters. To verify that they actually have that role, conditions on adjacency between the supposed fillets/chamfers and the main side/bottom faces and their proportions are checked.

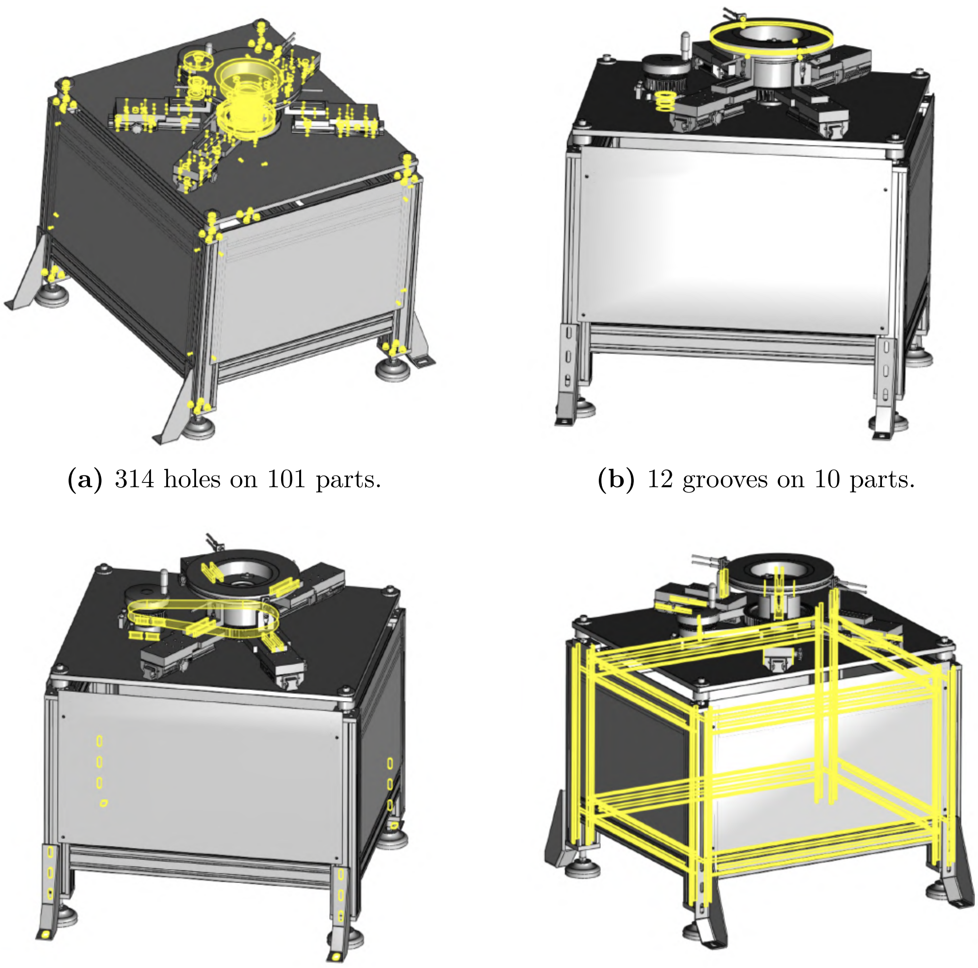
From the system implementation point of view, the feature recognition algorithm is iterated on the normalized B-rep of all the parts of the assembly. For each part the lists of features and the associated dimensional properties are stored, so that their accessibility and usability in the following steps are easy.

Figure 4.3 and Figure 4.4 show examples of the outcomes of the feature recognition process. They show, respectively, the CAD models of a gearbox and a disks break test bed where the features recognized on each part are highlighted on the whole. For a clearer visualization, the holes of the gearbox are shown in Figure 4.3a and the pockets and the slots are reported in Figure 4.3b, while grooves are not present. As for the disks break test bed, holes, grooves, pockets, and slots are separately highlighted respectively in Figures 4.4a, 4.4b, 4.4c, and 4.4d.



(a) 412 holes on 155 parts.
(b) 70 pockets and slots on 70 parts.

Figure 4.3: Features recognized on the CAD model of a gearbox of 426 parts.



(a) 314 holes on 101 parts.
(b) 12 grooves on 10 parts.
(c) 29 pockets on 10 parts.
(d) 92 slots on 23 parts.

Figure 4.4: Features recognized on the CAD model of a disks break test bed of 153 parts.

4.1.4 Contact detection

Contact detection is a crucial task that must be performed at this early stage because all subsequent modules rely largely on its outcome. It is the first process that evaluates the relative properties between two parts, rather than the geometric and topological features of the single parts.

In particular, taking as input the CAD model with normalized B-rep, along with the list of features associated with each part, this module evaluates for each pair of parts the existence of couplings and mountings, as they are defined in Section 3.3.2, and stores them if detected. Two lists, respectively the lists of couplings and the lists of mountings, are the returned output.

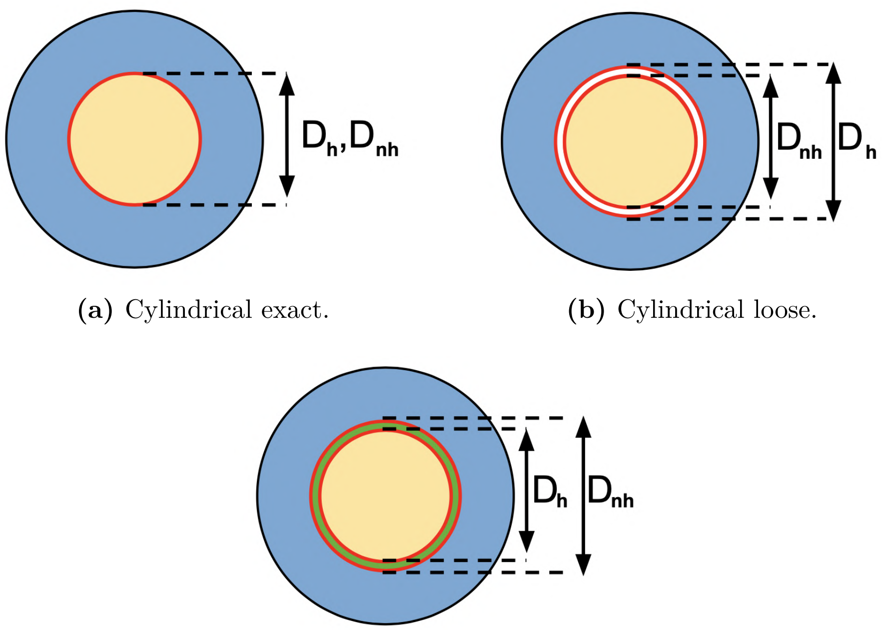
Couplings computation

First, the computation of the list of couplings, which will be referred as $COUPLINGS$, is carried out.

From the implementation point of view, for each pair of faces belonging to different components of the assembly, the surfaces' relative position is analyzed. That is, if two parts have a pair of faces lying on the same canonical surface, partially or totally overlapped in the 2D space defined by the common surface parameterization, they are considered in contact. Contacts can be planar, cylindrical or conical according to the geometric type of the surface. Contacts are fundamentally computed by checking if the two faces lie on surfaces of the same type and with certain properties i.e. are both on planes, parallel with each other and with opposite normal vectors, are both on cylinders with same axis and center, are both on coaxial cones with same apex angles.
To generalize as much as possible the method and especially to tackle misplacement of parts due to numerical errors, possibly due to import/export of the model from different systems and conversion in different formats, it is not required that the faces lie on the exact same surface, rather a tolerance associated with the maximal distance between the two surfaces is set as $tol_{contact} = 0.1$. That is to say, the distance between two planes must be less than $tol_{contact}$, as well as the difference between the radii of the cylindrical and conical surfaces.

Cylindrical contacts detection is then further extended and this point deserves to be focused because strictly connected with the semantic interpretation of parts' relations. More in details, instead of using the just mentioned $tol_{contact}$, cylindrical contacts detection is carried out by accepting the coaxial cylindrical face pairs so that their diameters ratio is in a neighborhood of 1, precisely the maximum accepted

value for the diameters ratio is 1.25. It is to clarify that these tolerance values are chosen on the basis of practical considerations made by engineers.
From the CAD model standpoint, it means that cylindrical faces can both intersect with each other or have a clearance. From the interpretation point of view, this extension is conceived to deal with threads interference. In fact, as explained in Section 1.2.2 according to conventional representation practices, threaded contacts are frequently modeled as cylindrical interferences between the threaded portions of parts. As a consequence, only relying on the absolute $tol_{contact}$, these scenarios would be overlooked and many relevant information about parts' contact would be lost. In general, information on the presence of a thread attribute may be provided by the assembly data structure, but has been deliberately neglected because the presence is not guaranteed depending on the origin of the model. It is therefore preferred to use the attribute, when present, as a simple confirmation of the coupling identifier, rather than as a means of identification. Thus, the introduction of a maximum ratio between the diameters allows to overcome the problem and detect contacts addressing all the possible situations. In addition, a strength is that the information about the interference/clearance is also stored as attribute of the coupling, and this allows to further exploit it in the following to infer engineering meaningful knowledge from the only geometric properties of the contacts.



(a) Cylindrical exact.

(b) Cylindrical loose.

(c) Cylindrical interference.

Figure 4.5: Projection view of the three types of cylindrical contact. In red are colored the projections of the faces in contact. In green is indicated the volumetric intersection. Let D_h be the diameter of the hollow face and D_{nh} the diameter of the not hollow face.

In summary, three types of cylindrical contact are therefore distinguished, depending on the relation between the diameters D_h and D_{nh} respectively of the hollow and not hollow cylindrical faces involved in the contact. Namely, a cylindrical contact is exact if $D_h = D_{nh}$ (Fig. 4.5a), it is loose if $D_h > D_{nh}$ (Fig. 4.5b), and it is interference if $D_h < D_{nh}$ (Fig. 4.5c).

Once a pair of faces is detected in contact, a coupling $c(f_1, f_2)$ is created and added to the final list of couplings. The principal data are the two faces f_1 and f_2, however, thanks to the algorithm implemented and the evaluations done for detecting the contact, some data, that will be useful in the following, have been already computed and thus they are assigned as attribute of the coupling, so that they are accessible without repeating redundant computations. Among these:

- Parts: it reports the two parts P_1 and P_2 to which the faces belong.
- Contact type: it is a property that indicates whether the contact is planar, cylindrical, cylindrical interference, cylindrical loose, or conical.
- Contact perimeter: it is the list of the edges that define the perimeter of the contact area.
- Contact area: it is the value of the area of the portion of faces in contact.
- Contact orientation: it indicates the geometric properties of the contact. In particular, if the contact is planar it refers to the normal vector of the planes, while if the contact is cylindrical or conical it refers to the axis of the surfaces.
- Relative position: it is the transformation matrix that allows to change from the reference system of P_1 to the reference system of P_2.

Once all the pairs of faces are evaluated, the list $COUPLINGS$ is returned. It is evident that, at this point, several couplings can exist associated with the same pair of parts. As it will be better discussed in Section 4.3.2, these will be aggregated in a single object, i.e. the liaison, to provide a more complete and meaningful information about the contact between two parts.

Mountings computation

Relying on the just obtained list of couplings, then the computation of the list of mountings, denoted as $MOUNTINGS$, follows.

In particular, making reference to the definition of mounting given in Section 3.3.2, the algorithm cycles over all the planar couplings included in the list $COUPLINGS$ and assesses the existence of specific features meeting some requirements on the two associated parts.

Given a coupling $c(f_1, f_2)$ and its associated parts P_1 and P_2, the features previously detected on those two are considered. Namely, the features of interest for the mountings computation are holes, straight or curve through pockets, and polygonal pockets. Among the features found on each part P_i, only those having a loop of edges lying on the coupling's faces f_i are selected. Otherwise, the definition of mounting is not met. After the selection, each pair of features F_1-F_2, where F_1 is a feature of P_1 and F_2 feature of P_2, is analyzed. In particular, they must be aligned. The verification of the alignment depends on the type of features involved, thus, for sake of clarity, the possible situation are now listed.

- Hole-hole, polygonal pockets-polygonal pockets, hole-polygonal pockets. Notice that both holes and polygonal pockets are characterized by a center and an axis. Thus, a pair of these features is aligned when their axes are parallel and the vector between their centers is parallel to the axes.

- Hole-straight/curve pocket. First, it is to clarify that a pocket, both if it is straight and curve, is not simply characterized by a center and an axis, as an hole, rather it has a medial axis, i.e. a line or an arc between its two ends, and an axis which is the axis passing from the medial axis and parallel to the pocket side faces. As a consequence, an hole is considered aligned with a pocket when their axes are parallel and the axis of the hole intersects the medial axis of the pocket (i.e. if projected on the same plane, the center of the hole is a point of the medial axis). The case in which there is a polygonal pocket instead of the hole is equally treated.

- Straight/curve pocket-pocket. The alignment between two pockets, regardless of whether they are straight or curve, is verified when their axes are parallel and, if projected on the same plane, the medial axes intersect.

Once the pair F_1-F_2 is confirmed to be aligned, a mounting $m(f_1, f_2, F_1, F_2)$ is created and added to the final list of mountings. It contains as main data the two faces f_1 and f_2 underlying the coupling and the two associated features F_1 and F_2, described as lists of faces. Furthermore, according to the performed evaluations, many geometric information about the mounting has already been extracted, and for completeness it is stored as attribute of it. The most relevant and useful data, largely exploited in the next modules, are:

- Axis: it is the directional vector that indicates the axis according to which the features are aligned.
- Diameters: it indicates two reference diameters associated with the two features of the mounting. Namely, when the feature is an hole the diameter value corresponds with the hole's diameter; when the feature is a straight or curve pocket the diameter value corresponds with the width of the of the pocket, when instead the feature is a polygonal pocket the diameter value corresponds with its inscribed diameter.
- Depth: it is the value of the height of the mounting, given by the sum of the depth of each feature.

It is specified that more than one mounting can exist associated with the same coupling. This because several pairs of aligned features can be found lying on the same pair of adjacent faces.

When all the pairs of features and planar couplings have been evaluated and analyzed, the list $MOUNTINGS$ is completed and returned as output.

4.2 Part recognition module

The part recognition is the module that first allows to elevate the semantic meaning of an assembly passing from geometric and topological information to an actual engineering interpretation of the parts. This operation is a fundamental prerequisite for the liaisons computation and the enriched CAD model definition.

The main goal of this phase is to group the assembly's components into two sets based on their functions, namely distinguishing between custom designed parts and standard parts, as they are defined in Section 1.1.2. In particular, the objective is achieved by means of a rule-based approach which automatically recognizes standard parts among the assembly's parts thanks to their distinctive shape and positioning within the assembly, while the remaining parts are then designated as custom designed parts. In addition, the standard parts are more precisely classified in different categories and subcategories, as well as characterization through dimensional values typically used in mechanical engineer are returned. In the next sections the standard parts recognition algorithm is discussed in details.

4.2.1 The recognized categories: motivations and goals

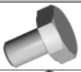
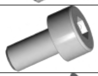
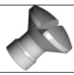
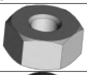
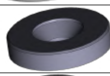
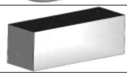
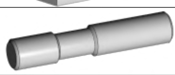
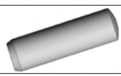
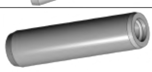
First, before going deep into the algorithm description, it is worth to point out which categories of standard parts have been addressed and justify the choice.

As it has been pointed out in Section 1.1.2, standardization organizations have normed a large number of components covering all the production sectors. Therefore, depending on the sector, the standard parts may vary in terms of classification, defining shape and dimensional characteristics. Moreover, depending on the product class and on the materials of the parts involved, the variety of included standard elements is generally limited to some recurrent standard parts and categories. Therefore, since this research is carried out in partnership with the Italian company Hyperlean which manages CAD models supplied by various Italian companies involved in the design and manufacturing of automatic machines, the presented recognition focuses on the standard parts largely employed in this type of products.

The categories currently considered are: **screws**, **nuts**, **O-ring**, **washers**, **circlips**, **keys**, **studs** and **pins**.
Some of these classes, i.e. O-ring and studs, refer to a single type of parts, while the others, i.e. screws, nuts, washers, circlips and pins, include a large variety of parts, which differ in features and usages. As a consequence, in the latter case, it is necessary to distinguish subcategories, in order to return more accurate results (see also Table 4.2). Namely, according to practical experience, screws are divided in eight subcategories depending on the head shape, that are: *hex head screws*, *socket hex head screws*, *socket hex countersunk head cap screws*, *cross recess countersunk flat head screws*, *cross recess countersunk raised head screws*, *cross recess raised cheese head screws*, *slotted pan countersunk head cap screws*, *slotted flat countersunk head cap screws*. Nuts include two subcategories, that are *hex nuts* and *hex cap nuts*, according to whether they have a through hole or a blind hole closed on one side by a domed end. Among washers are distinguished *flat washers* and *spring washers*, depending on whether the cylindrical shape is closed or open. Circlips consist of five subcategories according to the ring ends behavior and the internal shape, namely: *internal circlips*, *external circlips*, *snap rings*, *rings type G* and *rings type E*. Finally, pins are divided into two subcategories: *not holed pins* and *holed pins*.

It is evident that the work presented focuses on part categories made of a single solid body (i.e. standard parts defined as assemblies are not considered), which follows international standards for shape and dimensions. Even if, the list is not exhaustive, it is wide enough to cover the types typically found in mechanical as-

Table 4.2: Summary table of the standard parts categories considered and the associated extracted dimensions

Category	Subcategory	Image	Dimensions and geometric properties
Screws	Hex head		Nominal Diameter, Length, Head Height, Key Size, Center and Axis
	Socket hex head		Nominal Diameter, Length, Head Height, Key Size, Socket Depth, Center and Axis
	Socket hex countersunk head cap		Nominal Diameter, Length, Head Height, Key Size, Socket Depth, Center and Axis
	Cross recess countersunk flat head		Nominal Diameter, Length, Head Diameter, Groove Width, Cross Depth, Center and Axis
	Cross recess countersunk raised head		Nominal Diameter, Length, Head Height, Head Diameter, Groove Width, Cross Depth, Center and Axis
	Cross recess raised cheese head		Nominal Diameter, Length, Head Height, Head Diameter, Groove Width, Cross Depth, Center and Axis
	Slotted pan countersunk head cap		Nominal Diameter, Length, Head Height, Head Diameter, Slot Depth, Center and Axis
	Slotted flat countersunk head cap		Nominal Diameter, Length, Head Diameter, Slot Depth, Center and Axis
Nuts	Hex		Nominal Diameter, Head Height, Key Size, Center and Axis
	Hex cap		Nominal Diameter, Head Height, Key Size, Center and Axis
O-ring	-		Diameter, Chord, Center and Axis
Washers	Flat		Nominal Diameter, Head Height, Key Size, Center and Axis
	Spring		Thickness, Inner Diameter, Outer Diameter, Center and Axis
Circlips	Internal		Internal Diameter, External Diameter, Thickness, Center and Axis
	External		Internal Diameter, External Diameter, Thickness, Center and Axis
	Snap ring		Internal Diameter, External Diameter, Thickness, Center and Axis
	Ring type G		Internal Diameter, External Diameter, Thickness, Center and Axis
	Ring type E		Internal Diameter, External Diameter, Thickness, Center and Axis
Keys	Key type A		Height, Length and Width
	Key type B		Height, Length and Width
Studs	-		Nominal Diameter, Length, Threads Length, Center and Axis
Pins	Not holed		Length, Diameter, Center and Axis
	Holed		Length, Diameter, Hole Nominal Diameter, Center and Axis

semblies having a crucial importance in the assembly process, and sufficient to prove the robustness and scalability of the approach.

The considered categories and the associated subcategories are summarized in Table 4.2. The table also reports in the last column the characterizing dimensions and properties for each standard parts' subcategory that are usually considered by engineers in the design and useful for the assembly process. The values of these parameters are extracted measuring and evaluating the geometry of the solid model of the recognized part. For instance, the key size associated with hex head screws corresponds to the diameter of the circle inscribed in the hexagonal plane face, the nominal diameter corresponds to the diameter of the cylindrical face of the stem, or else height, length and width of keys correspond with the three dimensions of the model. In addition, an image is provided for each subcategory in the third column of Table 4.2 to better visualize the types of components. The images are representative for each class, but these can be actually modeled in alternative ways, with more or less details.

It is to underline that the recognition of standard parts is very challenging. Due to the possible idealization and simplification of the 3D models of mechanical components mentioned in Section 1.2.1, a part can have different representations or else different parts can appear quasi-identical in shape. As a consequence, to avoid the misclassification of parts, similarly to [80], the method proposed follows a multi-step approach consisting of a single part analysis and a context analysis.
Both phases strongly rely on the geometric processing outcomes. In particular, normalization process is essential for single parts analysis, feature recognition and contact detection stand at the base of the context analysis. A distinctive feature of the approach is that both single parts analysis and context analysis are rule based. Namely, for each category of standard parts shape rules and context rules have been defined that a part must respect if it belongs to that category. The combination of these two types of rules allows to uniquely identify a standard part. It is to underline that, from the point of view of industrial applications, the rule set (especially concerning sizes) can be customized in such a way to restrict the scope classification to the standard parts of interest. In the following sections the two steps are further detailed.

4.2.2 Single part analysis

The single part analysis aims at providing a preliminary detection of standard elements in a CAD model based on their shape and the proportions between their dimensions.

More specifically, as introduced in Section 3.1, the strategy adopted is to gather up engineering knowledge of mechanical components, catalogs on standards and design rules for each category and their subcategories, and then single out their most typifying aspects.
In other words, those engineering characteristics have been identified, both relative to shape and dimensions, that a component must necessarily have when it belongs to one of the categories. In addition, common rules usually followed by designers to model standard parts are taken into account.

The table of the geometric requirements

The collected properties are then translated into appropriate geometric requirements, which are summarized in Table 4.3. In particular, two types of geometric requirements are considered. The first concerns the necessary presence of specific types of faces (i.e. planar, cylindrical, conical, toroidal or spherical). The latter concerns the arrangements and the relations between faces and edges (i.e. positioning at one end of the part, symmetries, coaxiality, dimensional ranges, etc.).

The rules are ordered in each row of the table hierarchically based on the significance of the condition. Namely, reading the table from left to right, the conditions that are more restrictive are reported first, followed by those that are less discriminatory (i.e. those related to the distinction between subcategories). This order corresponds to the one adopted in the eight (one for each macro class) procedures for the standard parts recognition.

In the table, BB indicates the bounding box of the part computed aligned to an intrinsic frame of the part containing all its faces, e.g. in the case of axisymmetric parts the z axis corresponds to the axis of symmetry. A face, respectively a set of faces, is considered at an extreme of the BB when it, respectively a face of the set, is the face of the part having the barycentre closest to a BB's face. The $\{p_i\}$ and $\{c_i\}$ represent the sets of planar and cylindrical faces sorted in descending order with respect to the property indicated in the cells. For instance, the elements p_j and p_k (respectively c_j and c_k) indicate the elements of the set with the j-th and k-th largest value of the considered property.

Table 4.3: List of the geometric requirements for each standard parts subcategory. Let n be the number of faces of the analyzed part and let " be the symbol that indicates the same requirement as the above cell.

	Necessary conditions on presence of faces and their organization							
Hex nut	$9 \leq n \leq 35$	6 planar faces forming a regular polygon P_6	1 cylindrical closed face		other faces symmetric to the axis of P_6		P_6 is convex	the closed cylindrical face is concave
Hex cap nut	"	"	"	1 spherical face	"	the spherical face is at one extrema of the BB	"	"
Hex screw	$10 \leq n \leq 40$	"	"		"	P_6 is at one extrema of the BB	"	
Socket hex screw	"	"	"		"	"	P_6 is concave	
Socket hex countersunk screw	"	"	"		"	"	"	
Cross recess countersunk flat screw	$10 \leq n \leq 50$	4 pairs of frontal planar faces in a concave cluster C_4	"		other faces symmetric to the axis of the cylindrical face	C_4 is at one extrema of the BB		
Cross recess countersunk raised screw	"	"	"		"	"		
Cross recess raised cheese screw	"	"	"		"	"		
Slotted pan countersunk screw	"	1 pair of frontal planar faces in a concave cluster C_1	"		"	C_1 is at one extrema of the BB		
Slotted flat countersunk screw	"	"	"		"	"		
O-ring	$1 \leq n \leq 5$	$\geq$1 toroidal face	other faces: cylindrical, convex toroidal, planar ring		all faces coaxial with each other	$\geq$1 toroidal face convex		all the edges must be arcs of a circle
Washer	$4 \leq n \leq 10$	2 planar faces	2 cylindrical faces		all the faces coaxial with each other	the planar faces are rings and parallel	cylindrical faces are 1 concave and 1 convex	

	Necessary conditions on presence of faces and their organization							
Internal circlip	$6 \leq N \leq 30$	≥ 2 planar faces ($\{p_i\}$ ordered by area)	≥ 2 cylindrical opened faces ($\{c_i\}$ ordered by opening angle)	2 closed cylindrical faces	p_1, p_2 are 1) anti-parallel 2) BB(p_1,p_2) = BB(part)	c_1 is convex and c_2 is concave	c_1 and c_2 coaxial, with 1) opening angles $\geq 200°$ 2) axis paralllel to the normal of p_1, p_2	the 2 closed cylindrical faces concave with same radius< radii of c_1, c_2
External circlip	"	"	"	"	"	c_1 is concave and c_2 is convex	"	
Snap ring/ Spring washer	"	"	"	0 closed cylindrical faces	"	c_1 is convex and c_2 is concave or vice versa	"	
Circlip type G	"	"	≥ 6 cylindrical opened faces ($\{c_i\}$ ordered by opening angle)	"	"	c_1 is concave and c_2 is convex	"	c_3, c_4 and c_5, c_6 opened cylindrical faces, with same radius, facing each other
Circlip type E	$8 \leq N \leq 30$	"	≥ 3 cylindrical opened faces	"	"	c_1 is convex	c_1 is with 1) opening angle $\geq 200°$ 2) axes parallel to the normal of p_1, p_2	2 equal opened concave cylindrical faces belonging to same surface and with same center and axis of c_1
Key type A	$6 \leq N \leq 20$	≥ 4 planar faces ($\{p_i\}$ ordered by area)	≥ 2 cylindrical opened faces ($\{c_i\}$ ordered by area)	0 closed cylindrical faces	p_1 and p_2 are 1) anti-parallel 2) same area	p_3 and p_4 are 1) anti-parallel 2) same area 3) perpendicular to p_1, p_2	c_1 and c_2 convex with 1) same are and parallel axis 2) axis parallel to the normal of p_1, p_2	
Key type B	"	≥ 6 planar faces ($\{p_i\}$ ordered by area)		"	"	"	p_5 and p_6 are 1) anti-parallel 2) same area 3) perpemdicular to p_1, p_2 and p_3, p_4	BB(p_1,..,p_6) = BB(part)
Stud	$3 \leq N \leq 20$	≥ 2 planar faces	≥ 1 cylindrical closed faces		other faces symmetric to the axis of the cylindrical face	cylindrical faces convex	only 2 planar faces are not ring and 1) antiparallel 2) at the 2 extrema of BB	
Pin	"	≤ 3 planar faces	"		"	"		
Holed pin	"	"	≥ 2 cylindrical closed faces		"	≥ 1 cylindrical face concave		

The iterative process

Following the table structure, the single part analysis is accomplished as an iterative process.

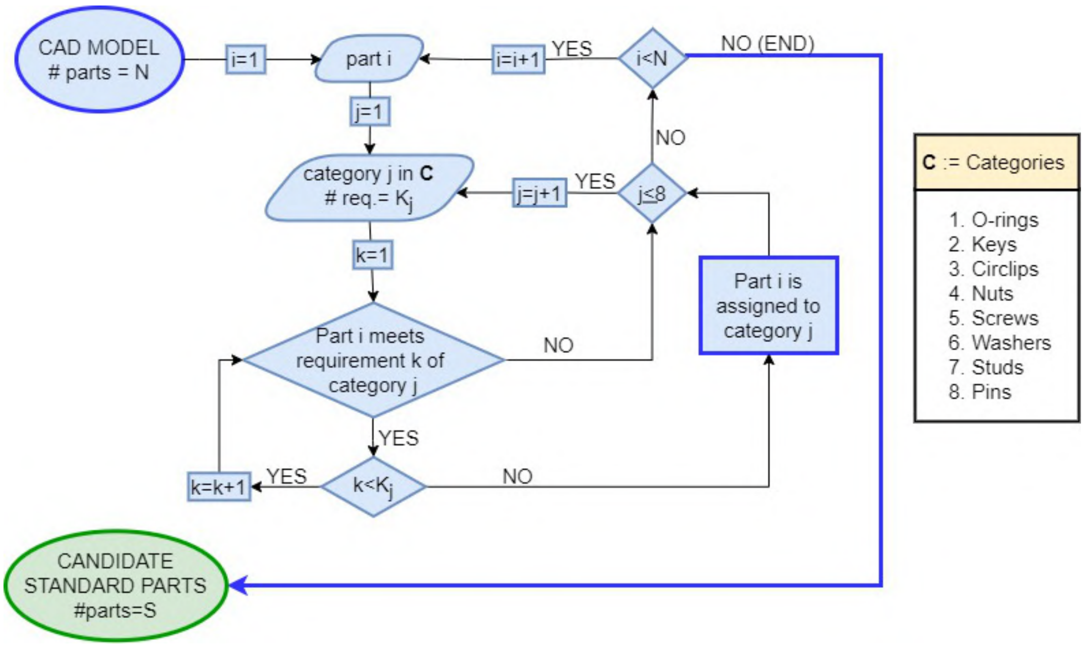
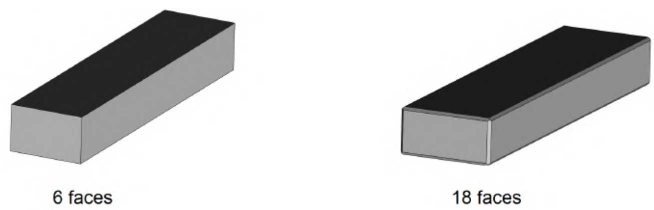


Figure 4.6: Flowchart of the algorithm implemented for the single parts analysis.

As shown in Fig. 4.6, the N parts of the assembly are investigated checking their possible membership to each of the eight considered standard parts categories collected in the set C. Namely, the identification of the i-th part membership to the j-th category, with $j = 1, \ldots, 8$, is carried out by verifying the fulfillment of the category requirements K_j. The eight implemented recognition functions are independent from one another and follow a schema that, proceeding by steps, analyses the geometric requirements the part has to satisfy to belong to one of the subcategories of the given category. Once the part characteristics do not satisfy a requirement k, the function returns false, without evaluating the succeeding features, and the membership to the next category, i.e. category $j + 1$, is then evaluated. It is evident from Table 4.3 that the functions can evaluate in parallel the membership to all the subcategories of the same category till the requirements are the same, but when different rules are found they split the evaluation according to subcategories adopting switch statements. To minimize the number of checks as much as possible, each function evaluates the requirements as they are reported in Table 4.3, ensuring that the part meets the fundamental and most representative features of the category, and then evaluates the properties associated with more precise specifications allowing the distinction between subcategories.



(a) Keys of type B modeled with respectively 6 and 18 faces.

(b) Cross recess countersunk raised head screws modeled with respectively 18 and 41 faces.

Figure 4.7: Examples of standard parts belonging to same subcategory but with different number of faces and geometric details.

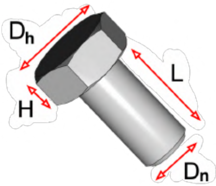
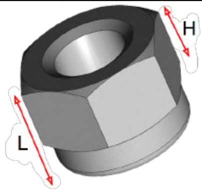
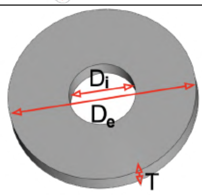
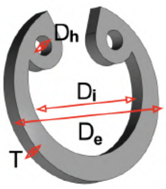
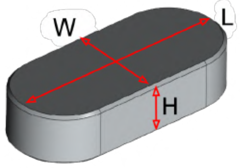
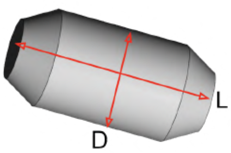
In general, first, the number n of faces is counted. Even after the pre-processing phase, the number of faces can vary depending on the level of detail with which the parts are modeled (Fig. 4.7). A minimum number of faces is mandatory, with reference to the most simplified model, and a maximum number is chosen considering all the possible chamfers, fillets and finishes. Thus, if the number of faces is in the established range, the algorithm proceeds to verify the type of surfaces of the faces and to evaluate their relative positions (e.g. parallel, perpendicular), the symmetry of the part and the existence of specific faces sequences. If the requirements' combination specified for the considered category is satisfied, the component is supposed to belong to the corresponding class. To confirm this assumption, the dimensions and their ratio are checked.

The dimensional requirements

For a better readability of the geometric requirements and to first allow their understanding, dimensional restrictions were omitted from Table 4.3, also because these are the last check for the recognition assessment and thus they can be separately discussed.

In particular, Table 4.4 points out the restrictions considered. Each row reports the dimensional requirements associated with a category of standard parts, as generally same requirements apply to all subcategories of the same category, otherwise it is specified. The second column shows the representative image of a part of the category, where the dimensions taken into account are highlighted, while the third

Table 4.4: Table of the dimensional requirements for the standard parts categories.

Category	Image	Dimensional requirements
Screws		$\frac{L}{H} \geq 3$ $D_h > D_n$
Nuts		$1 \leq \frac{L}{H} \leq 3$
Washers		$1.3 \leq \frac{D_e}{D_1} \leq 3$ $\frac{T}{D_i} < 0.65$
Circlips		$1.3 \leq \frac{D_e}{D_i} \leq 3$ $\frac{T}{D_i} < 0.5$ ($D_h < D_i$ if internal/external)
Keys		$\frac{W}{L} < 0.65$ $\frac{H}{L} < 0.65$
Pins		$1.5 \leq \frac{L}{D} \leq 11$

column actually lists the requirements.
To ensure generality and properly cover standard parts of variable sizes, the dimensional criteria concern the proportions evaluations, which are fixed by standards, rather than the verification of each individual dimensional value of the component. Thus, the here provided requirements refer to the evaluation of the ratio of pairs of measurements, which must be within a specified range established by analyzing the standard dimensions stated in the catalogs of mechanical parts. It is to underline that O-rings and studs are missing in the table since they are recognized with no dimensional constraints due to their greater variability in proportions. It does not

affect the quality of the recognition, because even if several parts are recognized at this stage, they will then be verified by the context analysis.

In order to guarantee the recognition of as many standard parts as possible, in particular also of the simplified ones, for the specification of the rules, standard parts modeled in different ways have been taken into consideration and from these the set of common rules has been derived. This choice, however, involves the fact that some categories could have similar geometric requirements, and thus can be confused.
In this regard, a part can be associated with multiple categories, and the further context analysis is conceived precisely to confirm, refute or discriminate the obtained recognition. In particular, the output of the single part analysis described here is a list containing all the parts of the CAD model which have been assigned to at least a standard part category, along with the associated details of the single or multiple recognition (i.e. subcategories and dimensions). The list is denoted as $CANDIDATE\_STANDARD\_PARTS$ and its cardinality is indicated as S, with $S \geq 0$.

Single part classification assessment

Before moving on to the context analysis details, the assessment of the single part analysis carried out on an ad hoc collected dataset of mechanical parts is reported and discussed.

These tests are not performed starting from an assembly, hence they deviate from the main objective of the semantic analysis of a product. However, results are useful to understand the behavior of the different categories and create a general sorting that orders them from the "best" recognized to the "worst" one. Results of test performed on industrial CAD models of mechanical products will be provided in the following (see Section 4.2.4), once the complete multi-step recognition approach is covered.

The dataset is purposely generated by collecting CAD models of the considered standard parts, and also not standard parts, from different online catalogs, such as GrabCAD [8], TraceParts [11], and PARTcommunity [10], and from an existing repository [9]. This because, at the best of the authors' knowledge, there is no dataset which focuses on standard parts classification as it is intended here. Recent works in literature provide datasets more suitable for the validation of the standard parts recognition here presented [17, 94], but these can not however cover all the subcategories addressed and do not explicitly deal with the different conventional representations.

In details, the dataset includes 825 CAD models in STEP format. Parts are organized in 22 directories, i.e. one for each of the considered (sub-)categories plus the "miscellanea" one in order to provide the ground truth to evaluate the results quality according to the precision-recall (PR) metric. The dataset is publicly available at http://standardPartRecognitionDataset.ge.imati.cnr.it.
For sake of clarity, in the following discussion and pictures, sub-categories results are grouped into their macro categories.

Category	n° parts
Screws	71
Nuts	182
Washers	39
Circlips	61
O-rings	35
Keys	41
Studs	39
Pins	31
Others	326

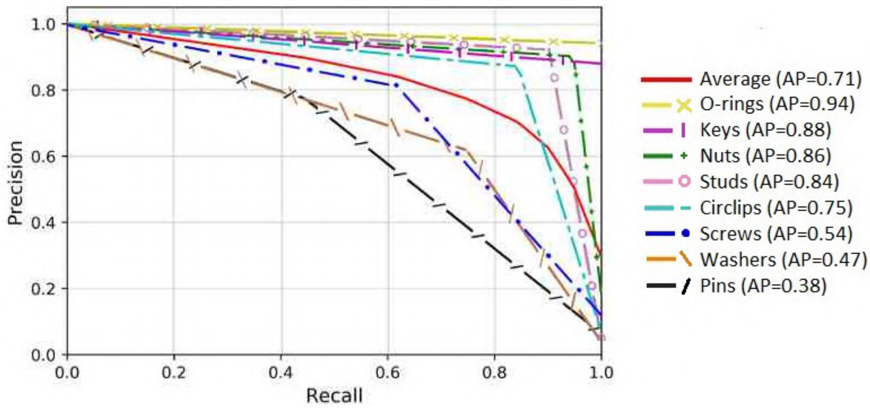
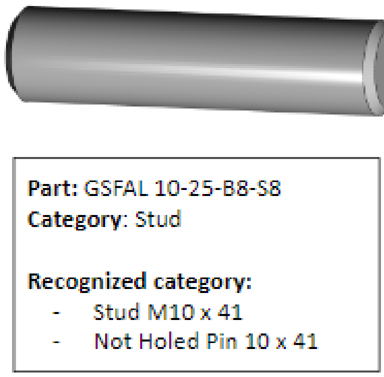


Figure 4.8: Dataset organization and plot of the precision-recall (PR) curves for each standard parts category.

In Fig. 4.8 the precision-recall curves for each category are plotted, and the average precision (AP) values are reported. The average curve is provided to evaluate the overall quality of the recognition algorithm. In general, a large area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate.

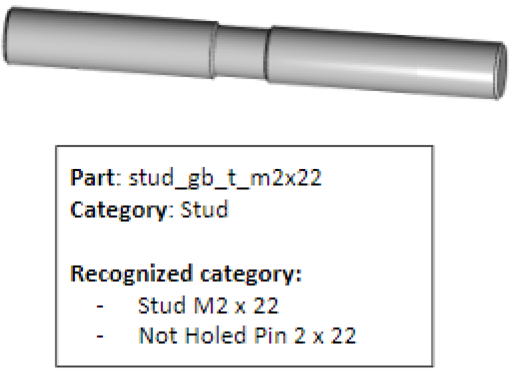
From the graphic it is evident that some categories are well recognized, others less. The curves associated with O-rings and keys are linear and delimit a large area under themselves. This because those classes are difficult to confuse with other standard parts classes, and their geometric requirements are few and simple. The curves of nuts, studs, and circlips are not linear, but the area under them is quite large, that is their recognition is however well performing. The remaining categories' curves instead decrease faster and indicate a worse recognition quality. This is justified by the fact that these categories can both be mistaken with other standard and non standard parts.

For instance, misleading interpretation of shafts as pins and/or studs can occur. Also, the recognition of studs often returns multiple results, i.e. both stud and not



(a) Stud.

(b) Pin.



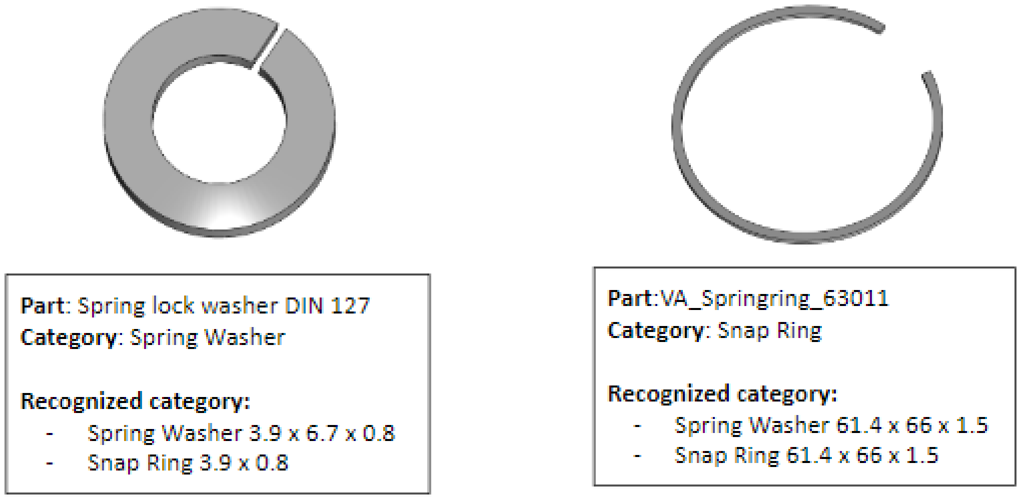
(c) Stud.

(d) Pin.

Figure 4.9: Examples of recognition results for studs and pins.

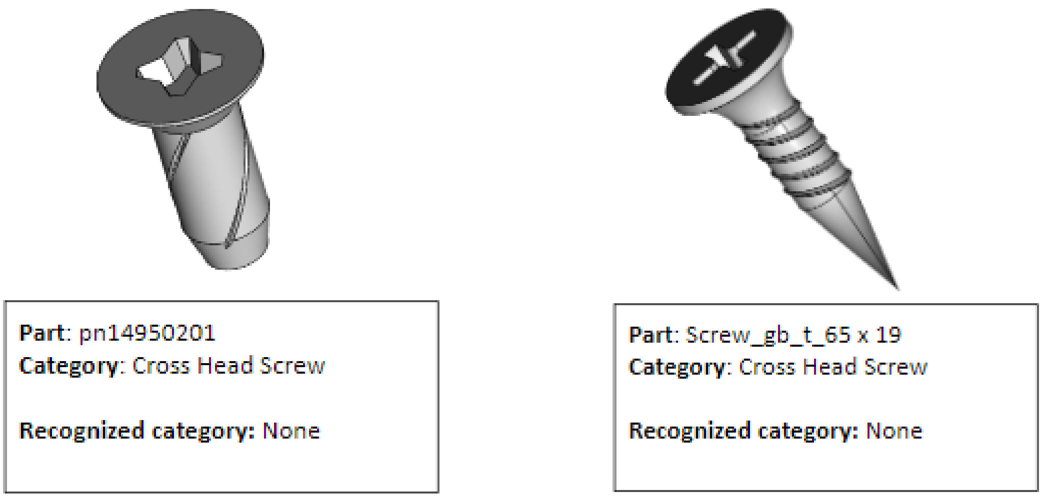
holed pin, thus increasing false positives associated with the macro category of pins but not affecting the PR curve associated with studs (Fig. 4.9a, 4.9c). The vice versa (i.e. pins also recognized as studs) instead happens less frequently (Fig. 4.9d). This situation can be explained by referring to Table 4.5: studs and pins show intersecting requirements, unless the last rule for studs, that only allows planar faces at the ends of the part, and thus is more restrictive. An example is the pin in Figure. 4.9b, which is classified as pin since it has spherical faces at its extremities. In addition, the holed pin are certainly not confused. Hence, with equal number of parts between studs and pins, without considering the subcategories, false positive associated with studs are less.

Spacers and cylindrical parts can be wrongly recognized as washers, as well as spring washers are assigned both to spring washers and snap rings, and vice versa (Fig. 4.10). In fact, by looking at Table 4.5 it is evident that snap rings and spring washers share the same requirements line. The only difference between these two



(a) Spring washer. **(b)** Snap ring.

Figure 4.10: Examples of the recognition results for washers and circlips.



(a) Screw. **(b)** Screw.

Figure 4.11: Examples of recognition results for screws.

subcategories stands in dimensional restriction. Namely, referring at Table 4.4, the admissible range for the ratio between the thickness and the inner diameter is smaller for snap rings, but it is contained in that for washers. As a consequence all the snap rings are also classified as spring washers, while some spring washers are uniquely classified.

Finally, hex screws and cap nuts can be confused in some cases, or else they can not be recognized due to modeling strategies not respecting standards, even if downloaded from repositories indicating they are standard parts (Fig. 4.11).

To conclude, the average precision-recall curve, which is the red curve in Figure 4.8, summarizes the just discussed results and indicates that the single part analysis approach provides good results, but not optimal. This confirms that shape criterion alone is not robust enough to uniquely identify standard parts especially because false positives are returned.

4.2.3 Context analysis

Since the classification based solely on shape can return misleading results, due to the shape similarity among the categories' elements, the context analysis is crucial to validate the $CANDIDATE\quad STANDARD\quad PARTS$ recognition.

The context rules

The key idea is to leverage typical engineering arrangements of components, mainly fasteners and locating elements, in real mechanical assemblies to infer admissible characteristics or mandatory requirements that the models of standard parts can or have to meet.

Following the distinction reported in Section 1.1.3, the considered arrangements concern both the relations between fasteners and the positioning of fasteners and locating elements in defined seats. Consistently with the system workflow, the guidelines validated in real objects have to be associated with the geometric and topological information of parts' models accessible up to this point, so that the process is automated. This goal can be achieved since the relation between fasteners can be translated in the existence of contacts between standard parts. The seats, then, can be easily recognized as geometric features (see Section 4.1.3), and thus the positioning in a seat will correspond with the contact between a standard part and a feature's faces.

As a consequence, two types of rules are distinguished in the algorithm, respectively the *contact with standard part* and the *contact with feature*. These are exploited as decision criteria to assess whether a part associated with a category from the single part analysis actually belongs to the category.

In Table 4.5 the decision criteria are schematized by means of a grid. In particular, the rows list the categories of standard parts to evaluate and the columns represent all the possible scenarios to check. To emphasize the distinction between the rules of contact with standard parts and of contact with features, columns are grouped into two blocks. In the first block of columns (in grey) the categories of

Table 4.5: Decision criteria used in the context analysis: possible contacts between standard parts (in grey) and contact with features associated with seats (in blue).

	Contact with standard part								Contact with feature				
	O-ring	**Key**	**Circlip**	**Nut**	**Screw**	**Washer**	**Stud**	**Pin**	**Hole**	**Groove**	**Slot**	**Through pocket**	**Pocket**
O-ring										✓			
Key											✓		✓
Circlip										✓			
Nut				✓	✓	✓	✓						
Screw				✓		✓			✓			✓	
Washer				✓	✓	✓	✓						
Stud				✓		✓			✓				
Pin									✓				

standard parts are presented, while in the latter block (in blue) the features of interest are reported. A cell is checked respectively when the contact with a standard part category of the column is admitted by the category associated with the row or when it has to be positioned in the seat corresponding with the feature of a column.

From the table it is evident that the two types of rules apply simultaneously to few categories. Indeed, only screws and studs meet both conditions of contact and positioning with other standard parts. They have to be inserted in holes due to their function, but they also can be tightened with nuts and washers. This second condition is mandatory for studs, otherwise they would be confused with pins, but it is not for screws. O-ring, keys, circlips, and pins, then, only satisfy positioning requirements. On the contrary, nuts and washers are the only categories that have to be in contact with other threaded fasteners to be correctly recognized. In fact they alone can not fulfill any function, and their positioning would be on convex cylindrical surfaces, so adding this condition to the contact with feature rules would be redundant.

Although each category has its own behavior, a general approach addressing the context analysis can be provided. That is, for each candidate standard part in the assembly, the parts in contact with it are examined. When these are classified as standard parts their category is considered, while for not classified parts the presence of specific features is verified. If a criterion is not required by the analyzed category the step is simply skipped, without affecting the process in any way.
In order to reduce the verification steps, the identified parts are analyzed in a suitable order according to the belonging candidate category, as described in the following section.

Sorting of categories

The contact between standard parts is the first criterion adopted to assess the correctness of the shape-based part category recognition.

To minimize the use of misleading information deriving from wrong classification, the order in which the candidate standard parts are evaluated is crucial. In details, as shown in Fig. 4.12, the context analysis algorithm cycles over the categories contained in the ordered set **C**, which have been sorted taking into account different factors.

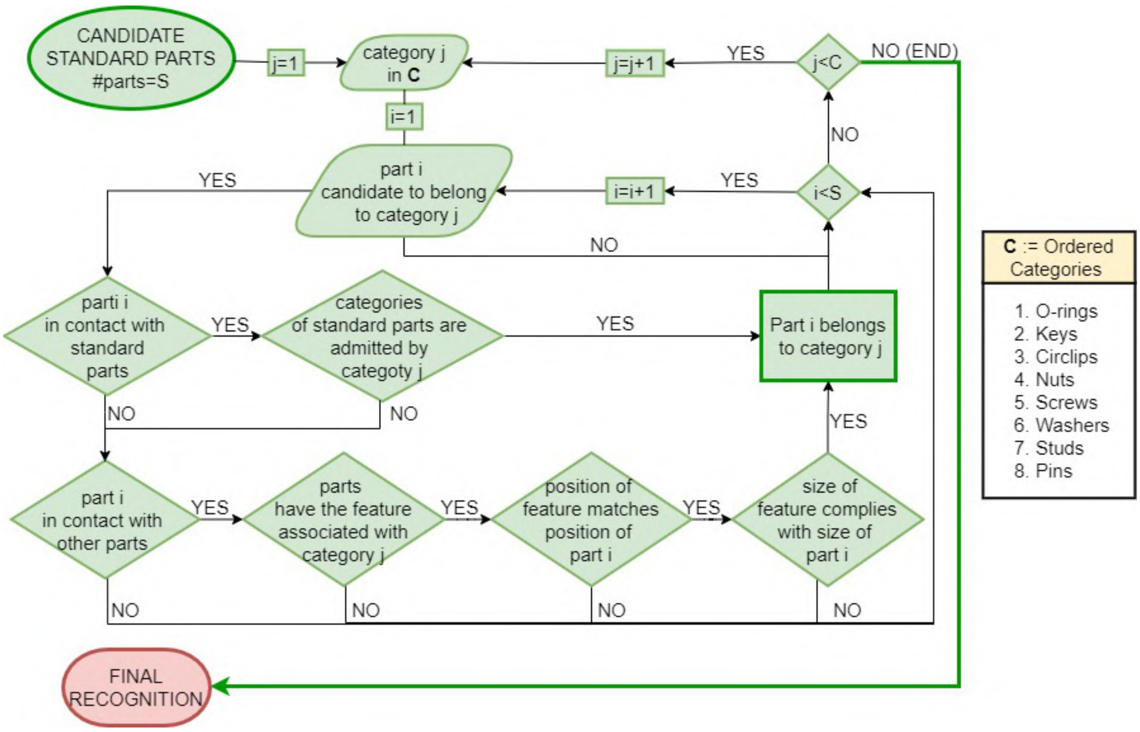


Figure 4.12: Flowchart of the algorithm implemented for the context analysis.

These factors include the possible contact with other standard parts and the probability of being confused with other categories. The categories having no relation with other standard parts and a shape that can hardly be mistaken with other categories are assessed first. Taking into account Table 4.5, these are O-rings, keys and circlips. In this way, their validation reduces the number of uncertain data, allowing for more reliable information in the subsequent evaluations. Then, the algorithm processes the parts identified as belonging to categories relying on the contact with other standard parts for their recognition, but that are also characterized by distinctive shapes, namely nuts and screws. Washers analysis follows, since their recognition mainly depends on the relation with the just confirmed screws and nuts. Finally studs and pins are taken into account. These are two categories featured with very generic and similar shapes, both inserted in holes. In this case, the relation with other standard parts, i.e. nuts and washers, rather than the positioning in defined features, allows to discriminate between them, therefore as reliable as possible data is needed.

The justification of the applied sorting can be found in results of the single part analysis validation. In fact the order in which categories will be addressed in the context analysis reflects the decreasing values of average precision shown in Fig 4.8, going from that more reliably recognized to the less.

Candidate standard parts validation

Figure 4.12 illustrates the validation process that proceeds analyzing the parts of the $CANDIDATE\ \ STANDARD\ \ PARTS$ list by category, in the order discussed above. Thus, once the category j is fixed, given a part i, with $i = 1, \ldots, S$ where S is the number of candidate standard parts returned by the single part analysis, if part i is a candidate to belong to category j, all the relations with its adjacent parts are analyzed. It must be specified that in the algorithm a general concept of contact between the parts is used. Thus, two parts are considered in contact, or adjacent, not only when they share the contact detected as described in Section 4.1.4, but also when intersect. In this way, it is possible to overcome problems arising from modeling or numerical errors, i.e. bad positioning or sizing, or modeling choices, as, for example when dealing with deformable components (Fig. 4.13).

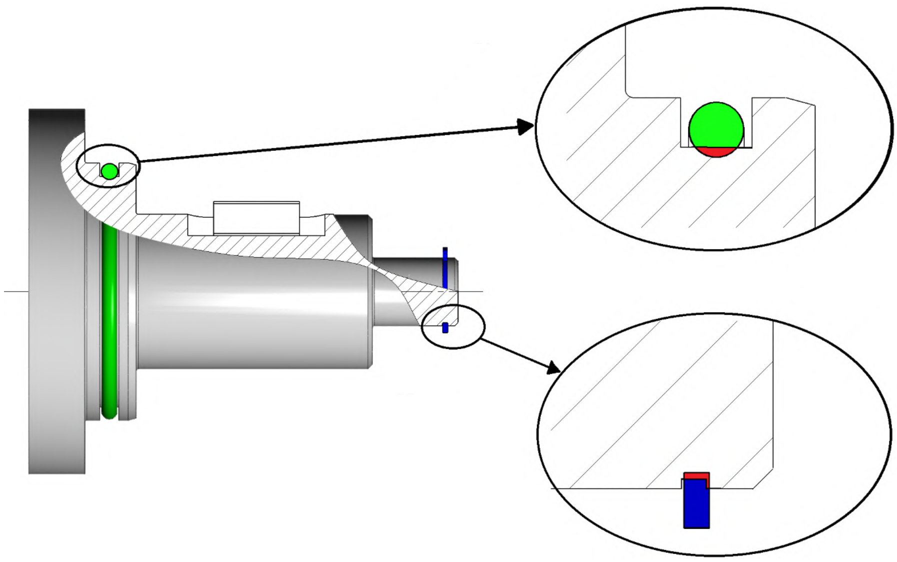


Figure 4.13: Examples of deformable standard parts, i.e. O-ring (green) and circlip (blue) that intersect the faces associated with the respective seats, i.e. grooves.

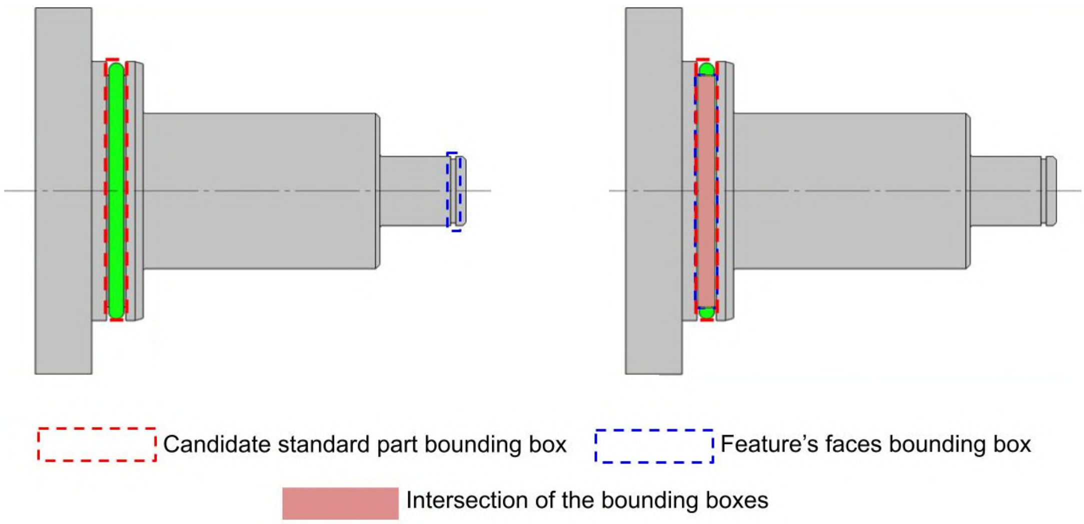


Figure 4.14: Example of the matching of the position of a candidate O-ring in respect to two grooves through bounding boxes intersection (2D view).

If the current category j allows contacts with other standard parts, adjacent standard parts are taken into account. If these belong to the admissible categories for the component category being validated, the membership of part i to category j is confirmed and the algorithm proceeds to next parts evaluation. If instead the adjacent standard parts belong to not admitted classes, or no adjacent standard parts are found, then the part positioning is verified.

The non-standard parts adjacent to the analyzed candidate standard part are now considered. For each of them, it is checked whether there is at least a feature associated with the current category j. However, the identification of a right feature on an adjacent part is not sufficient to confirm the class of the candidate standard part. The feature, in fact, could be the seat of another standard part, just as it could be a feature not associated with any standard part i. Further verification is needed regarding the matching of the position and the dimensions of the feature with those of the candidate standard part. First, the feature's faces and the candidate standard part bounding boxes must intersect (Fig. 4.14). If this condition is not met it means that the part is not included in the found feature and the classification is rejected. If instead the bounding boxes intersect, then the matching of the dimensions follows.

Since the candidate standard part must fit in the seat, some dimensions should coincide. The match is assessed within a threshold, in order to handle numerical and/or modeling errors, as well as the different modeling of deformable parts. The threshold is set on a reasonable ratio of the dimensions (i.e. dimension of the feature over that of the part or vice versa), which must be, in general, in the range $[0.9, 1]$, but the range is extended to $[0.8, 1]$ for parts associated with grooves due to the

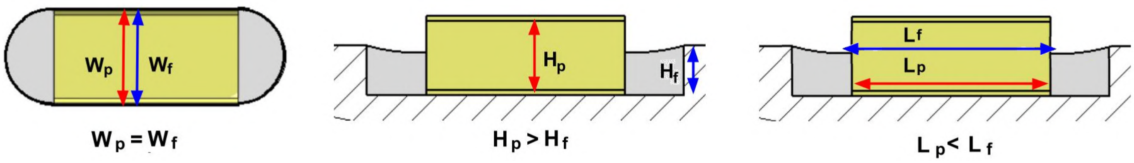


Figure 4.15: Example of the matching of the relevant dimensions of a candidate key with those of the intersecting circular-end pocket.

wide variation in their representation.
The relevant dimensions (e.g. width, length, height, diameter) of a part that must respect the ranges dictated by the feature size depend on the category analyzed. The example of a key is reported in Fig. 4.15. It shows that it must be inserted between a keyway and a keyseat, which are identified as circular-end pockets, rectangular pockets or slots. The width $\mathbf{W_p}$, the height $\mathbf{H_p}$ and the length $\mathbf{L_p}$ of the candidate key have to be compared with those of the adjacent feature, respectively $\mathbf{W_f}$, $\mathbf{H_f}$, and $\mathbf{L_f}$. In particular the width must coincide with the feature's. The key's height must be greater than the feature's, to ensure the key be linked also to another part. Finally, the length of the key must at most correspond to the feature's length, that in this particular case of a circular-end pocket corresponds with the length of the planar sides, otherwise it means that the key can not be contained in the feature.

In conclusion, as shown in Figure 4.12, if the i-th part of the *CANDIDATE STANDARD PARTS* list meets the requirements of contact with other standard parts, or the condition of being positioned in a specific seat with compliant dimensions, it can be established that the candidate standard part was correctly classified by the single parts analysis. Its category j is confirmed, and if other categories were assigned to the part, they are discarded.

The returned output, that is the actual output of the multi-step recognition approach and thus of the module described in this section, is the list of *STANDARD PARTS* of the CAD model assigned to a unique standard part category, each equipped with specifications for subcategories and dimensional values.

4.2.4 Standard parts recognition: results and discussion

While for single parts the creation of the ground truth for the evaluation of the method can largely exploit the classification provided by professional repositories for engineers, in the case of assemblies so far no public datasets exist appropriate for our aims.

The dataset

This section focuses on issues and challenges in creating a suitable ground truth not only to validate the standard parts recognition, but also the modules that will be next described.

The main characteristics a dataset should convey to evaluate the tools developed in this work can be summarized as follows. The dataset should include CAD models of complex mechanical assemblies with boundary representation in standard formats. It is preferable to have industrial models, or at least, realistic ones. That is, the models should not be ideal, in the sense that their design and the positioning of their parts are perfect or previously fixed and at the same time the defects should be compatible with those of industrial conventions and habits. For directly usable datasets, the standard parts, should be labeled, mechanical features and contacts information should be available, as well as the existence of meaningful groups of parts, such as subassemblies.

No public datasets exist providing CAD models equipped with such data, even because real industrial CAD models are usually subjected to confidentiality and hence they are not released. However, the use of not industrial models would limit the analysis to some scenario, neglecting instead some of particular interest in this research, and this is the first issue. Then, it is known that open source libraries (e.g. GrabCAD [8]) exist providing CAD models of objects of different types, among which also real mechanical products belonging to the treated classes. However, these do not contain the information cited above, and even in some cases neither the needed file format is available.

It can be assessed that, in general, it is hard to find CAD assembly benchmarks of complex objects made of many parts (i.e. more than 100). In recent years, some efforts have been made in collecting CAD models of assemblies and producing ad hoc datasets for different purposes (e.g. [47, 81, 113]). However, referring to the research here carried out, not all the models there collected are effective. As a matter of fact, besides the missing of labels, the available datasets are usually heterogeneous, in the sense that they include a too wide variety of classes of products and moreover they can involve simple objects, even single parts. Or else, when they provide mechanical models, it happens that the models are ideal or designed by students, and thus having shapes inconsistent with industrial practice or even too simplified.

The limits reported underline the difficulties in the creation of a meaningful ground truth in terms of number of elements and available information. Consequently, to validate the work and each of the developed modules, a set of CAD models in STEP format of mechanical assemblies mainly belonging to the class of mechanical and electronic equipment was collected.

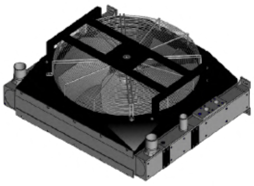
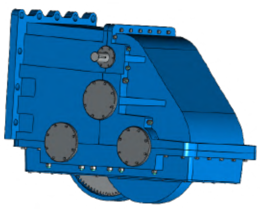
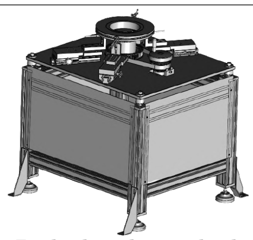
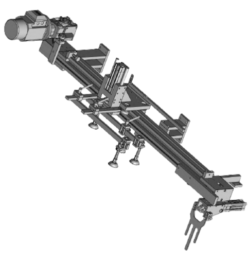
The dataset consists of 16 actual industrial CAD models supplied by different engineering companies that completely satisfy the design and structure requirements. In addition, further models have been also included in the dataset, which are carefully selected from the various online repositories to be suitable for the tools validation. This is done, on the one hand, to enrich the dataset increasing the number of models to be evaluated, on the other hand, to address as many scenarios as possible and analyze the differences between industrial and not industrial CAD models. Starting from that dataset, the results obtained by the presented methods are assessed with the help of engineers.
For sake of completeness, the whole dataset is provided in the Appendix A.

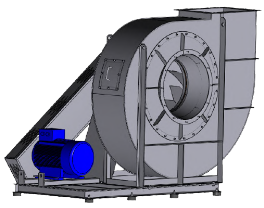
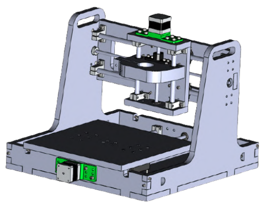
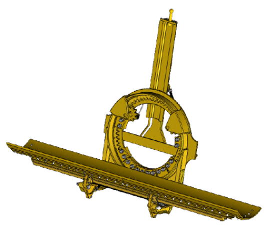
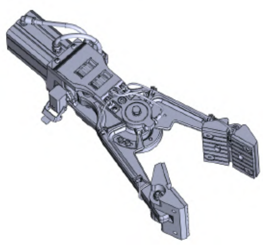
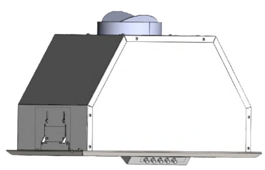
Standard parts recognition assessment

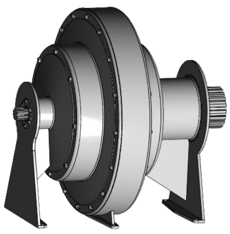
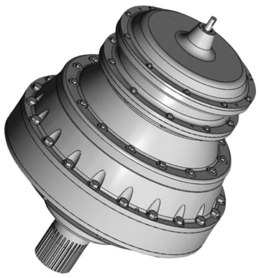
In Table 4.6 some of the industrial assemblies used to validate the approach are collected. The results obtained with single part and context analysis are organized in the table as follows. The fourth and the fifth columns show respectively the number of candidate standard parts recognized after the single part analysis and the number of standard parts recognized at the end of the multi-step approach. The two columns report both the total and the per-category number of recognized standard parts recognized.
That is, a set contains all the parts equally classified, both in terms of subcategory and dimensions. The last column, then, provides the number of a candidate standard parts/sets that were rejected/reassigned emphasizing the importance of the context analysis. The green and the red colors indicate the ones correctly and wrongly modified.

Results show that the provided multi-step approach is overall robust, and can recognize and correctly classify most of the standard parts included in mechanical assemblies. Comparing the list of candidate standard parts returned by the single parts analysis and the final list of standard parts, it is evident that the context analysis allows to overcome the issues arising from the evaluation of the shape only. In particular, in the first step, classes such as studs, keys, O-rings and washers are over-recognized. That is, several false positive are returned, due to the existence in mechanical assemblies of simple components that can be mistaken for standard parts if considered out of context. Looking at Table 4.6, these categories of standard parts are those that actually appear most among the *refused* items.

Table 4.6: CAD models used to validate the multi-step recognition approach. In green the parts correctly refused/assigned by the context analysis. In red the parts wrongly refused/assigned by the context analysis.

N.	CAD Model	n°of parts	n°of candidate standard parts	n°of standard parts	Candidate standard parts refused/assigned
1	 **Radiator**	201	108 standard parts grouped in 55 sets of elements with same subcategory and dimensions	56 standard parts grouped in 7 sets of elements with same subcategory and dimensions	Refused: - 42 candidate O-RINGS grouped in 42 sets - 10 candidate KEYS grouped in 6 sets
2	 **Gearbox**	426	286 standard parts grouped in 23 sets of elements with same subcategory and dimensions	284 standard parts grouped in 22 sets of elements with same subcategory and dimensions	Refused: - 2 candidate CIRCLIPS grouped in 1 set Assigned: - 13 candidate STUDS/PINS grouped in 1 set are assigned to studs - 42 candidate CIRCLIPS/WASHERS grouped in 3 sets are assigned to WASHERS
3	 **Disks break test bed**	153	44 standard parts grouped in 14 sets of elements with same subcategory and dimensions	38 standard parts grouped in 13 sets of elements with same subcategory and dimensions	Refused: - 6 candidate SCREWS grouped in 1 set
4	 **Linear axis for automation**	199	48 standard parts grouped in 13 sets of elements with same subcategory and dimensions	31 standard parts grouped in 7 sets of elements with same subcategory and dimensions	Refused: - 6 candidate STUDS grouped in 2 sets - 1 candidate WASHER - 2 candidate CIRCLIPS grouped in 1 set - 2 candidate WASHER grouped in 1 set - 4 candidate SCREW grouped in 1 set Assigned: - 1 candidate STUD/PIN is assigned to pin

N.	CAD Model	n°of parts	n°of candidate standard parts	n°of standard parts	Candidate standard parts refused/assigned
5	 **Fan assembly**	325	244 standard parts grouped in 31 sets of elements with same subcategory and dimensions	228 standard parts grouped in 18 sets of elements with same subcategory and dimensions	Refused: - 14 candidate KEYS grouped in 11 sets - 2 candidate CIRCLIP grouped in 2 sets
6	 **Cartesian slider**	237	113 standard parts grouped in 16 sets of elements with same subcategory and dimensions	104 standard parts grouped in 10 sets of elements with same subcategory and dimensions	Refused: - 9 candidate STUDS grouped in 6 sets
7	 **Agricultural steel-work assembly**	455	314 standard parts grouped in 18 sets of elements with same subcategory and dimensions	308 standard parts grouped in 16 sets of elements with same subcategory and dimensions	Refused: - 6 candidate STUDS grouped in 2 sets
8	 **Robotic gripper**	318	170 standard parts grouped in 33 sets of elements with same subcategory and dimensions	113 standard parts grouped in 29 sets of elements with same subcategory and dimensions	Refused: - 53 candidate STUDS grouped in 2 sets - 4 candidate CIRCLIP grouped in 2 sets Assigned: - 2 candidate STUDS/PINS grouped in 1 set are assigned to pin - 6 candidate CIRCLIPS/WASHERS grouped in 2 sets are assigned to WASHERS
9	 **Cooker hood**	106	32 standard parts grouped in 6 sets of elements with same subcategory and dimensions	31 standard parts grouped in 5 sets of elements with same subcategory and dimensions	Refused: - 1 candidate KEY

N.	CAD Model	n°of parts	n°of candidate standard parts	n°of standard parts	Candidate standard parts refused/assigned
10	 Axial reducer	306	220 standard parts grouped in 26 sets of elements with same subcategory and dimensions	173 standard parts grouped in 13 sets of elements with same subcategory and dimensions	Refused: - 6 candidate O-RING grouped in 6 sets - 16 candidate WASHER grouped in 2 sets - 14 candidate WASHER grouped in 1 set - 11 candidate CIRCLIPS grouped in 4 sets Assigned: - 1 candidate STUD/PIN is assigned to PIN
11	 Planetary gearbox	490	310 standard parts grouped in 39 sets of elements with same subcategory and dimensions	285 standard parts grouped in 32 sets of elements with same subcategory and dimensions	Refused: - 25 candidate WASHER grouped in 7 sets Assigned: - 10 candidate STUD/PIN grouped in 4 sets are assigned to PIN

For example, the case of the radiator in model N.1 is significant. From the table it is evident that about half of the candidate standard parts is not confirmed by the context analysis. Those are 42 metal rings actually forming a grid and 10 metal blocks forming the body of the product. The "42 candidate O-RINGS grouped in 42 sets" (i.e. one set for each part because they all have different diameter) are simply modeled with a single convex torodial face, and, thus, according to the geometric requirements collected in Table 4.3, they are recognized as O-rings after the single parts analysis. However, the rings of the radiator are not mounted in any circular grooves, but they are welded on a radial metal structure. As a consequence the context analysis refuses the recognition and the 42 parts are correctly discarded. (Fig. 4.16).

Still discussing the model N.1, the "10 candidate KEYS grouped in 6 sets" (i.e. there are 2 sets made of 2 parts with same height, length, and width and other 4 parts each with different dimensions) which are correctly refused by the context analysis, in reality are simple structural components having the shape of a parallelepiped

(Fig. 4.17). In fact, they have 6 planar faces, divided in three pairs of equal and anti-parallel faces, two by two perpendicular with each other. Checking Table 4.3, these properties meet the requirements of keys of type B. Although, by applying the context analysis, it results that the 10 components are not keys because they are not in the corresponding seats.

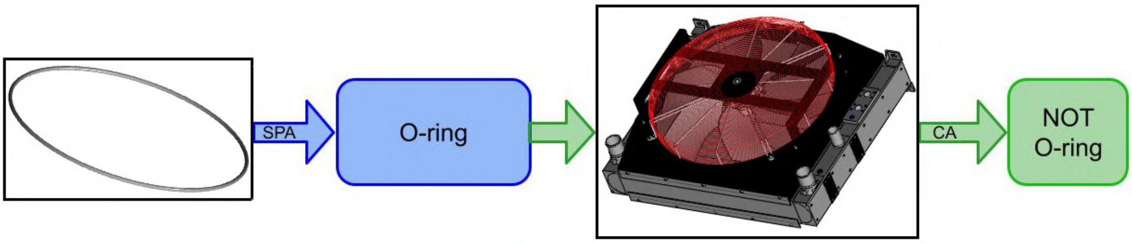


Figure 4.16: Example of toroidal part recognized as O-ring by the single part analysis (SPA), but refused by the context analysis (CA).

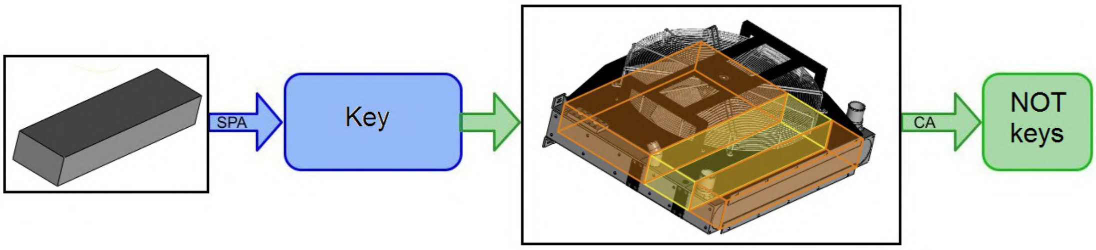


Figure 4.17: Example of parallelepiped-like parts with different dimensions recognized as keys by the single part analysis (SPA), but refused by the context analysis (CA).

The other strength of the context analysis is therefore the ability to disambiguate the assignment of a part to multiple categories that have similar shape requirements.

The gearbox in model N.2 represents a remarkable example of this scenario. The single parts analysis, in fact, returns 55 components associated with more than one category. In particular, there are "13 candidate STUDS/PINS grouped in 1 set" (i.e. all the 13 parts have same dimensions) that are parts recognized both as studs and not holed pins. In fact, they have 9 faces such that 4 are planar, 2 of which are not ring, 3 are cylindrical closed not hollow, and the others are symmetric to the axis of the cylinders. By referring to Table 4.3, the components meet the requirements both of studs and pins. However, the context analysis correctly assigns the 13 parts to the category of studs, since contact with nuts is found (Fig. 4.18).
The other "42 candidate CIRCLIP/WASHERS grouped in 3 sets" (i.e. there are 3 sets of respectively 28, 8, and 6 parts having same diameters and thickness) are thus recognized both as snap rings and spring washers, due to the identical requirements

for those two categories reported in Table 4.3. But finally the 42 components are assigned to washers because of context: they are not inserted in grooves, rather they are in contact with screws (Fig. 4.19).

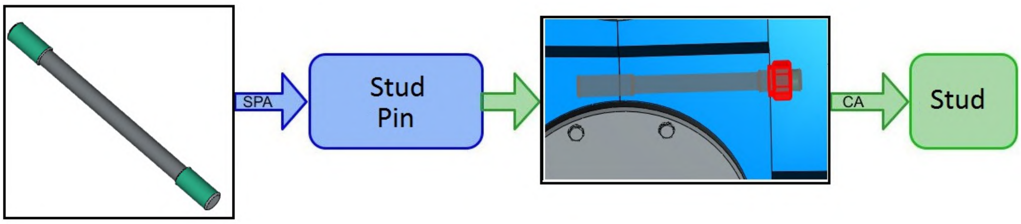


Figure 4.18: Example of part recognized both as stud and pin by the single part analysis (SPA), but assigned to stud by the context analysis (CA).

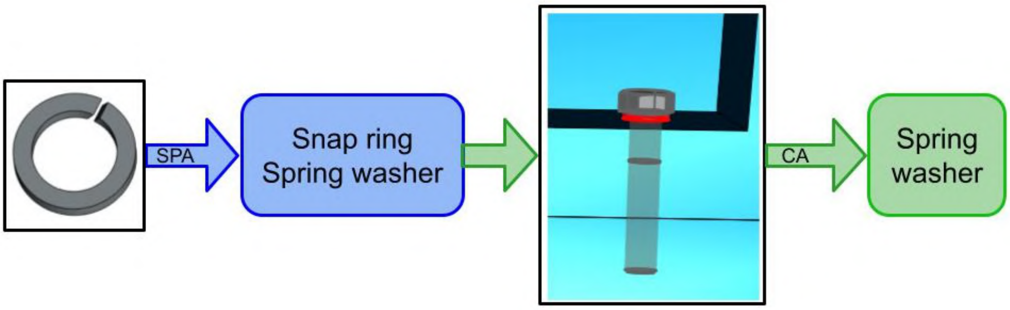
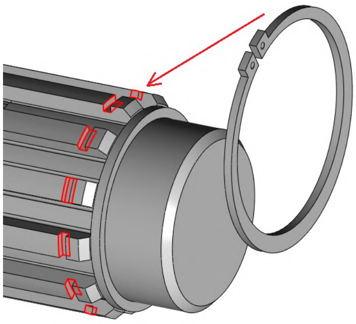
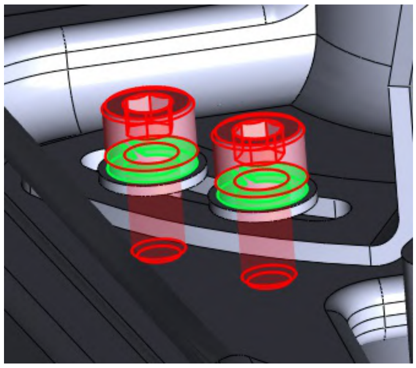


Figure 4.19: Example of part recognized both as snap ring and spring washer by the single part analysis (SPA), but assigned to washers by the context analysis (CA).

Despite the generally good results, the context analysis in some cases fails. In fact, it can happen that standard parts resulting from the single parts analysis are wrongly rejected. However, these problems in general arise from modeling errors that are outside the tolerance threshold in the representation of features and parts (e.g. parallelism or perpendicularity between faces not detected), or from a incorrect positioning of parts in relation to the associated features (e.g. their bounding boxes never intersect within the considered tolerance), or else because some standard parts are not even recognized by the single parts analysis and thus the context rules fail. The axial reducer in model N.10 is an example. Namely, 11 circlips are refused because misplaced with respect to the associated groves or placed in grooves not modeled according to standards (Fig. 4.20a). Moreover, since the model is a CAD assembly created by students and thus it does not strictly follow all standards, screws are missing and consequently a set of 14 washers is refused because contacts with the respective screws is not validated. Another example is the model N.8 where, instead, a set of screws is not recognized for shape, since they do not respect some proportional requirements, i.e. the ratio between the height of the head and the length of the stem is out of the standards. Consequently the washers tightened on those screws, even if they are recognized by the single parts analysis, are wrongly refused after the context evaluation (Fig. 4.20b).



(a) Original B-rep model with custom topological entities.



(b) Normalized B-rep model with maximal topological entities.

Figure 4.20: Examples of standard parts not correctly recognized: (a) circlip refused for not standard modeled groove. (b) Screws not recognized for shape and thus washer refused for context.

Hence, wrong classifications result when modeling errors occur, but they do not totally depend on context analysis requirements, which instead turn out to be sufficiently robust. It can be broadly assessed that the standard parts that are rejected at this stage are lower than those that are correctly recognized, and therefore the context analysis is certainly promising and can overcome most of the issues arising from the only shape-based classification.

Visualization

In Fig. 4.21 an example of how the developed tool displays the classification results is presented. In the *STANDARD PARTS* form, the candidate standard parts and the final recognized standard parts are listed in the first and in the second column respectively. In the first list the non confirmed candidate parts are also highlighted. In both columns, the identified components are grouped according to subcategory and dimensions, i.e. all the repetitions of the same standard part are grouped together. By clicking on one item the associated components are highlighted in the CAD model to provide an overview of their arrangement within the assembly. Finally the extracted engineering dimensions of the elements of the set are provided on the third column of the same form and an example of the dimensions extracted for a set of keys is further shown in Fig. 4.22.

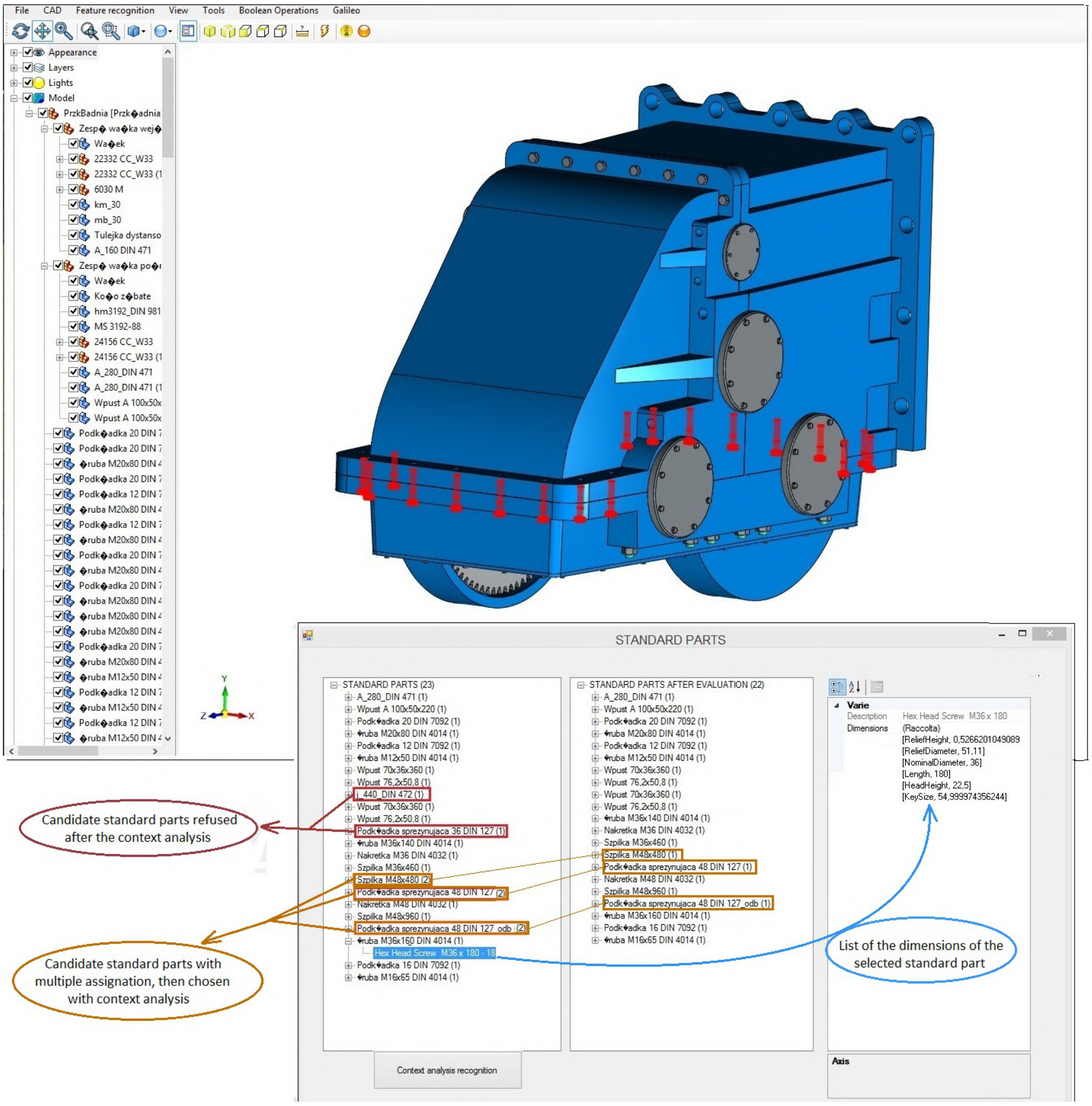


Figure 4.21: Visualization of the multi-step recognition results in the developed tool.

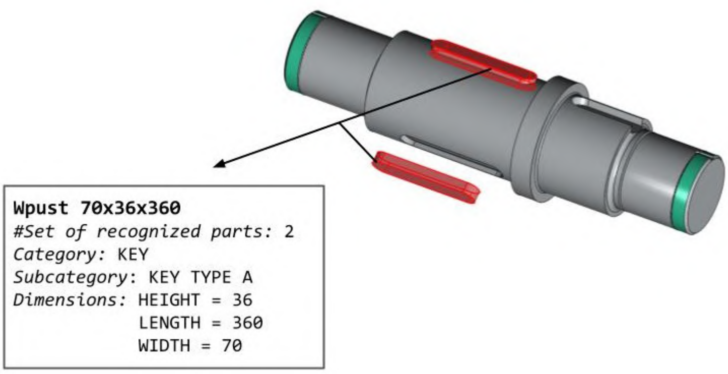


Figure 4.22: Example of dimensions extracted for a set of two keys.

4.3 Mounting analysis module

Once the recognition phase is completed, each part of the CAD model has been analyzed and assigned to a category of standard parts or to the custom designed parts set. This classification is the first semantic information actually extracted, since it assigns to parts a functionality and an engineering meaning, that is they can be fasteners, locating elements or structural parts.
In addition, due to the geometric processing outcomes, the list of $COUPLINGS$ is already accessible, that provides the existing contacts between pairs of parts, as well as the list $MOUNTINGS$, defined as attributes of the coupling (see Section 4.1.4).

These three types of data stand at the basis of the liaisons, as defined in Section 3.3. In fact, the new structure relies on an improved understanding of contacts. In particular, only custom designed parts can establish a liaison; all other parts are afterwards assigned as contacts properties. This seeks to replicate the general engineering reasoning and reflect the mechanical schemes applied in real products (refer to Section 1.1).

Therefore, this module focuses on inspection and restructuring of the available knowledge with the final goal of returning the representation of the CAD models based on liaisons.

4.3.1 Fastening functional sets

A further operation concerning standard parts is carried out in order to enhance their functional interpretation.

It has been mentioned that typical arrangements of fasteners and locating components tightened together can exist in mechanical products (see Section 1.1.3). The knowledge of these groups of parts is of particular interest since they allow some parts to fulfill specific function that they can not perform on their own. As a matter of fact, when used alone or combined with others in a set, mechanical components assume different roles within the assembly and give a different meaning to the connection they create. For instance, the presence of a screw to connect two parts or of a screw blocked by nuts and spaced by washers strongly affects the order and the number of operations needed to create the connection, as well as its strength. Also, a nut not screwed on a threaded shank does not comply with its function of fastening, so it has to be better investigated.
Identify the presence of such arrangements in the CAD model of an assembly and

store them as functional sets to include in the liaisons definition result thus fundamental steps. These steps have to be carried out in the data extraction phase to enhance the semantic information associated to the model, since the knowledge of functional sets can influence the subsequent more complex tasks, e.g. design optimization, assembly/disassembly sequence planning, as well as cost estimation.

According to the categories of standard parts addressed, the possible sets would be covered by the following two cases: screw-nuts-washers and stud-nuts-washers. In both cases at least a nut or washer must be found, otherwise reference is made to the use of only one fastener, but, when present, the number of nuts and washers is variable (Fig. 4.23).

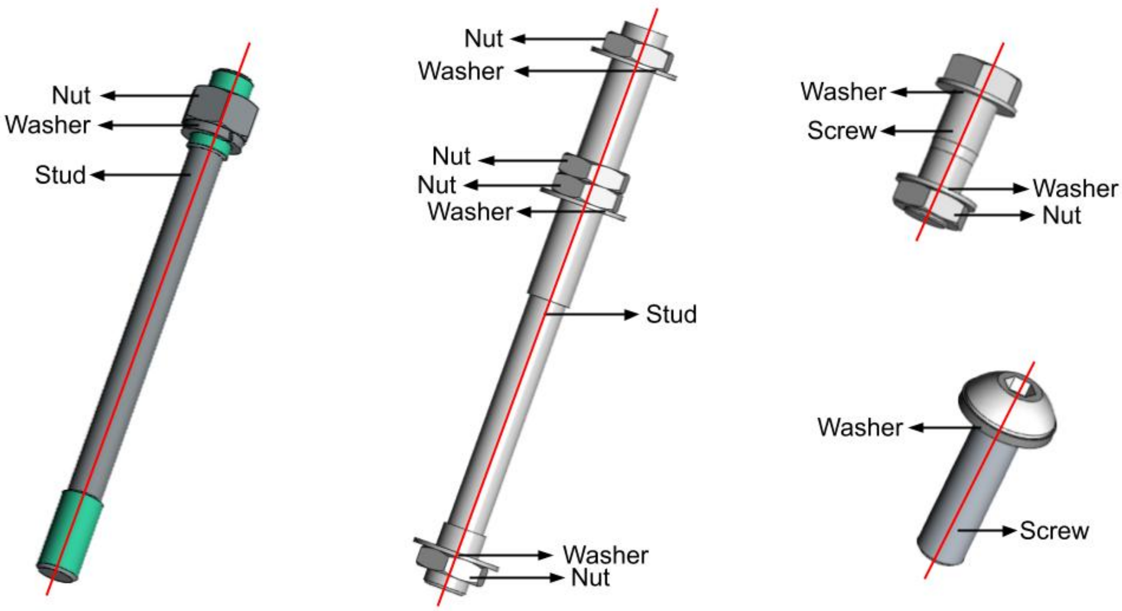


Figure 4.23: Examples of fastening functional sets.

An interesting feature of these components is that they are axisymmetric and thus to be correctly mounted with each other their centers have to be aligned and the axes parallel.

From the computational point of view, the identification of the fastening functional sets is achieved by the analysis of the couplings and the mountings between standard parts. As shown in Algorithm 1, given a standard part S, such that its category is screw or stud, the list C of couplings is scrolled through to identify all the couplings involving S. Among the identified couplings, if there are some where the second part is also a standard part, i.e. washer or nut, these are considered and the second part is named as R. If there is a coupling, i.e. the components are in contact, it follows almost immediately and may be assumed that they belong to the same functional set. However, to avoid trivial mistakes the best practice is to further evaluate the geometric properties of the parts.

Algorithm 1 Fastening set computation

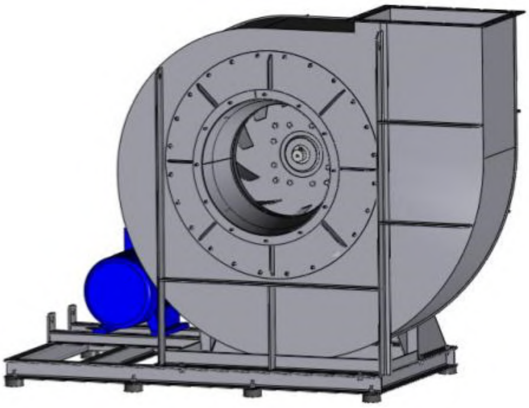
```
Data: S:=screw/stud; C := {c_i(P_1, P_2)} list of couplings;
Result: F fastening set
F := {S};
foreach c(P_1, P_2) ∈ C do
    if P_1 ≠ S ∧ P_2 ≠ S then
        next
    end
    if P_1 = S then
        R := P_2;
    else
        R := P_1;
    end
    if R is nut or R is washer then
        a_R := axis R; a_S := axis S;
        c_R := center R; c_S := center S;
        if a_R ∥ a_S and vector(c_R − c_S) ∥ a_R then
            add R to F;
        end
    end
end
if F ≠ {S} then
    return F = {S, R_1, ..., R_k}
end
```

Since the axis and the center of standard parts are known, it is sufficient to read and compare them. In details, if the axis $\mathbf{a}_S$ of the screw/stud is parallel with the axis $\mathbf{a}_R$ of the other standard part, and the centers c_S and c_R are aligned, the part R is added to the fastening set identified by the screw/nut.

Once all the couplings between the current screw/stud S and the other standard parts have been analyzed, the functional fastening set $F(S, R_1, \ldots, R_k)$ is defined and given as result, with $R_1, \ldots, R_K$ the parts returned by the algorithm. $F(S, R_1, \ldots, R_k)$ is equipped with its own center and axis, besides the properties associated with each single standard part constituting it, that remain accessible.

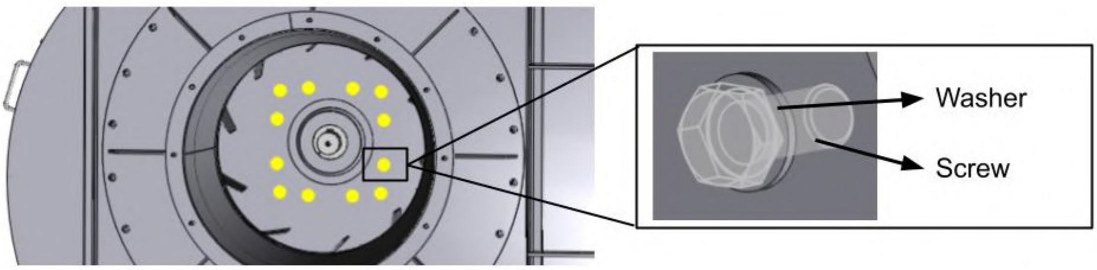
The process is iterated for each screw and nut of the CAD model, and the final output is actually a reorganization of the standard parts. In details, standard parts will include both the components not grouped in any set, and the just computed fastening sets.

Example For sake of completeness, the computation of the fastening functional sets for a fan assembly starting from the list of its standard parts is reported.

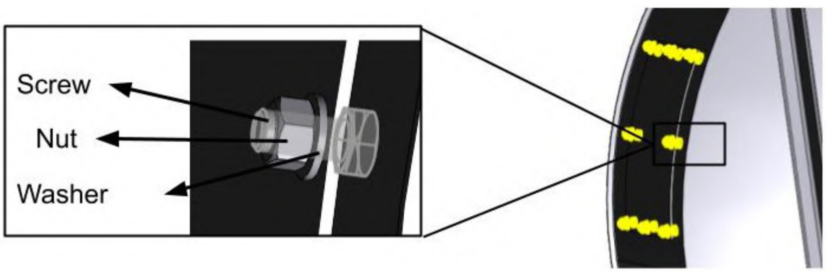


TOT standard parts	228
TOT standard parts+fastening sets	**87**
Single parts	12
Sets of 2 parts	20
Sets of 3 parts	44
Sets of 4 parts	11

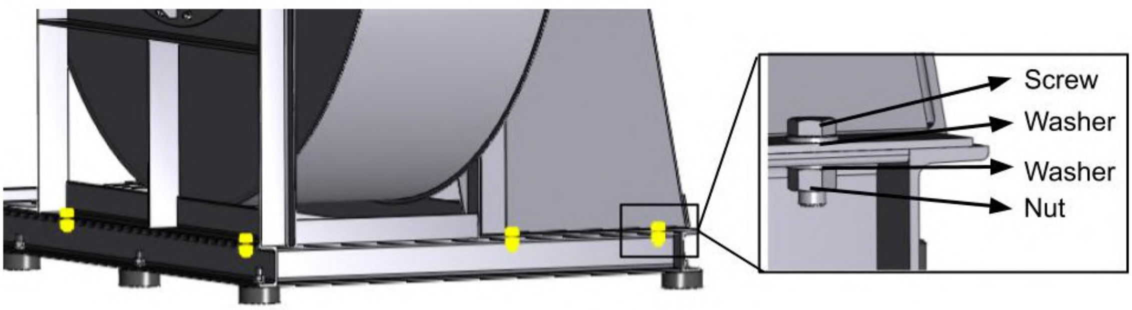
Figure 4.24: CAD model of a fan assembly and summary of the identified fastening sets.



(a) Set of 2 standard parts.



(b) Set of 3 standard parts.



(c) Set of 4 standard parts.

Figure 4.25: Some examples of the fastening sets of the fan assembly.

The fan assembly model is chosen as example since it is relevant in this context. In fact, the standard parts returned by the recognition approach are 228 (see model N.5 in Table 4.6), distinguished between screws, nuts and washers. As shown in Figure 4.24, after the fastening sets computation the number of elements in the generalized list of standard parts, which includes both the individual standard parts and the fastening sets, is reduced to 87. In particular, 20 sets of two parts (e.g. Fig. 4.25a), 44 sets of three parts (e.g. Fig. 4.25b), and 11 sets of four parts (e.g. Fig. 4.25c) are computed, while only 12 parts remain unpaired. This scenario underlines the importance of the fastening sets, since it is evident that most of the standard parts of the assembly need to be treated in conjunction with each others to correctly understand their function.

As it will be better explained in next section, the knowledge of fastening sets is also crucial in the liaison computation to correctly assign standard parts to each liaison without excluding any of them.

4.3.2 Computation of liaisons

At this point, all the necessary data are effectively available to generate the liaisons and the list containing them.

According to the definition given in Section 3.3.3, a liaison is defined between two custom designed parts in contact, and for each pair of these there exists at most a single liaison. Thus, the list of couplings is first considered, each element of which is representative of a pair of faces in contact belonging to two general parts of the assembly. Given a coupling $c_i(f_1, f_2)$, the membership of the associated parts P_1 and P_2 in the custom designed parts is evaluated. In the affirmative, a liaison $l(P_1, P_2, C, M, S)$ is created and the coupling c_i is added as item of the list C of the couplings associated with that liaison, while the lists M and S remain empty. This happens unless a coupling between two different contact faces of P_1 and P_2 was already found and thus a liaison identified by the two current parts already exists. In the latter scenario, only the coupling c_i is added to the list C.
The examination is repeated for each coupling, and at the end of the process all the defined liaisons are collected in a list L. However, for each liaison, the lists of the mountings and of the standard parts still have to be filled in. Like what was just accomplished, the list of mountings is scrolled and each mounting between two custom designed parts is assigned to the relative liaison. As for standard parts, instead, the reorganized list is considered, which includes both single components and fastening sets (see Section 4.3.1). By going through the couplings, for each

standard part all the custom designed parts in contact with it are stored, as well as for each fastening set the custom designed parts in contact with at least one of the composing parts. If the custom designed part identified are at least two, each possible pair of them is considered and if a liaison exists identified by one of these pairs of components, the standard part/fastening set is assigned to it.

These seem quite long operations since every item of the lists of couplings and mountings have to be analyzed. Although, all the needed information, such as the types of parts or the existence of contacts, is already computed and easily accessible as properties associated with each coupling/mounting. As a result, simple queries about the existence or not of an element in a list are performed, which do not require high computational costs and time.

At the end on the liaisons creation, it can happen that some standard parts remain not assigned to any liaison. The reasons are different and are discussed in the following, pointing out whether these are situations that can be solved by some reasoning on conventional representations and engineering knowledge, or issues that are not really easy to deal with:

- A standard part is excluded from liaisons when it is in contact with only one custom designed part. This situation is attributable to the issue of incomplete models or missing components (see Section 1.3.1). In fact, a typical example is the case of a fastener used to join a part of the assembly, with which is thus in contact, to an external component that instead is not represented in the analyzed model (Fig. 4.26). This problem can not be overcome, in the sense that the standard part will remain out of the liaisons list, rather the information is stored and may be useful in next evaluations and tasks or it can be used as a warning of incompleteness of the model.

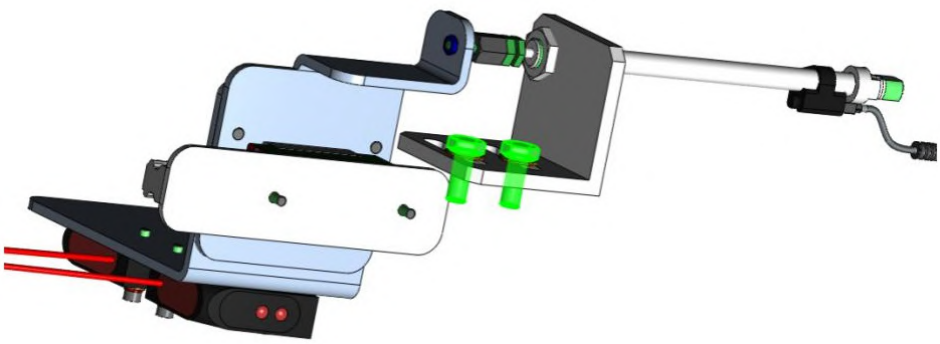


Figure 4.26: Screws not included in liaison because one of the custom designed parts is missing.

It must be clear that washers and nuts do not need to be associated with that scenario even if they usually are in contact with at most one custom designed part from the modeling standpoint (see Fig. 4.25). This is because they are considered within the fastening sets, therefore it is sufficient that a part of the set (i.e. the screw or stud) is in contact with two custom designed parts to assign all the parts of the set to a liaison.

- The standard part fastens two adjacent structural parts, but the contact between the fastening part and one of the two structural parts is not identified due to modeling errors, i.e. displacement of parts that generates clearances and intersections, or designer choices, e.g. not modeled holes (Fig. 4.27). The second is an acceptable scenario, since in some cases the holes are not preformed into the parts when they are manufactured because they are generated by the insertion of fasteners, and therefore are not designed in the CAD models of the parts. To overcome the misleading situation, it is possible to evaluate the standard parts not included in liaisons and check if there are some intersection with other custom designed parts.

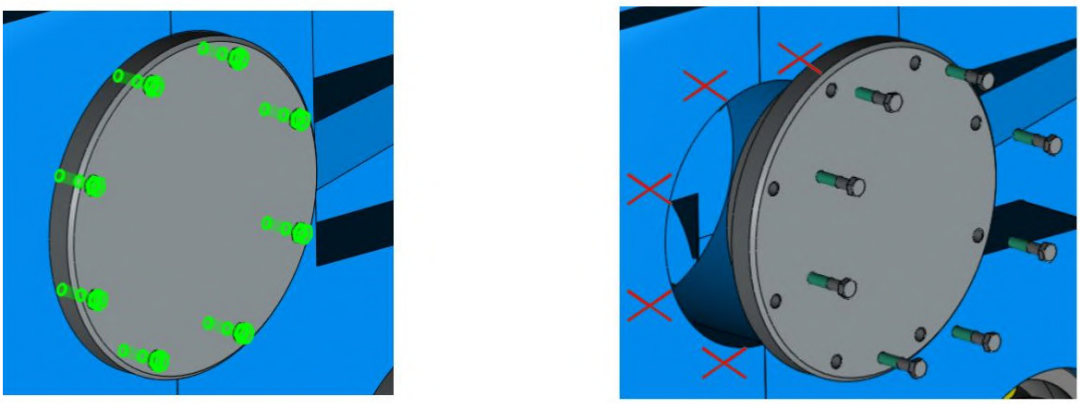


Figure 4.27: Screws not included in liaison because one of the custom designed parts has not modeled holes.

- A standard part or a fastening set is in contact with two custom designed parts and links them together, but these two components are not in contact with each other, i.e. no liaisons exist between them (Fig. 4.28). This situation is fixed by creating a new virtual liaison between the two components, which is only equipped with the standard parts involved. It is clear that this will not be a real liaison in the sense of the definition given in Section 3.3.3, but it is significant to consider and save the connection generated by such standard parts.

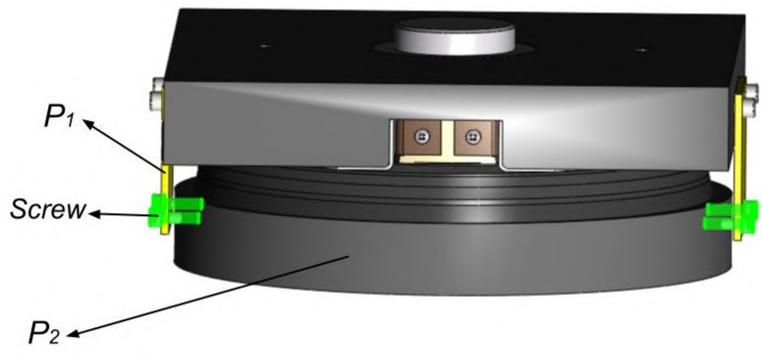


Figure 4.28: Screws connecting two parts P_1 and P_2 that do not underlie a liaison because they are not in contact.

- The standard part is an O-ring and it is positioned between two custom designed parts in contact with each other, i.e. underlying a liaison. However, especially when the O-ring is modeled as a single toroidal surface, the contact between the standard part and one or both the custom designed parts is not detected because it is not a surface contact or it generates a volumetric intersection (see Fig. 4.13). This issue can be solved by exploiting the information used in the standard parts recognition relative to the matching between standard parts and features. That is, the O-ring can be assigned to the liaison involving the two parts having the grooves recognized as seats of it.

In conclusion, the liaisons computation phase returns as final output the list L of liaisons along with a support list that indicates if some standard part has not been assigned to any liaison. In particular, the list L provides pairs of parts in contact equipped with properties aimed at enhancing their semantic understanding. The properties of each liaison are deduced from the analysis and processing of the data of the couplings associated, and are given as accessible data of the liaison.

Visualization and inspection

To allow a more practical analysis of the new assembly representation and the browsing of the provided data, an additional interface of the software of the company Hyperlean has been realized. It consists of two main forms. In the *Viewer* form, the CAD model is shown and can be managed as in common CAD systems, sided by the original tree structure (i.e. the form 1 in Fig. 4.29). The *Liaison assembly* form shows the new organization of the assembly based on the liaison data structure (i.e. the form 2 in Fig. 4.29) and allows the inspection of the extracted properties (i.e. the forms 3 and 4 in Fig. 4.29).

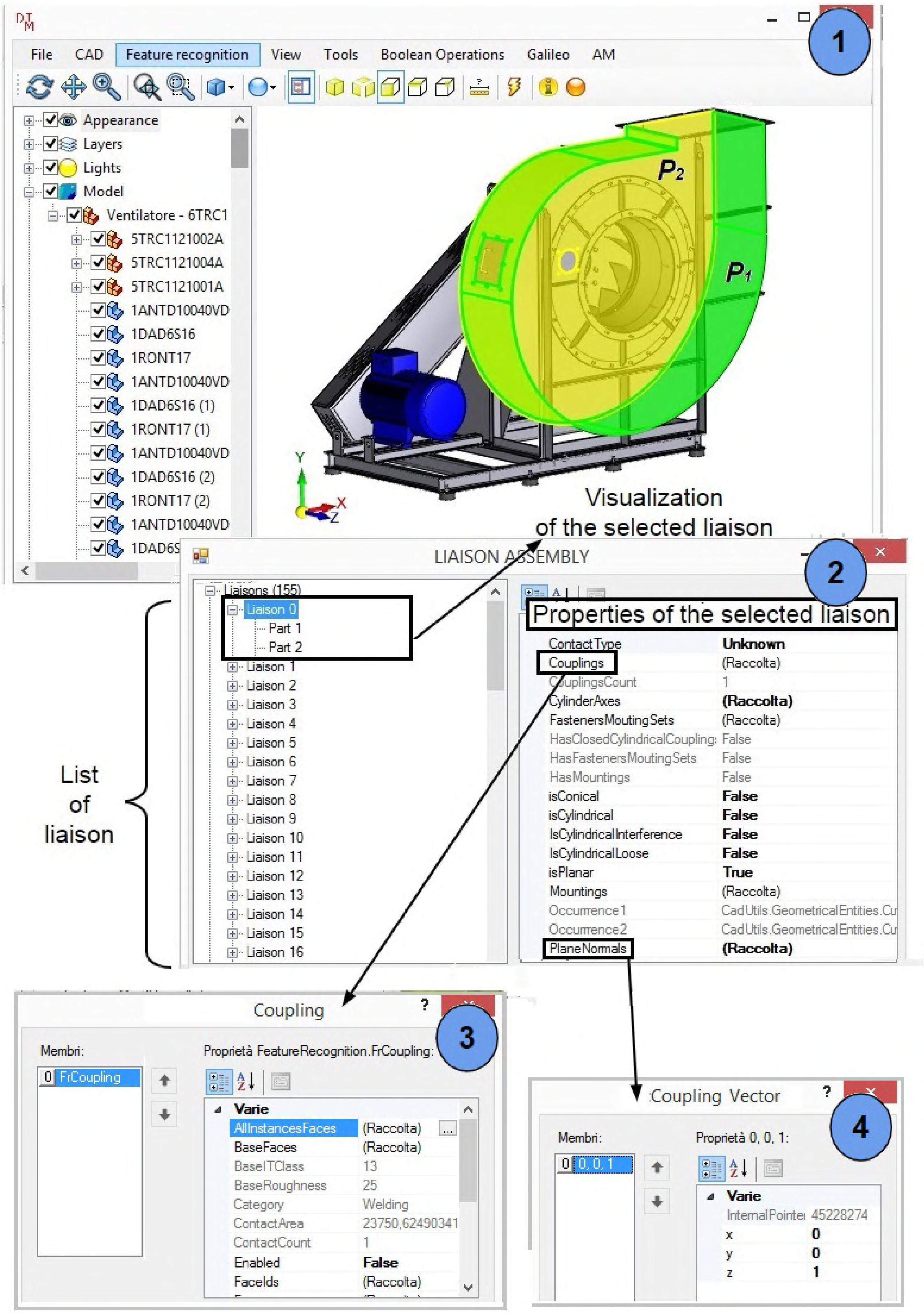


Figure 4.29: Interface developed for the visualization of the enriched CAD model representation for a fan assembly. Form 1 is the *Viewer*; form 2 is the *Liaison assembly*; forms 3 and 4 are those associated with the liaisons properties.

More in details, the enriched CAD model representation is presented as a new single-level tree, where each leaf is a liaison. By clicking on a liaison leaf, it is expanded and the composing parts and standard parts are listed below. In addition, the properties and the accessible data (e.g. couplings, mountings, contact information, etc.) are visualized on the right side of the same form and can be further selected to visualize their specifications. Also, the *Liaison assembly* form is linked with the *Viewer* in the sense that by clicking on an item of the liaisons tree (e.g. a liaison, a part, a standard part set, etc.) its corresponding components are highlighted in the *Viewer* form.

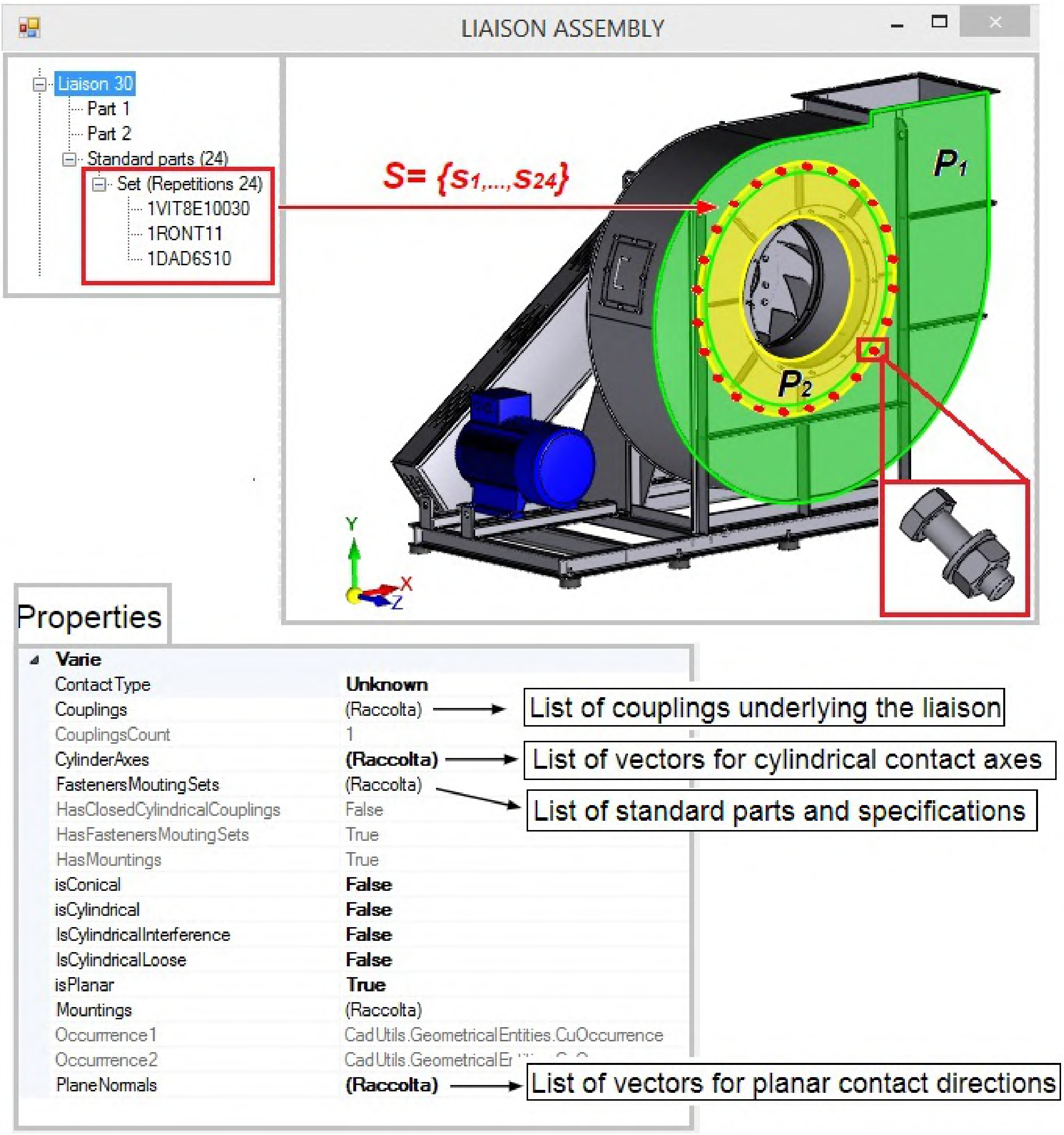


Figure 4.30: Example of visualization of a liaison of the fan assembly having repeated standard parts set.

Examples of visualization of the liaisons associated with the CAD model of a fan assembly are presented in Figures 4.29 and 4.30. In the specific, the input model is made of 325 parts, 228 of which are recognized as standard parts (see Section 4.2.4). The total number of liaisons created to completely describe the fan assembly is 155, which is a very small amount compared to the pairs of parts in contact that would be detected if considering all the 325 parts at the same level, regardless of the standard parts, and this is beneficial in the next modules. In the *Liaison assembly* interface the list of the 155 liaisons is visualized. In Figure 4.29 the first liaison is selected, which consists of the two parts P_1 and P_2, respectively colored green and yellow in the *Viewer*. From the forms it can be inferred that the liaison involves a single coupling, which is planar and the normal vector associated with the planar face of P_1 in contact with an opposite planar face of P_2 is parallel to the z axis. Figure 4.30, instead, shows the case where a liaison with standard parts is selected. More precisely, the liaison includes 24 copies of the same standard parts set consisting of screw, nut, and washer. To avoid redundant information, the liaison tree contains only a representative item indicating the standard parts sets composition, along with the number of its repetitions. All the sets are colored in the *Viewer* and their geometric attributes (e.g. centers and axes) are readable in the properties section.

In this way the CAD model of a complex assembly can be inspected through the browsing of its liaisons. The visualization of pairs of parts in contact, along with their contact information, is more significant in respect with the visualization of the single parts. Many details useful for the understanding of the components' relationship and their behavior within the assembly are in fact provided and highlighted. For instance, the aggregation of the standard parts in functional sets, as well as the grouping of the repetitions of the same standard part or standard part set emphasize the role of them in connecting two custom designed parts. Such a visualization, along with the possibility to interact with the liaisons' elements, is intuitive and enhance the CAD model analysis. The interface presented here will be exploited also in next modules to show their results.

4.4 Conclusions

This chapter discusses the developed methods for the definition of the new enriched representation of mechanical CAD models based on liaisons.

Considering that the final 3D model of a complex product is frequently achieved through an intensive design process possibly including third parties' components, it is assumed that the input B-rep is typically an heterogeneous structure in terms of shape details and missing explicit information on the technological data of the components and their relationships. As a consequence, the main challenge is in the development of tools that can automatically infer these type of data without the need for human intervention.

It follows that the reported work is not a simple extraction of the data, but rather an elaborate process of analysis and interpretation that aims to convert low-level data into high-level semantic knowledge by means of the definition of rules and requirements on the geometry and topology of the assembly. The rule-based approach has been chosen because of the considered engineering context, which is well structured, where mechanical parts have to satisfy well established laws and schemes. The major achievements obtained in this direction can be summarized in the subsequent points:

- association of geometric features with engineering seats for the placing of specific types of components;
- association of adjacency between surfaces with mechanical contacts;
- association of geometric shapes and dimensions with mechanical classes of parts;
- creation of a single data structure that collects all the information.

Such a comprehensive semantic analysis of a complex CAD model, which evaluates parts within the assembly, rather than singularly, and exploits the context, along with engineering knowledge and geometric processing, for sake of generality to deal with as much as possible design conventions, is innovative and very promising. However, the data extraction phase can be further improved under various aspects. On the one hand, the features recognition module can be extended to the detection of seats not fitting into standard features, but still effective in the industry. On the other hand, with regard to its scalability, the standard parts recognition module can be applied to cover additional standard element categories and subcategories by defining new rules and a corresponding new recognition function for each new class.

Nevertheless, the developed modules are sufficient to automatically extract and provide the crucial information for a reliable understanding of a mechanical CAD model, especially those information usually manually given by experts.

In particular, the modules are preparatory for facing with assembly tasks in a novel way to overcome some of the most restrictive weaknesses emphasized in the literature review. A large variety of operations can be addressed, and some example of application are provided in the next chapter, i.e. subassembly identification, assembly sequence planning, and design for assembly analysis.

5

Phase 2: Data Exploitation

How to manage an industrial CAD model of a mechanical assembly in B-rep format and the methodologies applied to extract all the data necessary for the computation of liaisons were discussed in the previous chapter.
The leverage of the semantically enriched representation of the CAD model to deal with some of the widespread and challenging assembly tasks is the focus of this chapter. In particular, innovative methodologies are studied and developed to address subassembly identification and assembly sequence planning problems relying on usually overlooked engineering meaningful information, along with the benefits of using automatic extraction of data in design for assembly approaches is investigated. Results and examples of application are shown after the discussion of the implementation structure for each task.

5.1 Subassembly identification module

The subassembly identification module aims at identifying meaningful subsets allowing to address assembly tasks in a innovative way relying on the just introduced list of liaisons.
The novelty stands in taking into account that the treated object is actually a real mechanical assembly, which respects precise engineering rules, and to leverage the semantics of components and their relations, rather than only geometrically evaluate the existence of a contact between two parts. As for the standard part recognition, for the cluster identification the guiding idea is to exploit engineering knowledge to assess and point out the primary design rules generally respected in mechanical assemblies and match them in the geometry and topology of the CAD models.

5.1.1 Introduction to clusters

To allow a correct understanding of the here proposed approach, some notions have to be introduced clarifying how they are intended in the described process.

Definition of cluster

First, in this section the concept of cluster is used, instead that of subassembly. In particular, a cluster is defined as follows:

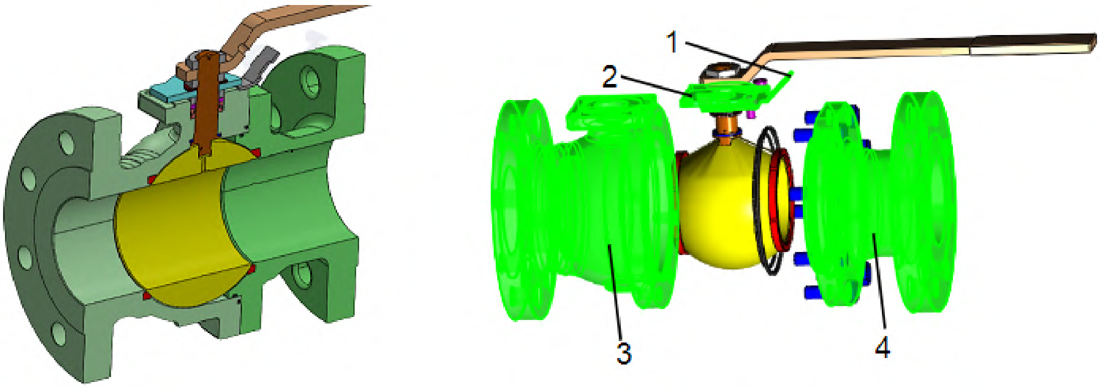
Cluster

Given a mechanical assembly, a connected set of parts which share some characteristics constitutes a cluster. The characteristics can regard common mounting techniques (e.g. screwing, welding, gluing, etc.), as well as shape features (e.g. a certain type of symmetry, etc.).

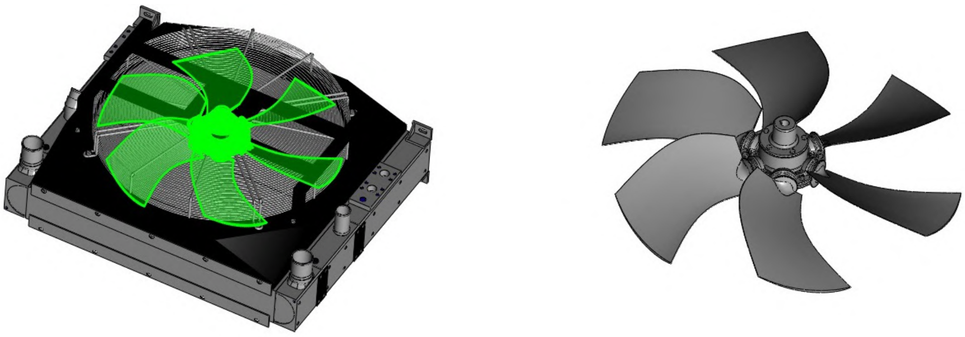
Although it may appear that the concepts of cluster and subassembly are overlapping, they are fundamentally different.

Namely, as defined in Section 1.1 a subassembly is here referred as a mechanical unit that can be mounted independently from the rest of the assembly and, once fabricated, is then inserted in the main product. On the contrary, a cluster does not have such strict mounting requirements, in the sense that it has not necessarily to be a single block to be inserted in the final assembly. Rather it requires that the composing parts meet some engineering meaningful properties.

For a better understanding, Figure 5.1 shows an example of a cluster identified in the CAD model of a ball valve and an example of subassembly included in the



(a) Section view and exploded view of a ball valve where the parts of the cluster are highlighted.



(b) Assembly of a radiator where the subassembly is highlighted and a detail of the subassembly.

Figure 5.1: CAD models of a (a) ball valve and a of a (b) radiator to show respectively examples of cluster and subassembly.

CAD model of a radiator. More precisely, in Figure 5.1a a set of four adjacent parts sharing the characteristic of being mounted by fasteners and representing the cover of the valve is highlighted. It is evident that this group of parts can not be referred as subassembly. In fact, it can not be separately mounted and then inserted in the ball valve assembly, since it has to contain the ball mechanism inside. In Figure 5.1b, instead, a complete rotor is highlighted as example of subassembly. It involves the combination of several components mounted with different techniques that can be assembled independently and constitute a stable unit.
Finally, a further consideration is that if a subassembly is made of parts sharing some characteristics, it is also a cluster, and in this case the two concepts are equivalent.

Types of clusters

The developed approach aims to detect different types of assembly clusters. Each type is defined according to the characteristics of the constituting parts. These are

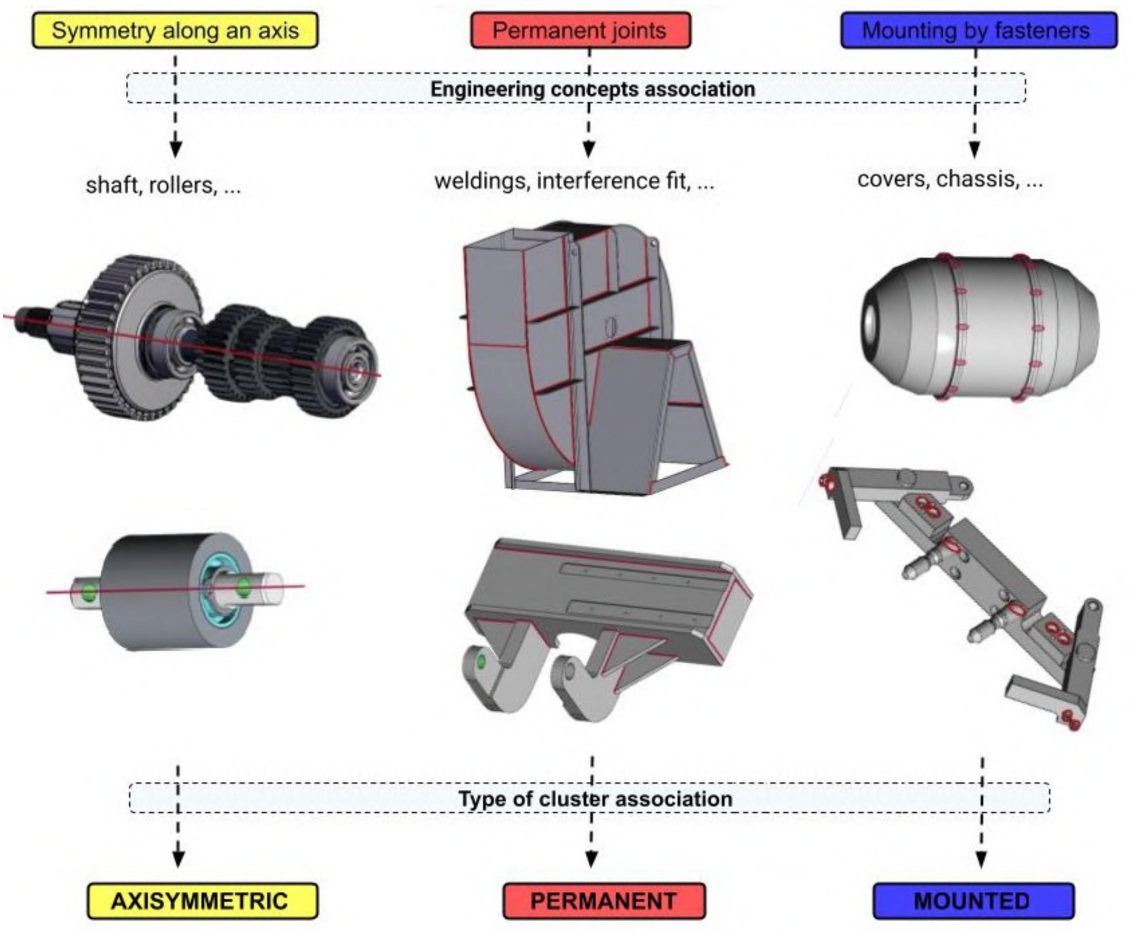


Figure 5.2: Examples of the three types of clusters and their interpretation through engineering concepts.

established upon heuristics, each one associated with a precise engineering concept corresponding to mechanical assembling processes.
Three general heuristics have been defined, which can be related to different types of mechanical assembling. These are the symmetry along an axis, the presence of permanent joints without the use of connecting elements, and the presence of fasteners and mountings (Fig. 5.2); they can be identified exploiting the parts' semantic and liaisons' data.

Symmetry along an axis A frequently encountered situation in mechanical assemblies is the aggregation of components along a common axis. It can be referred both to sets of parts mounted along a shaft and sets of concentric parts (e.g. crankshaft, roller, pulley, etc.). These components have a specific function inside the assembly, namely they transmit power or movement. Furthermore, from an engineering point of view, they have a distinctive structure and include characteristic

components. In fact, the groups of parts aggregated along an axis mostly consist of axisymmetric parts (as they are defined in Section 4.1.2), among which gears and bearings are particularly widespread, connected by clips and gaskets. An exception is then given by the keys, which although not being axisymmetric are typical of these groups (e.g. to block the shaft with a gear). Besides, the mounting technique used in these situations expects to thread by sliding or to interference fit hollow parts into the axis or into the central part.
It results evident that the identification of clusters of parts aggregated along a common axis would be very helpful in enhancing the semantic understanding of a CAD model. It can be exploited in several tasks, such as SI and ASP, because these clusters respect a precise assembly sequence and may be considered subassemblies themselves.

As a consequence one type of cluster we define in our approach is actually that of *axisymmetric clusters*. These clusters have indeed a specific engineering and semantic meaning and the features of the parts from which they are composed are easily accessible from the liaison graph already computed. Figure 5.2 shows examples of axisymmetric clusters.

Permanent joints - Absence of connection elements A further engineering concept, exploited in the clustering method, is the permanent joint, which includes welding, gluing and interference fit, as already discussed in Section 1.1.1. Those are assembling processes performed to stuck the parts together, defining a strong and irreversible relation between them. As a consequence, units jointed in a permanent way are stable and behave as a single independent object. From a structural point of view, the parts usually involved in a permanent connection process have geometrically simple shapes, e.g. sheet metal and plates. The contacts between each pair of parts, in most cases, concerns few planar faces, and above all no extra connection elements are involved. In an engineering prospective, welding, gluing and interference fit processes do not interfere with the assembly of the other components, but rather they are executed in a preliminary phase. In addition, one of the main functions of these groups in mechanical assemblies is to serve as basis on which the other subassemblies are then mounted. Therefore, it can be assessed that the knowledge of the assembly's permanent units can significantly contribute to the subassembly identification. For example, a suggested strategy in some SI works is even to collapse all the welded parts in a single part. However, the gap is in the recognition of welds, gluing and interference fit which is not automatic, but rather it requires human intervention. This because, most of the time, the information are not represented at all in CAD models, neither as annotations nor as solid modeled beads.

Among the different types of cluster the *permanent clusters* are thus defined, see Figure 5.2. The identification of these clusters is algorithmically performed starting from the liaisons graph and it especially relays on contacts features.

Mounting by fasteners The third basic aspect of primary importance is the mounting by fasteners. The presence of fasteners, such as screws, bolts, studs and pins, connecting two or more assembly's parts is a very meaningful feature which embodies a stable but non-permanent joint between the parts (see Section 1.1.1). On the one hand mounting is used to connect different subassemblies, on the other a set of parts connected by fasteners can constitute an useful cluster. Referring to the second situation, the groups of mounted parts, for instance, may be associated to the external cover of an assembly or its chassis. Moreover, as far as the structure is concerned, the components which can be found in mounted groups do not have to meet many requirements of shape, they just need to be drilled in order to allow the placement of the connectors. Thus the recognition of mounted groups would be substantial in SI processes.

Hence, due to the massive use of fasteners in mechanical engineering and their relevance in the assemblies, the last type of cluster considered is that of *mounted clusters*. To identify them in CAD assembly models the parts classification developed and the liaisons' data are sufficient. Examples of mounted clusters are shown in Figure 5.2.

5.1.2 The clustering algorithm

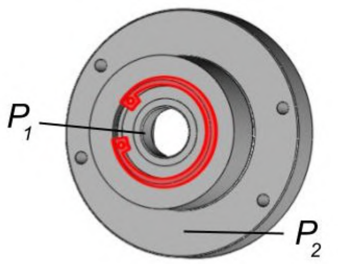
The implemented heuristic method aims to divide a CAD assembly model into disjoint clusters of parts, accordingly to the three types of cluster described in section 5.1.1.

The algorithm consists of three steps: selection of the parts which satisfy the requirements for each type of cluster, cluster evaluation, and cluster refinement. Notice that both parts' selection and cluster computation are based on liaisons analysis, however they have been defined as two separate phases. This choice is justified as the implemented algorithm is integrated in the industrial software and existing libraries are exploited for computational simplicity.

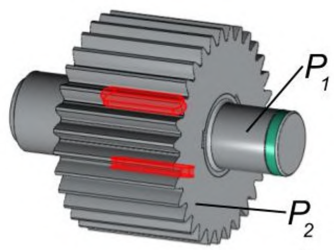
Parts' selection

Starting from the new representation of the CAD model that is based on the liaisons list, and having available all the data stored in each liaison, the first step of the clustering approach is to select the parts that meet the conditions established for each type of cluster. To automate the selection, the engineering requirements are described in terms of the extracted information, i.e. parts' geometric characteristics and their contact features, and a series of rules are defined to evaluate the liaison's parts membership to each type of cluster.
In the following a brief description of the rules for each type of cluster is reported.

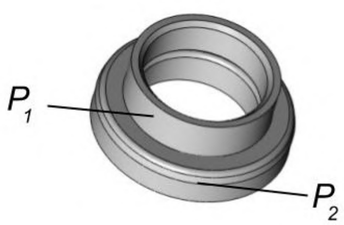
- **Rules for axisymmetric clusters ($R_{Axisymmetric}$).** Liaison must be between parts classified as axisymmetric according to their shape type (see Section 4.1.2). The only standard part categories accepted are the axisymmetric ones, such as circlips or O-rings, and keys, and their presence is not mandatory (Fig. 5.3a, 5.3b). Threaded connectors are excluded, otherwise it falls into the case of mounted clusters. For the same reason, at most one mounting can exist between the two parts (Fig. 5.3c), as the presence of a pattern of holes suggests a mounting operation. Finally, the cylindrical contacts must be between closed cylindrical surfaces (Fig. 5.3d).



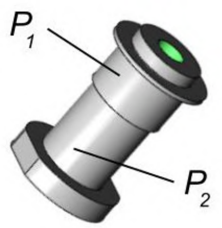
(a) Axisymmetric parts with closed cylindrical contact and a circlip.



(b) Axisymmetric parts with closed cylindrical contact and keys.



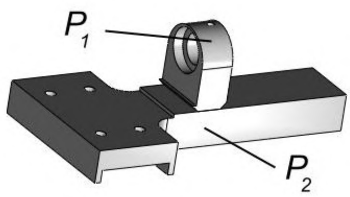
(c) Axisymmetric parts with a mounting.



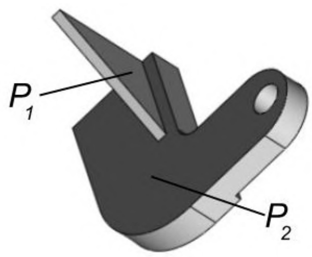
(d) Axisymmetric parts with closed cylindrical contact.

Figure 5.3: Examples of liaisons that meet $R_{Axisymmetric}$.

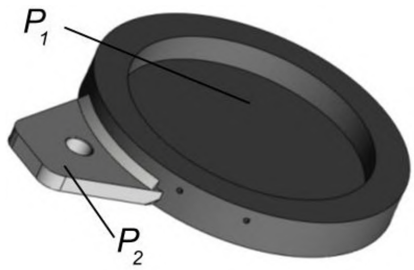
- **Rules for permanent clusters ($R_{Permanent}$).** At most only one part of the liaison can be axisymmetric. Fasteners and mountings are not admitted since they are strongly connected with mounting features (Fig. 5.4a, 5.4b). If one of the two parts is axisymmetric, an additional condition is imposed for contacts, i.e. the cylindrical contacts, if any, do not have to be closed (Fig. 5.4c). A closed cylindrical contact is attributable to mounting by sliding, and thus typical of axisymmetric clusters.



(a) Not axisymmetric parts with planar contact.



(b) Not axisymmetric parts with planar contact.

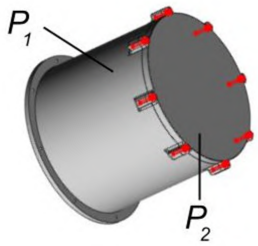


(c) Axisymmetric and not parts with not closed cylindrical contact.

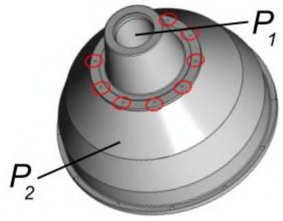
Figure 5.4: Examples of liaisons that meet $R_{Permanent}$.

- **Rules for mounted clusters ($R_{Mounted}$).** No restriction on the part type is imposed. If both the parts are axisymmetric, it has to be considered the situation excluded in the axisymmetric clusters analysis; that is to say, only threaded fasteners are accepted (Fig. 5.5a) and at least two mountings must be recognized (Fig. 5.5b). In the case only one part is axisymmetric, all the type of fasteners are accepted and at least a mounting must be recognized (Fig. 5.5c). Finally, if none of the parts are axisymmetric, at least one mounting must be recognized and all the types of fasteners are accepted (Fig. 5.5d).

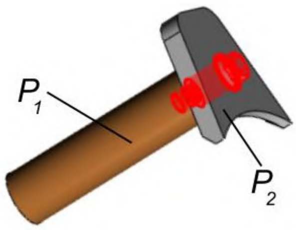
For a better understanding, the rules are summarized in Table 5.1. Namely, the columns report the properties of a liaison that are considered in the parts selection process, organized in the order they are evaluated. Each row, then, stands for a



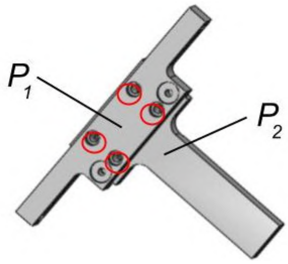
(a) Asymmetric parts with screws.



(b) Asymmetric parts with mountings.



(c) Axisymmetric and not parts with a screw.



(d) Not axisymmetric parts with mountings.

Figure 5.5: Examples of liaisons that meet $R_{Mounted}$.

specific rule for the corresponding type of cluster.

The selection algorithm considers each liaison $l(P_1, P_2, C, M, S)$ of the liaison list and evaluates, through conditional statements, the properties of P_1 and P_2, the categories of the standard parts contained in S, the cardinality of the set of mountings M, as well as the properties of the couplings in C. Notice that the first discriminant factor is the shape type of the two involved parts.
The cases in which one, both ore none of the components in contact are axisymmetric are distinguished. Then, the standard parts and the mountings sets are investigated. The presence of certain categories of fasteners or locating elements is decisive to distinguish between a cluster type rather than an other. Also, the checking of the number of mountings, together with the previous information, is important. For instance, it allows to identify axisymmetric parts mounted by fasteners when fasteners are not included in the CAD model due to model simplifications (see Section 1.3.1). Once the analyzed liaison meets all the requirements for a given type of cluster, the parts P_1 and P_2 are both added to the list of parts candidate for that cluster.
At the end of the selection process, the output consists of three lists of parts, namely $P_{Axisymmetric}$, $P_{Permanent}$, and $P_{Mounted}$, one for each type of cluster.

		Properties of a liaison $l(P_1, P_2, C, M, S)$				
		P_1 is axisymmetric	**P_2 is axisymmetric**	**Accepted categories of standard parts in S**	$\#M$	**Restriction on couplings in C**
Rules for	**Axisymmetric cluster** ($R_{Axisymmetric}$)	✓	✓	circlips, O-rings, keys	0 or 1	Cylindrical couplings: closed
	Permanent clusters ($R_{Permanent}$)	✓	×	-	0	Cylindrical coupling: not closed
		×	×	-	0	-
	Mounted cluster ($R_{Mounted}$)	✓	✓	screws, nuts, washers, studs, pins	>1	-
		✓	×	screws, nuts, washers, studs, pins	>0	-
		×	×	screws, nuts, washers, studs, pins	>0	-

Table 5.1: Table of the rules defined to evaluate liaison's parts membership to each type of cluster.

Clusters evaluation

Once the selection process is completed, the parts included in each list can generate a single cluster or multiple distinct clusters. In fact, according to the selection algorithm, each component is in contact with at least another component of the list, but it can not be ensured that all the components form a unique group of connected parts. Moreover, lists are not necessarily disjoint because a part can indeed belong to multiple lists. This is the case of a component present in at least two liaisons, and each liaison satisfies the rules for different types of cluster.

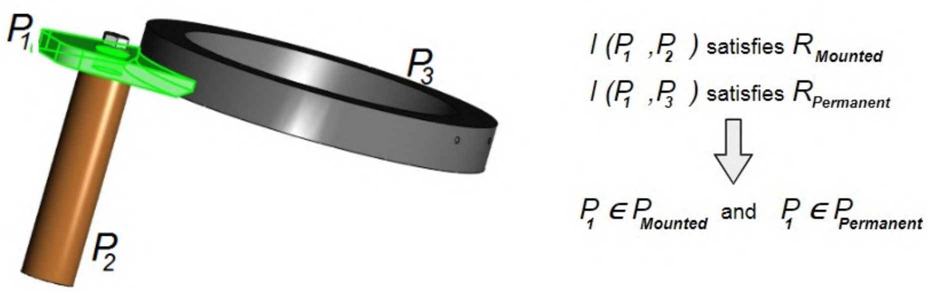


Figure 5.6: Example of part belonging to two liaisons that satisfy different rules, and thus assigned to two different lists.

An example is shown in Figure 5.6, where the highlighted part P_1 is in contact with both P_2, the pin, and P_3, the circular basis, thus two liaisons exist including this component. That are $l(P_1, P_2)$, characterized by a linking screw and hence satisfying the rules $R_{Mounted}$, and $l(P_1, P_3)$, which instead meets the rules $R_{Permanent}$ since it has neither standard parts nor mountings, but only an open cylindrical coupling. As a consequence, P_1 is simultaneously assigned to the lists $P_{Mounted}$ and $P_{Permanent}$.

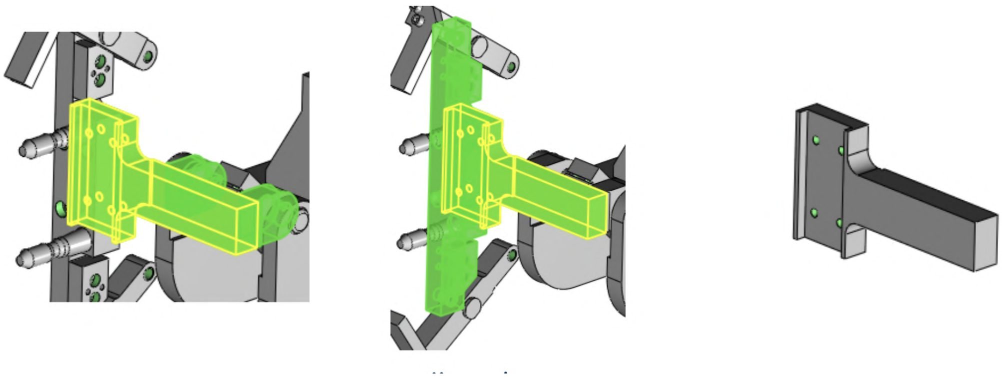
To address the first issue, and therefore separate the parts contained in the three lists into disjoint groups, a further processing step is required to aggregate parts by contact in the $P_{Axisymmetric}$, $P_{Permanent}$, and $P_{Mounted}$ lists, thus providing connected sets of parts, i.e. the clusters. In practice, this implies that if two parts P_i and P_j of the same list are in contact, i.e. a liaison between them is found, they are assigned to the same cluster and three different scenarios are envisaged. Respectively, if none of the already defined clusters contains neither P_i nor P_j, a new cluster is created containing both of them. If instead a cluster already exists that includes P_i or P_j, the missing part is assigned to the cluster. Finally, in case there are two different clusters containing the two parts, those clusters are merged into one.

At the end of the aggregation process, the parts of the lists $P_{Axisymmetric}$, $P_{Permanent}$, and $P_{Mounted}$ are organized into clusters, then all collected in the $CLUSTERS$ list. Precisely, clusters are disjoint when evaluated by type, but, according to the previous observation, one or more components may be shared between clusters of different types, because the problem of the intersections between the three lists has not yet been addressed (Fig. 5.7). Therefore, to provide disjoint clusters a final refinement step is needed.

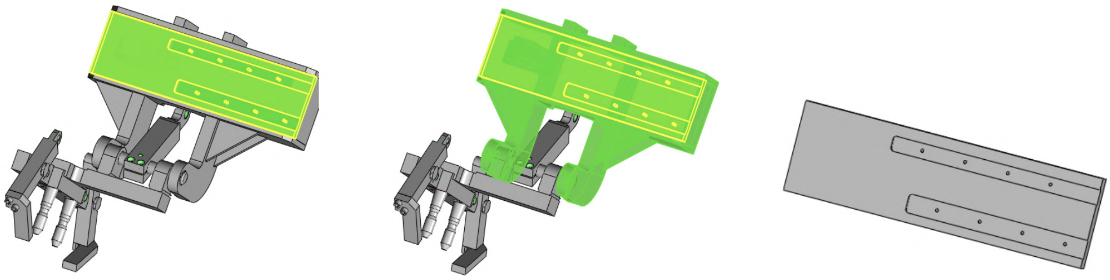
Clusters refinement

The cluster refinement phase evaluates the intersections between the generated clusters and updates the collection $CLUSTERS$. In particular, it detects the parts included in more than one cluster, defined as $Shared := \{P_1, \ldots, P_s\}$, and establishes their membership to only one cluster.

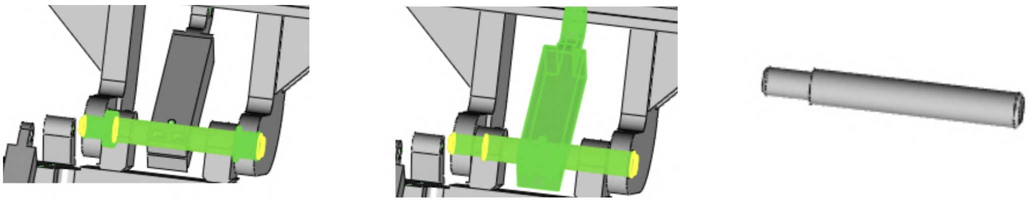
The difficulty lies in understanding which type of cluster, and therefore which engineering concept, to give precedence to. Rules to algorithmically make the assignment of the shared components are specified based on engineering considerations and knowledge of assembling processes. To this end, it can be assessed that axisymmetric clusters, in general, are correctly grouped, due to the distinctive requirements parts have to satisfy. Therefore, if a component is shared between an axisymmetric cluster and another type of cluster it would be assigned to the first, thus giving more weight to axisymmetric characterizations. The precedence selection between mounted and permanent clusters is instead more cumbersome. In fact, on the one



(a) Permanent cluster (b) Mounted cluster (c) Part in common



(d) Mounted cluster (e) Permanent cluster (f) Part in common



(g) Axisymmetric cluster (h) Permanent cluster (i) Part in common

Figure 5.7: Examples of clusters of different type sharing a common part.

hand it was mentioned in Section 5.1.1 that permanent joints are usually performed before the mounting operations. As a consequence, it would be immediate to affirm that if a component is included both in a mounted and a permanent cluster, it is reasonable to assign it to the permanent one. On the other hand, permanent clusters as they are generated, can include a wide variety of parts that share a simple coupling, while mounted clusters are obtained through more strict rules. Thus their detection is more reliable and the assignation to mounted clusters would be preferable. According to these premises, some tests have been performed on different type of assemblies, both giving priority to permanent clusters and mounted clusters. It has been evaluated that the correctness of one sort rather than the other depends mainly on the class of product. In particular, products characterized by an external cover (e.g. gearboxes, valves, etc.) are correctly clustered by assigning parts shared between permanent and mounted clusters to the latter. As for the other types of products, the precedence to permanent clusters is reliable.

Algorithm 2 Clusters refinement

Data: $Shared := \{P_1, \ldots, P_s\}$ shared parts; $s \geq 0$; $CLUSTERS$ clusters;
Result: $CLUSTERS$ list of disjoint clusters
if s=*0* **then**
  return $CLUSTERS$
end
while $s \neq 0$ **do**
  Assign P_1 to one of the clusters containing it;
  Remove P_1 from the other clusters;
  Update $CLUSTERS$;
  Update $Shared$;
end

As schematized in Algorithm 2, the refinement process is carried out as a recursive operation, that evaluates the components P_i contained in the list *Shared* one at time until the list is empty. After that P_i is assigned to a single cluster according to the rules discussed above, P_i is removed from the other clusters. These are updated and recomputed, since the removal of a component may cut a cluster into two separate sets. Finally, the list *Shared* is also updated and the refinement is repeated considering the remaining parts in common with multiple clusters.

A list of disjoint clusters, each associated to one of the three defined types, i.e. axisymmetric, permanent, and mounted, is returned as final output.
It is to emphasize that, in general, the matching between the list of clusters before the refinement and that returned as output is a quite strong hint to define an assembly order among the clusters. The parts in common with multiple clusters, indeed, imply a connection between the clusters. When assigning the component to only one cluster, it is indirectly affirmed that the cluster will be assembled first and then mounted on the other thanks to the part they share.
Furthermore, it must be noted that some clusters may only include a single component, and thus they are not actual cluster if intended as set of parts. This scenario is not an error, rather it is a direct consequence of the refinement phase. In fact, once a part in common is assigned to a cluster, it is also removed from the others that shared it. The removal can disconnect the clusters into smaller ones, and in the most trivial case they result in a single element. However, in the perspective of SI and ASP, it is useful to maintain cluster labeling even when dealing with single parts. This because the classification into a specific type provides information on assembly techniques.

5.1.3 Cluster detection assessment

The assessment of the clustering algorithm is carried out on the same collection of CAD models of mechanical assemblies employed in the standard parts recognition assessment (see Section 4.2.4). It is a good dataset to test the method since its heterogeneity in model classes and the presence of different types of clusters.
It is to underline that, even for cluster detection, the creation of a ground truth is one of the main issues for the same reasons illustrated in Section 4.2.4. Namely, CAD models do not include reliable information on clusters or subassemblies that can be exploited to validate the results automatically. As a consequence, the validity of the outputs returned by the algorithm is checked with the help of engineers. In addition, it was argued that the subassemblies provided in the tree structure of the CAD model are not always significant and hence can not be taken for granted (see Section 1.2). However, when they are provided and their validity is confirmed by experts, these can be exploited as data to evaluate clusters and matching the results.

Results

In this section, the application of the algorithm on some of the tested CAD models is discussed. The clusters obtained before and after the refinement phase are shown, as well as the application of the refinement phase is evaluated both giving precedence to permanent clusters over mounted ones and vice versa.

Gripper mechanism. The first model is an industrial CAD assembly of a gripper mechanism. It consists of 48 parts, 6 of which are classified as fasteners, but many not modeled fasteners are inferable from the mountings analysis too.

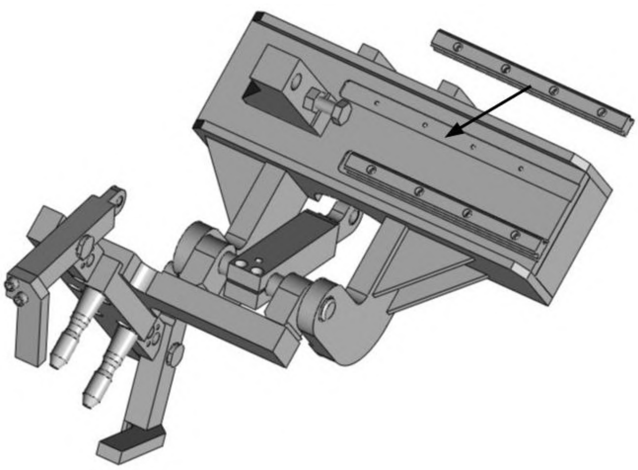
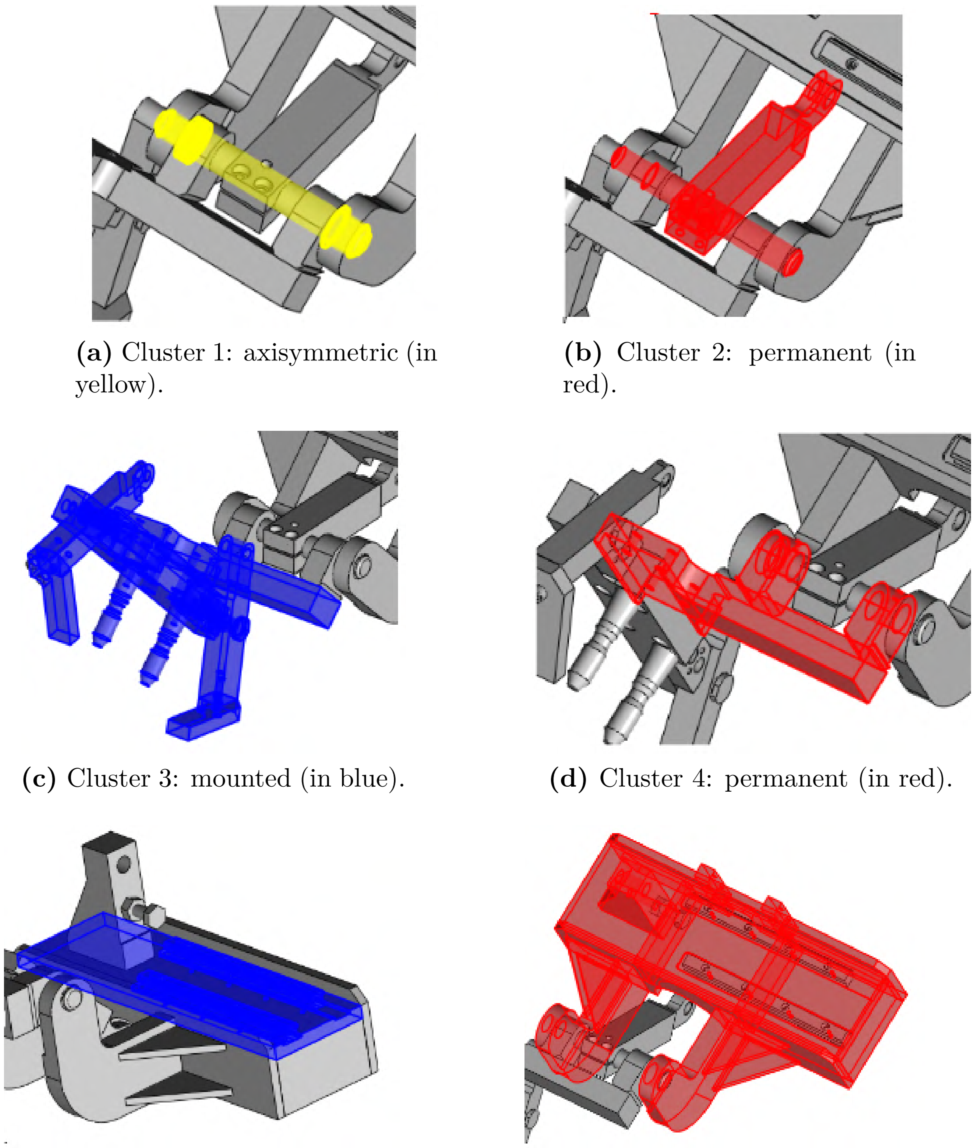


Figure 5.8: CAD model of a gripper mechanism.

An example of missing fasteners is visible in Figure 5.8, where the two rails and the sheet metal on which they are mounted show respectively four empty mountings not filled with the screws.



(a) Cluster 1: axisymmetric (in yellow).

(b) Cluster 2: permanent (in red).

(c) Cluster 3: mounted (in blue).

(d) Cluster 4: permanent (in red).

(e) Cluster 5: permanent (in red).

(f) Cluster 6: permanent (in red).

Figure 5.9: Clusters detected in the gripper mechanism.

In Fig. 5.9 the outcome after the clustering phase is shown. The detected clusters are six, more precisely: 1 is axisymmetric (Fig. 5.9a); 2, 4 and 6 are permanent (Fig. 5.9b, 5.9d, 5.9f); 3 and 5 are mounted (Fig. 5.9c, 5.9e). Three parts in common with multiple clusters are then identified. These are shown in Figure 5.10 highlighted with a color derived from the overlapping of the colors of the belonging clusters. A part is in common with the axisymmetric cluster 1 and the permanent cluster 2 (in orange), while the other two are in common with mounted and permanent clusters (in violet), respectively clusters 3 and 4, 5 and 6.

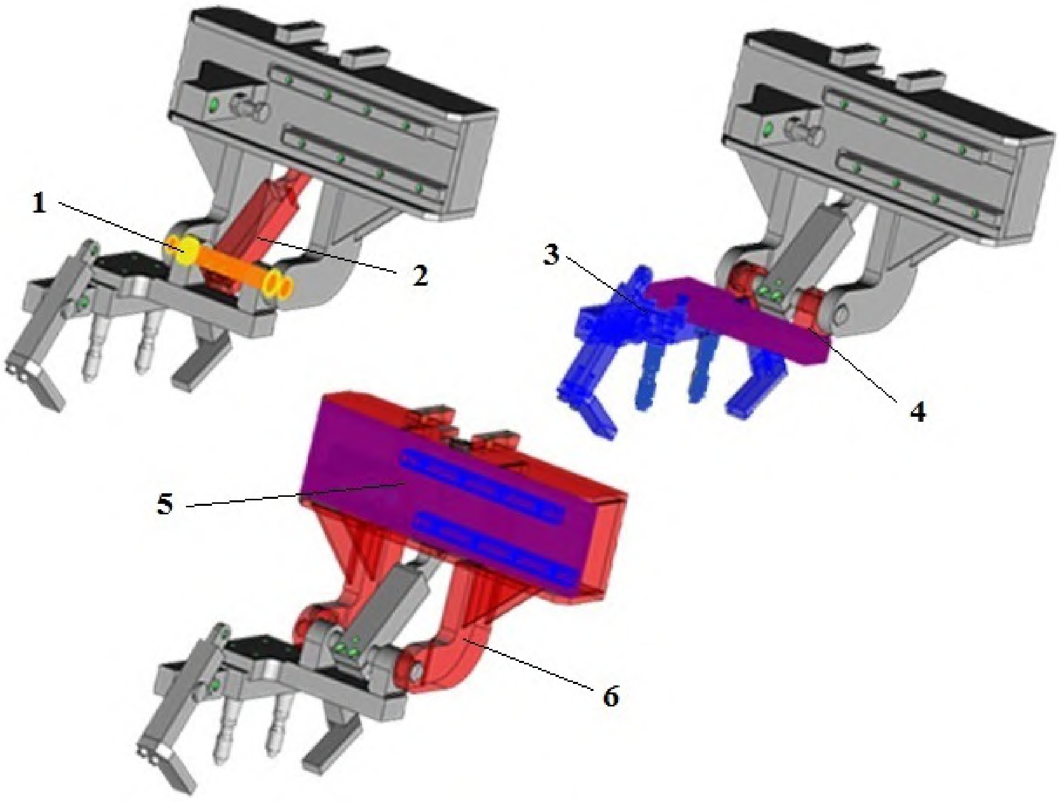


Figure 5.10: Shared parts between the clusters of the gripper mechanism. A part is shared between axisymmetric and permanent clusters (in orange), two parts are shared between mounted and permanent clusters (in violet).

During the refinement phase the shared parts are assigned to a single cluster. For the assignation, axisymmetric clusters have always the priority on the others. Thus, as for the first shared part it is assigned to the axisymmetric cluster 1 and removed from the permanent cluster 2. On the contrary, for the other shared parts, both the refinement phase giving priority to permanent and that giving priority to mounted clusters has been tested. The precedence to permanent clusters results optimal for this type of product. In fact, in this case, assigning the second and third shared parts to the permanent clusters 4 and 6 returns significant clusters, which also correspond with the subassemblies provided in the tree structure of the CAD model. Figure 5.11 shows the disjoint clusters after the refinement phase giving precedence to permanent clusters on that mounted, revealing, for example, that cluster 5 is splitted in two separate clusters after its update.

Additional useful information, inferred from the refinement phase, about the clusters assembly precedence are displayed in figure: for instance, cluster 3 will be mounted by fasteners on the welded cluster 4, since they shared a part, as well as single-parts clusters 5a and 5b will be mounted on cluster 6; cluster 2, instead, will

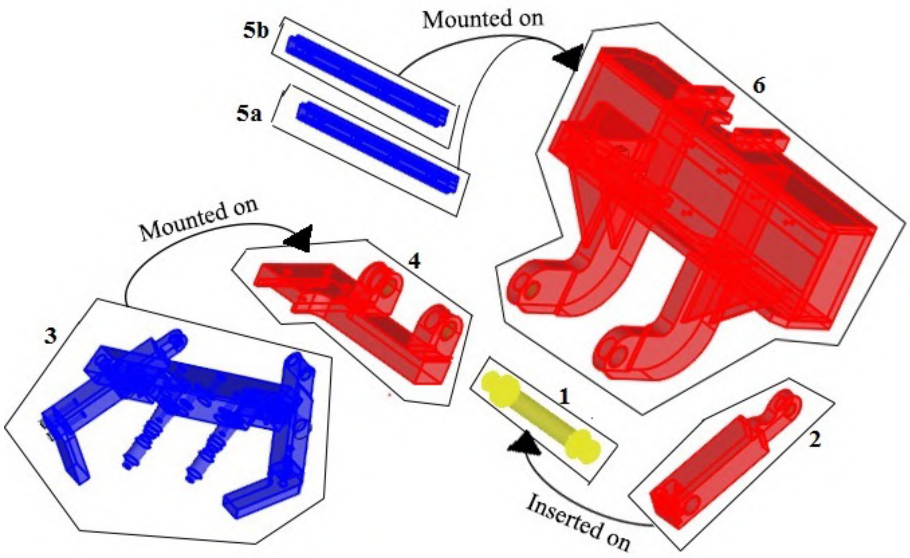


Figure 5.11: The gripper mechanism's disjoint clusters after the refinement phase giving precedence to permanent clusters.

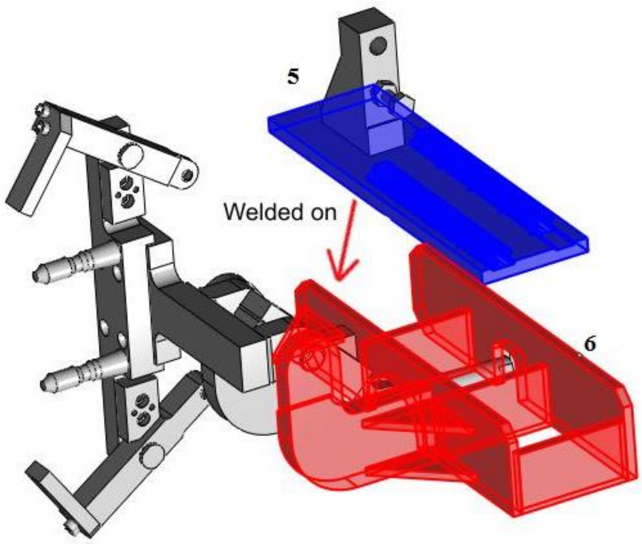


Figure 5.12: Example of gripper mechanism's disjoint clusters after the refinement phase giving precedence to mounted clusters.

be inserted on the axisymmetric cluster 1.
As a counterexample, in Figure 5.12 are shown the disjoint clusters instead obtained assigning the part shared between clusters 5 and 6 to the mounted cluster. It is evident that the resulting clusters are less reliable and feasible in respect with the previous, in fact cluster 6 is incomplete without the upper sheet metal, while the two rails can be realistically mounted after on it.

Rotor wind turbine The clustering of the model of a rotor wind turbine is provided (Fig. 5.13). It is not an industrial CAD model, but it is a conceptual model relevant in the cluster detection algorithm due to its characteristics. It includes an external cover mounted by fasteners containing different crankshafts and simplified bearings. The model is made of 46 parts, none of which are standard, since all the fasteners are omitted and simply represented by the presence of mountings.

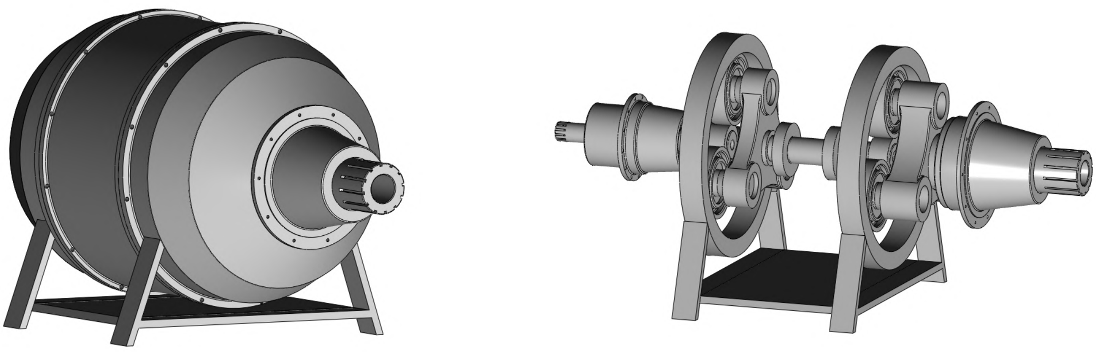
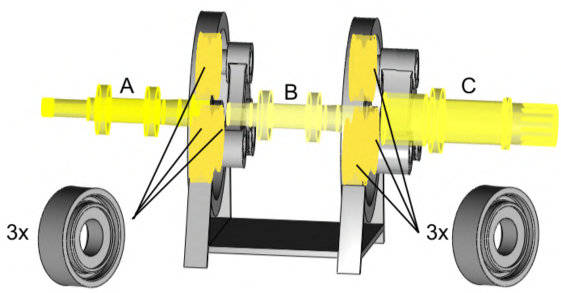
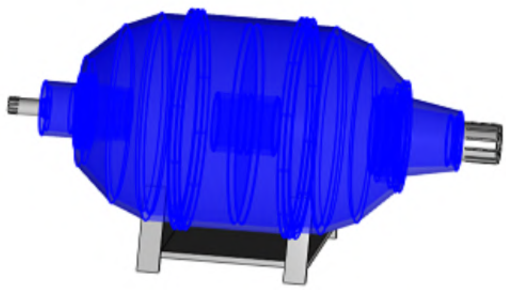


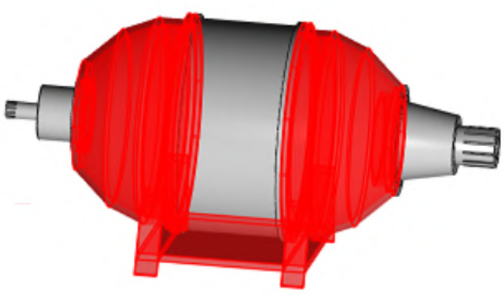
Figure 5.13: CAD model of a rotor wind turbine.



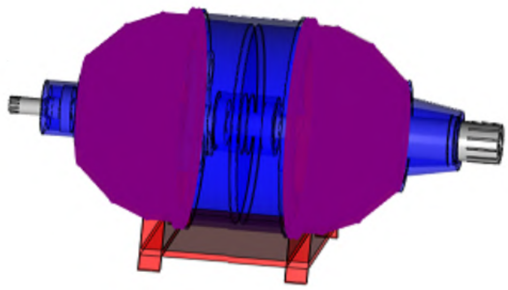
(a) 9 axisymmetric clusters (in yellow).



(b) Mounted cluster (in blue).



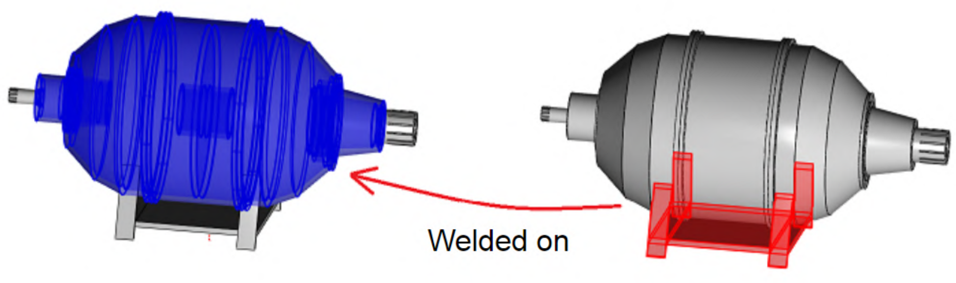
(c) Permanent cluster (in red).



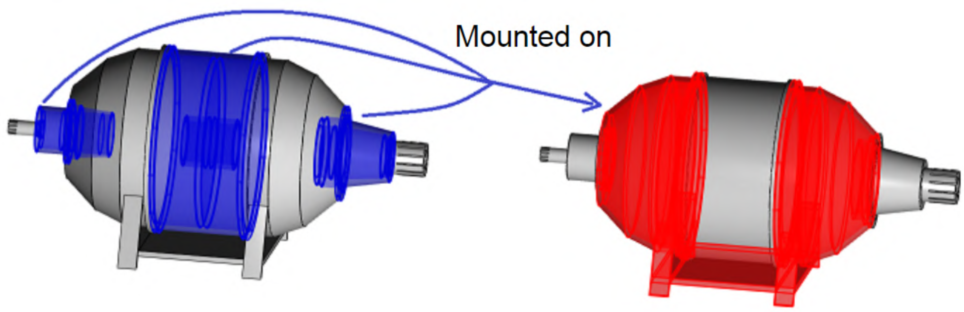
(d) Shared parts (in violet).

Figure 5.14: Rotor wind turbine after the clustering phase.

The results returned after the clustering phase are shown in Figure 5.14. In details, 9 axisymmetric clusters are detected, which do not share parts with any other cluster (Fig. 5.14a). These are 3 different crankshafts (A, B, and C) and 6 bearings that due to their simplified aspect appear as groups of concentric parts. Then, a mounted (Fig. 5.14b) and a permanent (Fig. 5.14c) cluster are returned, which share two parts (Fig. 5.14d), thus the refinement phase is required.



(a) Assignation of shared parts giving precedence to mounted clusters.



(b) Assignation of shared parts giving precedence to permanent clusters.

Figure 5.15: Disjoint mounted (in blue) and permanent (in red) clusters of the rotor wind turbine after the refinement phase.

Both the refinement phase giving priority to mounted and that giving priority to permanent clusters has been tested on the rotor wind turbine model. Figure 5.15 shows the two different outcomes obtained, confirming the fact that for assemblies having an external cover the assignation of parts shared between mounted and permanent clusters to mounted cluster is optimal. Indeed, in this way the entire cover is returned as single cluster (Fig. 5.15a), while working with the other order would return not meaningful clusters (Fig. 5.15b).

Gearbox The example of a gearbox is now reported. It consists of 426 parts, a larger number than previous models, 284 of which are recognized as standard parts, as it was demonstrated in Section 4.2.4 (see Table 4.6). As shown in Figure 5.16, the gearbox includes an external shell containing four main shafts with gears, bearings and other components mounted on them.

The clustering first step returns 5 axisymmetric clusters, 6 mounted clusters,

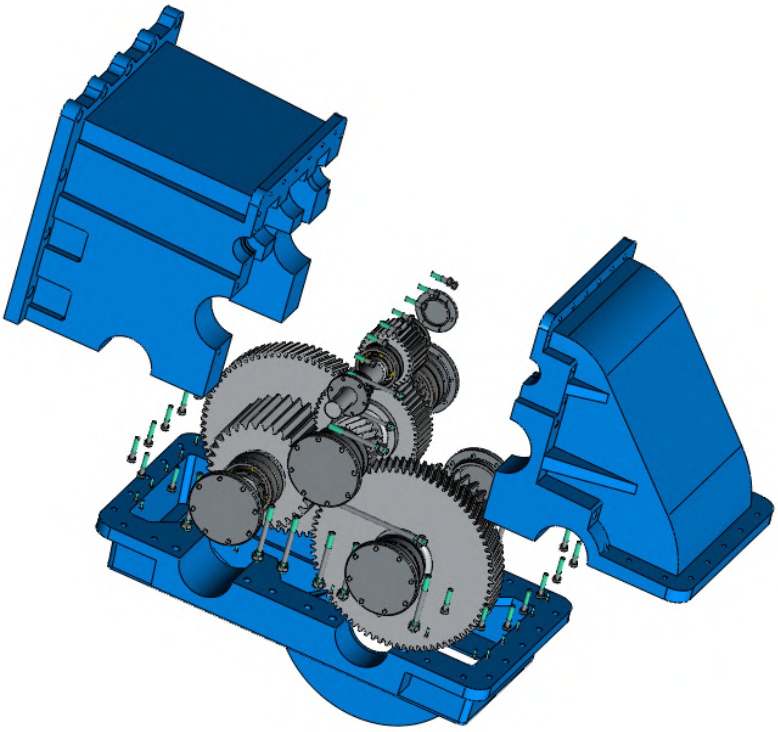


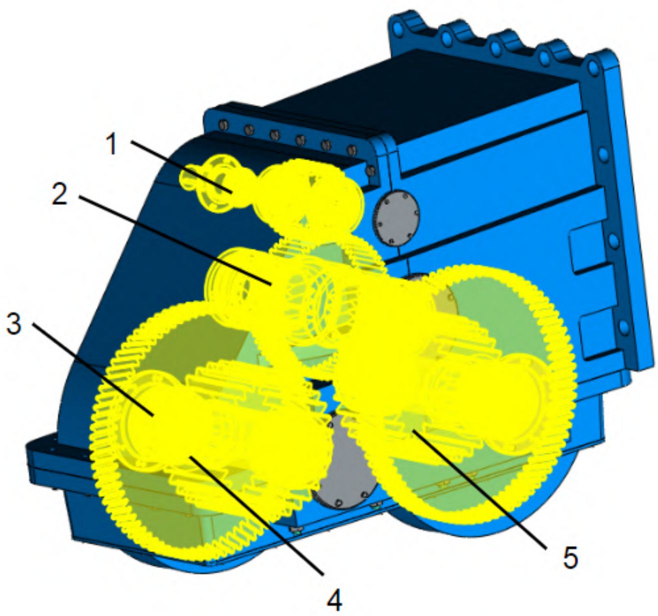
Figure 5.16: Exploded view of the CAD model of a gearbox.

and 4 permanent clusters. In particular, the asymmetric clusters are 5 (Fig. 5.17), instead of the 4 expected considering the presence of the 4 shafts and the parts aggregated. This because in the input model a bearing is not in contact with the corresponding shaft, and thus two separate clusters are returned, namely the clusters 3 and 4 shown in the Figures 5.17d and 5.17e.

The mounted clusters, instead, involve a group of 4 parts that corresponds with the external cover of the gearbox (Fig. 5.18) and 5 repetitions of a threaded locking nuts and a small element connected by a screw.
As far the permanent clusters, a group of 41 parts is detected, that covers most of the model's custom designed parts, along with the circlips not recognized as standard parts (see Table 4.6) that are thus not treated as locating components. Also other three sets of two parts given by a gear and an adjacent part are included in the permanent clusters.

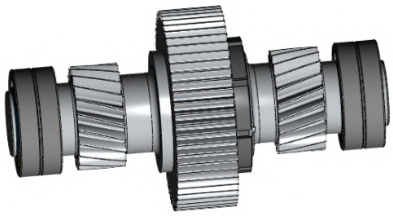
It is evident that several parts would be shared between more clusters, especially between the bigger permanent cluster and the clusters of the other types. Namely, 28 shared parts are found, some examples of which are shown in Figure 5.19. This scenario underlines the need for the refinement phase.

Both the refinement phase giving priority to permanent and to mounted clusters,



(a) Highlight of the axisymmetric clusters in the gearbox model.

(b) Cluster 1.



(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

(f) Cluster 5.

Figure 5.17: Axisymmetric clusters detected in the gearbox.

after having confirmed the axisymmetric ones, are tested. From the result, it appears that the order axisymmetric-mounted-permanent for the assignation of shared parts is the optimal for this type of product, as assumed in Section 5.1.2. In fact, the axisymmetric and the mounted are returned not changed, and thus they remain

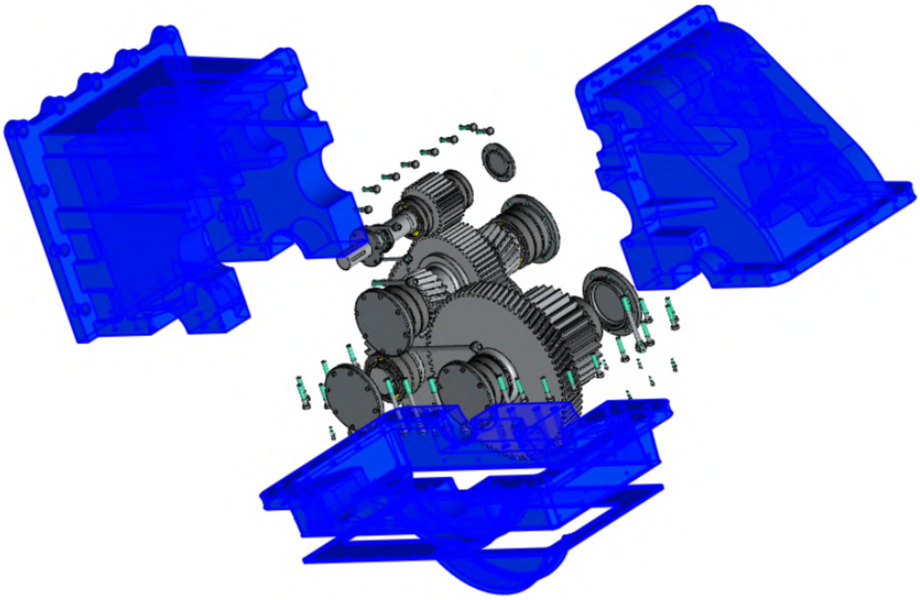


Figure 5.18: The mounted cluster corresponding with the external cover of the gearbox.

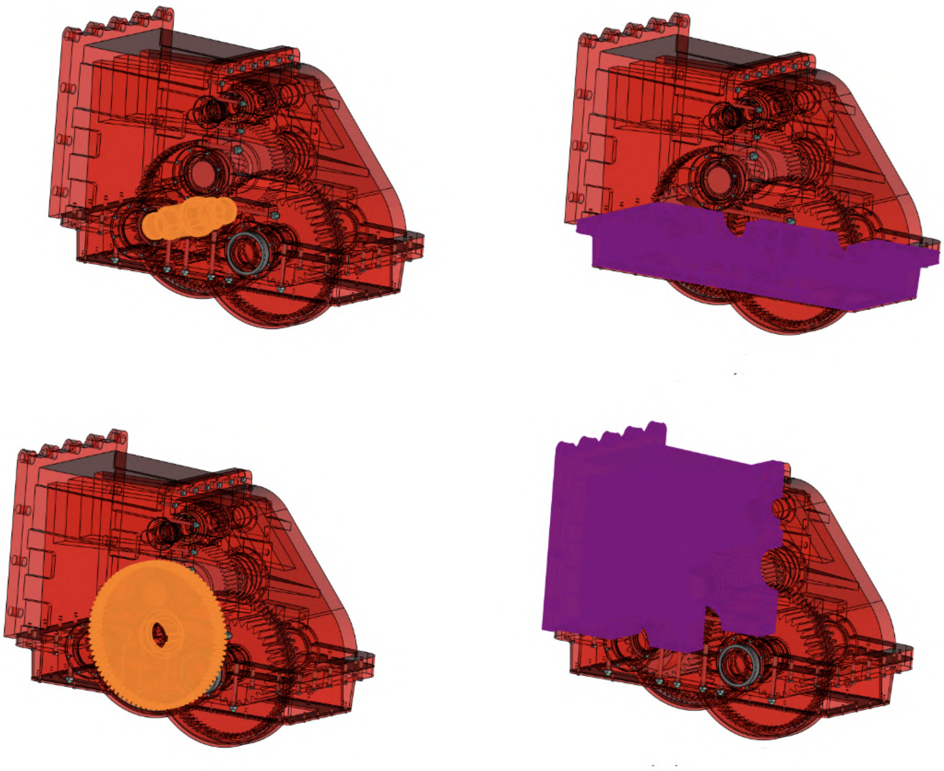


Figure 5.19: Examples of parts shared between axisymmetric and permanent clusters (in orange) and between mounted and permanent clusters (in violet).

reliable. On the contrary, the list of permanent clusters is modified, namely it is returned empty. In details those clusters of two parts are no longer returned since gears are correctly assigned to axisymmetric clusters, while the group of 41 parts, that was not meaningful nor realistic, is broken down and all the components are assigned to axisymmetric or mounted clusters.

In this particular case, the tree structure of the CAD model of the gearbox is organized in subassemblies that overall reflect the outcomes of the cluster detection. This information can be exploited to further confirm the validity and usability of the clusters identified from an industrial standpoint.

General discussion of results

The main characteristics taken into account for the test models are the number of parts, which varies in the range from 35 to 426, the number of standard parts and the presence of modeled or not modeled fasteners. The number of parts does not affect the quality of the clustering, but it only increases the computation time in the data extraction phase.
The more standard parts are modeled and correctly recognized, instead, the more results are reliable. This because not recognized fasteners would be treated as custom designed parts and this can cause misleading interpretation of the contacts, especially wrong permanent joints detection, and consequently false permanent clusters are returned. Absence of fasteners together with modeling errors (e.g. holes misalignment) generate similar issues, since mountings are not recognized and thus the liaisons satisfy requirements for permanent clusters.

In general, it can be assessed that the resulting groups are promising and, even if more than half of the total custom designed parts can results in common with multiple clusters after the clustering phase, most of the final clusters are meaningful and meet the theoretical expectations. For example, all the parts of assemblies' covers are correctly grouped in mounted clusters, as well as welded subassemblies are properly identified as permanents clusters. We can affirm that, among the three types of clusters, the axisymmetric ones are the most reliably identified, thanks to their precise requirements. The only drawback is that some extra axisymmetric clusters, made up of screws, nuts and spacers not classified as standard parts, can be provided, but this is consistent with the implementation, and actually they are not to be considered false positives. As for the other two types of clusters, some false permanent clusters can be identified, according to the mentioned issues, and consequently some mounted clusters are missing.

5.2 Assembly sequence detection module

The assembly sequence detection module aims at addressing one of the most relevant tasks in the industrial manufacturing field, i.e. the assembly sequence planning, starting from the assembly clusters. At this stage it focuses on the axisymmetric ones. By exploiting the liaisons data along with the properties of the just introduced clusters, it is possible to automatically address the ASP problem considering meaningful engineering information that is usually overlooked and returning more reliable results not only based on geometric reasoning.

5.2.1 The importance of collision detection in ASP

An assembly-by-disassembly strategy is adopted (see Section 2.3.1), which assumes that the assembly sequence can be derived exploiting the reverse of the disassembly sequence. Thus, the starting point would be the CAD model of the complete product, and the goal becomes the evaluation of the order according to which parts can be disassembled without being obstructed. This limits the range of assembly motions to consider, since constraints are imposed on components by their final positioning. Otherwise, if the actual assembly task was considered, each part should be placed in the space without restrictions on the possible directions and choice of the assembly path, and thus computationally finding a feasible solution without experts intervention would become almost impossible. Anyway, even adopting the assembly-by-disassembly strategy, each pair of parts has to be evaluated, and this operation becomes increasingly complex with the increasing of the number of parts composing the assembly. As a consequence, the idea is to exploit the organization of the assembly into meaningful clusters, as those discussed in Section 5.1 and first compute the sequence for each cluster, to reduce the number of parts to deal with simultaneously.

According to the above premises, the approach presented takes the enriched CAD model based on the liaisons structure and organized in clusters as input and computes an assembly sequence for each cluster by evaluating the sequence of removal of its components. The liaisons clearly convey meaningful data both relative to contacts between parts and assembly operations (e.g. the presence of fasteners). What is still needed to know to return a possible sequence are the obstructions between parts during their disassembly, i.e. when moved from their final position inside the cluster to their total removal from it.
The extraction of this kind of information, commonly denoted as *collision detection*, is the primarily topic addressed in this section.

Collision detection

Collision detection is the fundamental phase in ASP/DSP that evaluates the movement of each part of an assembly along some directions to identify the collision-free paths according to which it can be removed from the assembly without intersecting other components. In general, matrices, usually known as *precedence matrices*, containing the detected collisions are returned as output.

As discussed in Section 2.3, current collision detection methods need to be improved to deal with real industrial models. In fact, on the one hand it is often assumed that the precedence matrices are already given or they are manually defined by dragging components in a physical simulation environment with the help of commercial CAD software. On the other hand, most of the proposed approach for collision detection are based on weak assumptions, that abstract much from real engineering situations.

The most relevant limits include: movement directions limited to the orthogonal reference axis x,y, and z, weak or no exploitation of the characteristics and semantic meaning of mechanical parts, limited/no consideration of the presence of functional sets.

The proposed approach aims to tackle these limitations and provide a solution that considers significant directions of movement, the functionality of the parts and their geometric characteristics.

5.2.2 Preliminaries on the developed collision detection approach

To show the advantages of the extracted information for the sequence detection, in this thesis the axisymmetric clusters have been considered. These clusters have distinctive features that allow to address the problem on realistic models.

The axis of symmetry suggests the direction of assembly of the parts and not being bound to the orthogonal axes avoids trivial errors; for example, those issues deriving from the use of a rotated reference system or from the diagonal positioning of the axisymmetric sets in the assembly. Moreover, the availability of semantic information about some of the standard parts usually included in axisymmetric clusters allows to optimize the collision detection. The awareness of the presence of deformable components (e.g. circlips, O-rings) or locating elements (e.g. keys) allows to overcome ambiguous results that would be returned from the mere geometric analysis of precedences.

The algorithm implemented for the automatic detection of collisions between axisymmetric cluster's components to be disassembled is outlined in the following. It strongly exploits the characteristics of this type of clusters in order to reduce and simplify the interference analysis. In fact, due to their nature, the 3D collision detection can be reconducted to a 2D profiles intersection problem.
Thus, for axisymmetric clusters, the identification of the direction of movement according to which the parts have to be (dis-)assembled and thus the collisions have to be evaluated is automatic. It is indicated as **d** and corresponds with the directional vector of the axis of symmetry of the cluster, and thus also of all its composing parts.

The general schema is meant to consider the cluster's parts one at time. Each part has to be ideally moved along the disassembly direction **d** and whether it intersects any of the other parts is evaluated.
Since the automatic simulation of the disassembly is the most challenging and onerous task, the collision analysis is performed in two steps. The first is a qualitative analysis, aimed at excluding trivial cases, it involves the use of contacts' knowledge and bounding boxes. The latter is a more accurate, but computationally complex too, analysis which exploits geometric techniques to solve the collision detection.

The expected output is the precedence matrix **M**. It is a n-by-n square matrix, with n the number of custom designed parts of the axisymmetric object and where each element m_{ij} corresponds to a pair of parts of the cluster, namely the pair given by the i-th part P_i and the j-th part P_j .
Each m_{ij} with $i, j = 1, ..., n$ reports information about the relative obstructions between the two associated parts. It can assume a values between 0,1 or 2 according to the four reported scenarios:

- $m_{ij} = 0$ if $i = j$;
- $m_{ij} = 0$ if P_i does not intersect P_j when moved along the direction **d**;
- $m_{ij} = 1$ if P_i intersects part P_j when moved along the direction **d**;
- $m_{ij} = 2$ if there is a blocking contact between P_i and P_j (where a blocking contact refers to a contact between planar faces with normal vectors parallel with **d**.)

Based on how it is constructed, the matrix **M** implicitly contains much relevant knowledge on parts' precedences that can be leveraged in the disassembly sequences detection. In particular, the rows represent the movement of the parts along the direction $\mathbf{d}$, while the columns represent the movement of the parts along the opposite direction $-\mathbf{d}$. If a row/column is zero, it means that the associated part can be disassembled in the direction $\mathbf{d}/-\mathbf{d}$ with no obstructions. If a row/column has instead some non-zero elements, it implies that the parts associated with the non-zero elements have to be removed before the part associated with the row/column.

The algorithms implemented to address both the steps of the collision detection are detailed respectively in Section 5.2.3 and Section 5.2.4. For sake of simplicity, in the following it will be used the convention of indicating as p the part whose movement is simulated, and q the part that may obstruct the removal of p.

5.2.3 Collision detection filtering phase implementation

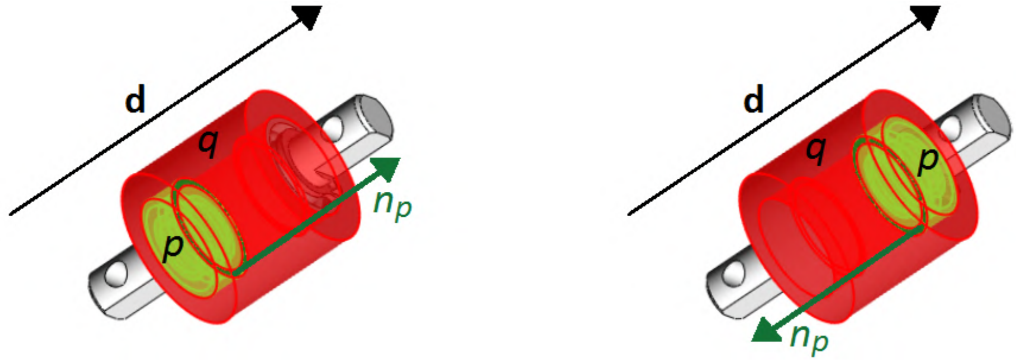
This phase of the collision analysis is basically a static evaluation to reduce the number of cases that have to be further checked and thus to fix some entries of the matrix M with no complex computations. It leverages the knowledge provided by the liaisons and the organization of the components within the cluster. According to the space occupied by parts and their relative positions, considerations are made both on the certainty that a part is blocked and on the possibility of moving a component without interferences. Namely the contacts analysis and the bounding boxes analysis are developed.

Contacts analysis

When the collision of a part p, moved along direction $\mathbf{d}$, with a part q have to be investigated, the evaluation of liaisons can be crucial to avoid calculations.
In fact, if a liaison between p and q exists, i.e. $l(p, q)$, it means that there is one or more couplings between the two parts. If at least one of the couplings embedded in the liaison meets certain requirements it can be concluded that the contact between p and q is blocking an thus the collision would be necessarily detected.

In particular, if a coupling $c(f_p, f_q)$ involving p and q exists such that the faces f_p and f_q are planar, then the orientation of the faces is crucial to define if one part is definitely blocked in a certain direction by the other. In the considered case, i.e. under the axisymmetric hypothesis, if the planar faces in contact have normal vectors $\mathbf{n}_p$ and $\mathbf{n}_q$ respectively parallel to the direction of movement $\mathbf{d}$ and to the opposite direction $-\mathbf{d}$, or vice versa, it can be concluded that the contact is blocking (Fig. 5.20), and the associated element of M can be fixed. In details:

- $\mathbf{n}_p // \mathbf{d}$: p is blocked by q in direction $\mathbf{d}$ and thus can not be moved along $\mathbf{d}$



(a) $\mathbf{n}_p // \mathbf{d}$ (b) $\mathbf{n}_p // -\mathbf{d}$

Figure 5.20: Examples of blocking contacts.

(Fig. 5.20a). The matrix element associated with p and q is set to 2.

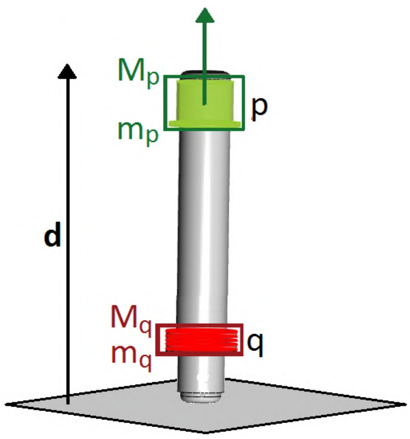
- $\mathbf{n}_p // -\mathbf{d}$: p is blocked by q in direction $-\mathbf{d}$ and thus can not be moved along $-\mathbf{d}$ (Fig. 5.20b). For the matrix properties, this information is reported in the matrix by setting the element associated with q and p to 2.

Bounding boxes analysis

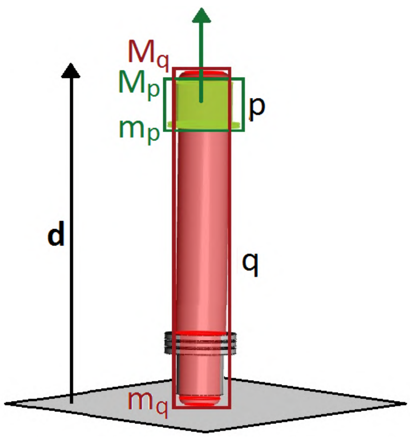
The second analysis that characterizes the filtering phase of the collision detection exploits the bounding boxes of p and q to assess if p can be definitely removed without interfering with q.
To simplify the analysis, the bounding boxes of the parts are defined in a reference system such that the direction $\mathbf{d}$ corresponds to the z axis.
To exclude an intersection between p and q when p is moved along $\mathbf{d}$, and, thus, to set the matrix element associated with p and q to 0 is necessary and sufficient to compare the minimum m_p and maximum M_Q of z coordinates of the two bounding boxes. In particular:

- $m_p \geq M_q$: it means that p is above q relative to direction $\mathbf{d}$, and they will never intersect (Fig. 5.21a). The matrix element associated with p and q is fixed at 0.

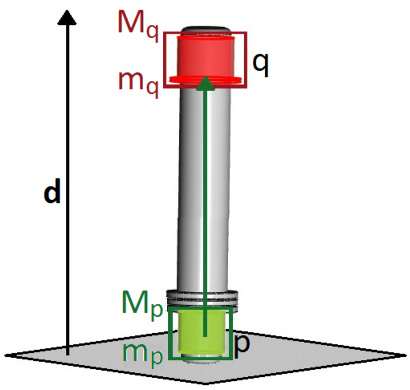
- $m_p \leq M_q$: it can indicate both that p is overlapped with q (Fig. 5.21b) or p precedes q (Fig. 5.21c) relative to direction $\mathbf{d}$, and they may intersect when p is moved along $\mathbf{d}$. The matrix element associated with p and q remains undefined and is then solved by reasoning on the intersection of their planar sections, as described in the next chapter.



(a) $m_p \geq M_q$



(b) $m_p \leq M_q$



(c) $m_p \leq M_q$

Figure 5.21: Possible relative positions between two bounding boxes.

5.2.4 Accurate collision detection phase implementation

This phase expects that the movement of a part is simulated exploiting only its CAD model.

Two possible approaches, which are based on fundamentals of projection geometry to solve the collision detection in a reduced 2D space, are described and discussed in the following. Namely, the first uses *planar projections*, the other instead relies on *planar sections*. They are both implemented and can be used in the collision detection process for axisymmetric clusters, with the difference that the first is more general and can be extended to all type of clusters, while the latter makes extensive use of axisymmetric hypothesis.

Planar projection approach

The planar projection approach consists in the analysis of the intersections of the planar domains given by the parts' projections on a plane with normal vector parallel to direction **d**. In this way, the silhouettes of parts p and q are compared detecting when a part can be removed sliding on the other with no obstructions, even if bounding boxes intersection analysis fails.

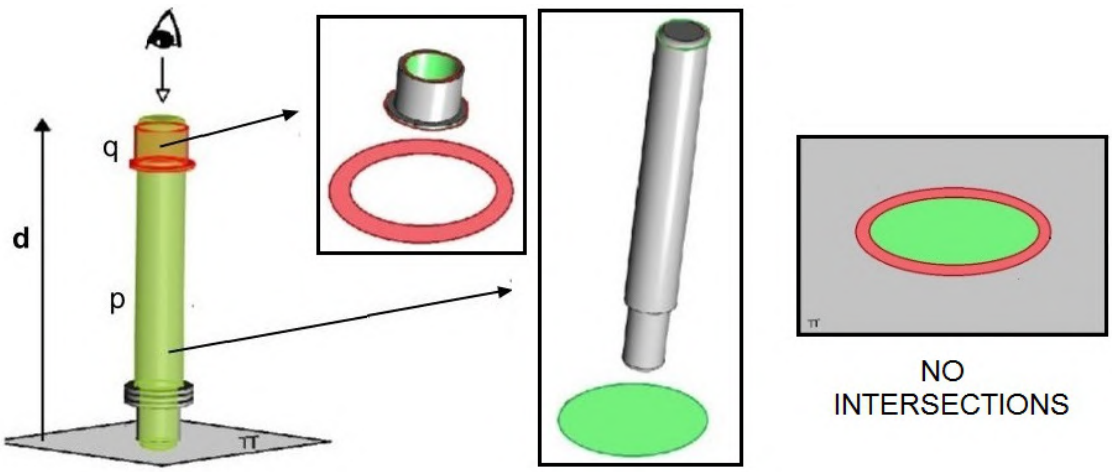
The planar projections analysis consists of two sub-steps: the first analyses if the movement of p in direction **d** is obstructed by q; the latter analyses if the movement of q in direction $-\mathbf{d}$ is obstructed by p. The two analyses are complementary and refer to the same element of the matrix, once read either according to the rows or to the columns.

Step 1 This first step of the projections analysis assesses if p collides with q when disassembled along **d**. Consequently, the shape of the portion of q, if present, which is below p in relation to **d** is not relevant, since p will be moved in the opposite direction. As shown in Figure 5.22c, where part p is green and part q red, only the portion of q standing above p in relation to **d** can cause intersections between p and q during the disassembling. Thus, the silhouette of the portion of q standing below p should not be projected, since it can lead to mistakes.

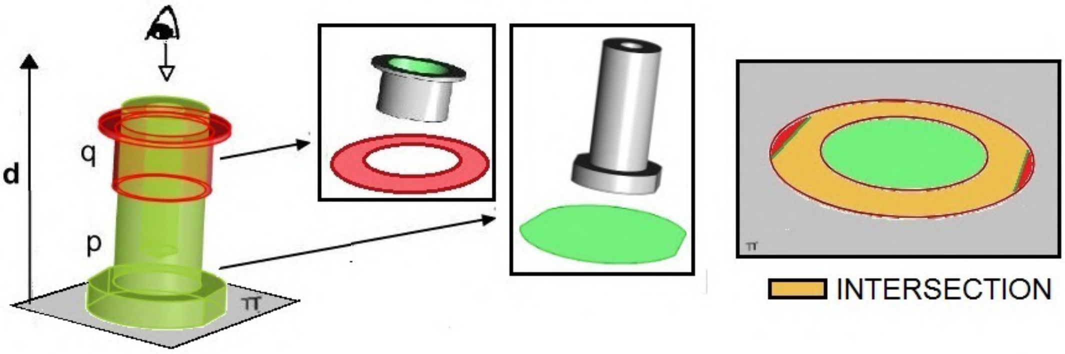
To do this, after setting a reference system such that the direction **d** corresponds to the z axis, the projection plane π is defined with normal vector parallel to the direction **d** (i.e. π is parallel to the xy plane of the system and $\mathbf{d} = z$) and origin with z coordinate aligned with the vertex of p with minimum z coordinate.

Parts p e q are then projected on π according to $-\mathbf{d}$ and thus, only the silhouettes of the portions of the parts standing in the half-space associated with π and $+\mathbf{d}$ are obtained. As a consequence, part p is always totally projected, since it lies on π and extends in the positive direction of **d**. Part q, instead, is totally projected when it is entirely above p, i.e. the minimum z coordinate of part q in the reference system is positive (Fig. 5.22a, 5.22b), but it is partially projected when it is crossed by π and it is in both the half-spaces, i.e. the minimum z coordinate of part q in the reference system is negative, and thus the redundant portion of q is not considered, reducing errors (Fig. 5.22c).

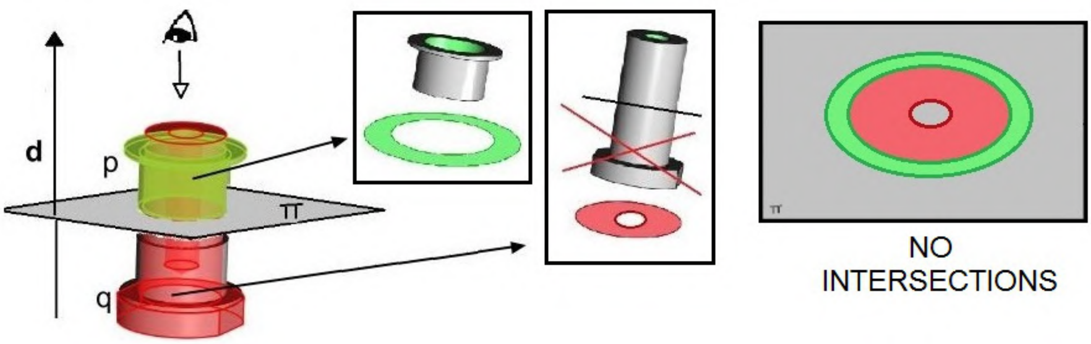
Geometrically, given a part, the output of the projection is a set of co-planar curves that corresponds to the silhouette of the considered portion of the part. The set contains at least one curve, the bigger external profile of the part, but it



(a) Parts p and q totally projected with empty intersection.



(b) Parts p and q totally projected with not empty intersection.



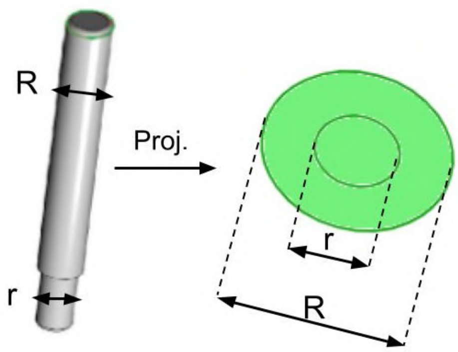
(c) Parts p totally projected and part q projected only for the portion above part p with empty intersection.

Figure 5.22: Examples of projections of parts p (in green) and q (in red) on plane π along direction $-\mathbf{d}$ to evaluate the disassembly of p in direction $\mathbf{d}$ and visualization of the intersection of the planar domains.

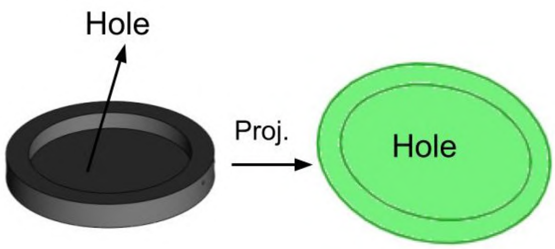
can include other internal curves corresponding to holes (basically the pass-through holes with axis parallel to $\mathbf{d}$). Roughly speaking, what is expected as a result of the projection of a part on π is what it would be seen if looking at the part in direction $-\mathbf{d}$. Once part p and q have been projected, their resulting curves define two planar domains which lie on the same plane π. The situation that can occur are the following:

- If the planar domains do not intersect each other, it means that part p can be moved along $\mathbf{d}$ without colliding with q. It can be concluded that the matrix element associated with p and q is set to 0 (Fig. 5.22a, 5.22c).
- If the planar domains intersection is not empty, it implies that p may collide with q when moved along $\mathbf{d}$. The matrix element associated with p and q remains undefined (Fig. 5.22b).

The second situation and its result have to be better justified. In particular, false obstructions may be obtained through the above described planar projection analysis. In fact, the curves returned after a part projection correspond to the bigger external profile of the part and the smaller internal holes. For instance, if the external surface of a part is characterized by two or more cylindrical faces with different radii, only the profile of the face with a bigger radius is reported in the planar domain generated by the projection, as it is shown in Figure 5.23a. Similarly, blind holes are not detected (Fig. 5.23b), as well as only the smaller radius of countersink or counterbore holes is visible by means of the projection.



(a) Projection of a part with two cylindrical faces where only the face with bigger radius is reported.

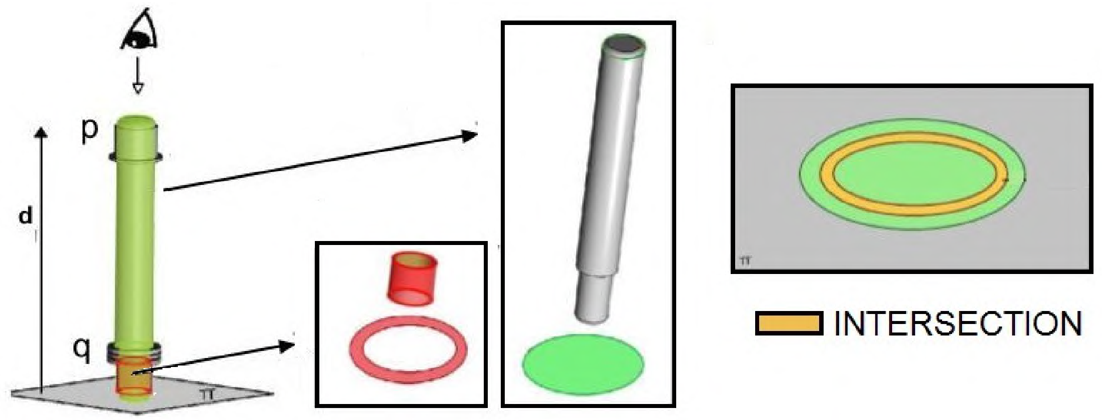


(b) Projection of a part with a blind hole where the presence of the hole is not reported.

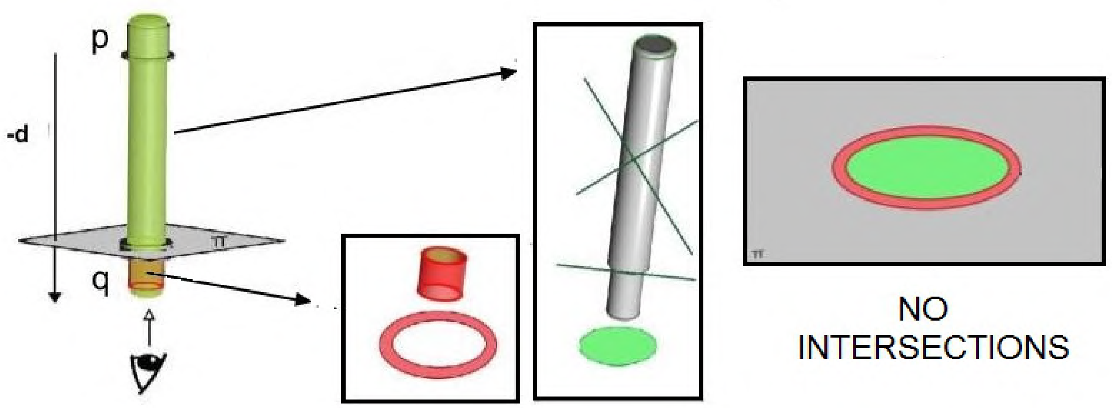
Figure 5.23: Examples of projections where some parts' details are not reported in the planar domain.

The planar domains therefore are often bigger than the exact profiles of the parts. As a consequence some intersections between the planar domains of the parts can result even if the bodies do not obstruct each other.
The problem is solved by performing the second step, i.e. simulating the movement of q in direction $-\mathbf{d}$.

Step 2 If the previous step returns an intersection between the projections of p and q, the obstruction is re-evaluated to confirm the intersection or assert the freedom of movement for p versus q. To this aim, since the columns of the matrix $\mathbf{M}$ represent the movement of the parts along the opposite direction $-\mathbf{d}$, this second step of the planar analysis assesses if part q collides with p when disassembled along $-\mathbf{d}$.



(a) Projection along $-\mathbf{d}$ returns false intersection.



(b) Projection along $\mathbf{d}$ returns correct empty intersection.

Figure 5.24: Example of false obstruction resulting from the projections along direction $-\mathbf{d}$ to evaluate the disassembly of p (in green) in direction $\mathbf{d}$, but solved by projecting along direction $\mathbf{d}$ to simulate the disassembly of q (in red) in direction $-\mathbf{d}$.

The calculations are equivalent to the ones executed before, but with the roles of part p and q inverted, as well as the normal of the projection plane π. That is to say, π has normal parallel to $-\mathbf{d}$ and q lies on π, and thus it is always totally projected, while p can be totally or partially projected according to its position relative to q.

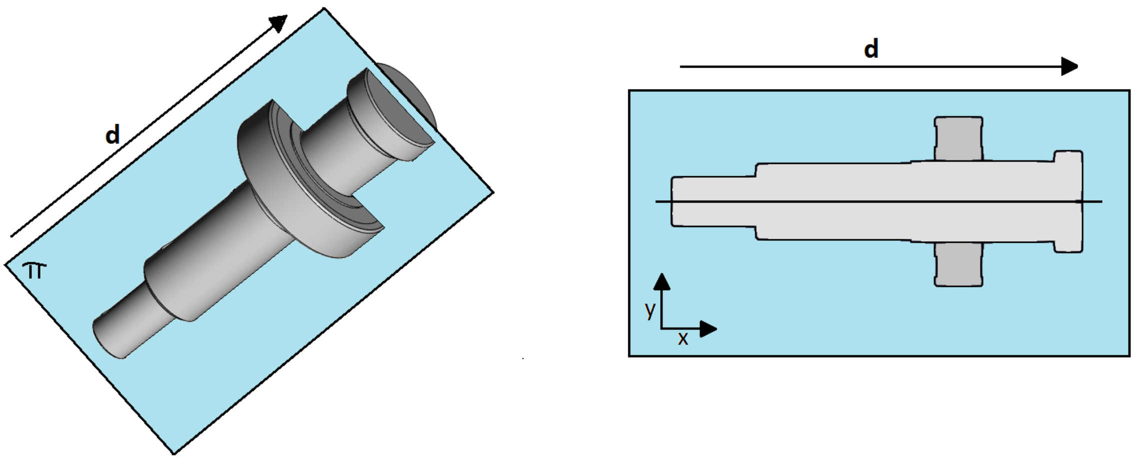
If no intersection is found between the planar domains, it means that q can be disassembled along $-\mathbf{d}$ without intersecting p. Complementary, p can be disassembled along $\mathbf{d}$ without intersecting q, even if the previous step has identified some intersections, which where false-positive (Fig. 5.24). Thus, the matrix element associated with part p and q is set to 0.

If some intersections are found between the planar domains, it means that q collides with p when moved along $-\mathbf{d}$. In this case the matrix element associated with p and q is set to 1.

Anyway, some false obstructions can result even after the two steps of the planar analysis due to the presence of some modeling details (e.g. fillets).

Planar section approach

The planar section approach consists in the analysis of the intersections of the planar sections obtained by cutting the parts p and q along their axis with a plane parallel to direction $\mathbf{d}$. It must be specified that by section it is actually intended the profile of the part corresponding to the maximum radial footprint found along the entire contour. In fact, a mere section could be in an area with a slot, and hence return an incorrect profile.
To simulate, then, the movement of p in direction $\mathbf{d}$, its section is extruded for a certain length so that it can be brought over q.

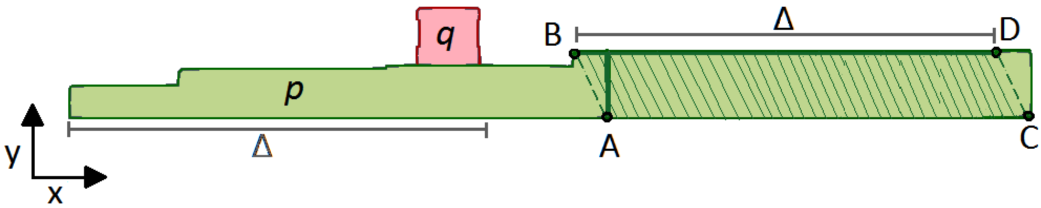


(a) Parts' 3D models with section plane π.

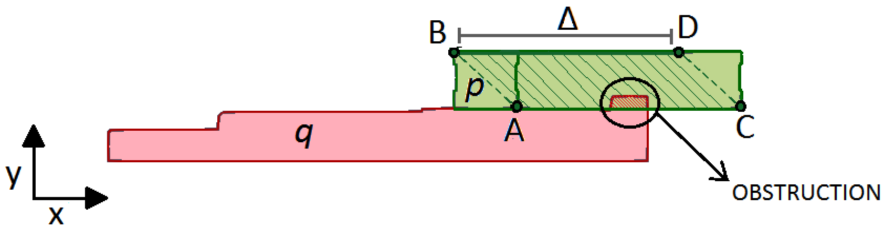
(b) Planar sections of the 3D models.

Figure 5.25: Example of a pair of parts (i.e. bearing and shaft) that require planar sections analysis.

To do this, the cutting plane π is defined with a normal vector perpendicular to the direction $\mathbf{d}$, passing through the parts' axis, and a reference system is set such that π corresponds at the xy plane (Fig. 5.25). The analyzed components p and q are then cut, and the respective sections are obtained. In the rest of the analysis only half of the sections are considered. In fact, thanks to the symmetry conditions of axisymmetric parts, the behavior of half of a section corresponds with the behavior of the whole part, and thus a planar analysis of a restricted portion of the components is sufficient to evaluate obstructions. After the sections computation, the movement of p along direction $\mathbf{d}$ has to be simulated to verify if q obstructs its disassembly. At this aim, the section of p is extruded in direction $\mathbf{d}$ until the space occupied by q is taken.



(a) The bearing (q) does not obstruct the shaft (p) movement.



(b) The shaft (q) obstructs the bearing (p) movement.

Figure 5.26: Evaluation of the obstruction of the bearing when the shaft is moved along $\mathbf{d}$, and vice versa.

Geometrically, this operation is addressed by searching for two points A and B on the profile of the section of p. Respectively, A corresponds to the vertex with the maximum x (if there are multiple, the one with the minimum y is chosen), while B corresponds to the vertex with the maximum y (if there are multiple, the one with the minimum x is chosen) (Fig. 5.26). Two new points C and D are then obtained translating A and B in the direction parallel to $\mathbf{d}$ for a distance Δ such that p will result completely extracted from q. The planar domain given by $ABDC$ covers the maximal space occupied by part p during its disassembly movement. As a consequence:

- if the planar domain given by $ABDC$ does not intersect the planar domain given by the section of part q, then q does not obstruct p when moved along $\mathbf{d}$ (Fig. 5.26a). Consequently, the matrix element associated with p and q is set to 0;

- if the planar domain given by $ABDC$ intersects the planar domain given by the section of part q, it means that the parts collide during the disassembly (Fig. 5.26b), and then the matrix element associated with p and q is set to 1.

It can happen that the sections of p and q intersect themselves already before doing the extrusion. It can be due to modeling issues or numerical errors. This may return a false obstruction during the analysis (Fig. 5.27). To avoid errors, it must be verified the presence of intersections between the two original sections. If there is any, and the ratio of its area to the total area of the planar domains is below a given tolerance, it will not be considered as an obstruction in the analysis.

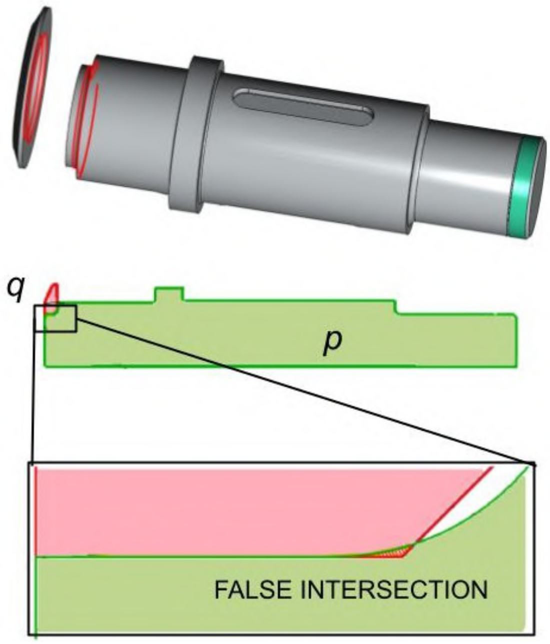
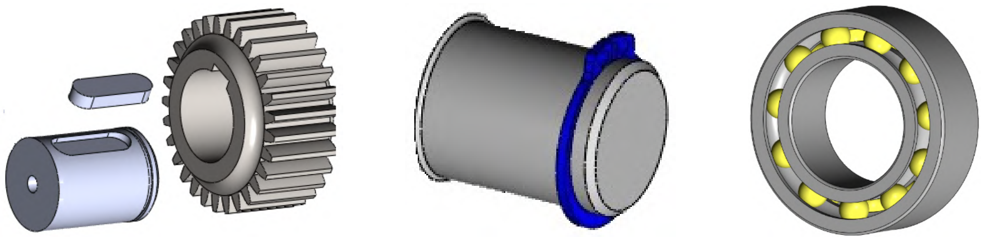


Figure 5.27: Example of intersection between the sections of two parts even before the extrusion of one of the two.

The two approaches, i.e. planar projection and planar section evaluation, can be indistinctly used to evaluate the obstructions between the parts of axisymmetric clusters since results are promising in both cases. As a consequence, the tool developed gives the opportunity to choose which method to apply (as it is shown in Fig. 5.31). From the computation point of view, the planar section is preferable since fewer steps are required. However, the planar projection analysis deserves to be implemented and be available in preparation for the generalization of the collision detection for all clusters type.

5.2.5 Particular cases

The approaches described to evaluate the collision of a part of a cluster with the others during its removal in general return correct results. However, in industrial CAD models of mechanical products situations can occur in which the collision analysis, both when applying the planar projection or the planar section approach, fails due to the distinctive shape of some parts and their modeling that can be inconsistent with real scenarios.
Specifically, when treating axisymmetric clusters, three particular cases can be listed: the use of non-axisymmetric standard parts positioned between two components, i.e. keys (Fig. 5.28a), the use of deformable standard parts with blocking function, i.e. circlips and O-rings (Fig. 5.28b), and the existence of parts that need to be dealt with as a single component, i.e. bearings (Fig. 5.28c). Each single case in investigated in the following.

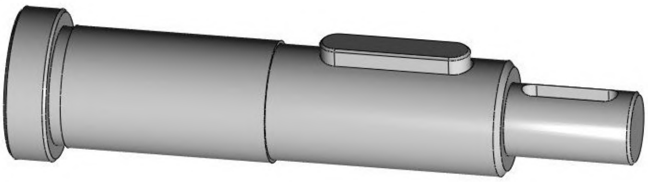


(a) Key. **(b)** Circlip. **(c)** Bearing.

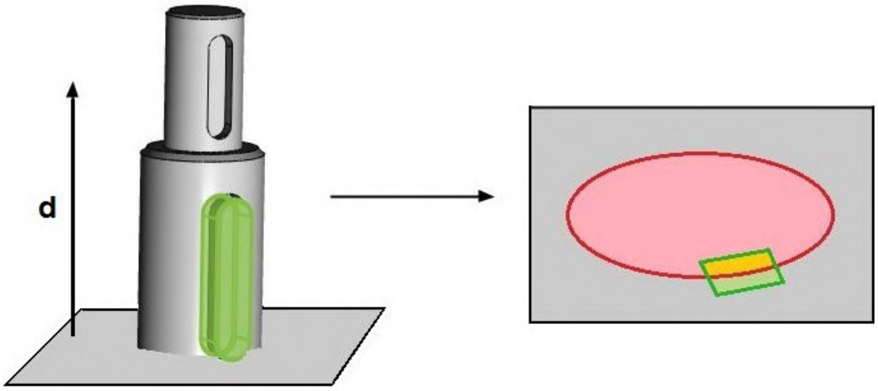
Figure 5.28: Examples of particular cases for the precedence computation.

Non-axisymmetric standard parts. Axisymmetric subassemblies often involve the use of keys to restrain/align pulleys or gears to shafts. Keys, however, are not axisymmetric parts and generally need to be mounted perpendicularly to the axis direction. Thus, both the planar projection and the planar section analysis are not suitable for evaluating their movement. In the first case, in fact, if considering the projections of a key and of the shaft containing it along the direction **d** parallel to the axis of the shaft, the intersection will be not empty (Fig. 5.29). This result is correct based on how the components are arranged, i.e. the key is blocked within an hollow feature carved into the shaft, but it can not be exploited in inferring information on the precedence between the two parts. In the second case, the hypothesis of symmetry is missing, and therefore the section of the key, as it is computed here, is not feasible.

Thanks to the data extraction process, keys are known, as well as the parts they connects and the features associated with keyseat and keyways. As a consequence, to



(a) Key inserted into a shaft.



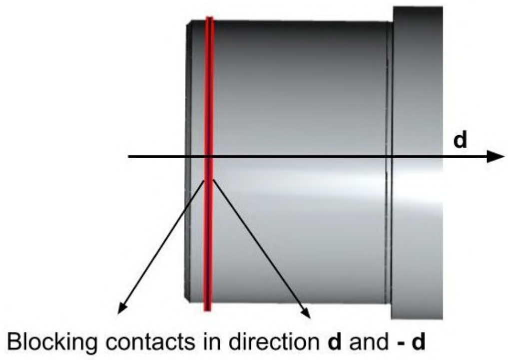
(b) Planar projection analysis where the key (in green) is the part to be moved in direction **d**.

Figure 5.29: Examples of collision analysis for a key and a shaft containing it.

avoid the issues generated by keys in the collision analysis and then in the precedence matrix computation, the obstructions between a key and any other part of the cluster are not evaluated, and keys are not included among the precedence matrix items. The presence of a key, in fact, does not affect the assembly of components except for the two parts it is employed between. Hence, it would be sufficient to separately store the relations between keys and the parts they connect, for instance in an extra matrix and, then, exploit this information in the assembly sequence definition. At this purpose, a general set of rules can be identified and considered for the generation of disassembly precedence constraints according to which a key is almost always disassembled after the external part and before the inner shaft [76].

Deformable standard parts. In axisymmetric subassemblies, circlips and O-rings are commonly used to retain components, but they are standard parts with a particular behavior since are deformable. Namely, these parts are axisymmetric-like and are fitted into their seats, i.e. in the grooves, by enlarging and sliding. In particular when they are modeled with resting shape, the collision analysis between deformable parts and the parts in which they are inserted will return not realistic outcomes. In fact, in most cases, only with the contact analysis two blocking contacts would be detected, both in direction **d** and −**d** (Fig. 5.30b). These are given by the contact of the deformable component with the faces of the associated groove. The mere interpretation of this result implies that the deformable part is stuck into the groove, but it is not meaningful from the precedences standpoint. Even when the contact is not detected for parts positioning issues or modeling errors, and the narrow collision analysis is carried out, results are misleading. This because both projections and section evaluations will return not empty intersection between the planar domains.

(a) Circlip inserted into a groove.



(b) Blocking contacts between the circlip and the faces of the groove.

Figure 5.30: Example of contact analysis for a circlip mounted on a shaft.

The knowledge of liaisons and thus the awareness of circlips and O-rings, along with with their positioning, can be exploited to make some considerations. In particular, to overcome misleading results of the collision detection analysis, the suggested strategy is to simulate the dilation of the deformable components by translating their sections perpendicular to the axis direction for a distance h equal to the depth of the groove, as if the parts were enlarged and then removed by sliding. This would avoid the wrong intersection.

Bearings. Bearings are multi-part components typical of mechanical products for movement transmission. On the whole, bearings are axisymmetric objects, mounted by sliding on a shaft. However, they consist of several parts, all axisymmetric, but some of which (i.e. the rolling elements) have axis not corresponding with the main axis of the shaft (Fig. 5.28c), and thus with the axis of the cluster they are included in. In the collisions analysis, considering each part of a bearing singularly would be ineffective. This because the bearing is one of that components that must be assembled before it is mounted on a (sub-)assembly. Hence, treating the single parts composing it as a independent ones is redundant and misleading.

Therefore, to realistically simulate the engineering way of working and achieve reliable results, while also reducing the size of the precedences matrix, bearings' parts must be treated as if they were merged into one object. Thus, the inner parts that are not coaxial with the cluster along with the cage would not be handled in the collision detection and do not affect it, while the only inner and outer rings together produce the planar domains that must be evaluated in the phase of projection or planar section.

In literature, this topic is never addressed specifically, in the sense that assemblies involving bearings are rarely considered in the ASP context, or else the knowledge of bearings and, hence, their usage as single components is taken for granted. On the contrary, under the hypotheses assumed here, the awareness of bearings is not straightforward nor trivial. In addition, the strategy above proposed to address the problem clearly foresees the identification of bearings. This task, that should be part of the data extraction phase, has not been implemented, but some manual shortcuts and the tool given by Lupinetti et al. [80] have been applied to identify the parts of bearings and group them into an object in order to test the efficacy in analyzing bearings movements as single parts.

For sake of completeness, it must be argued that the merging of bearings' parts also implies the re-computation of the liaisons list. In particular, if the two parts of a liaison are both included in the bearing, the liaison is removed. If instead, a part of a liaison is a component of a bearing, the liaison is modified by substituting the single part with the whole bearing. In this way, duplication of the same liaison may be found, and thus redundant copies will be removed. This considerations make the point of the importance in considering some multi-part component as a single element. This is beneficial not only from the ASP standpoint, but also in the creation of the enriched CAD model, since it allows to reduce the number of liaisons but enhancing their semantic.

In summary, the exploitation of engineering knowledge combined with the available geometric and non-geometric information on assembly's parts are promising elements that can allow to predict and thus tackle some challenging scenarios.

5.2.6 The precedence matrix computation

The previous sections describe the collision detection methods devised for the disassembly of parts in axisymmetric clusters. Here, the overall process for the matrix computation is illustrated.

Starting from the list of clusters returned after the cluster detection phase (see Section 5.1) and selecting one among the axisymmetric, the precedence matrix computation is run.

As shown in Algorithm 3, each pair of parts (P_i, P_j) where $i, j = 0, \dots, n$ with n the number of parts of the cluster, is considered and evaluated with the aim of setting the value of the element m_{ij} of the matrix $\mathbf{M}$. It is assumed that bearings are already grouped into a unique component, and that when at least a part among P_i and P_j is classified as key the associated element is nullified, consistently with the Section 5.2.5.

Algorithm 3 Collision detection

```
Data: PARTS:=list of parts; n ≥ 0 number of parts
Result: M precedence matrix
for i = 0 : n do
    for j = 0 : n do
        if i = j then
            m_ij := 0
            Continue for
        end
        if P_i is KEY or P_j is KEY then
            m_ij := −
            Continue for
        end
        p := P_i; q := P_j; ;                                /* Filtering phase */
        if p in contact with q then
            if contact is blocking then
                m_ij := 2
                Continue for
            end
        end
        B_p := bounding box of p; B_q := bounding box of q
        if min_z(B_p) > min_z(B_q) then
            m_ij := 0
            Continue for
        end
        S_p := section of p; S_q := section of q ;           /* Accurate phase */
        S'_p := extrusion of S_p
        if S'_p ∩ S_q ≠ ∅ then
            m_ij := 1
        else
            m_ij := 0
        end
    end
end
```

The two analyses involved in the filtering phase, i.e. the contact analysis and the bounding box analysis are executed and when a blocking contact or a collision free path is found, the value of m_{ij} is fixed respectively at 2 or 0. The remaining pairs are subject to the accurate collision detection phase. Since the context is that of axisymmetric clusters, the planar section method is applied by default, but the end user can also choose to apply the planar projection method (Fig. 5.31).

Finally, starting from $\mathbf{M}$, a first attempt in deriving the disassembly sequence of the analyzed cluster can be done by reordering the matrix rows and columns. In particular, based on the assumption that a zero row implies that the corresponding part can be disassembled with no collisions from the cluster in direction $\mathbf{d}$, and a zero column implies that the part can be instead freely disassembled in direction $-\mathbf{d}$ (see Section 5.2.2), the idea is to remove at each time a zero row/column of $\mathbf{M}$ and reevaluate the obtained sub-matrix. In this way a possible order according to which parts can be disassembled is obtained, i.e. it is given by the order in which the zero rows/columns are selected. It is a sequence feasible from the collision-free path point of view, but its feasibility is not ensured from the engineering standpoint. A more thorough assessment must be made in relation to the adopted assembly tool and to the relative positioning of the parts (for example if one is mounted concentrically on the other or only side by side) to obtain a realistic and optimal sequence, but this will be part of future works.

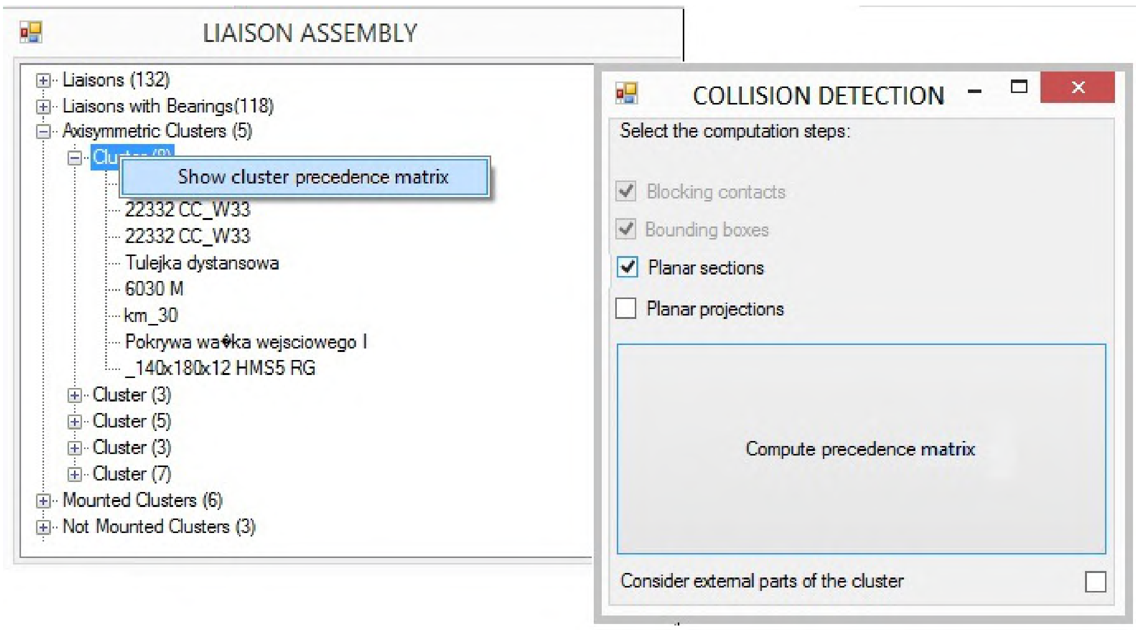


Figure 5.31: User interfaces for the selection of the axisymmetric clusters of which computing the precedence matrix and the selection of the collision detection steps.

User interface

The precedence matrix computation starts from the form where the list of clusters is presented. The user can select one of the axisymmetric clusters and ask for the precedence matrix computation (Fig. 5.31).

After the collision detection evaluation the final matrix is presented in a further tab. To allow an intuitive reading of the results, the tab providing the matrix is connected with the viewer form, where for simplicity only the analyzed cluster is now visualized (Fig. 5.32). Namely, selecting a cell of the matrix, the two associated parts are highlighted in the 3D model. In particular, the part associated with the row, that is the part whose movement along **d** is simulated, is colored green, while the part associated with the column, that is the part that may obstruct the movement, is colored red. This choice allows to focus the attention on each pair of parts and also provides a visual confirmation of the results.

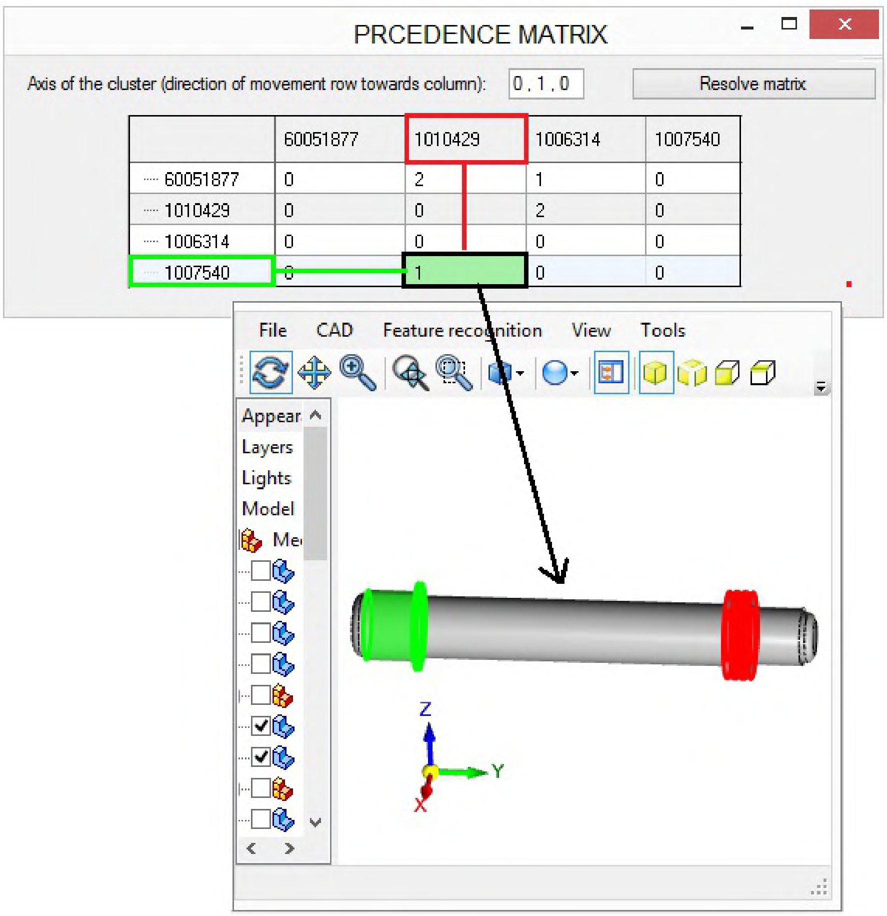


Figure 5.32: Interface for the visualization of the precedence matrix and highlight in the viewer of the parts corresponding with the selected cell.

5.2.7 Collision detection evaluation

In this section the application of the collision detection approach to compute the precedence matrix for axisymmetric clusters is evaluated on a test case.
In particular, the CAD model of the gearbox already treated in Section 5.1.3 is considered. It is a significant assembly since the clustering detection returns 5 asymmetric clusters included within the gearbox (see Fig. 5.17) which involve both standard parts, such as keys and circlips, and bearings modeled as multi-part components.

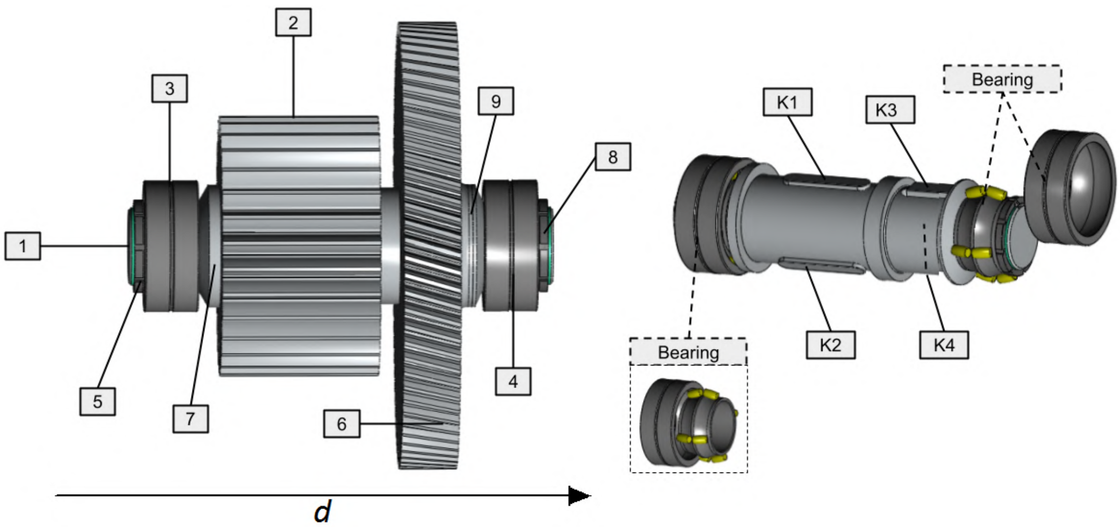


Figure 5.33: Axisymmetric cluster 5 detected in the gearbox CAD model (see also Fig. 5.17). On the left: visualization of the complete cluster. On the right: focus on the bearings and keys mounted on the shaft (part 1).

	1	2	3	4	5	6	7	8	9
1	0	0	0	1	0	2	0	1	1
2	2	0	0	1	0	0	0	0	1
3	1	1	0	1	0	1	2	1	1
4	0	0	0	0	0	0	0	2	0
5	1	0	2	0	0	0	1	0	1
6	0	0	0	1	0	0	0	0	2
7	1	2	0	1	0	1	0	1	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	2	0	0	0	1	0

= M

(a) Precedence matrix.

	K1	K2	K3	K4
1	x	x	x	x
2	x	x	-	-
3	-	-	-	-
4	-	-	-	-
5	-	-	-	-
6	-	-	x	x
7	-	-	-	-
8	-	-	-	-
9	-	-	-	-

= K

(b) Matrix of the keys' relations.

Figure 5.34: Results of the collision analysis applied to the axisymmetric cluster 5 of the gearbox.

In the following, the collision analysis for one of the identified clusters is outlined, namely the cluster 5 (as it is referred in Fig. 5.17). It consists of 13 parts, 2 of which are bearings and 4 are keys (Fig. 5.33).
As explained in Section 5.2.5, bearings' parts are merged into single components. That is, each bearing is actually composed of 10 parts (i.e. rolling elements and inner and outer rings) but for consistency with the method they are tackled as a unique object during the collision analysis. Keys, instead, are not included in the precedence matrix **M**, but they are separately associated with the parts they fasten together. Thus, the matrix **M** computed with the collision analysis algorithm is a 9-by-9 square matrix (Fig. 5.34a), and a 9-by-4 matrix, denoted as **K**, is also reported to show the relations between the keys and the relative shaft and/or gears (Fig. 5.34b).

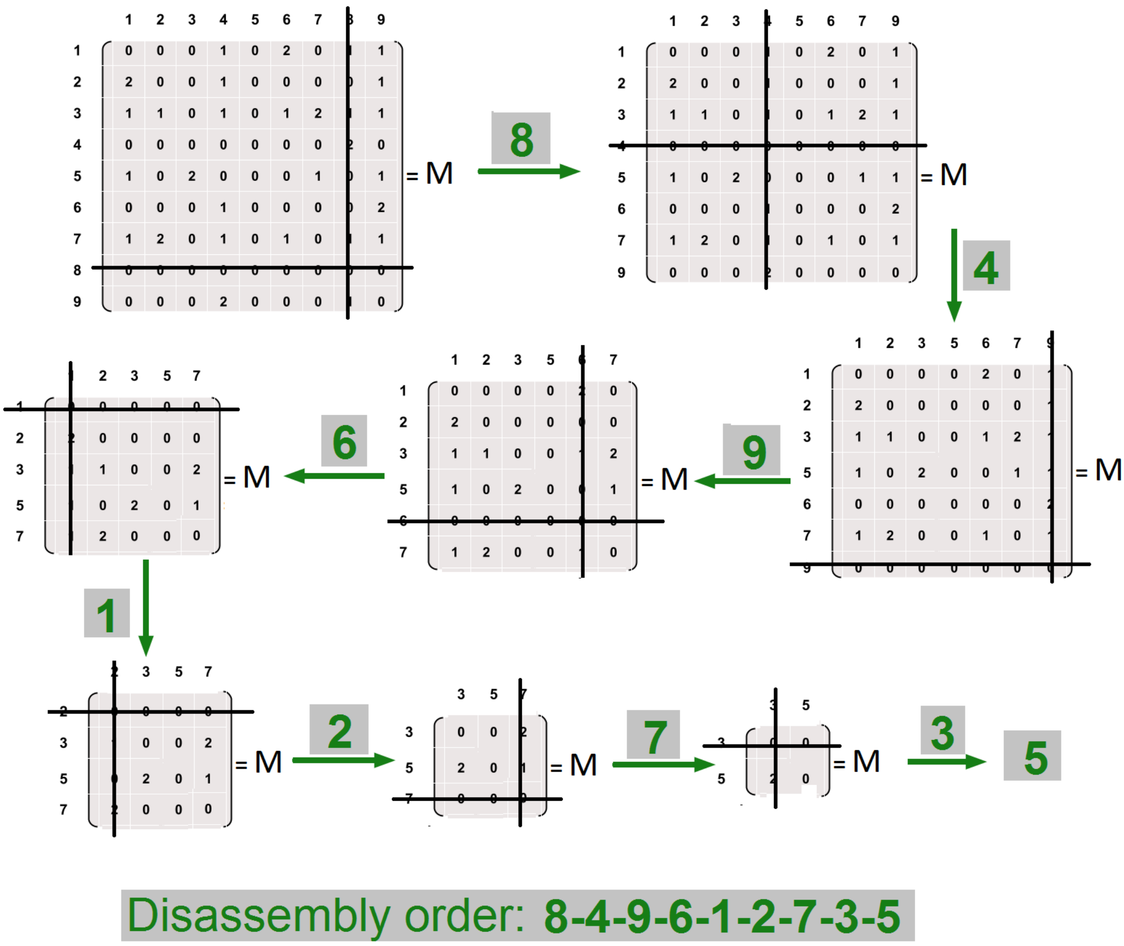


Figure 5.35: Example of disassembly sequence obtained through the analysis of the zero rows.

The provided approach results promising. The obstructions between the subassembly's components are correctly detected and the computation is totally automatic, without the need of human intervention. Moreover, the actual pairs of parts that have to be evaluated by means of the sections analysis is reduced from 169 (i.e. assuming the all 13 parts should be analyzed) to 36 thanks to the awareness of keys as well as to the contacts and bounding boxes phase, allowing to reduce the required computational effort. For sake of completeness, also the possible disassembly sequence obtained by iteratively removing at each time a zero row of M is provided in Figure 5.35.

5.3 Design for assembly module

To validate the usefulness of the enhancement of CAD models with additional semantic information in other contexts than the assembly planning, the Design for Assembly process has been considered, identifying how it can be improved using the extracted data. In particular, it has been investigated how the semantic information returned by the data extraction phase (see Chapter 4) and collected in the liaisons can be leveraged to automatically calculate DFA indices, which is an operation usually based on manual procedures.

5.3.1 The need of automate DFA

DFA is a systematic procedure largely employed in the mass manufacturing production aiming to ease and improve the assembly of a product, both in terms of design quality and assembly time and cost. The efficiency in the application of the approach is captured by the renowned DFA Index [31], expressing the ratio between an ideal assembly time based on a minimum number of components N_{min}, a minimum theoretical assembly time per component t_{min}, and the actual assembly time TA.

$$DFA \quad Index = \frac{t_{min} * N_{min}}{TA} \tag{5.1}$$

A higher DFA index reflects an improved design solution and, accordingly to equation 5.1, it can be clearly met by both reducing the number of components and improving the execution time of the single assembly task.

Regarding the first point, a limited number of parts is beneficial since it significantly lightens numerous company activities which need to be potentially performed for each single part code: design, drafting, CAE simulation, coding, acquisition, manufacturing planning, logistic chain organization, maintenance organization, etc.. In particular, B&D methodology [31] strongly pushes on the reduction of the num-

ber of components.
The goal can be pursued in a twofold manner. At first, parts can be eliminated whose function can be realized in an alternative more efficient manner. Typically, this refers to fasteners such as screws, nuts, washers, studs, etc., which can be substituted by solutions requiring much lower assembly times, such as snap fits. Secondly, separated parts can be consolidated in single components whenever possible, subject to three specific conditions:

- parts are realized with the same material;
- parts are not subjected to relative displacement;
- parts are not required to be separated for assembly or disassembly operations.

As second strategy, assembly time can be reduced by design strategies which leverage improved part geometries and arrangement. It refers to part symmetries, unambiguous contact conditions, wide visibility on the parts to be assembled. In a similar way, in Lucas and Hull approach [79] the efforts required for assembly tasks has been investigated, dividing the required time between handling and fitting operations. Handling refers to tasks required for feeding and manipulating parts. It depends on size and weight of the part, manipulation difficulties and orientation ambiguities. Fitting includes the time to insert and secure each part. Here, again, the main drivers to consider are the number of components and the method used to fix them.

From this brief overview of DFA strategies, it is evident how the type and the geometry of the parts and their relative position in the assembly are the most important aspects considered by the majority of the analyses and consequent actions to improve the product design. It is important to observe that the reported approaches have been mainly developed for low volume production batches. This is the case where manual assembly is normally adopted and most significant results in terms of gained efficiency can be pursued by an efficient application of DFA.
However, the mentioned analyses are basically manually operated. The scientific literature lacks of methods for the automation of the required product evaluation. Also, from an industrial point of view, few commercial tools exist on the market to support DFA analysis. Most of the software, such as DFMA® from Boothroyd Dewhurst, Inc. [13] are basically supporting systems requiring a manual input of the data regarding the characteristics of the parts of the product. Some other tools leverage the shape analysis of parts and the geometric feature recognition algorithms, such as DFMPro® [14]. However, in the last case the scope of the analysis is mostly

limited to a single part in a Design for Manufacturing perspective rather than an entire assembly investigation.

In this context, it is evident that significant benefits can be obtained if DFA supporting tools are fed by data automatically extracted from the product CAD models. This would allow a wider adoption of the DFA strategies, including in a mass production scenario, leading to improved product quality and cost, thus better addressing the current compressed time constraints in the product development process.

To this end, the liaisons' data are promising because they include most of the necessary information to drive automated DFA analysis. For example, the above mentioned techniques to reduce the number of parts can be supported by the standard parts recognition (see Section 4.2) and clusters detection (see Section 5.1), as well as the design strategies to reduce the assembly time can take advantage of the geometric processing phase outcomes (see Section 4.1). In the following, the association between the fundamental DFA parameters and the geometric and non-geometric attributes automatically extracted and then stored in the enriched CAD model is discussed.

5.3.2 The DFA indices mapping

With the final aim of providing a tool capable of performing an automated DFA analysis, work has been done to identify which information obtained by means of the data extraction phase (see Chapter 4) and contained in the results of the data exploitation modules already discussed (see Sections 5.1 and 5.2) can be leveraged to evaluate aspects relevant to DFA.
In practice, the first crucial step carried out is the mapping between the aspects necessary for estimating and improving assembly times and applying optimized design guidelines, and thus aimed at increasing the DFA index (see equation 5.1), and the attributes extracted and available from the geometry and topology of the CAD model of the analyzed product.

Table 5.2 reports the associations that have been done. For sake of clarity, the table is divided into two blocks. In the first set of columns, in orange, the data related to the real assembly organization and mounting tasks are listed, while in the latter columns, in green, the needed information associated with the CAD model of the assembly is reported.

Moreover, for a better contextualization and understanding of each purpose or characteristics of the parts to be evaluated, i.e. the items of the *Aspect relevant*

Table 5.2: Table of the mapping of the DFA parameters to the geometric and non-geometric attributes that can be extracted from the CAD model of a mechanical assembly.

Phase			Aspect relevant to DFA	Level of geometric analysis			Attributes
General	Handling	Insertion		Feature	Component	Assembly	
✓			Identification of component to be eliminated		✓		Standard parts type
✓			Foster the reduction of standard parts type		✓		Standard parts type and diemnsions
✓			Foster the use of standard parts and not custom designed ones		✓		Similar parts retrieval from repositories
✓			Identification of component to be integrated with others		✓		Material
						✓	Type of liaison, Type of cluster, Assembly sequence
	✓		Easiness to grasping, manipulating, handling		✓		Bounding box size, Material, Thickness dimensions
						✓	Type of liaison, Product structure, Accessibility
	✓		Thickness		✓		Dimensions
	✓		Size		✓		Largest bounding box size
	✓	✓	Symmetry angle (alpha + beta)[31]		✓		Symmetry and axes, angle to repeat orientation along main axes [31]
	✓		Need for tweezers or special tools for grasping		✓		Bounding box size, Material
						✓	Component accessibility
	✓		Need for optical magnification		✓		Largest bounding box size
	✓		Additional grasping difficulties		✓		Material, Surface roughness, Surface treatments
	✓		Parts severely nest or tangle		✓		Material, Min thickness, Slenderness
	✓		Heaviness of the part		✓		Weight
	✓		Two person, hand or mechanical assistance		✓		Weight, Bounding box size
						✓	Assembly sequence
		✓	Holding down required to maintain orientation and location after assembly			✓	Product structure, Type of liaison, Type of cluster
		✓	Easiness to align and position	✓			Axisymmetric features, Presence of chamfers, Holes
					✓		Angles to repeat orientation along principal inertial axes
		✓	Resistance to insertion			✓	Mating faces, type of liaison
		✓	Part, tools and hands can reach the location			✓	Product structure, Accessibility
		✓	Restricted view on the location			✓	Product structure, Obstructions, Visibility map
		✓	Fastening process type	✓			Presence features for fasteners seats, Pattern of features
					✓		Standard parts type, Custom designed parts type, Material
						✓	Type of liaisons, Type of cluster
		✓	Screw tightening after insertion	✓			Type of features, Aligned threaded holes
					✓		Standard parts type
						✓	Type of liaison
		✓	Plastic deformation after insertion		✓		Material
						✓	Type of liaison, Type of cluster
		✓	Riveting after insertion			✓	Type of liaison

to DFA column, these are assigned to one or more *Phases*, that indicate the assembly phase they affect. In details, three distinct phases are distinguished, that are: general, handling and insertion. The general phase refers to the design tasks that aim to simplify the structure of the product analyzed. The handling phase, instead, includes the operations related to the manipulation of the parts. Finally, the insertion phase involves the actual assembly tasks, namely the positioning and mounting of a part within the assembly.

As for the geometric and non-geometric attributes listed in the column *Attributes* and matched with each DFA aspect, these are divided into categories based on the level of geometric analysis required to extract them from the CAD model. The levels of geometric analysis identified are three, precisely: feature, component, and assembly. The feature level analysis includes those characteristics related to single portions of parts. The component level analysis provides for all the data implicitly or explicitly associated to each single part. Information on parts relations and relative positioning are instead obtained with the assembly level analysis.

By making reference to the system developed and the tools discussed so far, it appears that most of the attributes reported in Table 5.2 are actually already gathered in the enriched CAD model as properties of the liaisons or can be easily inferred and further stored during the computational processes as they are computed and used in different algorithms (e.g. the bounding boxes). In the case of the attributes of feature level, they mainly correspond with the outcomes of the feature recognition process (Section 4.1.3). Also a large portion of the attributes belonging to the component level are strictly related with the geometric processing phase (Section 4.1.2) and the standard parts recognition (Section 4.2). Assembly level attributes, instead, can be found in liaisons, but also in the clusters detection (Section 5.1) and collisions detection (Section 5.2) results. In summary, under the hypotheses imposed here (see Section 1.3.1), only few attributes can not be automatically derived from the CAD model, that are those physical or semantic properties not at all related to the parts' geometry, such as material, weight, and surface roughness.

These considerations validate the possibility of automatically addressing the DFA methodologies and developing tools not only to extract the required parameters, but also to support the designer works with alerts and suggestions.

5.3.3 Test case

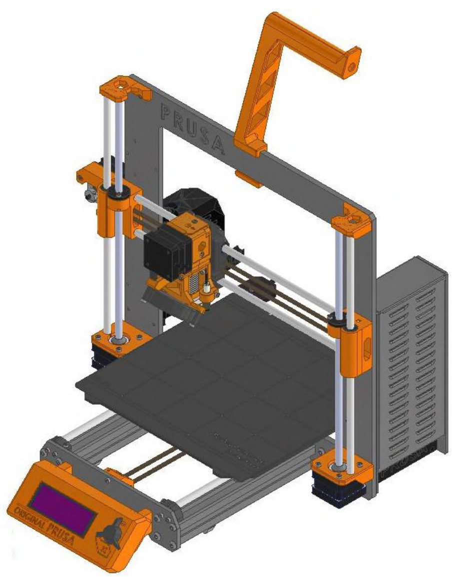


Figure 5.36: Original Prusa i3 MK3S+ 3D printer. CAD model of the printer assembly rebuilt in Autodesk Inventor 2022 from the material available in the website [12].

In this section the CAD model of a 3D printer for home use, the Original Prusa i3 MK3S+ model (Fig. 5.36), is taken as an example. Such a product constitutes an ideal test case to identify possible directions of design improvements combining the automatic semantic enrichment of the CAD model and the DFA suggestions. In the Prusa website [12], the CAD models of the single parts of the machine are accessible for autonomous printing as tessellated geometry and, moreover, their categories are provided by means of their name. Solid B-rep models have been modeled and assembled together in order to have a CAD model of the whole printer and its subassemblies, as available for industrial products.

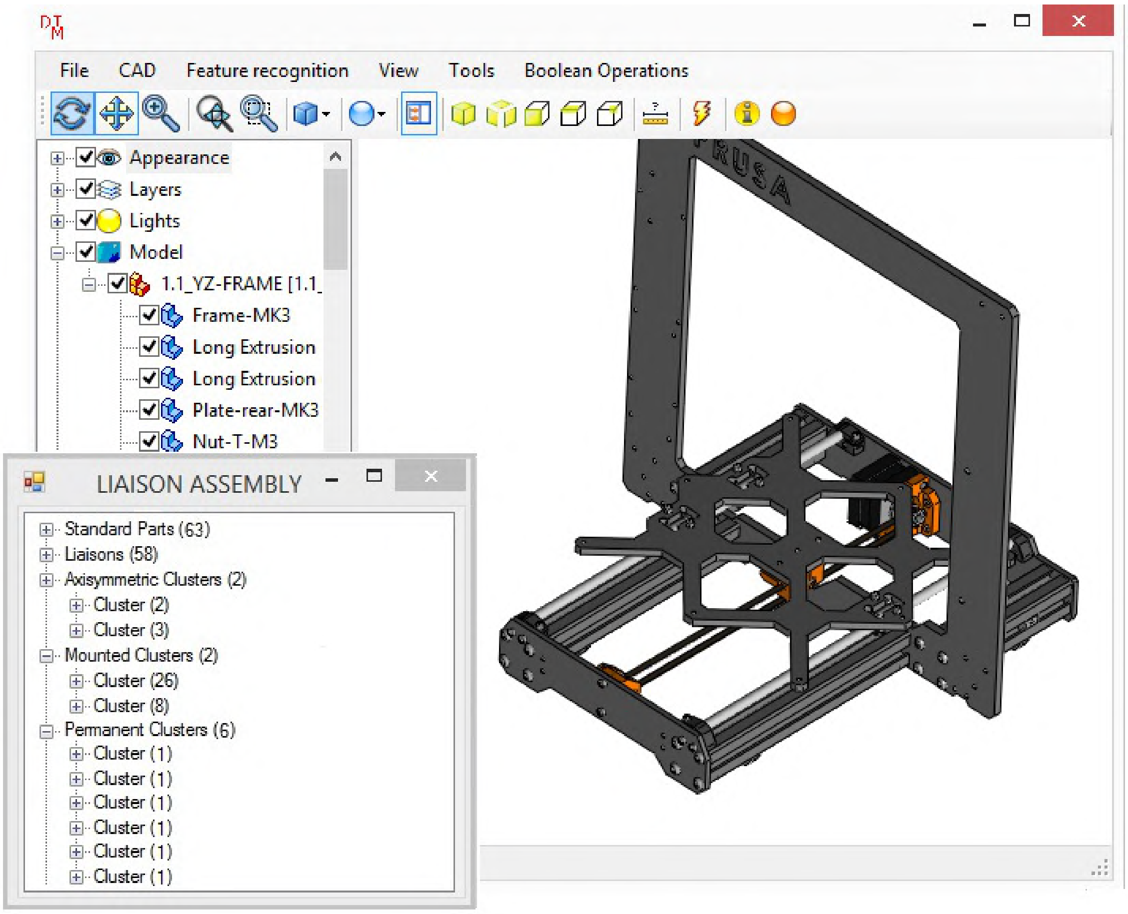


Figure 5.37: CAD model of the YZ Frame subassembly of the 3D printer in the developed system interface and visualization of the data involved in the enriched model representation based on liaisons.

In the following, a portion of the 3D printer, namely the *YZ Frame* subassembly (Fig. 5.37), is deeply investigated. The developed modules associated with the data extraction and the clustering task are applied on it to compute the enriched CAD model based on liaisons and, then, demonstrate the benefits that can be obtained by leveraging the inferred semantic information in the DFA analysis.

Table 5.3: Summary table of the information extracted for each part of the *YZ Frame* subassembly by applying the developed modules (Rp: part repetitions).

Part	Rp	Shape type	Standard Part	Holes	Grooves	Slots	Pockets
Screw-Hex-M5x16r	32	Axisymmetric	Socket Hex Head Screw M5x16	-	-	-	-
Screw-Hex-M3x10	22	Axisymmetric	Socket Hex Head Screw M3x10	-	-	-	-
Nut-Hex-M3	6	Axisymmetric	Hex Nut M3x2.5	1	-	-	-
Screw-Hex-M3x12	6	Axisymmetric	Socket Hex Head Screw M3x12	-	-	-	-
Screw-Hex-M3x30	1	Axisymmetric	Socket Hex Head Screw M3x30	-	-	-	-
Screw-Hex-M3x18	1	Axisymmetric	Socket Hex Head Screw M3x18	-	-	-	-
Nut-Hex-Lock-M3	8	Axisymmetric	Hex Nut M3x4	1	-	-	-
Nut-T-M3	2	Plate	-	1	-	-	-
Nut-Square-M3	12	Plate	-	1	-	-	-
Frame-MK3	1	Plate	-	50	-	-	-
Long Extrusion	2	Plate	-	5	-	-	-
Plate-rear-MK3	1	Plate	-	14	-	-	-
Short Extrusion	2	Plate	-	5	-	-	-
Plate-front-MK3	1	Plate	-	18	-	-	-
Foot	4	Plate	-	-	-	-	-
Y-belt-idler	1	Plate	-	4	-	-	1
Y-rod-holder	4	Plate	-	7	-	-	-
Idler-roller-623h	1	Axisymmetric	-	1	1	-	-
Y-rod	2	Axisymmetric	-	-	-	-	-
Y-belt	1	Plate	-	-	-	1	-
Y-motor	1	Plate	-	4	-	-	-
Y-motor-holder	1	Plate	-	4	-	-	-
Y-motor-pulley	1	Axisymmetric	-	2	1	-	-
Y-carriage	1	Plate	-	17	-	-	-
Bearing-LM8UU	3	Axisymmetric	-	1	4	-	-
Clip-bearing	3	Sheet Metal	-	2	-	-	-
Y-belt-tensioner	1	Plate	-	2	-	-	-
Y-belt-holder	1	Plate	-	3	-	-	-

In details, the YZ Frame subassembly is made of 122 components. For each component the geometric processing module returns the shape type and the features, along with the contact between each pair of parts. Then, the part recognition approach is carried out and 76 parts of the model are classified as standard. They are grouped in 7 sets according to subcategory and dimensions, namely 5 sets of hex head screws with different nominal diameters and lengths and 2 sets of hex nuts characterized by different heights. It is to clarify that, instead, 12 square nuts and 2 T-nuts included in the model are not recognized because belonging to standard part categories not yet treated. As for the fastening functional sets computation, 14 pairs of screw-nut are detected (i.e. each recognized nut is mounted on a screw), while the other 50 standard parts will remain single. Table 5.3 summarizes the extracted information for each part of the analyzed product.
Going ahead with the definition of the enriched CAD model representing the YZ Frame subassembly, it involves 58 liaisons generated starting from the remaining 46 parts which are not classified as standard. Finally, the clustering phase provides 2 axisymmetric clusters of respectively 2 and 3 parts, each corresponding with a rod and the bearings mounted on it, 2 mounted clusters of 26 and 8 parts, representing

the building structure and the carriage, and 6 permanent clusters each of 1 part, that are the belt and the foots which have to be singularly assembled on the mounted clusters. The list of clusters is also shown in Figure 5.37.

All the geometric and semantic information automatically extracted from the YZ Frame subassembly of the 3D printer has been given to engineers in order to accomplish a DFA analysis of the product assembly process.
Table 5.4 shows an example of the tables build applying the B&D method on the subassemblies of the printer, and then, to the final assembly [30]. The table also reports the procedures for computing DFA indices. Moreover, suggestions provided by experts for modifications of the design solution emerge to improve the assembly process and relative time. The process proved to be fast and promising for the designer.

Table 5.4: Portion of DFA analysis of the YZ Frame subassembly according to the B&D method (Rp,i: Part multiplicity; MH Code: Manual Handling Code; TH,i: Handling Time; MI Code: Manual Insertion Code; TI,i: Insertion Time; TA,i: Operation Time; TMP: Estimation for Theoretical Minimum Parts [0 - removable, 1 - non removable]).

Part	Rp,i	MH Code	TH,i[sec]	MI Code	TI,i[sec]	TA,i[sec]	Ci [€]	TMP	*Design notes*
Frame	1	95	4	06	5.5	9.5	0.12	1	*Need for equipment to keep the frame vertical during the assembly operations of aluminum extrusions*
Longer Extrusions	2	00	1.13	08	6.5	15.26	0.19	1	
Shorter Extrusions	2	00	1.13	08	6.5	15.26	0.19	0	*Consider to use 2 aluminum extrusions instead of 4. Extrusion design should be modified accordingly by possibly increasing the section from 30x30 to 30x60 millimeters*
Screw M5x16	32	10	1.5	06-38-92	16.5	288	3.60	0	*To be eliminated (see previous note)*
Front Plate	1	30	1.95	06	5.5	7.45	0.09	1	
Nut M3nE (PSU)	2	30	1.95	03	3.5	10.9	0.14	1	*To be eliminated if PSU holding system to the frame is redesigned. It is really necessary the power supply to be connected to the Frame?*
Back Plate	1	30	1.95	06	5.5	7.45	0.09	1	
Anti-vibration feet	4	20	1.8	11	5	27.2	0.34	1	
Nut M3n	2	07	2.65	03	3.5	12.3	0.15	0	*To be eliminated if threaded holes are provided in the front plate*
Nut Nylon M3nN	1	17	3.06	03	3.5	6.56	0.08	1	
Bearing Housing 623h	1	11	1.8	06	5.5	7.3	0.09	1	
Screw M32x18	1	10	1.5	38	6	7.3	0.09	1	
Y-belt-idler	1	30	1.95	08	6.5	8.45	0.11	0	*To be integrated in a redesigned front plate*
...	...	...	...	...	...	...	...	...	

5.4 Conclusions

The applications provided in this chapter seek to strengthen the effectiveness of the automatic semantic interpretation and the benefits that can be derived from the use of an enriched representation of the CAD model of a mechanical product. The availability of many engineering data and the awareness of the types of parts included in a CAD model allow the development of methodologies to address complex assembly tasks in innovative ways. The proposition is to consider and automatically detect engineering notions on parts and their relations, which are usually overlooked or manually provided, to automatically and more realistically solve some of the complex and challenging problems of the production process.

This chapter, in particular, investigates two of the tasks most studied in the literature, which are strictly connected with each other, namely the subassembly identification and the assembly sequence planning, along with the more comprehensive topic of the design for assembly analysis.

As for subassembly identification, it results effective the grouping of parts based on liaisons properties, rather than only evaluating their adjacency. An heuristic approach is implemented that returns a list of sets of components sharing some features, defined as clusters. Clusters are groundbreaking since they include more semantic information on the assembly process and mounting techniques than that provided by common subassembly, and this is promising. However, an aspect to be taken into account, for instance in view of a parallel assembly line configuration, is that not all clusters can be independently mounted and then inserted in the final assembly, because a cluster may be totally contained in another (e.g. the case of a shaft included in an external cover).

Then, for the assembly sequence planning the focus is on the axisymmetric clusters' part precedence computation, and, in particular, a collision detection algorithm is presented. It stands out from traditional methods because it is automatic and takes into account realistic scenarios such as the presence of specific fasteners or locating components also making more realistic choices in the assembly directions identification.

Finally, it has been chosen to analyze the usefulness of the extracted data for the DFA process because it is a crucial task in the industrial manufacturing, but it is almost always manually carried out and its improvement would be beneficial under various aspects. In particular, the enriched CAD model, the liaison structures and the algorithms that lead to their calculation prove to be excellent tools for the DFA automation since they can made available most of the information required for the optimization of the DFA index thus avoiding the need of manual data insertion. Thus, this final module is just a confirmation of the overall validity of the development of a standalone system and usefulness of the data extracted, from the low-level to the high-level ones. All the results can be exploited both singularly or combined with each other to overcome common issue or enhance existing methodologies.

6

CONCLUSIONS AND PERSPECTIVES

Awareness of semantic information related to mechanical components and engineering knowledge about assembly techniques have proven to be fundamental ingredients in the development of technologies to assist and simplify the automatic analysis of the manufacturing process.

The understanding of the categories of parts that compose a mechanical assembly, as well as of their principal function, the type of connections underlying two or more components, as well as the rules that characterize them, are increasingly used in research works dealing with assembly tasks and their automation. However, most of the data are usually given for granted, or manually provided by experts, making appear simple operations that are not at all immediate when they have to be faced due to lack of data, heterogeneity in the representation of the parts, and modeling errors.

In this context the general purpose of this thesis was the definition of improved and automated methodologies to address some assembly tasks in real industrial contexts, thus avoiding ideal condition hypothesis as the majority of the works present in literature.

The work has been carried out in the frame of an industrial Ph.D. project in partnership with the Italian engineering software company Hyperlean, therefore the results have been integrated into its products and already present functionalities have been exploited for the development of the proposed solutions.

First, subassembly identification and assembly sequence planning approaches were investigated and some algorithms were developed. In particular, it has been initially worked on the identification and characterization of the relations and assembly precedence among parts, as well as on the definition of contact matrices and associated graphs. Some semantic data, such as the semi-automatic identification of weldings and fasteners (i.e. by exploiting names, codes, and attributes), were leveraged to simplify and cut the CAD assembly graph to obtain meaningful subsets of parts. However, many challenges arose, especially related to the use of industrial CAD models belonging to different classes of products and the relative extraction and management of data.

To overcome these limitations and provide a reliable and standalone system not dependent on human intervention, able to realistically perform assembly tasks on complex industrial CAD models, taking into account their engineering aspects, the enrichment of the input CAD model results a crucial operation to handle.

As a consequence, a consistent part of the work is focused on the extraction of high-level information on the assembly parts and their relations only relying on the geometric and topological data included in the CAD model combined with engineering and design knowledge.

This is done through an approach that can extract essential and engineering meaningful information (e.g. the membership to a specific standard class of parts, their dimensions, the seats in which they are inserted, the type of connection, the assembly techniques, etc.) by means of rules on the geometry of parts inferred from typical engineering rules, mounting schemes, and design strategies. Then, a new data structure is defined called *liaison* to intuitively and comprehensively describe the relations between two parts of the assembly exploiting the extracted semantic data. Liaisons are the basis of the enriched product model representation that is the starting point to address the complex assembly tasks. To show the strength of the data extraction phase and the usefulness of the liaisons, innovative approaches for SI, ASP, and DFA are finally introduced.

In the following, some aspects to be improved in the different implemented modules, their possible extensions, along with future perspectives are highlighted. For the sake of clarity, the discussion is divided by considering first the aspects related to data extraction and management modules and then those related to the use of data and applications.

Data extraction and management

Some effort can be done to improve or extend the data extraction modules under several aspects. The most relevant are now pointed out.

- As for the part recognition, the list of the considered categories and subcategories of standard parts is mainly defined according to practical experience.

That is to say, it reflects the most widespread types of fasteners and locating components that are generally employed in mechanical assemblies and that are encountered in the analyzed CAD models. Anyhow, it is the author's proposition to undertake the extension of the classes treated. This requires the analysis of the elements of the new categories and the specification of rules for the identification of their salient characteristics, in terms of shape and dimensions dictated by international standards, as well as context rules.
In addition, to avoid possible overlapping or contradictory rules for standard parts, the recognition rules would be connected to the product sector considered, i.e. it could be useless or even produce bad results applying rules related to standard components for the wood sector to mechanical parts.

- Another aspect that would be very beneficial in the semantic enrichment of a CAD model is the extension of the part recognition to not standard parts. In particular, this need arises because, in the course of the work, some components have been mentioned whose awareness is important in the interpretation of the assembly parts as well as in the simplification of the product model representation. Among these, gears and bearings, as well as springs, pulleys, or belts. Some effort has already been done in this direction, implementing some not rigorous algorithms for the gears and bearings recognition, but the formal extension of the recognition approach has to be addressed. This is possible since these components have distinctive shape features and follow typical arrangements within the assembly.

- The rule-based approach used in part recognition has proven to be promising. On the one hand, because of the considered engineering context which is well structured where parts have to satisfy well established laws; on the other hand, it is easily extendable to consider other standard part classes and a generalized formulation for not standard components can be proposed. However, in the future, the opportunity of using different knowledge formalization methodologies, for example by adopting an approach based on ontology, will be evaluated and the performances will be compared.

- It was shown that the recognition of standard parts sometimes fails in the context analysis step because the parts are positioned in seats not fitting into standard features, but these are however engineering reasonable scenarios. As a consequence, an extension of the feature recognition approach would be beneficial.

- The introduction of the liaison data structure suggests an innovative and intuitive way to represent the relations between two parts, taking into account many details usually ignored but crucial in the assembly understanding and processing. A future improvement concerning liaisons consists in their generalization admitting that the parts underlying a liaison are components. This would allow a further simplification of the product data structure, reducing the number of items involved in the liaison list. Moreover, it would be useful to analyze and treat as a single object group of parts that are known to be a stable cluster/subassembly (e.g. welded clusters, functional sets, etc.).

Use of data and applications

Such a thorough analysis of CAD models and the availability of semantic data allowed the development of innovative approaches to support complex assembly tasks. However, concerning the developed modules some extensions can be provided, and further applications can be found.

- The introduction of the concept of cluster results very promising from the model semantic interpretation point of view. Clusters, in fact, allow the detection of groups of parts that describe a precise mechanical operation or an engineering concept. These also suggest relevant information that can be exploited in the (dis-)assembly planning, such as possible directions of movement, as well as some precedences between the clusters themselves or between the clusters and certain components. However, the next improvements will focus on the assignation of a part to a cluster rather than another in the refinement phase, which actually is not always optimal. In this regard, the definition of more accurate rules possibly taking into consideration the specific mechanical assembly class treated (e.g. engines, mechanical arms, valves, etc.) may be the solution. This would be also supported by the study of new heuristics to take into account and consider according to the type of product analyzed.

  Furthermore, an interesting extension, that would be more demanding, is the identification of subassemblies, as they are defined in literature, starting from the knowledge of clusters. This would require the analysis of the clusters with respect to each other and their aggregation in order to obtain stable components.

- The application of the assembly sequence planning approach to single clusters results an affordable solution, especially in the collision analysis, since engineering meaningful information can be exploited to overcome some existing limitations. So far, the axisymmetric clusters have been studied, due to their particular features and their overall reliable detection. Future works expect to extend the methodology to the other type of clusters, starting from the mounted one, where the information given by fasteners and mountings would be crucial to identify the possible directions of movement.

- The combination of the processes carried out and the strategies adopted to deal with industrial CAD models and to address different parts' representation conventions and modeling practice can be promising from the perspective of the design for manufacturing (DFM) tasks, besides in the already mentioned DFA field. In particular, the results obtained, both those correct, the false positive, and the false negative can be leveraged as starting point for the detection of modeling errors, displacement of parts and features, and missing components. For example, the identification of parts not included in any liaison suggests that there are parts with no contacts (neither by surface analysis, nor by volumes intersection), and this can be exploited to detect modeling errors and/or incomplete assemblies. Similarly, the analysis of the standard parts not included in liaisons allows to infer the missing of holes/pockets or their not alignment. Also, information about the rejected candidate standard parts could provide hints of missing parts or wrong features.

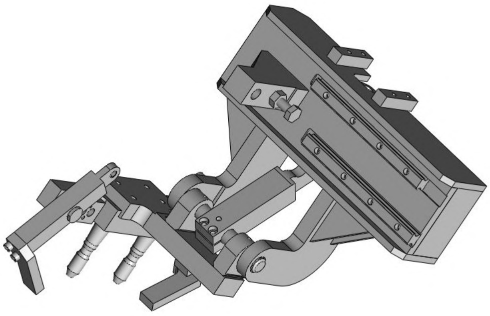
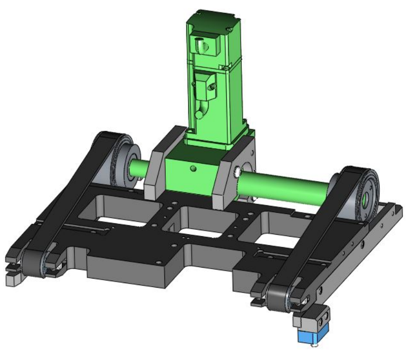
- Finally, the work done so far fits well and can find application in the Industry 4.0 context. On the one hand, mixed reality technologies can facilitate the visualization and inspection of the results. Systems able to label parts, indicate assembly techniques involved in the connections between two components, as well as highlight specific clusters in a 3D simulation or in an augmented environment would be beneficial in different fields, such as education, product maintenance, and product recycling. On the other hand, the automatic detection of groups of parts independent from one another and the knowledge of assembly precedence and possible mounting sequences can be given as input to enhance parallel assembly and human-robot collaboration.

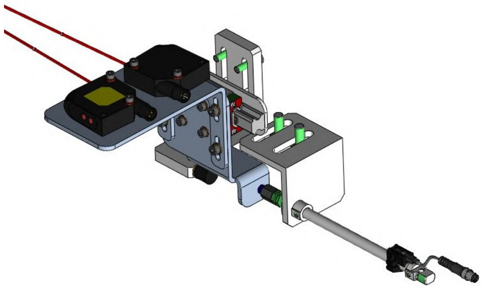
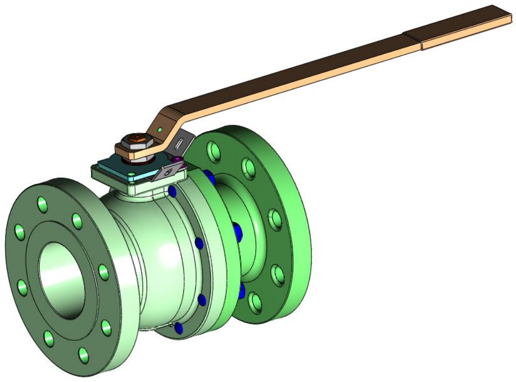
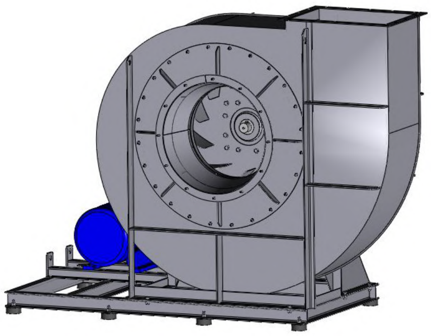
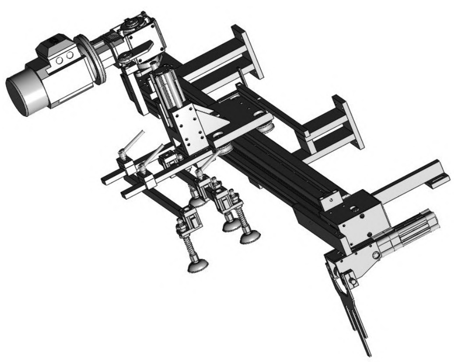
Appendices

A

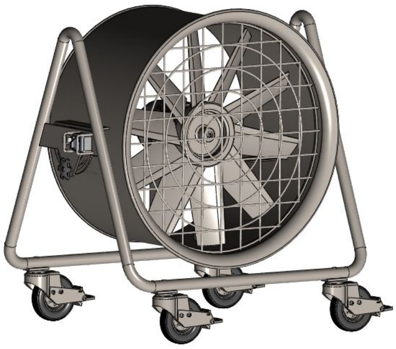
THE DATASET

▶ **Industrial CAD models supplied by engineering companies:**

M1)	Gripper mechanism		66 parts
M2)	Bridge bearing		70 parts
M3)	Belts drive mechanism		55 parts

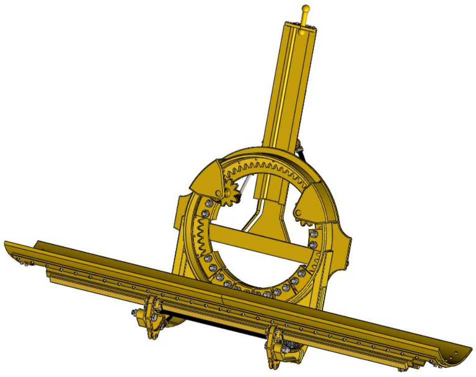
M4)	Laser assembly		62 parts
M5)	Ball valve		35 parts
M6)	Fan assembly		325 parts
M7)	Linear axis for automation		199 parts

M8) Industrial drum fan



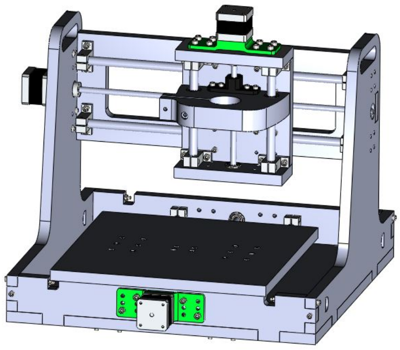
290 parts

M9) Agricultural steel-work assembly



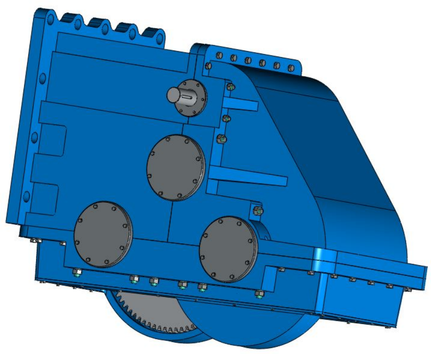
455 parts

M10) Cartesian slider

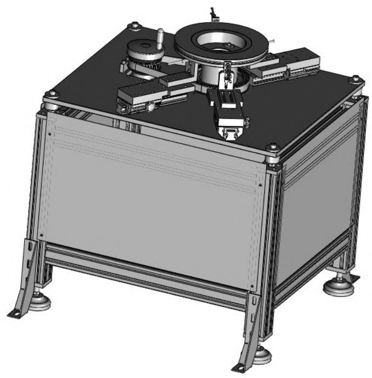
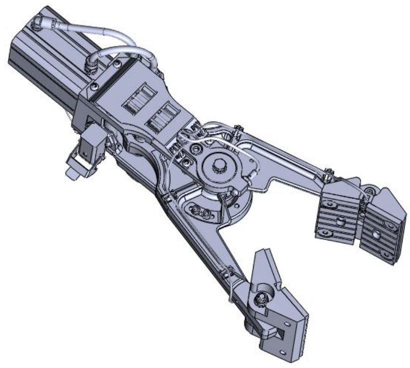
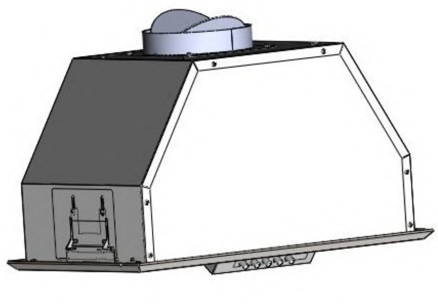
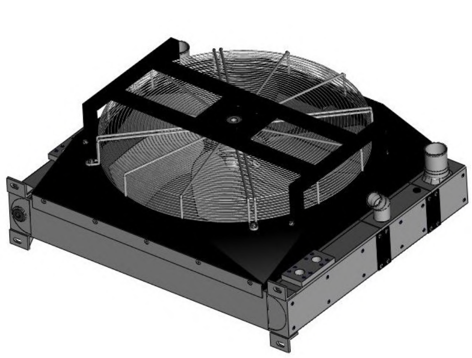


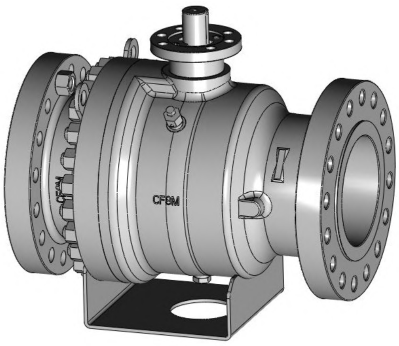
237 parts

M11) Gearbox

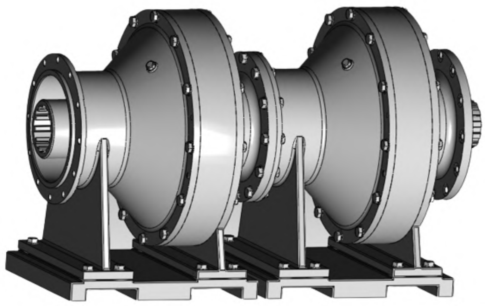


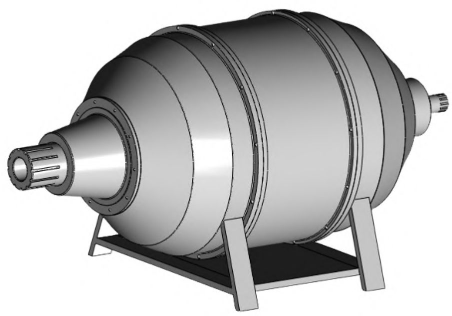
426 parts

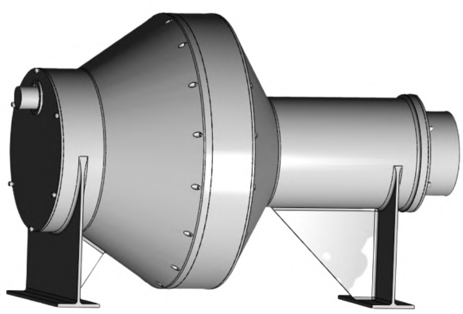
M12)	Disks break test bed		153 parts
M13)	Robotic gripper		318 parts
M14)	Cooker hood		106 parts
M15)	Radiator		201 parts

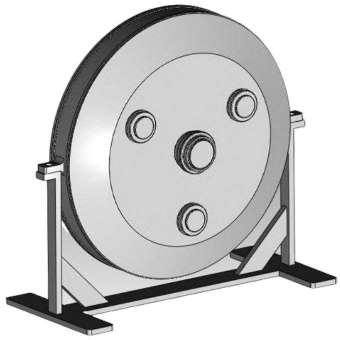
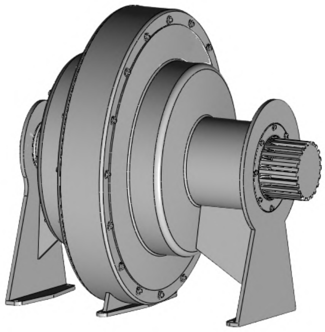
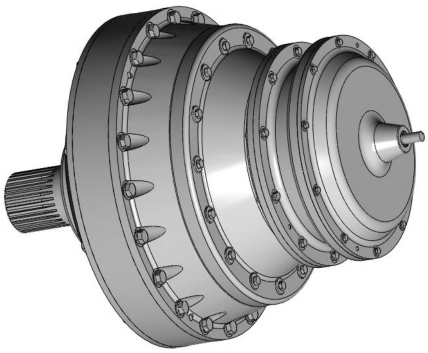
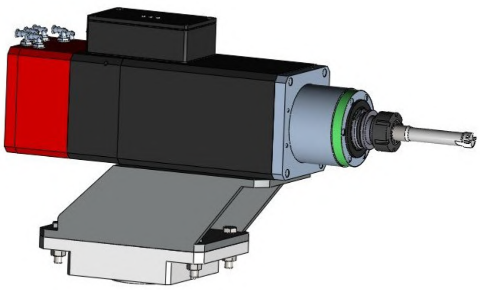
M16) Ball valve 2  74 parts

▶ **CAD models selected from online repositories:**

M17) Rotor wind turbine (3D assembly repository [7])  378 parts

M18) Rotor wind turbine 2 (3D assembly repository [7])  46 parts

M19) Rotor wind turbine 3 (3D assembly repository [7])  167 parts

M20)	Rotor wind turbine 4 (3D assembly repository [7])		92 parts
M21)	Axial reducer (3D assembly repository [7])		306 parts
M22)	Planetary gearbox (GrabCAD [8])		490 parts
M23)	Spindle assembly (GrabCAD [8])		95 parts

M24)	Gearbox 3 (GrabCAD [8])		107 parts
M25)	3D printer YZ frame (Prusa 3D web page [12])		122 parts

Acknowledgments

First, I would like to express my deepest gratitude to the Institute of Applied Mathematics and Information Technology (IMATI) of Genoa and the company Hyperlean of Ancona that made possible the accomplishment of this research project and funded it under the CNR-Confindustria agreement.

Special thanks to Dr. Franca Giannini and Dr. Marina Monti, my supervisors at IMATI, who after following me as a research fellow decided to support and guide me in this Ph.D., and continue to do so after. A big thank also to Prof. Roberto Raffaeli, my supervisor at Hyperlean, that was fundamental to introduce me and try to make me understand the world of mechanical engineering.

I would like to extend my sincere thanks to all members of the IMATI institute, and in particular to the "IMATTI" group, for the working, eating, drinking, and laughing times spent together. Finally, thanks to my family and my parents.

Bibliography

[1] Autodesk Inventor® software by Autodesk Inc. https://www.autodesk.com/products/inventor/, last accessed November 02, 2022.

[2] CATIA® software by Dassault Systemes Corporation. https://www.3ds.com/products-services/catia/, last accessed November 02, 2022.

[3] Creo® software by Parametric Technology Corporation. https://www.ptc.com/, last accessed November 02, 2022.

[4] SOLIDWORKS® software by Dassault Systemes Corporation. https://www.solidworks.com/, last accessed November 02, 2022.

[5] La Bulloneria s.r.l. https://www.labulloneria.it/products/1/2196/it/viti_a_testa_cilindrica_con_esagono, last accessed November 06, 2022.

[6] Rhinoceros® software by Robert McNeel & Associates. https://www.rhino3d.com/opennurbs/, last accessed November 06, 2022.

[7] 3D Assembly Repository by IMATI-CNR. http://3dassemblyrepository.ge.imati.cnr.it, last accessed November 16, 2022.

[8] GrabCAD by Stratasys Inc. https://grabcad.com/, last accessed November 16, 2022.

[9] Partclassifier dataset by IMATI-CNR . http://partsclassifier.ge.imati.cnr.it, last accessed November 16, 2022.

[10] PARTcommunity by CADENAS . https://b2b.partcommunity.com/, last accessed November 16, 2022.

[11] TraceParts by TraceParts S.A.S. . https://www.traceparts.com/, last accessed November 16, 2022.

[12] Original Prusa i3 MK3S+ web page. https://www.prusa3d.com/product/original-prusa-i3-mk3s-3d-printer-3/, last accessed November 24, 2022.

[13] DFMA® software by Boothroyd Dewhurst Inc. https://www.dfma.com/software/dfma.asp, last accessed October 28, 2022.

[14] DFMPro® software by HCL Technologies. https://dfmpro.com/, last accessed October 28, 2022.

[15] S. Abdul-Ghafour, P. Ghodous, B. Shariat, E. Perna, and F. Khosrowshahi. Semantic interoperability of knowledge in feature-based CAD models. *Computer-Aided Design*, 56:45–57, 2014.

[16] D. Agrawal, S. Kumara, and D. Finke. Automated assembly sequence planning and subassembly detection. In *IIE Annual Conference. Proceedings*, pages 781–788. Institute of Industrial and Systems Engineers (IISE), 2014.

[17] A. Angrish, B. Craver, and B. Starly. "FabSearch": A 3D CAD Model-Based Search Engine for Sourcing Manufacturing Services. *Journal of Computing and Information Science in Engineering*, 19(4), 2019.

[18] B. Babic, N. Nesic, and Z. Miljkovic. A review of automated feature recognition with rule-based pattern recognition. *Computers in industry*, 59(4):321–337, 2008.

[19] M. Bahubalendruni, U. Sudhakar, and K. Lakshmi. Subassembly detection and optimal assembly sequence generation through elephant search algorithm. *International Journal of Mathematical, Engineering and Management Sciences*, 4(4):998–1007, 2019.

[20] M. R. Bahubalendruni and B. B. Biswal. A review on assembly sequence generation and its automation. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(5):824–838, 2016.

[21] M. R. Bahubalendruni and B. B. Biswal. An efficient stable subassembly identification method towards assembly sequence generation. *National Academy science letters*, 41(6):375–378, 2018.

[22] G. Bala Murali, B. Deepak, M. Raju Bahubalendruni, and B. Biswal. Optimal assembly sequence planning towards design for assembly using simulated annealing technique. In *International conference on research into design*, pages 397–407. Springer, 2017.

[23] D. F. Baldwin, T. E. Abell, M.-C. Lui, T. L. De Fazio, and D. E. Whitney. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. *IEEE transactions on robotics and automation*, 7(1):78–94, 1991.

[24] R. Barbau, S. Krima, S. Rachuri, A. Narayanan, X. Fiorentini, S. Foufou, and R. D. Sriram. OntoSTEP: Enriching product model data using ontologies. *Computer-Aided Design*, 44(6):575–590, 2012.

[25] I. Belhadj, M. Trigui, and A. Benamara. Subassembly generation algorithm from a CAD model. *The International Journal of Advanced Manufacturing Technology*, 87(9):2829–2840, 2016.

[26] D. Ben-Arieh and B. Kramer. Computer-aided process planning for assembly: generation of assembly operations sequence. *The international journal of production research*, 32(3):643–656, 1994.

[27] R. Ben Hadj, M. Trigui, and N. Aifaoui. Toward an integrated CAD assembly sequence planning solution. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 229(16):2987–3001, 2015.

[28] S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno. Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design*, 38(9):1002–1019, 2006. Shape Similarity Detection and Search for CAD/CAE Applications.

[29] T. Bloch and R. Sacks. Comparing machine learning and rule-based inferencing for semantic enrichment of bim models. *Automation in Construction*, 91:256–272, 2018.

[30] B. Bonino, F. Giannini, M. Monti, R. Raffaeli, and G. Berselli. Geometric analysis of product cad models to support design for assembly. In *International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing*, pages 698–710. Springer, 2023.

[31] G. Boothroyd, P. Dewhurst, and W. A. Knight. *Product design for manufacture and assembly*. CRC press, 2010.

[32] F. Boussuge, J. Léon, S. Hahmann, L. Fine, and I. R. Alpes. An analysis of the transformation requirements for digital mock-ups of structural assembly simulations. *Civil-Comp Proceedings*, 100, 2012.

[33] J. D. Camba, M. Contero, and P. Company. Parametric CAD modeling: An analysis of strategies for design reusability. *Computer-Aided Design*, 74:18–31, 2016.

[34] Y. Cao, X. Kou, and S. Cao. A sub-assembly identification algorithm for assembly sequence planning. In *2015 International Industrial Informatics and Computer Engineering Conference*. Atlantis Press, 2015.

[35] A. Cardone, S. K. Gupta, and M. Karnik. A survey of shape similarity assessment algorithms for product design and manufacturing applications. *J. Comput. Inf. Sci. Eng.*, 3(2):109–118, 2003.

[36] W.-C. Chen, P.-H. Tai, W.-J. Deng, and L.-F. Hsieh. A three-stage integrated approach for assembly sequence planning using neural networks. *Expert Systems with Applications*, 34(3):1777–1786, 2008.

[37] M.-C. Chiu and C.-H. Chu. Review of sustainable product design from life cycle perspectives. *International Journal of Precision Engineering and Manufacturing*, 13(7):1259–1272, 2012.

[38] Y.-K. Choi, D. M. Lee, and Y. B. Cho. An approach to multi-criteria assembly sequence planning using genetic algorithms. *The International Journal of Advanced Manufacturing Technology*, 42(1):180–188, 2009.

[39] A. Cicek and M. Gülesin. A part recognition based computer aided assembly system. *Computers in industry*, 58(8-9):733–746, 2007.

[40] Y. Cohen, M. Faccio, F. Pilati, and X. Yao. Design and management of digital manufacturing and assembly systems in the Industry 4.0 era. *The International Journal of Advanced Manufacturing Technology*, 105(9):3565–3577, 2019.

[41] M. Crnjac, I. Veža, and N. Banduka. From concept to the introduction of Industry 4.0. *International Journal of Industrial Engineering and Management*, 8(1):21, 2017.

[42] T. De Fazio and D. Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Journal on Robotics and Automation*, 3(6):640–658, 1987.

[43] L. H. De Mello and A. C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. In *1989 IEEE International Conference on Robotics and Automation*, pages 56–57. IEEE Computer Society, 1989.

[44] L. H. De Mello and A. C. Sanderson. AND/OR graph representation of assembly plans. *IEEE Transactions on robotics and automation*, 6(2):188–199, 1990.

[45] L. H. De Mello and A. C. Sanderson. Representations of mechanical assembly sequences. *IEEE Trans. Robotics Autom.*, 7(2):211–227, 1991.

[46] B. Deepak, G. Bala Murali, M. R. Bahubalendruni, and B. Biswal. Assembly sequence planning using soft computing methods: a review. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 233(3):653–683, 2019.

[47] J. Dekhtiar, A. Durupt, M. Bricogne, B. Eynard, H. Rowson, and D. Kiritsis. Deep learning for big data applications in CAD and PLM – Research review, opportunities and case study. *Computers in Industry*, 100:227 – 243, 2018.

[48] P. Di Stefano, F. Bianconi, and L. Di Angelo. An approach for feature semantics recognition in geometric models. *Computer-Aided Design*, 36(10):993–1009, 2004.

[49] G. Dini and M. Santochi. Automated sequencing and subassembly detection in assembly planning. *CIRP Annals*, 41(1):1 – 4, 1992.

[50] T. Dong, R. Tong, L. Zhang, and J. Dong. A knowledge-based approach to assembly sequence planning. *The International Journal of Advanced Manufacturing Technology*, 32(11-12):1232–1244, 2007.

[51] S. S. Fernández-Miranda, M. Marcos, M. E. Peralta, and F. Aguayo. The challenge of integrating industry 4.0 in the degree of mechanical engineering. *Procedia manufacturing*, 13:1229–1236, 2017.

[52] G. Foucault and J.-C. Léon. Enriching assembly CAD models with functional and mechanical informations to ease cae. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 44113, pages 341–351, 2010.

[53] G. Foucault, A. Shahwan, J.-C. Léon, and L. Fine. What is the content of a DMU? analysis and proposal of improvements. *AIP-PRIMECA 2011 - Produits, Procédés et Systémes Industriels : intégration Réel-Virtuel*, Le Mont Dore, France, 2011.

[54] Z. Guo, D. Zhou, Q. Zhou, X. Zhang, J. Geng, S. Zeng, C. Lv, and A. Hao. Applications of virtual reality in maintenance during the industrial product lifecycle: A systematic review. *Journal of Manufacturing Systems*, 56:525–538, 2020.

[55] R. B. Hadj, I. Belhadj, M. Trigui, and N. Aifaoui. Assembly sequences plan generation using features simplification. *Advances in Engineering Software*, 119:1–11, 2018.

[56] D. Halperin, J.-C. Latombe, and R. H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26(3):577–601, 2000.

[57] A. Hamrol, J. Gawlik, and J. Sładek. Mechanical engineering in industry 4.0. *Management and Production Engineering Review*, 10:14–28, 2019.

[58] Z. Han, R. Mo, H. Yang, and L. Hao. Structure-function correlations analysis and functional semantic annotation of mechanical CAD assembly model. *Assembly Automation*, 2019.

[59] L. Hao, R. Mo, Z. Han, and B. Wei. Functional semantics annotation of assembly model using the fusion of bag of relationships model and spectral technology. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 13(4):JAMDSM0082–JAMDSM0082, 2019.

[60] K.-M. Hu, B. Wang, J.-H. Yong, and J.-C. Paul. Relaxed lightweight assembly retrieval using vector space model. *Computer-Aided Design*, 45(3):739–750, 2013.

[61] S. J. Hu, X. Zhu, H. Wang, and Y. Koren. Product variety and manufacturing complexity in assembly systems and supply chains. *CIRP annals*, 57(1):45–48, 2008.

[62] C. Y. Ip and W. C. Regli. A 3D object classifier for discriminating manufacturing processes. *Computers & Graphics*, 30(6):903–916, 2006.

[63] C. Y. Ip, W. C. Regli, L. Sieger, and A. Shokoufandeh. Automated learning of model classifications. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 322–327, 2003.

[64] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, 2005.

[65] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani. Developing an engineering shape benchmark for CAD models. *Computer-Aided Design*, 38(9):939–953, 2006.

[66] S. Jayanti, Y. Kalyanaraman, and K. Ramani. Shape-based clustering for 3D CAD objects: A comparative study of effectiveness. *Computer-Aided Design*, 41(12):999–1007, 2009.

[67] C. Jian, L. Liang, K. Qiu, and M. Zhang. An improved memory networks based product model classification method. *International Journal of Computer Integrated Manufacturing*, 34(3):293–306, 2021.

[68] P. Jiménez. Survey on assembly sequencing: a combinatorial and geometrical perspective. *Journal of Intelligent Manufacturing*, 24(2):235–250, 2013.

[69] J.-G. Kang and P. Xirouchakis. Disassembly sequencing for maintenance: a survey. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 220(10):1697–1716, 2006.

[70] J. Ko, E. Nazarian, H. Wang, and J. Abell. An assembly decomposition model for subassembly planning considering imperfect inspection to reduce assembly defect rates. *Journal of Manufacturing Systems*, 32(3):412 – 416, 2013. Assembly Technologies and Systems.

[71] X. Kou, Y. Cao, Q. Wang, and H. Qiao. Sub-assembly recognition algorithm and performance analysis in assembly sequence planning. *The International Journal of Advanced Manufacturing Technology*, pages 1–11, 2019.

[72] A. F. Lambert and S. M. Gupta. *Disassembly modeling for assembly, maintenance, reuse and recycling*. CRC press, 2004.

[73] A. J. Lambert. Disassembly sequencing: a survey. *International Journal of Production Research*, 41(16):3721–3759, 2003.

[74] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014.

[75] D. Lee, J. Kang, and P. Xirouchakis. Disassembly planning and scheduling: review and further research. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 215(5):695–709, 2001.

[76] J. Li, Q. Wang, P. Huang, and H. Shen. A novel connector-knowledge-based approach for disassembly precedence constraint generation. *The International Journal of Advanced Manufacturing Technology*, 49(1):293–304, 2010.

[77] M. Li, Y. Zhang, J. Fuh, and Z. Qiu. Toward effective mechanical design reuse: CAD model retrieval based on general and partial shapes. *Journal of Mechanical Design*, 131(12), 2009.

[78] Z.-B. Liu, S.-H. Bu, K. Zhou, S.-M. Gao, J.-W. Han, and J. Wu. A survey on partial retrieval of 3D shapes. *Journal of Computer Science and Technology*, 28:836–851, 2013.

[79] L. E. S. Ltd. Design for manufacture and assembly practitioners manual, version 10, 1993.

[80] K. Lupinetti, F. Giannini, M. Monti, and J.-P. Pernot. Identification of functional components in mechanical assemblies. *Procedia CIRP*, 60:542–547, 2017.

[81] K. Lupinetti, F. Giannini, M. Monti, and J.-P. Pernot. A 3D CAD Assembly Benchmark. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019.

[82] K. Lupinetti, F. Giannini, M. Monti, M. Rucco, and J.-P. Pernot. Identification of functional sets in mechanical assembly models. In *International Conference on Innovative Design and Manufacturing*, pages 1–6, 2017.

[83] K. Lupinetti, J.-P. Pernot, M. Monti, and F. Giannini. Content-based CAD assembly model retrieval: Survey and future challenges. *Computer-Aided Design*, 113:62–81, 2019.

[84] G. Lyu, X. Chu, and D. Xue. Product modeling from knowledge, distributed computing and lifecycle perspectives: A literature review. *Computers in Industry*, 84:1–13, 2017.

[85] B. Manda, P. Bhaskare, and R. Muthuganapathy. A convolutional neural network approach to the classification of engineering models. *IEEE Access*, 9:22711–22723, 2021.

[86] R. M. Marian, L. H. Luong, and K. Abhary. A genetic algorithm for the optimisation of assembly sequences. *Computers & Industrial Engineering*, 50(4):503–527, 2006.

[87] E. A. Nasr, A. A. Khan, A. M. Alahmari, and H. Hussein. A feature recognition system using geometric reasoning. *Procedia CIRP*, 18:238–243, 2014.

[88] A. Neb and J. Göke. Generation of assembly restrictions and evaluation of assembly criteria from 3D assembly models by collision analysis. *Procedia CIRP*, 97:33–38, 2021.

[89] S. Nzetchou, A. Durupt, S. Remy, and B. Eynard. Semantic enrichment approach for low-level CAD models managed in PLM context: Literature review and research prospect. *Computers in Industry*, 135:103575, 2022.

[90] Y. Peilin, C. Xiaonan, and P. Xuanming. Study on connections and subassemblies in assembly. *Journal of Xi'an Jiaotong University*, 11, 2004.

[91] J.-P. Pernot, F. Giannini, and C. Petton. Thin part identification for CAD model classification. *Engineering Computations*, 32(1):62–85, 2015.

[92] S. Ramnath, P. Haghighi, A. Venkiteswaran, and J. J. Shah. Interoperability of CAD geometry and product manufacturing information for computer integrated manufacturing. *International Journal of Computer Integrated Manufacturing*, 33(2):116–132, 2020.

[93] M. F. F. Rashid, W. Hutabarat, and A. Tiwari. A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. *The International Journal of Advanced Manufacturing Technology*, 59(1):335–349, 2012.

[94] M. Rucco, F. Giannini, K. Lupinetti, and M. Monti. A methodology for part classification with supervised machine learning. *AI EDAM*, 33(1):100–113, 2019.

[95] J. J. Shah, D. Anderson, Y. S. Kim, and S. Joshi. A discourse on geometric feature recognition from CAD models. *Journal of Computing and Information Science in Engineering*, 1(1):41–51, 2001.

[96] A. Shahwan, J.-C. Léon, G. Foucault, M. Trlin, and O. Palombi. Qualitative behavioral reasoning from components' interfaces to components' functions for DMU adaption to FE analyses. *Computer-Aided Design*, 45(2):383–394, 2013. Solid and Physical Modeling 2012.

[97] H. Shan, S. Zhou, and Z. Sun. Research on assembly sequence planning based on genetic simulated annealing algorithm and ant colony optimization algorithm. *Assembly Automation*, 2009.

[98] X. Shi, X. Tian, G. Wang, D. Zhao, and M. Zhang. Semantic-based subassembly identification considering non-geometric structure attributes and assembly process factors. *The International Journal of Advanced Manufacturing Technology*, 110(1):439–455, 2020.

[99] C. Sinanoğlu and H. R. Börklü. An assembly sequence-planning system for mechanical parts using neural network. *Assembly Automation*, 2005.

[100] G. C. Smith and S. S.-F. Smith. An enhanced genetic algorithm for automated assembly planning. *Robotics and Computer-Integrated Manufacturing*, 18(5-6):355–364, 2002.

[101] S. S.-F. Smith, G. C. Smith, and X. Liao. Automatic stable assembly sequence generation and evaluation. *Journal of Manufacturing systems*, 20(4):225, 2001.

[102] Q. Su. Computer aided geometric feasible assembly sequence planning and optimizing. *The International Journal of Advanced Manufacturing Technology*, 33(1):48–57, 2007.

[103] A. K. Swain, D. Sen, and B. Gurumoorthy. Extended liaison as an interface between product and process model in assembly. *Robotics and Computer-Integrated Manufacturing*, 30(5):527–545, 2014.

[104] S. Tao and M. Hu. A contact relation analysis approach to assembly sequence planning for assembly models. *Computer-Aided Design and Applications*, 14(6):720–733, 2017.

[105] M. Trigui, I. Belhadj, and A. Benamara. Disassembly plan approach based on subassembly concept. *The International Journal of Advanced Manufacturing Technology*, 90(1):219–231, 2017.

[106] M. Trigui, R. BenHadj, and N. Aifaoui. An interoperability CAD assembly sequence plan approach. *The International Journal of Advanced Manufacturing Technology*, 79(9):1465–1476, 2015.

[107] R. Viganò and G. Osorio Gómez. Automatic assembly sequence exploration without precedence definition. *International Journal on Interactive Design and Manufacturing*, 7(2):79–89, 2013.

[108] H. Wang, Q. Peng, J. Zhang, and P. Gu. Selective disassembly planning for the end-of-life product. *Procedia Cirp*, 60:512–517, 2017.

[109] Y. Wang and J. Liu. Subassembly identification for assembly sequence planning. *The International Journal of Advanced Manufacturing Technology*, 68(1):781–793, 2013.

[110] Y. Wang, J. Liu, and L. Li. Assembly sequences merging based on assembly unit partitioning. *The International Journal of Advanced Manufacturing Technology*, 45(7-8):808–820, 2009.

[111] J. Watson and T. Hermans. Assembly planning by subassembly decomposition using blocking reduction. *IEEE Robotics and Automation Letters*, 4(4):4054–4061, 2019.

[112] L. Wei and H. Yuanjun. Representation and retrieval of 3D CAD models in parts library. *The International Journal of Advanced Manufacturing Technology*, 36(9):950–958, 2008.

[113] K. D. Willis, P. K. Jayaraman, H. Chu, Y. Tian, Y. Li, D. Grandi, A. Sanghi, L. Tran, J. G. Lambourne, A. Solar-Lezama, and W. Matusik. JoinABLe: Learning bottom-up assembly of parametric CAD joints. *arXiv preprint arXiv:2111.12772*, 2021.

[114] G. Zachmann. Rapid collision detection by dynamically aligned DOP-trees. In *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No. 98CB36180)*, pages 90–97. IEEE, 1998.

[115] C. Zhang, G. Zhou, Q. Lu, and F. Chang. Generating significant subassemblies from 3D assembly models for design reuse. *International Journal of Production Research*, 56(14):4744–4761, 2018.

[116] N. Zhang, Z. Liu, C. Qiu, J. Cheng, and J. Tan. Disassembly sequence planning using a fast and effective precedence-based disassembly subset-generation method. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 234(3):513–526, 2020.

[117] P. Zheng, H. Wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, and X. Xu. Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. *Frontiers of Mechanical Engineering*, 13(2):137–150, 2018.