



Feature Recognition for Structural Aerospace Sheet Metal Parts

Seyedmorteza Ghaffarishahri¹ , Louis Rivest² 

¹École de technologie supérieure (ÉTS), Université du Québec,
seyedmorteza.ghaffarishahri.1@ens.etsmtl.ca

²École de technologie supérieure (ÉTS), Université du Québec, louis.rivest@etsmtl.ca

Corresponding author: Seyedmorteza Ghaffarishahri, seyedmorteza.ghaffarishahri.1@ens.etsmtl.ca

Abstract. With the advances in the aerospace industry, specialized tools, e.g. specialized modelling tools for aerospace sheet metal, have been developed to help at various stages of the product lifecycle. Although structural sheet metal parts form a significant portion of airplanes, there is no specialized automated feature recognition (AFR) method dedicated to them. AFR provides unparalleled contributions to various tasks in product lifecycle management, e.g. computer aided process planning, data retrieval and model difference identification. Despite the presence of a number of AFR methods for sheet metal parts, none of them are tuned to recognize the design semantics of the aerospace industry. This work proposes the first AFR method to recognize aerospace sheet metal features and design semantics to elevate the level of abstraction of the information from 3D STEP models. The proposed approach is to first preprocess the 3D STEP model in order to classify the topological elements of the boundary model (B rep model) and create relevant novel face sets and subtypes of faces, face boundaries and edges. Then, rule-based steps are used to recognize aerospace sheet metal features. The extracted features are described by their geometry, their relationship with other features and their pertinent parameters. As a result, the engineering semantics of 3D B-rep models of aerospace sheet metal parts are extracted and could be used for many applications like design reuse and model comparison.

Keywords: Feature definition, aerospace sheet metal, feature recognition, boundary representation, STEP.

DOI: <https://doi.org/10.14733/cadaps.2020.16-43>

1 INTRODUCTION

Numerous CAD solutions, including CATIA [1, 2], SOLIDWORKS [3], IronCAD [4], NX 12 for Design [5], Solid Edge [6], BricsCAD Platinum [7], Creo Parametric [8], TurboCAD [9], Onshape [10], Alibre Design [11], Autodesk Inventor [12] and Ansys SpaceClaim [13] provide feature-based CAD to model sheet metal parts. Amongst the aforementioned CAD solutions, CATIA and NX 12 for Design provide modelling tools specialized for aerospace sheet metal (ASM) design. These

modelling tools provide exclusive features that are commonly used in the design of ASM parts. Such exclusive features include curved flanges, joggles and stringer cutouts, to name a few. In addition, these feature-based ASM modelling tools are specialized to facilitate the design process in the aerospace industry.

The ASM models themselves, however, are not necessarily exchanged in native formats to avoid cross-platform and cross-version conflicts. Instead, they are exchanged via their boundary representation (B-rep) standardized by using Standard for the Exchange of Product (STEP) model data, one of the most effective and efficient information exchange methods [14]. However, the specific features commonly used in the design of ASM parts require the use of feature-based ASM design tools, which cannot be exchanged via STEP. An ASM feature-recognition method for STEP models – a B-rep model – could therefore significantly raise the level of design information exchanged via STEP. This work aims to propose an automated feature recognition (AFR) method for recognizing features from ASM B-rep models based on the definition of non-facetted manifold solid B-rep in ISO 10303-42 [15]. Our long-term goal is to use features to elevate the level of information provided for downstream applications such as 3D model difference identification. This way, we will be able to provide the user with, for example, the difference between two models in terms of their flange length.

2 PREVIOUS WORKS

While numerous feature recognition/extraction methods have been proposed over the last decades [16-20], the ones dedicated to sheet metal features are comparatively far less numerous. While there are AFR methods for sheet metal models such as the one proposed by Nnaji et al. [21], the majority of the methods can be divided into shear AFR methods, generic deformation AFR methods and freeform AFR methods. Being far from the scope of our work, the previous work on freeform AFR methods [22-24] is not reviewed here. Jagirdar et al. [25] proposed an AFR method for identifying shear features based on geometric and topological reasoning. Shear features are sheet metal parts features that are created by shearing operations like blanking, notching, piercing and cutoff. Although the method is focused on the recognition of shear features, it also recognizes features that are formed by shear and deformation operations, such as bridges. Devarajan et al [26] proposed an AFR method for shear features that is based on profile offsetting for layout punching tool path specification. The offsetting method reveals the portions of parts that cannot be removed by punching, so these portions would need specific tools to be produced. A change in punch diameter could alter the recognized features and thereby introduce inconsistent output. Kannan and Shunmugam [27] used a center-plane model to calculate the shear layout of sheet metal parts. The shear layout could then be used for recognizing shear features.

Sheet metal generic deformation features could be the result of either pure deformation or shear and deformation operations; hence, there are always footprints of deformation in their structures. Liu et al. [28] presented a fundamental study that addressed some of the main issues, including feature intersection and array features, although the geometry was limited to cylindrical and planar faces. Feature intersection occurs when a few features intersect such that one of them is split or misses certain topological entities. An array feature is made up of repeated features. Noticeably, these issues have not been considered in the subsequent works.

Kannan and Shunmugam [27, 29] took two significant steps in sheet metal AFR: resting the method on STEP AP 203 for promoting its applicability and proposing the calculation and utilization of a center-plane model. Their proposition to use a center-plane model is twofold: it proves that a reduction in topographical information improves computation performance and that it does not necessarily reduce useful information; however, it adds a step to the AFR method. Before Kannan and Shunmugam, Jagirdar et al [30] assumed that a sheet metal model is represented by its center-plane and accordingly proposed their AFR method. They considered the intersection of features in sheet metal models, as limited to only “cross-bend” features (features that pass through a bend), and proposed a technique to identify the intersecting features.

Even though their proposed technique was not general, subsequent studies have detained the problem of intersecting features entirely; making their research quite unique. Gupta and Gurumoorthy [31], in one of the most recent studies, propose a general solution for the recognition of generic deformation features. In their paper, cylindrical, conical, spherical and toroidal faces are considered as transitive entities to characterize deformation. This method is more geometrically general than any other published works.

3 TERMS DEFINITIONS AND PREMISES

Before describing the proposed AFR method, there are a number of concepts to be explained. The structure of B-rep models and the ASM feature taxonomy explain the input and output of this work, respectively. The feature definition and description briefly explain the authors' viewpoint on features to facilitate a better understanding of the proposed method.

3.1 Structure of B-rep models

The structure of B-rep models that is used in this study is included in the definition provided by ISO 10303-42 [15]. Here, the B-rep model refers to an exact explicit boundary representation, and thus, a non-faceted representation of an ASM part. The B-rep CAD model **manifold_solid_brep** contains geometric elements, including **surface**, **curve**, and **point**, as well as topological elements, including **connected_face_set**, **closed_shell**, **face**, **face_surface**, **face_bound**, **face_outer_bound**, **loop**, **edge_loop**, **path**, **edge**, **oriented_edge**, **edge_curve**, **vertex**, and **vertex_point**. The **curve** is assumed, based on studying real ASM parts, to be **lines**, elementary **conics** and general parametric polynomial curves, such as **b_spline_curve**. The **surface** is assumed to be only an **elementary_surface** which could be a **plane**, **cylindrical_surface**, **conical_surface**, **spherical_surface**, **toroidal_surface** and **b_spline_surfaces**. A **connected_face_set** is a set of faces such that the domain of the faces together with their bounding edges and vertices is connected [15]. A **face_surface** is a subtype of a **face** in which the geometry is defined by an associated surface. A **face_bound** is a topological entity, constructed by stringing together connected (oriented) edges beginning and ending at the same vertex, and is intended to be used for bounding a face. A **face_outer_bound** is a special subtype of **face_bound** that carries the additional semantics of defining an outer boundary on the face. An **edge_curve** is a special subtype of an **edge**, which has its geometry fully defined. A **vertex_point** is a subtype of **vertex** that has its geometry defined by a point.

3.2 Feature definition and description

In this work, we define a feature as: *a portion of a geometry model that is significant in at least one of the phases of the product's lifecycle and that can be described by its attributes*. Accordingly, the specifications of a part and the information conveyed by its model that are not reflected through its geometry, i.e. material, material properties, color and coating, are not features, but rather are the characteristics of the part itself.

The attributes of a feature are geometry, feature relationships and parameters. For example, the geometry of a hole is its faces, and its relation to its parent feature is represented by the edges connecting it to its parent feature. The parameters of the hole are its location (defined by an axis) and diameter (derived from its geometry). The geometry therefore links features with their B-rep. The feature relationships expose the feature structure of geometry models. In this work, the feature relationships are demonstrated to be rooted in the topological adjacency of the related features. In addition, the parameters of features carry the design intent or engineering semantics related to the features. The parameters can be numerical or non-numerical information and must be extracted from the geometry model, e.g. a bead's height and its type (straight or curved). It should be noted that some features have different types (e.g. flanges can be curved or planar and closed, open or perpendicular) and the types of each feature are considered as one of their parameters.

Parent-child relationships has been implemented to rationalize the relation of features in sheet metal parts [22, 28]. There is a *parent-child* relationship between the features of ASM parts. Child features are created based on their parent feature. A child feature could be the parent feature of another child feature. For example, a flange and a hole that are made on a web are the child features of the web, whilst the flange per se can be the parent feature of another hole on itself.

3.3 ASM feature taxonomy

The features in ASM parts included in this study were observed by studying design guidelines and 168 diverse structural sheet metal parts of fuselage and cockpit. A structural system is comprised of a thin-skinned shell which is stiffened by longitudinal stringers supported by transverse frames to form a semi-monocoque structure [32]. This semi-monocoque is very efficient and has a high strength to weight ratio. The parts that we have studied for this paper were all produced by brake-forming or hydro-forming, thus the skin panel parts and stringers were omitted. Brake-formed and hydro-formed parts are used to form frames, bulkheads, passengers and cargo floor structures [32] and cockpit components.

An aerospace sheet metal part is manufactured in two main steps: trimming the blank of the part from a sheet of metal and deforming the blank where needed. The generic features of ASM parts are organized, in this paper, into *web*, *trim features* and *deformation features*. Figure 1 shows the proposed taxonomy of ASM model features.

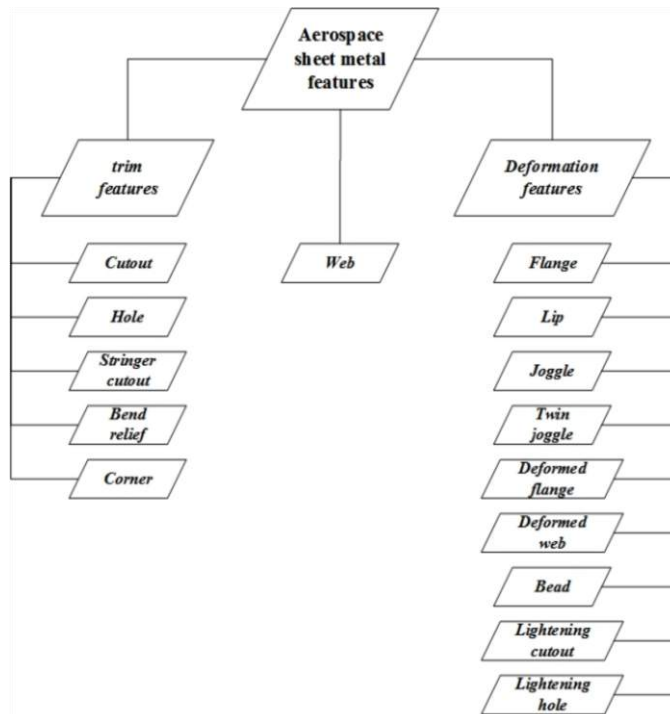


Figure 1: The proposed taxonomy of features in the ASM parts.

The *web* is distinct among ASM features, shaped by cutting the blank and by the deforming operations that create other features on it. Figure 2 illustrates an example where a web boundary is defined by cutting the blank, indicated by the dashed-blue line, and the deforming operations creating the other features on it, indicated by the solid-red line. It should be noted that at a later step of the AFR process the web boundary is examined so as to yield other features such as

corners and corner reliefs. A web is assumed to be the planar portion of an ASM part with the highest surface area (Figure 2), and for 165 of our 168 samples (98%) the assumption is valid.

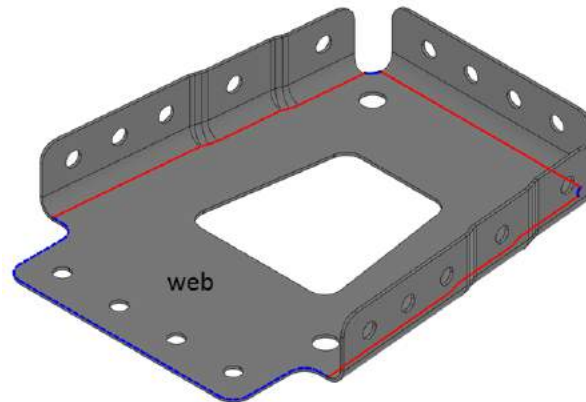
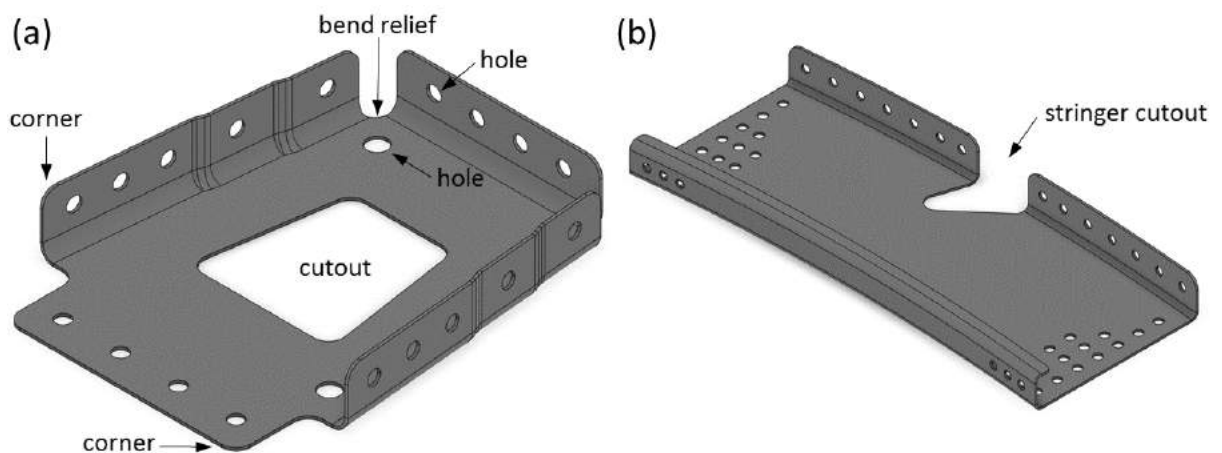


Figure 2: Illustration of the web as the feature with the highest surface area.

The trim features include *cutout*, *hole*, *stringer cutout*, *bend relief* and *corner*, as illustrated in Figure 3, and are the result of trimming. *Cutouts* are formed by removing a portion of their parent feature, provided that the boundary of the parent feature is not changed. *Holes* are circular cutouts. *Stringer cutouts* are formed by modifying the boundary of the parent feature, i.e. the web, and splitting the flanges or the deformed flanges resulted from a twin joggle, as illustrated by Figures 3 (b) and (c). Stringer cutouts are created to make room for a stringer to pass through. The *bend reliefs* are cutouts to avoid sharp adjacency between flanges, which causes cracking. Given that bend reliefs are designed based on design guidelines, the length of the relief cuts is calculable. The *corners* are formed by rounding off sharp convex corners, often at the vicinity of holes; corners are concentric with their corresponding hole. Although both corner reliefs and corners could be considered as parts of their parent features' boundary, they convey design intentions, necessitating distinguishing them as features.



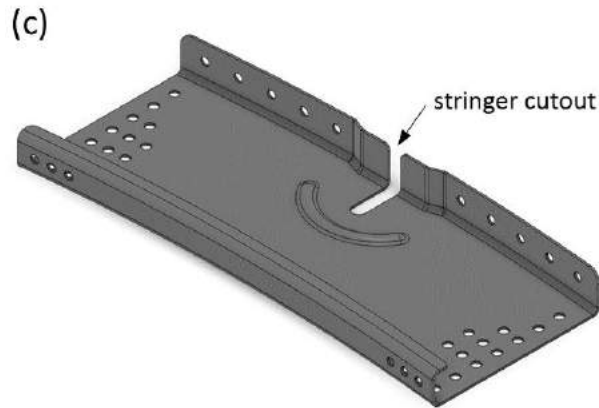
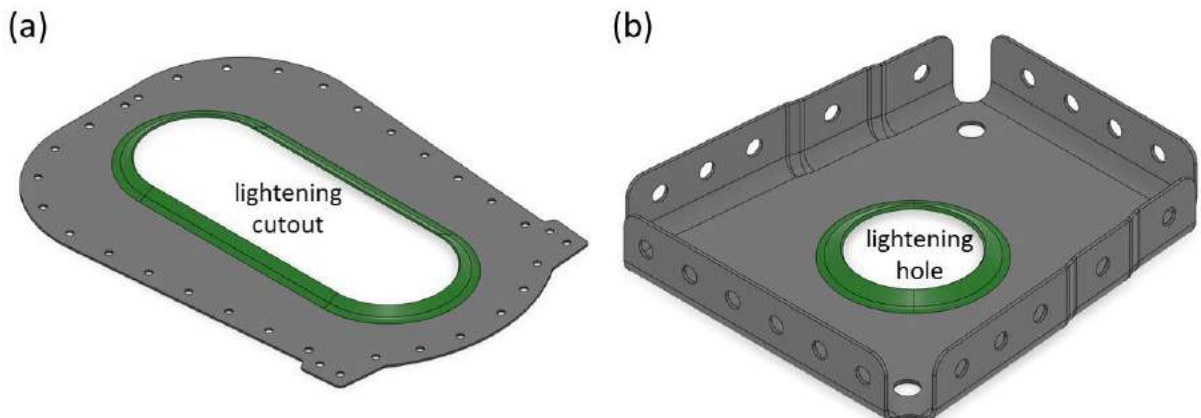


Figure 3: Illustration of trim features.

The deformation features include *lightening cutout*, *lightening hole*, *flange*, *lip*, *joggle*, *twin joggle*, *deformed flange*, *deformed web* and *bead*, as illustrated in Figures 4 and 5. They are created by deforming a portion of the parent feature. It is assumed in this study that all of the bends are created with a constant bend radius. The *lightening cutouts* and *lightening holes* are shaped by removing a portion of the parent feature and forming stiffening lips at the boundary of the removed portion, illustrated in Figure 4 (a) and (b). The *flange* is materialized on the external boundary of its parent feature and is always the child of a web or of another flange. It can be characterized as planar or curved; assembly or stiffening; immediate or return; single or combined; and perpendicular, open or closed, as illustrated in Figure 4 (c) to (h). Flanges types are defined according to the usual practice, except single and combined types of flanges which will help understanding the proposed feature recognition method.

Assembly flanges and stiffening flanges differ due to their functionality, which is reflected by the presence or absence of holes for joining them to other parts. The immediate flanges are materialized on the web, and the return flanges are materialized on their immediate parent flange. A Single flange is a flange that is modeled resting on a unique supporting geometry. On the other hand, a combined flange is materialized from distinct portions resting on distinct supporting geometries that combine to form it. Perpendicular, open or closed flanges are determined based on the angle between the flange and its parent feature.



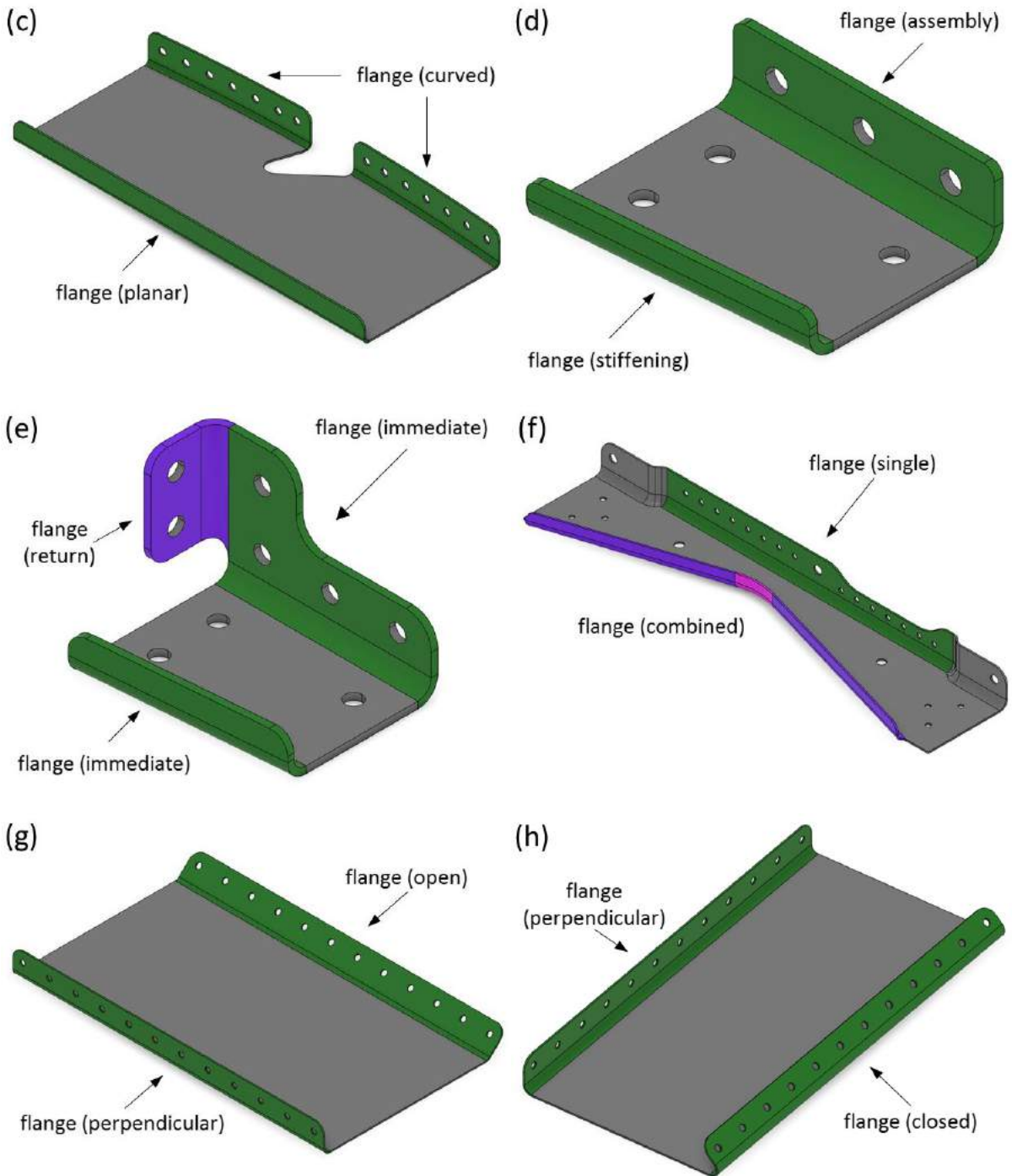
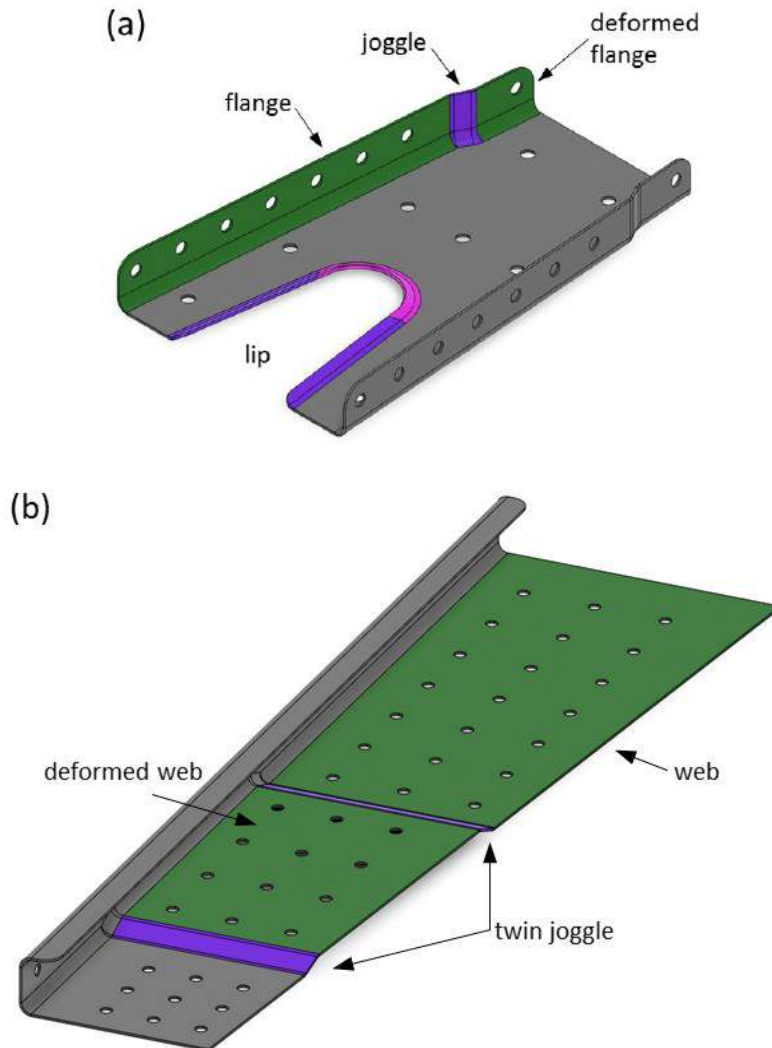


Figure 4: Illustration of deformation features: (a) a lightening cutout and (b) a lightening hole; (c) a planar and a curved flange; (d) an assembly and a stiffening flange; (e) an immediate and a

return flange; (f) a single and a combined flange; (g) a perpendicular and an open flange; and (h) a perpendicular and a closed flange.

Lips are shape-wise similar to combined-open-immediate-stiffening flanges, as illustrated in Figure 5 (a). *Joggles* and *twin joggles* are step-deformations that produce recesses on the web or on the flanges. The recessed portions of the parent feature's themselves are considered as features that are called either a *deformed web* or a *deformed flange* depending on the parent feature of the joggle. Figures 5 (a) and (b) provide examples of joggles and deformed flanges, and twin joggles and a deformed web, respectively. *Beads* result from protruding a portion of the web, and can be characterized as straight or curved as illustrated in Figures 5 (c) and (d).



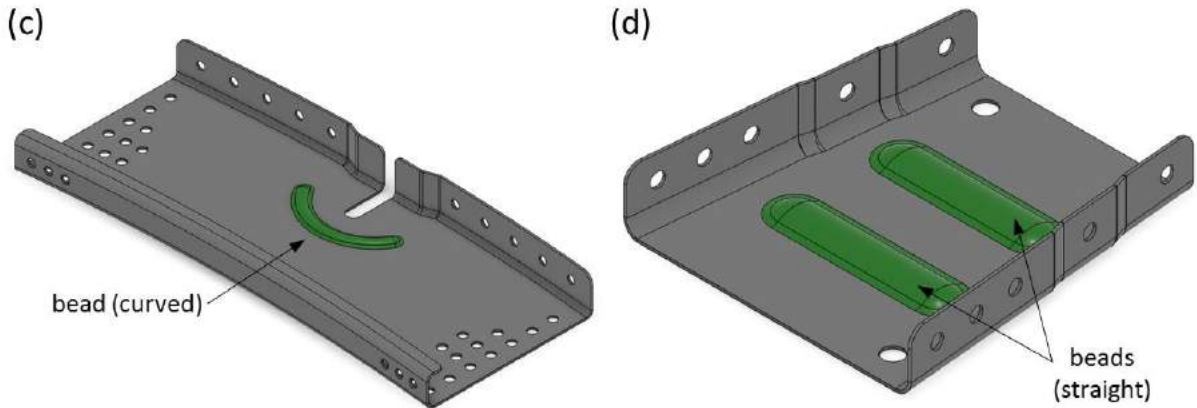


Figure 5: Illustration of: (a) a lip, a joggle and a deformed flange; (b) a twin joggle and a deformed web; and (c) and (d) a straight bead and a curved bead.

4 PROPOSED AUTOMATED FEATURE RECOGNITION METHOD

The aerospace sheet metal (ASM) automated feature recognition (AFR) method described in this work is comprised of two major steps: (1) classifying and grouping the elements of 3D B-rep model of the ASM parts and (2) recognizing the ASM features. Figure 6 provides a flowchart of the proposed method.

4.1 Classifying and grouping of B-rep elements

To prepare the 3D B-rep models for feature recognition, specific B-rep topological elements must be classified into novel subtypes and eligible faces are grouped into novel sets that will be used in feature recognition process. The classification and grouping of **faces**, **face_bounds** and **edges** is explained in the following section. After classifying **faces** (substep 1.1), they are grouped into **joggle_face_sets** (substep 1.2), which are sets of faces relating to joggles. At the end, **face_bounds** and **edges** are classified into their subtypes (substep 1.3). The output of this first major step thereafter becomes the input of the second major step, feature recognition.

4.1.1 Substep 1.1: Classifying faces

Classifying faces includes a series of classification actions on faces according to their continuity at their shared edges and their adjacency, and a series of faces class alteration actions if certain conditions are met.

First, every **face** from the B-rep model is classified as either **sheet_faces** or **trim_faces**. Figure 7 (a) shows an example to illustrate the distinction between **sheet_faces** (gray) and **trim_faces** (red). The **trim_faces** (red) are related to the faces of an ASM part that are created by trimming, and one of their dimensions is always equal to the part thickness. The **sheet_faces** (gray) are related to the faces of an ASM part that are not being trimmed, and they constitute the two sides of an ASM part.

A **sheet_face** is G1 continuous to all its adjacent **sheet_faces**. Any face that is not G1 continuous with a **sheet_face** is a **trim_face**. Two internally continuous C1 faces are geometrically continuous (G1) if their tangent planes coincide at every point along their shared edge [33]. Since the B-rep model is created by a CAD modelling solution, as opposed to being constructed from 3D scanning, it can be assumed that all of the faces are internally continuous C1,

and if the two faces are geometrically continuous at any arbitrary point of their shared edge, they have G1 continuity along the entirety of their shared edge.

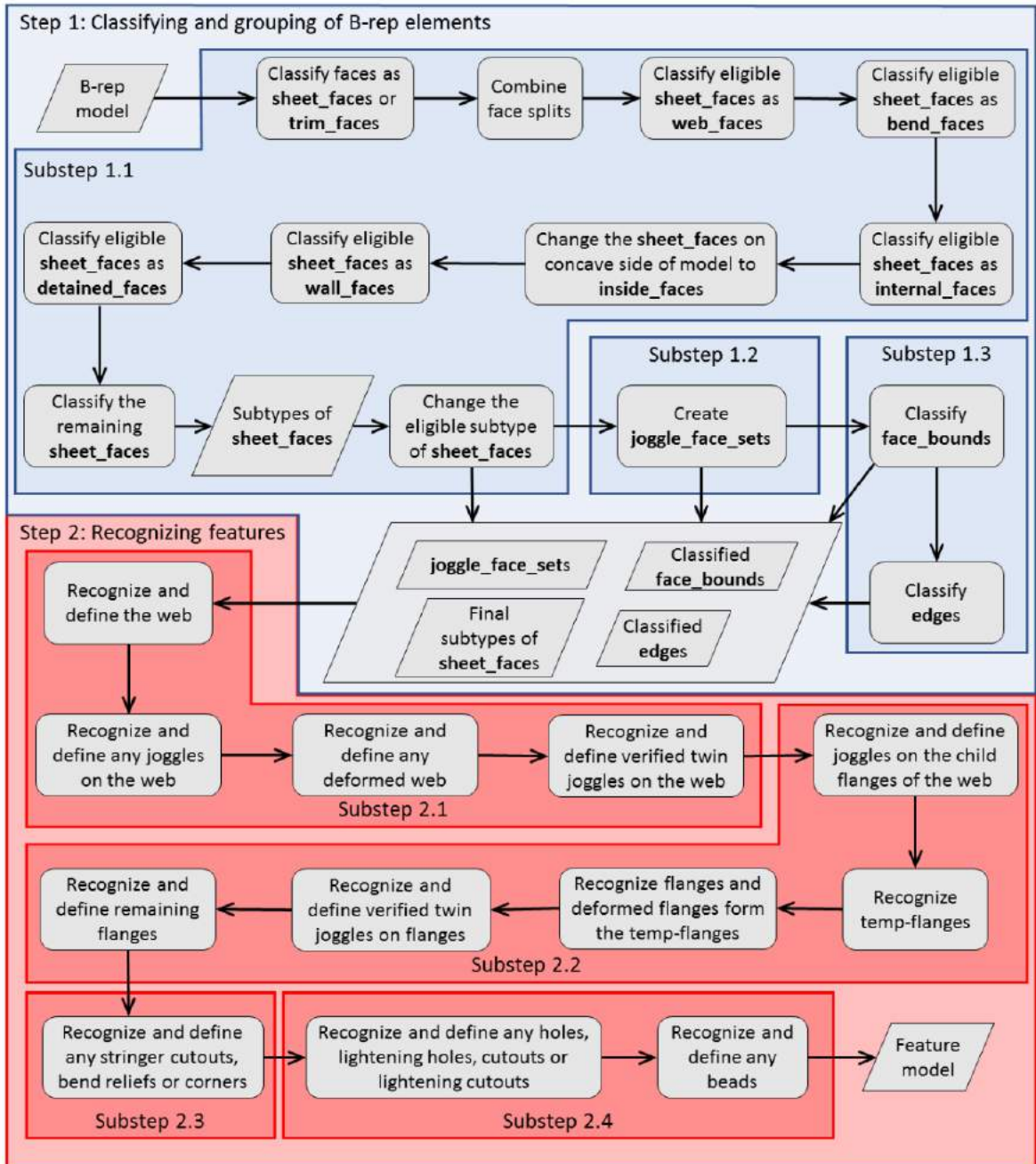


Figure 6: Flowchart of the proposed method.

Classifying faces as sheet_faces or trim_faces:

In a very first step, the two largest faces of the B-rep model are classified as **sheet_faces**. All the adjacent faces of these faces are subsequently evaluated to identify if they are in G1 continuity with these **sheet_faces**. The G1 continuous ones are classified as **sheet_faces**. The adjacent faces without G1 continuity are classified as **trim_faces**. This G1 continuity evaluation continues until all faces in the B-rep model are classified as either **sheet_faces** or **trim_faces**.

Combining face splits:

Before the next step, the **sheet_faces** must be checked for any face splitting. All of the **sheet_faces** must be evaluated to identify the adjacent ones that are associated to the same **surface**. These **sheet_faces** must be combined by removing one of their shared edges. Figure 7 (b) shows how the combination of split faces affects the faces of the lightening hole.

Classifying eligible sheet_faces as web_faces:

The **sheet_faces** are primarily classified as **web_faces**, **bend_faces**, **wall_faces**, **detained_faces** and **internal_faces**. The **sheet_faces** with the two highest surface area are classified as **web_faces**. **Web_faces** are a pair of planar **sheet_faces** that have the highest surface area. Figure 7 (c) shows the **web_faces** of the example in Figure 7 (a). The classification of the remaining **sheet_faces** to the other subtypes is performed by starting from the **web_faces** and traversing to **trim_faces** until all **sheet_faces** are classified appropriately.

Classifying eligible sheet_faces as bend_faces:

All the **sheet_faces** adjacent to the **web_faces**, via their **face_outer_bounds**, are then classified as **bend_faces**, as illustrated in yellow in Figure 7 (d).

Classifying eligible sheet_faces as internal_faces:

The **sheet_faces** adjacent to **web_faces**, via their non-**face_outer_bounds**, are classified as **internal_faces**. The **sheet_faces** adjacent to **internal_faces**, and that are not yet classified, are also classified as **internal_faces**, as illustrated in pink in Figure 7 (e).

Changing the sheet_faces on concave side of model to inside_faces:

At this stage, the areas of each of the preceding **web_faces** and their adjacent **bend_faces** and **internal_faces** are calculated. The **web_face** and its adjacent **bend_faces** and **internal_faces** that have the smallest combined area are changed – or reclassified – to **inside_faces**. The **inside_faces**, as illustrated in black in Figure 7 (f), correspond with the side of the ASM part that is concave. **Inside_faces** are not used in the rest of the process, while the remaining **sheet_faces**, on the convex side, are used in the following steps.

Classifying eligible sheet_faces as wall_faces:

All the **sheet_faces** adjacent to the preceding **bend_faces** are checked to verify that they are only adjacent to one **bend_face**. If so, they are classified as **wall_faces**. The **wall_faces** are the **sheet_faces** that are adjacent to only one previously classified **bend_face**, as illustrated in Figure 7 (g). **Wall_faces** must not be assumed to be linked to flanges, as they could be involved in forming the geometry of various features.

Classifying eligible sheet_faces as detained_faces:

If there are any **sheet_faces** adjacent to more than one preceding **bend_faces**, they are classified as **detained_faces**, as indicated in Figure 7 (h). **Detained_faces** are the **sheet_faces** that share edges with more than one previously classified **bend_face**.

Classifying the remaining sheet_faces:

At this stage, all the **sheet_faces** that are not classified as their subtypes and are adjacent to **wall_faces** via their **face_outer_bounds** are classified as **bend_faces**. The **sheet_faces** adjacent via the **wall_faces'** **non-face_outer_bounds** are classified as **internal_faces**. The remaining **sheet_faces** are evaluated according to the rules for **wall_face**, **detained_face**, **bend_face**, and **internal_face** to be classified consecutively until no **sheet_face** remains (Figure 7 (i)). In general, **bend_faces** are the **sheet_faces** that are adjacent to previously classified **wall_faces**, at the **face_outer_bound** of the **wall_faces** and **internal_faces** are the **sheet_faces** that are adjacent to **web_faces** or **wall_faces** at their **non-face_outer_bounds**.

Changing the subtype of eligible sheet_faces:

The **wall_faces**, **bend_faces** and **detained_faces** will be subsequently evaluated to verify if they are eligible to be changed. So far, these **sheet_faces** have been classified according to their adjacency. Now, if eligible, their classification is changed so that they appear in the appropriate sequence. The sequence that **bend_faces**, **connect_faces** or **wall_faces** eventually appear in needs to comply with two rules: **bend_faces** appear between **wall_faces**; and **connect_faces** appear between **bend_faces**. All the **detained_faces** will eventually be changed as either **connect_faces**, **bend_faces**, or **wall_faces**.

First, the **bend_faces** that are adjacent to other **bend_faces** are checked to identify those that are adjacent to only one other **bend_face**. If the adjacent **bend_face** is adjacent to more than one **bend_face**, it is changed to a **connect_face**. Since a **connect_face** actually 'connects' **bend_faces**, it is designated so, as illustrated by the **connect_faces** shown in green in Figure 7 (j). The change of the eligible **bend_faces** to **connect_faces** is continued until there are no **bend_faces** adjacent to another **bend_face**.

The **detained_faces** are then evaluated and the ones that are surrounded by preceding **bend_faces** are changed to **connect_faces**. The **connect_faces** are basically the **detained_faces** that are surrounded by **bend_faces** or the **bend_faces** that are adjacent to more than one **bend_face**. If there are any remaining **detained_faces**, the ones that are adjacent to preceding **connect_faces** are changed to **bend_faces** and the ones adjacent to preceding **bend_faces** are changed to **wall_faces**.

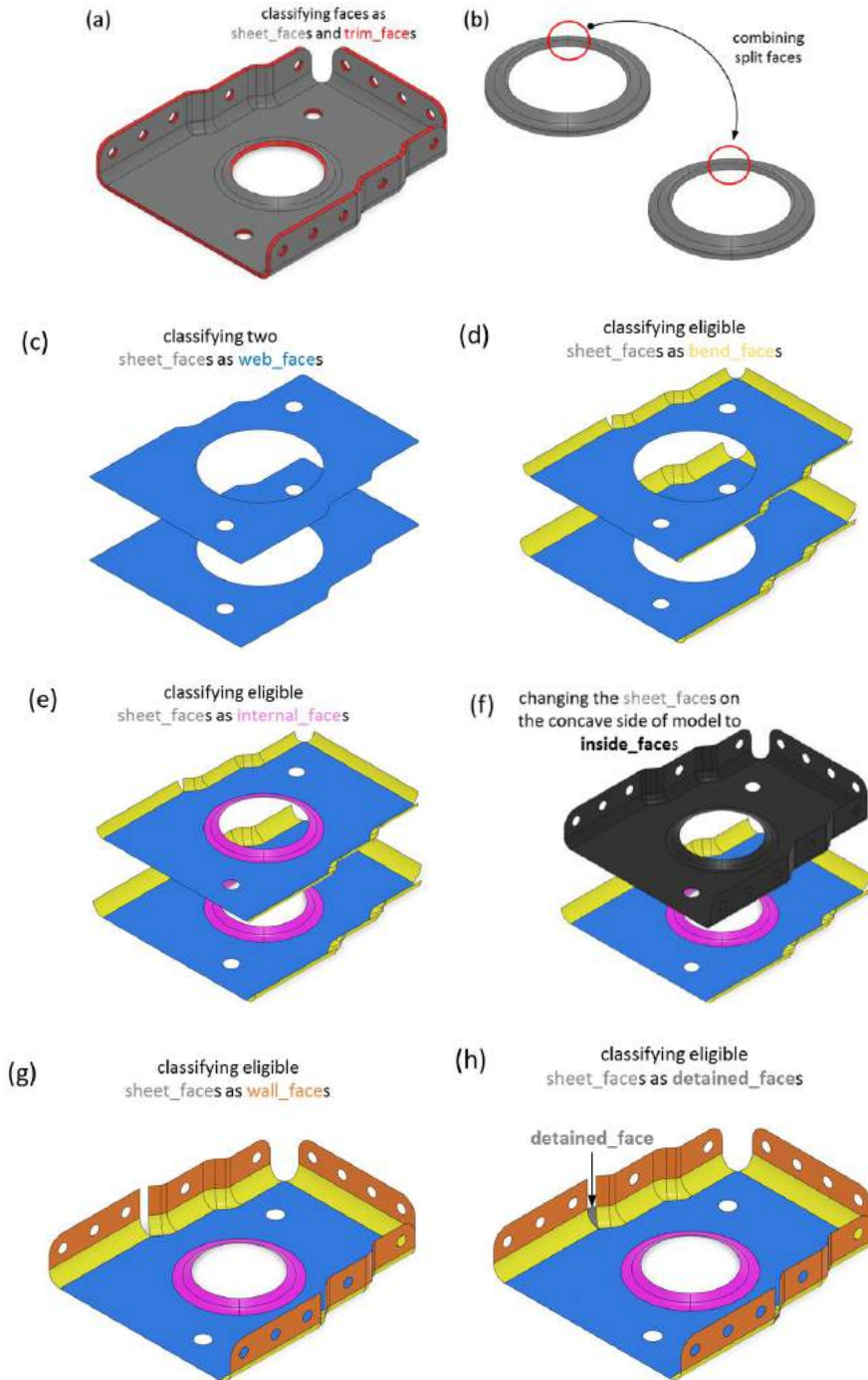
Finally, the **wall_faces** that are adjacent to other **wall_faces** are checked to identify the ones that are adjacent to only one other **wall_face**. If the adjacent **wall_face** is adjacent to more than one **wall_face**, it is changed to a **bend_face**. The change of the eligible **wall_faces** is continued until no **wall_faces** are adjacent to another **wall_face**. Figure 7 (j) illustrates the end result of the change of eligible **bend_faces** and **detained_faces** to **connect_faces**, and change of eligible **wall_faces** to **bend_faces**.

4.1.2 Substep 1.2: Creating joggle_face_sets

Now that **faces** are classified, eligible ones are grouped to create **joggle_face_sets**. Indeed, the presence of certain sets of two **connect_faces** is an indicator of the presence of a joggle feature on a web or flange. Observing different scenarios of joggles on webs or flanges reveals that it takes **connect_faces**, **bend_faces** and a **wall_face** to materialize a joggle. Therefore, it is proposed to create **joggle_face_sets** to be used in describing joggles. A **joggle_face_set** consists of two **connect_faces**, three **bend_faces** and one **wall_face**.

To create a **joggle_face_set**, a **bend_face** that is adjacent to only one **connect_face** is identified first, as indicated in Figure 8 (a). The **connect_face** adjacent to this **bend_face** is evaluated to verify if one of its adjacent **bend_faces** are adjacent to another **connect_face**. If

yes, the two identified **connect_faces** are included in the **joggle_face_set** as well as the **bend_face** between them. This **bend_face** is evaluated to identify its adjacent **wall_face** that is not adjacent to any of the aforementioned **connect_faces**. The **wall_face** is also included in the **joggle_face_set** in addition to all of its adjacent **bend_faces**. The **wall_face** and the **bend_faces** are indicated in Figure 8 (b).



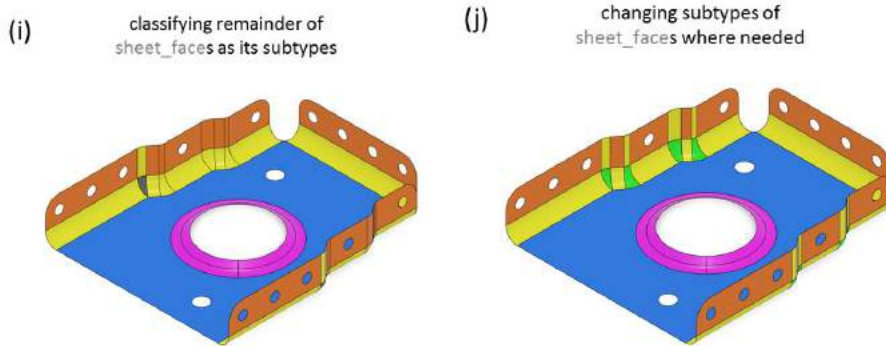


Figure 7: Illustration of the process of classifying faces.

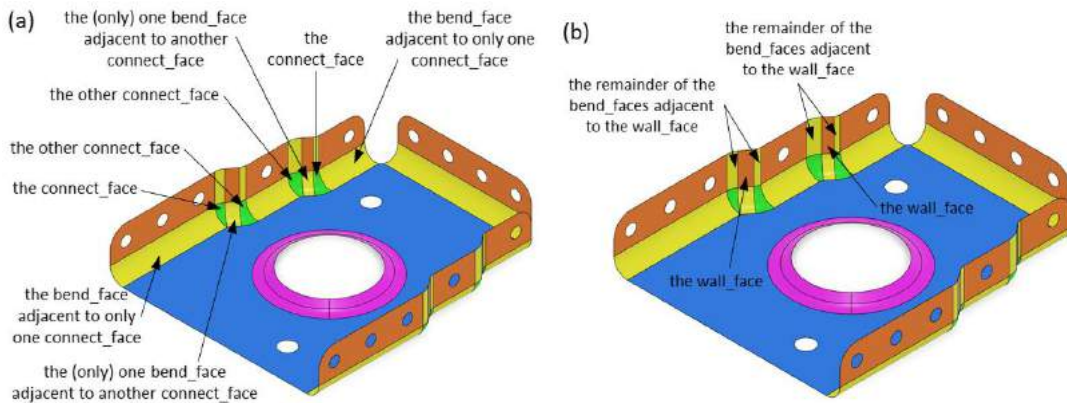


Figure 8: Illustration of the faces included in creating **joggle_face_sets**.

4.1.3 Substep 1.3: Classifying **face_bounds** and **edges**

Now that **faces** are classified and that **joggle_face_sets** are created, we classify **face_bounds** and **edges**.

Classifying face_bounds:

Similar to what was done with **faces**, the **face_bounds** are now classified into novel subtypes. The **face_outer_bound** is the only subtype of the **face_bound** in STEP B-rep models [15]. In our work, **face_bounds** are classified as **face_outer_bound**, **bead_bound**, **internal_bound** or **hole_bound**, as illustrated in Figure 9. If a **face_bound** is not a **face_outer_bound**, it is evaluated to verify if it surrounds seven **internal_faces**. In such a case, the **face_bound** is classified as a **bead_bound**. If a **face_bound** is neither a **face_outer_bound** nor a **bead_bound**, and it consists of **edges** associated to distinguishable **curves** [15], it is classified as an **internal_bound**. If a **face_bound** consists of **edges** associated to undistinguishable **curves**, it is classified as a **hole_bound**. Distinguishable **curves** are the ones that are not defined by identical underlying geometric equations, whilst undistinguishable **curves** are the ones that are defined by the same equations. A **hole_bound** is indeed a special type of **internal_bound** that carries the

additional semantics of defining an interior boundary related to a hole or a lightening hole. Thus, a **hole_bound** is comprised of **edges** that are associated to undistinguishable **curves**.

Classifying edges:

The **edges** are classified as novel subtypes based on the types of faces they are associated to and their geometry. They are classified as **trim_lin_edges**, **trim_nonlin_edges**, **untrim_lin_edges** or **untrim_nonlin_edges**, as illustrated in Figure 10. A **trim_lin_edge** is an edge that is shared between any subtype of **sheet_faces** and a **trim_face**, and is associated to a **line**. A **trim_nonlin_edge** is an edge that is shared between any subtype of **sheet_faces** and a **trim_face**, and is associated to any subtype of **curve** other than a **line**. An **untrim_lin_edge** is an edge that is not shared with any **trim_face** and is associated to a **line**. An **untrim_nonlin_edge** is an edge that is not shared with any **trim_face** and is associated to any subtype of **curve** other than a **line**. The classification of the edges is pivotal to recognizing features.

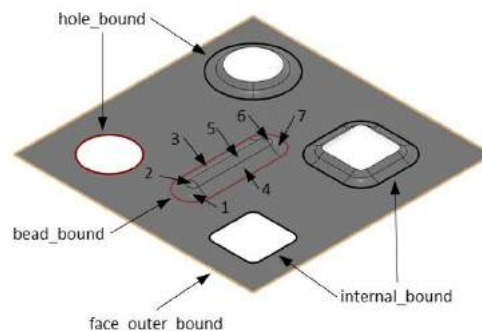


Figure 9: A fabricated example to illustrate subtypes of **face_bounds**.

4.2 Recognizing features

In previous section, a method for classifying and grouping **faces**, **face_bounds** and **edges** was proposed, as well as a method for grouping eligible faces to create **joggle_face_sets**. In this section, the method to use these entities for recognizing and constructing features and for extracting meaningful design information is proposed. Here, we explain the correlation of the geometric, topological and feature entities. ASM feature recognition involves identifying the topologic entities that comply with the conditions required for recognizing the corresponding feature. The parameters of features are calculated from the geometric and topological elements and the feature relationships are formed by the topological elements according to the features' descriptions.

Before describing the steps of feature recognition, it should be noted that some of the ASM features could have either parent or child relationships with other features, while some features could only have either a parent relationship or a child relationship with other features. For instance, a flange is always child of its parent feature and is itself the parent feature of its child features. On the other hand, features such as holes, lightening holes, cutouts, lightening cutouts, beads, corner and bend reliefs could only be the children of their parent feature and could never be the parent of any other feature.

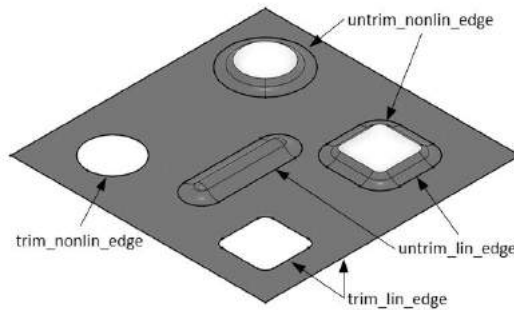


Figure 10: A fabricated example to illustrate subtypes of **edges**.

The web is a feature that exists in all ASM parts, and all other features of the ASM part are directly or indirectly its child features. The web does not have a parent feature, and thus it should be recognized in the first step. This step is also where it should be determined if the web is deformed by joggles and if so, that the deformed web is also recognized (section 4.2.1). Next, the existing flanges on the web, the joggles and the twin joggles on those flanges and the deformed flanges are recognized and fully characterized (section 4.2.2). In section 4.2.3., it is explained how the preceding flanges, deformed flanges and joggles are evaluated to recognize any existing bend reliefs, stringer cutouts or corners. Finally, in section 4.2.4., all the holes, lightening holes, cutouts, lightening cutouts and beads on the web and on the deformed web as well as the holes, cutouts and corners of preceding flanges and deformed flanges are recognized. Although a flange could possibly have child lightening holes, lightening cutouts and beads, such a case was not found in the industrial samples of this study and it is not considered in this work.

4.2.1 Substep 2.1: Recognizing webs, their child joggles and twin joggles, and deformed webs

The web is proposed to be recognized first. To recognize the web, identifying the **web_face** is sufficient, as it represents the geometry of the web. The feature relationships between the web and its child features are formed by the **bead_bounds** and **internal_bounds** and the **hole_bounds** of the **web_face**, and the edges that are shared between the **face_outer_bound** of the **web_face** and the adjacent **bend_faces** of the child features. The only parameter of the web is its supporting plane, which is the plane that the **web_face** occurs in.

After recognizing the web, to avoid mistakenly recognizing a joggle on the web as a flange, the presence of a joggle on the web is verified first. If the **web_face** shares an edge with a **bend_face** of a **joggle_face_set** but does not share an edge with any of the **connect_faces** of the **joggle_face_set**, the web is deformed by a joggle. If any of the **connect_faces** of the **joggle_face_set** share an edge with the **web_face**, the corresponding joggle is not pertinent to the web; it does not deform the web but rather deforms a child flange of the web. Figures 11 (a) and (b) illustrate a fabricated example in which the presence of a shared edge between the **web_face** and the **connect_faces** of a **joggle_face_set** distinguishes the joggle on a flange from the joggle on the web. Thus, to recognize a joggle on the web, the **web_face** is checked to verify whether it shares an edge with a **bend_face**, but not with any of the **connect_faces** of the **joggle_face_set**. The **joggle_face_set**, including its **bend_faces**, **connect_faces** and **wall_face**, form the geometry of the joggle. The feature relationships of a joggle-on-the-web to its parent web and to its child deformed web are formed by the shared edge between each corresponding **bend_face** of the **joggle_face_set** and the **web_face** of the web and **wall_face** related to the deformed web. Those shared edges are colored red in Figure 11 (b).

The parameters of joggles, including joggle vectors indicating depth and runout directions, the radius of the bends, the depth and the runout length, are extracted from the geometry of the

joggle, as shown in Figure 11 (c). It should be noted that, in the specific example in Figure 11 (c), the bend radii differ by 3 mm (the thickness of the part), so that in fact both bends on the actual part are defined as equal to the minimum bend radius value (6 mm) for the considered material and part thickness. In practice, these parameters, except depth vector, are obtained from a manufacturing standard and a standard number could represent them.

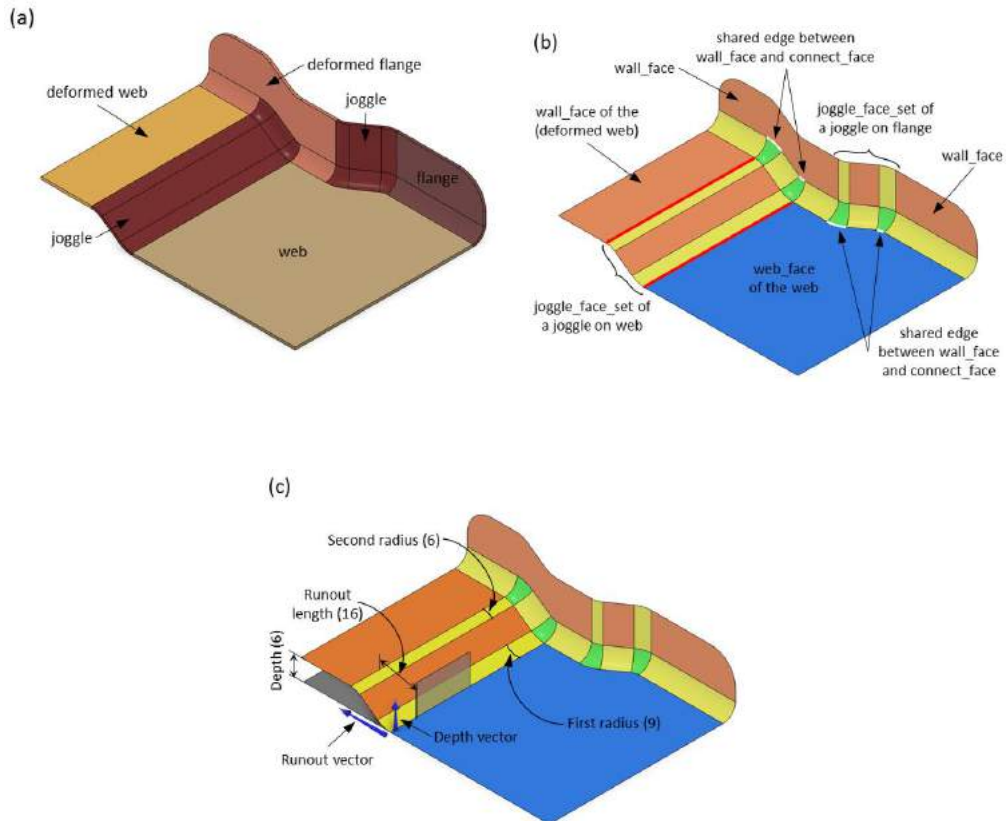


Figure 11: A fabricated example of how the presence of a shared edge between the **web_face** and the **connect_faces** of a **joggle_face_set** distinguishes a joggle on a flange from a joggle on the web (a) and (b); and the parameters of joggles (c).

Every joggle has one child feature: the deformed web or the deformed flange, if the joggle is on a flange. To recognize a deformed web, the **wall_face** that shares an edge with a **bend_face** of the preceding **joggle_face_set** but does not share an edge with any of the **connect_faces** of the **joggle_face_set** is identified, as shown in Figure 11 (b). The **wall_face** forms the geometry of the deformed web. The feature relationship between a deformed web and its parent joggle is formed by the shared edge between the **wall_face** and the **bend_face** of a **joggle_face_set**. Its feature relationship with its child features is formed by the **bead_bounds** and **internal_bounds** and **hole_bounds** of its **wall_face** and the edges that are shared between the **face_outer_bound** of the **wall_face** and the adjacent **bend_faces** of the child features. The parameters of a deformed web are its supporting plane, which is the plane that the **wall_face** occurs in, if we look at it from a geometric perspective. From a more semantic perspective, the deformed web is characterized by its distance to the web, which is equal to the depth of its parent joggle.

If the deformed web is itself deformed by a joggle or has a child joggle, the parent and child joggles of the deformed web are evaluated to recognize twin joggles, as illustrated by Figure 12 (a). Merging two joggles into one twin joggle means that the parent and child features of the two original joggles must also be merged. The deformed webs are evaluated to identify the one to be merged into the web. The geometry of the newly merged features is comprised of the combined geometry of both original features. For example, when the deformed web is being merged into the web, its **wall_face** is added to the geometry of the web. Figure 12 shows an example where two joggles (in Figure 12 (b)) are merged into a twin joggle and a deformed web merged into the web. Figure 12 (c) indicates how the **wall_face** of the deformed web that merges into the web is consequently associated to the web, or how the two **joggle_face_sets** associated to two individual joggles are associated to the one twin joggle.

The feature relationship of a twin joggle to its parent web is formed by the two shared edges between the **bend_faces** of a **joggle_face_set** and the **web_face** and **wall_face** of the web, as colored red in Figure 12 (c). The feature relationship of a twin joggle to its child deformed web is formed by the two shared edges between the **bend_faces** of the **joggle_face_sets** and the **wall_face** of the deformed web, as colored blue in Figure 12 (c). The parameters of the twin joggle are the same as those of a joggle, except that the runout vector is not included. In the industrial samples reviewed in this work, webs were only deformed by one joggle, two joggles on their two ends, or a twin joggle.

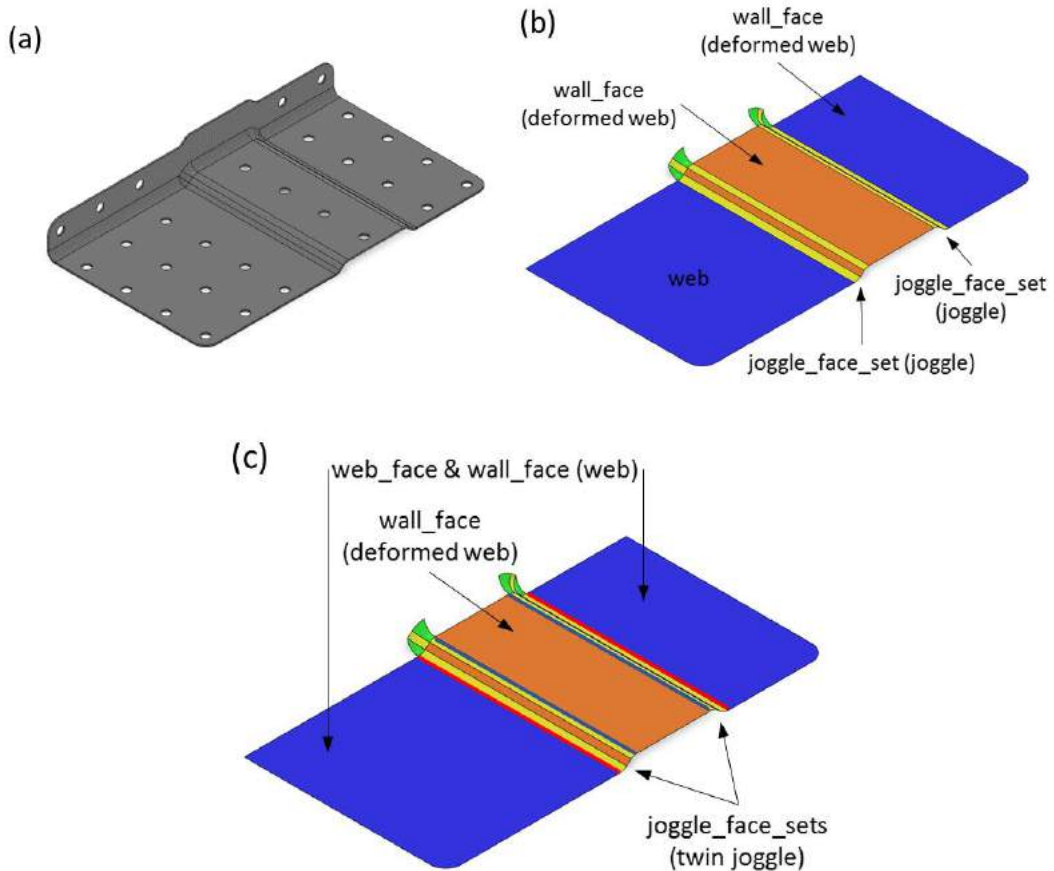


Figure 12: An example of two joggles merged into a twin joggle and a deformed web merged into the web.

4.2.2 Substep 2.2: Recognizing flanges, their child joggles and twin joggles, and deformed flanges

Once the web and its child joggles, twin joggles and deformed webs are recognized, the flanges and their potential child joggles, twin joggles and deformed flanges are recognized in a next step. The joggles on the flanges are proposed to be recognized before recognizing the flanges. To recognize the joggles on the child flanges of the web, the faces included in the web geometry are checked to verify if they share edges with a **bend_face** or with any of the **connect_faces** of any **joggle_face_set**. The **joggle_face_set**, including its **bend_faces**, **connect_faces** and **wall_face**, forms the joggle geometry. The parameters of a joggle on a flange are extracted from its geometry, as described for a joggle on the web. The feature relationships between the joggle and its parent flange and its child deformed flange are formed by the shared edge between each corresponding **bend_face** of the **joggle_face_set**, and the **wall_face** of the flange and the deformed flange. However, the feature relationships and parameters of a joggle-on-the-flange is dependent on determining its parent and child features, which are the flange and the deformed flange.

To recognize a flange and a deformed flange, the **bend_faces** of the **joggle_face_set** are checked to identify the two **wall_faces** that share an edge with any of them but that does not share an edge with any of the **connect_faces** of the **joggle_face_set**, as indicated in Figure 13. Each of these **wall_faces** share an edge with a **bend_face** that is not included in the **joggle_face_set** but does share an edge with the web geometry (**web_face**), as indicated in Figure 13. Each of the **wall_faces** and its adjacent **bend_face**, which is not included in the **joggle_face_set**, form a temporary feature entity, a temp-flange, until the flange and deformed flange are distinguished. The definition of the joggle, on a flange, provides the basis for distinguishing a flange from a deformed flange; since a joggle on a flange always creates a recess.

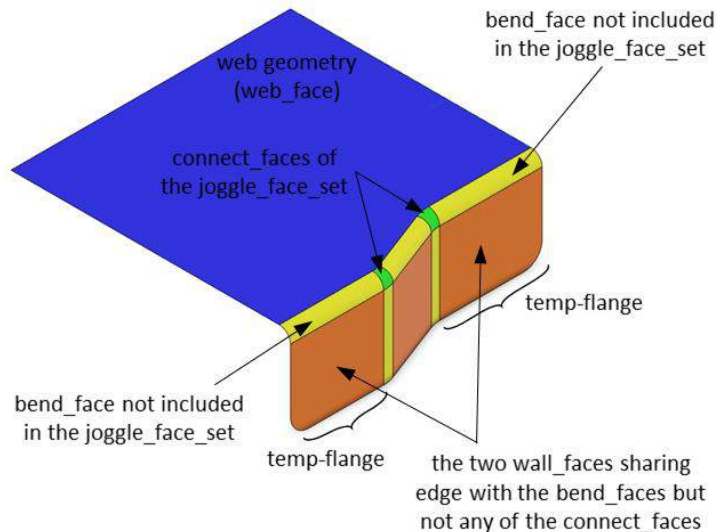


Figure 13: Illustration of the geometry of temp-flange.

If the cross product of the normal vector of a **wall_face** of the temp-flange with the normal vector of the **wall_face** of the joggle is in the same direction as the normal vector of the **web_face**, the order in which the parent feature of the joggle, the joggle and the child feature of the joggle share edges with the **web_face** is counterclockwise. If they are in the opposite direction, the order is clockwise. Figures 14 (a) and (b) give two examples of how the order of these features can be

determined. A temp-flange is converted to a deformed flange if it is a child feature of any joggle, or it is converted to a flange if it is only the parent feature to a joggle and is not itself a child feature of another joggle.

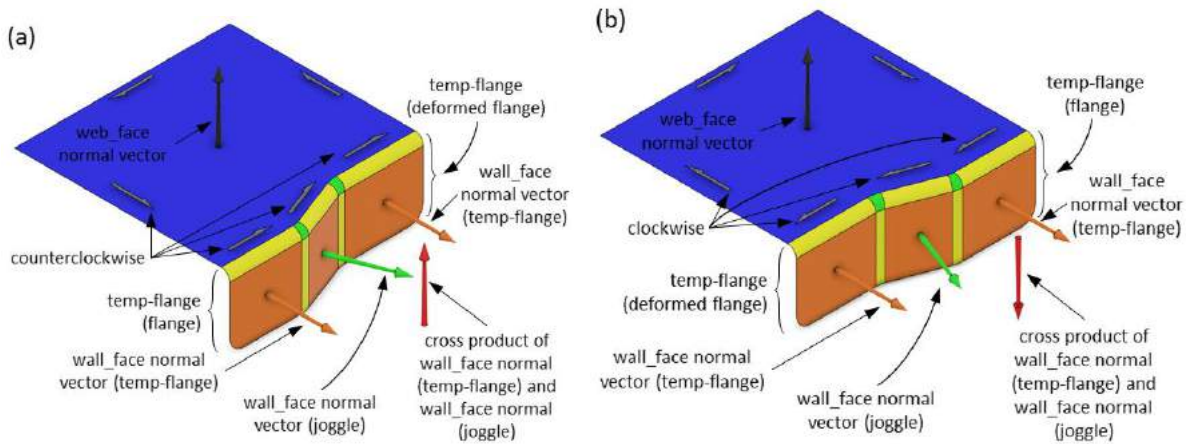


Figure 14: Two examples of determining the order in which the parent feature of the joggle, the joggle and the child feature of the joggle occur, to distinguish flange and deformed flange.

If a deformed flange is the child feature to two joggles, the parent joggles are evaluated for recognizing twin joggles. Similar to joggles on the web, if the joggles have identical parameters, but their runout vectors are in opposite directions, they are merged into one feature, the twin joggle. Merging two joggles into a twin joggle means that the two parent flanges of the two original joggles must be merged into one parent flange. The geometry of the newly merged features is composed of the combined geometry of both original features. For example, one of the flanges is removed and its geometry (**bend_face** and **wall_face**) are added to the geometry of the other flange. The feature relationship between a twin joggle and its parent flange is formed by the shared edges between the **bend_faces** of a **joggle_face_set** and the **wall_faces** of the parent flange. The feature relationship of a twin joggle to its child deformed flange is formed by the shared edges between the **bend_faces** of the **joggle_face_sets** and the **wall_face** of the deformed flange. The parameters of a twin joggle on a flange are the same as those of twin joggles on the web and are extracted from its geometry.

When all the flanges that are deformed by joggles are recognized, the other flanges are searched for. To recognize them, the **web_face** and **wall_faces** that are associated with the web and the flanges are checked to identify their adjacent **bend_faces** that is not included in the geometry of the previously-recognized features. Each **bend_face** that is found is checked to identify its adjacent **wall_face** that shares an edge with it and is also not included in the geometry of the previously recognized features. The **bend_face** and its adjacent **wall_face** form the geometry of a flange. In case the **bend_face** only shares an edge with a **wall_face**, that is part of the geometry of a previously-recognized flange, and a **connect_face**, that is not included in a **joggle_face_set**, the flanges are merged into one combined flange. The **bend_face** and the **connect_face** as well as the **bend_faces** and **wall_faces** of the adjacent flanges form the geometry of the flange. Figure 15 (a) shows an example for this case.

Once all flanges and deformed flanges are recognized, flanges are further characterized. The feature relationship between a flange and its parent feature is formed by the shared edge between the **bend_face** of the flange and the **web_face** or **wall_face** of the parent feature. The feature relationship between the flange and its child features is formed by the **bead_bounds** and the **internal_bounds** as well as the **hole_bounds** of the **wall_face** or the edges that are shared

between the **face_outer_bound** of the **wall_face** and the adjacent **bend_faces** of the child features.

The parameters of flanges, including their supporting geometry, bend radius, length and type are extracted from the geometry of the flange, as shown in Figure 15 (b). The planar and curved flanges are distinguished by evaluating their supporting geometry to verify if it is planar or non-planar. The assembly and stiffening flanges are distinguished by checking their **wall_faces** to verify if it has **hole_bound**. The immediate and return flanges are distinguished by checking their **bend_faces** to verify if they are adjacent to another **wall_face** or the **web_face**. The single and combined flanges are validated by checking their geometry for the presence of a **connect_face**. The perpendicular, open and closed flanges are distinguished by checking the angle between the normal vector of their **wall_face** and the supporting geometry of their parent feature.

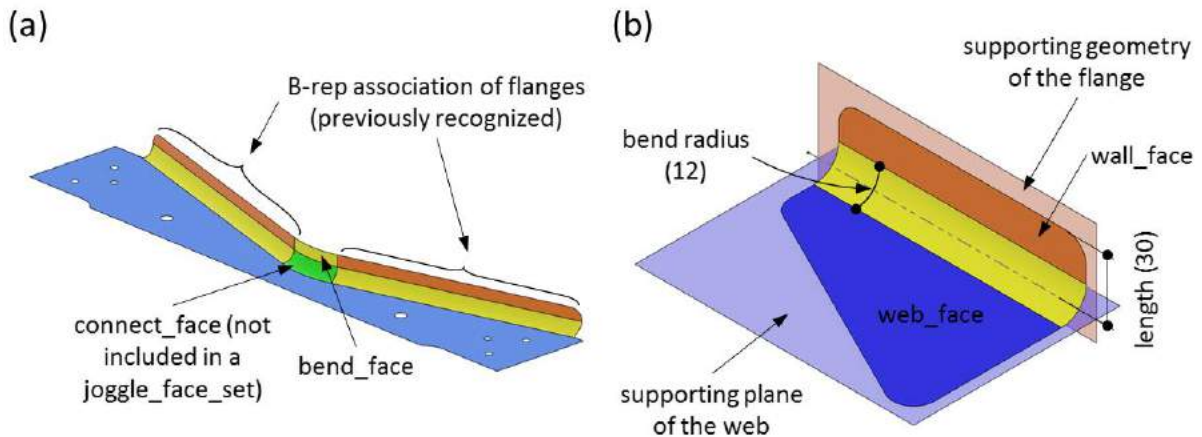


Figure 15: Illustration of: (a) recognizing a combined flange; (b) characterizing the parameters of a flange.

Now that flanges are characterized, deformed flanges are further characterized next. The feature relationship between a deformed flange and its parent feature is formed by the shared edge between its **wall_face** and the **bend_face** of the **joggle_face_set** of its parent joggle. Its feature relationships with its child features is formed by the **hole_bounds** of its **wall_face** and the edge that is shared between the **face_outer_bound** of the **wall_face** and the adjacent **bend_face** of the child feature. The parameters of the deformed flange are its supporting geometry, which is the plane or surface that the **wall_face** occurs in, and its distance to its related flange, which is equal to the depth of its parent joggle.

4.2.3 Substep 2.3: Recognizing stringer cutouts, bend reliefs and corners

After all the flanges and deformed flanges have been recognized and characterized, they are evaluated to identify some of their child features: stringer cutouts, bend reliefs and corners. To recognize the stringer cutouts, the deformed flanges that have identical supporting geometry, or the flanges that have identical supporting geometry, are identified, as indicated in Figures 16 (a) and (b). These flanges and deformed flanges are then evaluated to verify if there are **trim_nonlin_edges** and **trim_lin_edges** between them. Figures 16 (a) and (b) show two examples of stringer cutouts in which the **trim_nonlin_edges** and **trim_lin_edges** are indicated. If that is the case then the two flanges are merged into one and the edges form the geometry of a stringer cutout. When a stringer cutout is between two deformed flanges, they are merged into one deformed flange and their corresponding joggles and parent flanges are merged into a twin joggle and one flange. The feature relationship of the stringer cutout to its parent web is formed by the

face_outer_bound of the **web_face**. The parameter of the stringer cutout is its profile, which is formed by its edges.

To recognize a bend relief, the deformed flanges and flanges are evaluated to verify if their supporting geometry is not identical and they are apart by one **trim_nonlin_edge**, or a series of **trim_nonlin_edge** and **trim_lin_edges**, in the **face_outer_bound** of the **web_face** or the **wall_face** of their parent web. If the edges are G1 continuous with the **trim_nonlin_edges** of the **bend_faces** corresponding to the flanges and/or deformed flanges, it forms the geometry of a bend relief. Figure 16 (c) illustrates a fabricated example of a bend relief in which the **trim_nonlin_edges** are indicated. The feature relationship of a bend relief to its parent web or flange is formed by the **face_outer_bound** of the **web_face** or the **wall_face** that the **trim_nonlin_edge** is a part of. The parameter of the bend relief is its radius, calculated from the **trim_nonlin_edge**.

To recognize corners, the **trim_nonlin_edges** included in the **face_outer_bound** of the **web_face** or **wall_faces** are evaluated to identify the ones that are G1 continuous to at least one **trim_lin_edge**. Figure 16 (d) illustrates examples of corners in which the **trim_nonlin_edges** and the **trim_lin_edges** are indicated. The **trim_nonlin_edge** forms the geometry of the corner, and the **face_outer_bound** of the **web_face** or of a **wall_face** form the feature relationship of the corner with its parent web or flange. The parameter of the corner is its radius which is calculated from the **trim_nonlin_edge**.

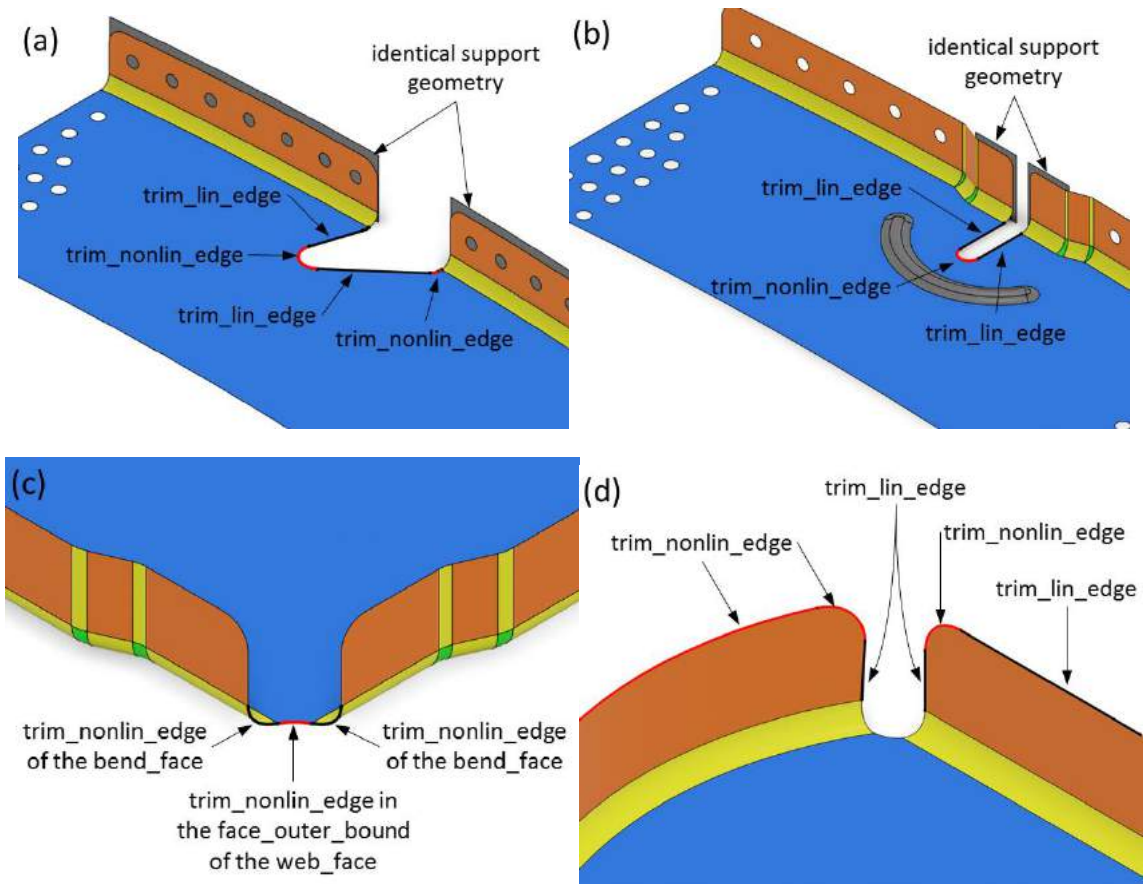


Figure 16: Analyzing edges to recognize stringer cutouts (a) and (b), bend reliefs (c) and corners (d).

4.2.4 Substep 2.4: Recognizing holes, lightening holes, cutouts, lightening cutouts and beads

In this last substep, all the holes, lightening holes, cutouts, lightening cutouts and beads on the web and the deformed web as well as the holes, cutouts and corners on the flanges and deformed flanges are recognized. To recognize these features, all of the **wall_faces** and **web_faces** are checked to identify their non-**face_outer_bounds**.

Finding a **hole_bound** on a **wall_face** or **web_face** that consists of **trim_nonlin_edges** verifies presence of a hole on the web or the feature that the **wall_face** is associated with. The **hole_bound** forms the geometry of the hole, and the **wall_face** or **web_face** form its feature relationship with its parent feature. The parameters of the hole are its location and its diameter; the latter is calculated from the edges included in the **hole_bound**. In practice, the holes diameter corresponds to a manufacturing standard and a standard note could represent. The location of the hole is represented by a point at the center of the **hole_bound**.

On the other hand, finding a **hole_bound** on a **wall_face** or **web_face** that consists of **untrim_nonlin_edges** verifies presence of a lightening hole. The **hole_bound** as well as the **internal_faces** that are surrounded by the **hole_bound** form the geometry of the lightening hole. The feature relationship of a lightening hole is the same as that of a hole. The parameters of a lightening hole include its location, diameter, bend radius and height. Such a combination of values would typically correspond to a manufacturing standard. These parameters are all calculated from the **internal_faces** in the geometry. Figure 17 illustrates the parameters of holes and lightening holes.

Cutouts and lightening cutouts are recognized in a procedure that is similar to how holes and lightening holes are recognized. Finding an **internal_bound** on a **wall_face** or a **web_face** that consists of **trim_nonlin_edges** and **trim_lin_edges** verifies the presence of a cutout on the web or the feature that the **wall_face** is associated with. On the other hand, if the **internal_bound** consists of **untrim_nonlin_edges** and **untrim_lin_edges**, it verifies the presence of a lightening cutout. The geometry of a cutout is formed by the **internal_bound** and the geometry of a lightening cutout is formed by the **internal_bound** as well as the **internal_faces** that are surrounded by it. The **wall_face** or **web_face** that the **internal_bound** is on forms the feature relationships of cutouts or lightening cutouts with their parent feature. The parameters of cutouts and lightening cutouts are their profile, which is formed by the **internal_bound**, bend radius and the height, which are calculated from the **internal_faces** included in its geometry. In practice, bend radius and height are designed according to a manufacturing standard (similar to lightening holes manufacturing standards) and a standard number could represent them. Figure 17 illustrates the geometrical parameters of a cutout and of a lightening cutout, but no references to manufacturing standards for confidentiality reasons.

To recognize a bead, the **web_face** or the **wall_face** associated to the web are evaluated to verify the presence of a **bead_bound**. The geometry of a bead is formed by the **bead_bound** and the **internal_faces** that are surrounded by it. The **wall_face** or **web_face** that the **bead_bound** is on forms the feature relationship of the bead with its parent feature. The parameters of a bead are its profile, which is formed by the **bead_bound**, its type (straight or curved) and its first and second radius (designed according to a manufacturing standard). An example is shown Figure 17.

5 AN EXAMPLE

In this section, a real part is used to illustrate the proposed feature recognition process. Figure 18 shows the part, illustrated in Figure 18 (a), and its feature recognition process progressively. The 3D B-rep model of the part, illustrated in Figure 18 (b), is imported. The first major step of classifying and grouping the elements of 3D B-rep model is shown from Figure 18 (c) to (g). First, its faces are classified as **sheet_faces** and **trim_faces**, shown in Figure 18 (c). Then, the **sheet_faces** (on the outside of the part) are classified as their appropriate subtypes, shown in Figure 18 (d). It should be noted that to remain concise the process of classifying **sheet_faces** to their subtypes is not illustrated in detail. Next, the faces are evaluated to identify the potentially existing **joggle_face_sets**, indicated by red circles in Figure 18 (e). The **face_bounds** and **edges**

of all **sheet_face** subtypes are also classified as their subtypes. As an example, the subtypes of **face_bounds** and **edges** of a **wall_face** are illustrated to be classified to their subtypes in Figures 18 (f) and (g).

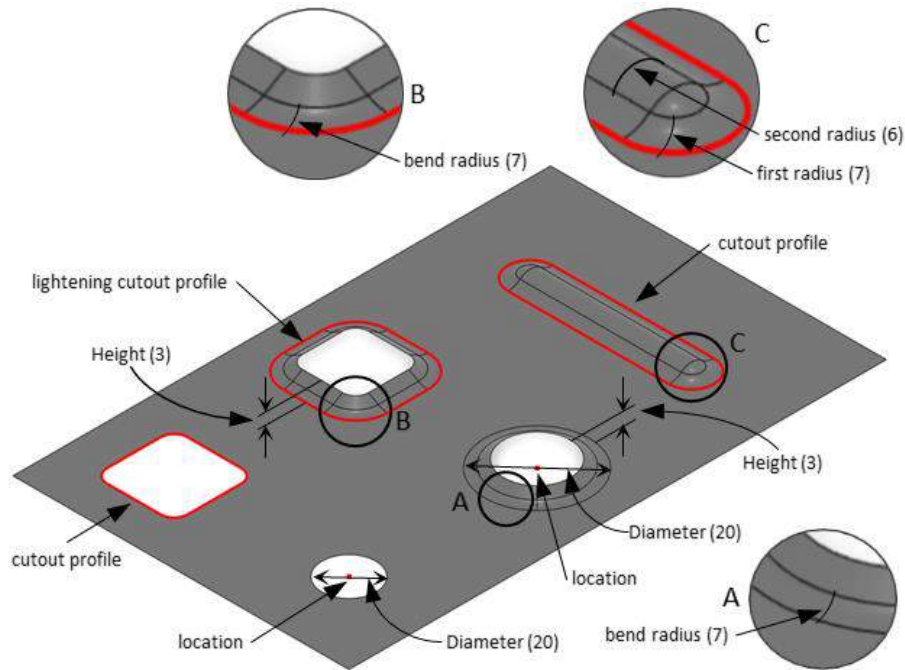
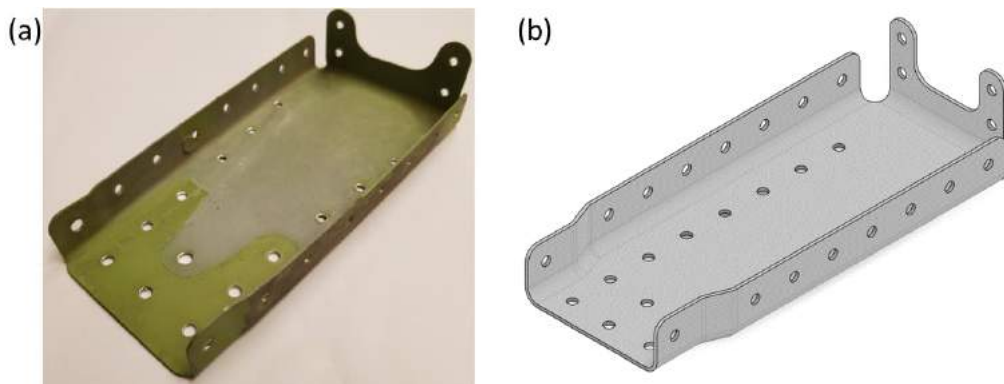
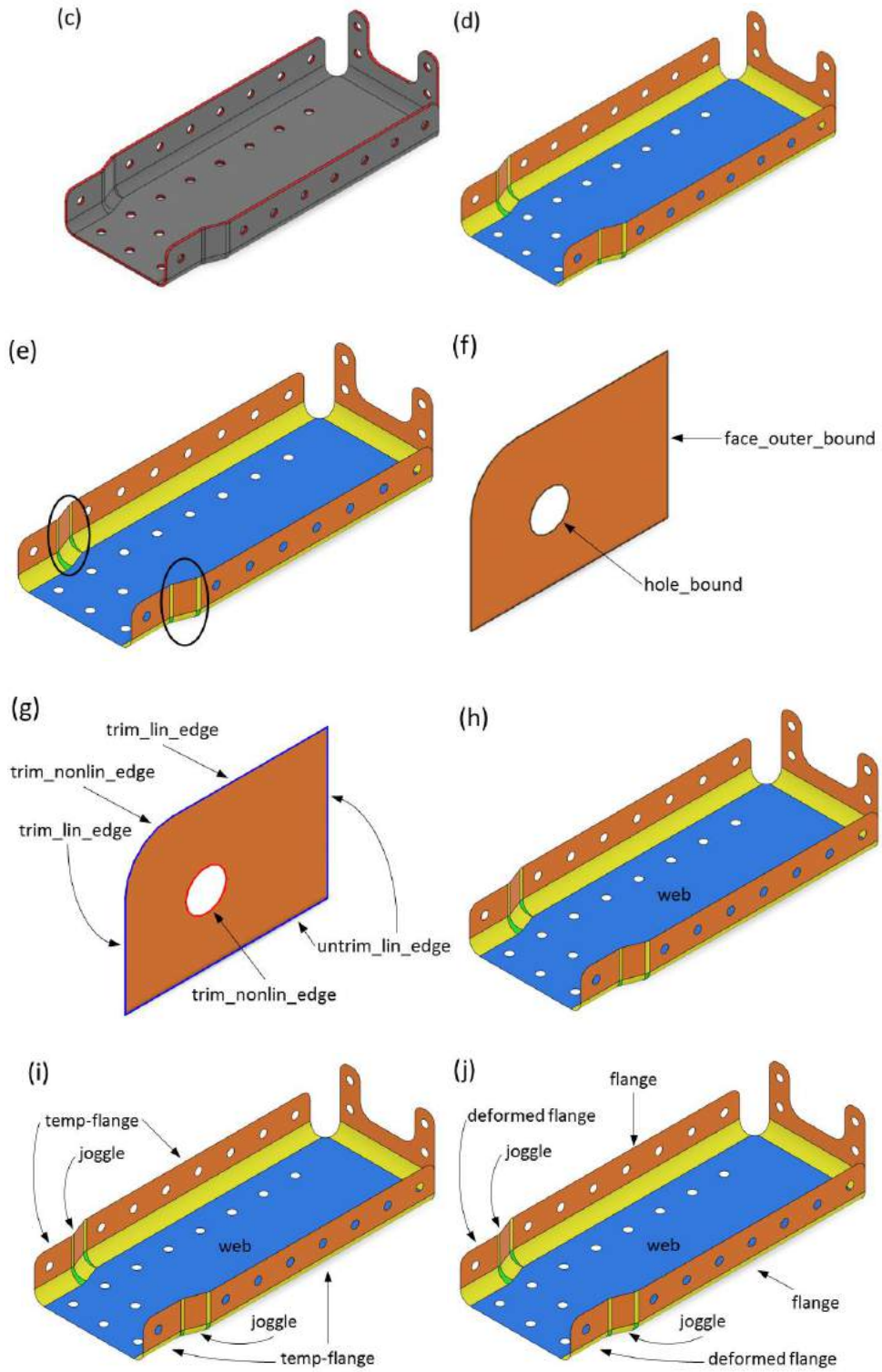


Figure 17: Parameters of a hole, a lightening hole, a cutout, a lightening cutout and a bead.

The second major step of recognizing features is shown from Figure 18 (h) to (l). First the web is recognized, shown in Figure 18 (h). Then, the joggles are recognized along with the temp-flanges, illustrated in Figure 18 (i). The temp-flanges are changed to flanges and deformed flanges, illustrated in Figure 18 (j). Next, the rest of the flanges (those without joggles) are recognized, shown in Figure 18 (k). Once all flanges and deformed flanges are recognized, their child bend reliefs and corners are also recognized. As an example, the bend reliefs and corners of two flanges are pointed out in Figure 18 (l). Eventually the holes are recognized, shown in Figure 18 (l).





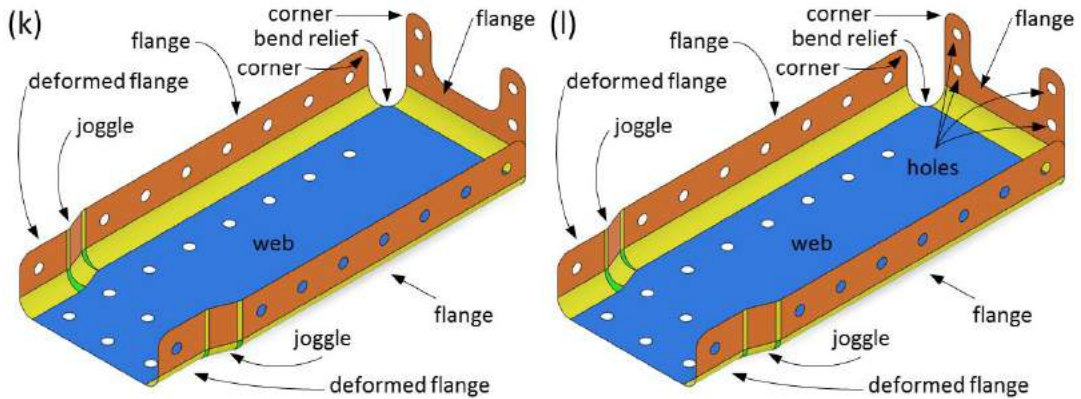


Figure 18: Illustration of feature recognition process for a real part model.

6 CONCLUSION

The main objective of this work is to propose a feature recognition method for structural aerospace sheet metal (ASM) part models, which has not been addressed by previous works. The proposed method is designed to cover all of the features observed from studying real-world structural ASM parts, models and designs, rather than generic sheet metal parts.

The nomenclature of the features and the terminology used in this work are based on the terminology used in the aerospace industry, rather than the terminology utilized in previous academic works. For example, a lightening hole and a lightening cutout are called a collar and an internal flange [27], or an extruded hole [22]. To the best of the authors' knowledge, features like web, joggle, twin joggle, stringer cutout, corner and bend relief, that are used in the design of structural ASM parts, have not been included in the literature, contributing to the novelty and practicality of this paper. On the other hand, features like hem and curl [29], lance and louver [31] are not observed in structural ASM parts and are therefore excluded from the scope of this paper.

A description of ASM features was provided in this work. The taxonomy of the features of structural ASM parts was presented, based on a detailed study of aerospace design guides and 168 actual parts. The studied sample parts were produced by brake-forming or hydro-forming. Skin panels were omitted since they belong to another class of parts. Based on the design guides and the studied samples, two assumptions were made:

1. a web is assumed to be the planar portion of an ASM part with the highest surface area, and
2. all the bends are assumed to have constant bend radii.

The proposed automated feature recognition method consists of two major steps: 1) classifying and grouping the elements of 3D B-rep model, and 2) recognizing aerospace sheet metal features. Through step 1, relevant topological elements of B-rep models are categorized to their subtypes so as to enable step 2.

The two above-mentioned assumptions are satisfied by a great majority of structural ASM parts. Indeed, only 2% of the studied parts did not meet these premises. However, devising an automated feature recognition method able to include these 2% would be a serious challenge to be embarked upon by future work.

The proposed feature recognition process describes the features of a part's model through their geometry, feature relationships and parameters. From there, it becomes possible to describe any ASM part by its features, so as to increase the semantics level of the dialog with the CAD user.

End users are thus envisioned to interact with the feature ASM models like they interact with native CAD models and their operations. Hence, combining the feature structure with the B-rep information could help build fully modifiable feature models. Such feature models could then easily be adapted for new designs. Our next research goal, however, is to compare similar structural ASM parts and express their differences in terms of meaningful features.

Seyedmorteza Ghaffarishahri, <https://orcid.org/0000-0001-9894-6539>

Louis Rivest, <https://orcid.org/0000-0003-0112-0090>

7 REFERENCES

- [1] CATIA 3DEXperience 2017, Dassault Systems.
- [2] CATIA R5-R6 2017, Dassault Systems.
- [3] SOLIDWORKS 2018, Dassault Systems.
- [4] IronCAD 2017, IronCAD LLC.
- [5] NX for Design 2017, Siemens PLM Software.
- [6] Solid Edge 2017, Siemens PLM Software.
- [7] BricsCAD Platinum 2017, Bricsys.
- [8] Creo Parametric 2017, PTC.
- [9] TurboCAD 2017, IMSI/Design.
- [10] Onshape, www.onshape.com, Onshape.
- [11] Alibre Design 2017, Alibre LLC.
- [12] Inventor 2018, Autodesk.
- [13] Ansys SpaceClaim 2017, SpaceClaim Corporation.
- [14] Pratt, M. J.: Introduction to ISO 10303—the STEP standard for product data exchange, *Journal of Computing and Information Science in Engineering*, 1(1), 2001, 102-103.
- [15] Industrial automation systems and integration -- Product data representation and exchange -- Part 42: Integrated generic resource: Geometric and topological representation - STEP 10303-42, International Organization for Standardization, 2003.
- [16] Li, Y. G.; Ding, Y. F.; Mou, W. P. and Guo, H.: Feature recognition technology for aircraft structural parts based on a holistic attribute adjacency graph, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 224(2), 2009, 271-278. <https://doi.org/10.1243/09544054JEM1634>
- [17] Langerak, T. R.: Local parameterization of freeform shapes using freeform feature recognition, *Computer-Aided Design*, 42(8), 2010, 682-692. <https://doi.org/10.1016/j.cad.2010.02.004>
- [18] Verma, A. K. and Rajotia, S.: A review of machining feature recognition methodologies, *International Journal of Computer Integrated Manufacturing*, 23(4), 2010, 353-368. <https://doi.org/10.1080/09511921003642121>
- [19] Babić, B. R.; Nešić, N. and Miljković, Z.: Automatic feature recognition using artificial neural networks to integrate design and manufacturing: Review of automatic feature recognition systems, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 25(3), 2011, 289-304. <https://doi.org/10.1017/S0890060410000545>
- [20] Wang, Q. and Yu, X.: Ontology based automatic feature recognition framework, *Computers in Industry*, 65(7), 2014, 1041-1052. <https://doi.org/10.1016/j.compind.2014.04.004>
- [21] Nnaji, B.O.; Kang, T.-S.; Yeh, S. and Chen, J.-P.: Feature reasoning for sheet metal components, *International Journal of Production Research*, 29(9), 1991, 1867-1896. <https://doi.org/10.1080/00207549108948055>
- [22] Sunil, V. and Pande, S.: Automatic recognition of features from freeform surface CAD models, *Computer-Aided Design*, 40(4), 2008, 502-517.

- [23] Zhang, C.-j.; Zhou, X.-h. and Li, C.-x.: Automatic recognition of intersecting features of freeform sheet metal parts, *Journal of Zhejiang University-SCIENCE A*, 10(10), 2009, 1439-1449. <https://doi.org/10.1631/jzus.A0820705>
- [24] Gupta, R. K. and Gurumoorthy, B.: Automatic extraction of free-form surface features (FFSFs), *Computer-Aided Design*, 44(2), 2012, 99-112. <https://doi.org/10.1016/j.cad.2011.09.012>
- [25] Jagirdar, R.; Jain, V. K.; Batra, J. L. and Dhande, S. G.: Feature recognition methodology for shearing operations for sheet metal components, *Computer Integrated Manufacturing Systems*, 8(1), 1995, 51-62. [https://doi.org/10.1016/0951-5240\(95\)92813-A](https://doi.org/10.1016/0951-5240(95)92813-A)
- [26] Devarajan, M.; Kamran, M. and Nnaji, B. O.: Profile offsetting for feature extraction and feature tool mapping in sheet metal, *International Journal of Production Research*, 35(6), 1997, 1593-1608. <https://doi.org/10.1080/002075497195146>
- [27] Kannan, T. R. and Shunmugam, M.S.: Processing of 3D sheet metal components in STEP AP-203 format. Part II: feature reasoning system, *International Journal of Production Research*, 47(5), 2009, 1287-1308. <https://doi.org/10.1080/00207540701510063>
- [28] Liu, Z.-j.; Li, J.-j.; Wang, Y.-l.; Li, C.-y. and Xiao, X.-z.: Automatically extracting sheet-metal features from solid model, *Journal of Zhejiang University-Science A*, 5(11), 2004, 1456-1465. <https://doi.org/10.1631/jzus.2004.14>
- [29] Kannan, T. R. and Shunmugam, M. S.: Processing of 3D sheet metal components in STEP AP-203 format. Part I: feature recognition system, *International Journal of Production Research*, 47(4), 2009, 941-964. <https://doi.org/10.1080/00207540701510055>
- [30] Jagirdar, R.; Jain, V. K. and Batra, J. L.: Characterization and identification of forming features for 3-D sheet metal components, *International Journal of Machine Tools and Manufacture*, 41(9), 2001, 1295-1322. [https://doi.org/10.1016/S0890-6955\(01\)00006-2](https://doi.org/10.1016/S0890-6955(01)00006-2)
- [31] Gupta, R. K. and Gurumoorthy, B.: Classification, representation, and automatic extraction of deformation features in sheet metal parts, *Computer-Aided Design*, 45(11), 2013, 1469-1484. <https://doi.org/10.1016/j.cad.2013.06.010>
- [32] Niu, C.: *Airframe structural design: practical design information and data on aircraft structures*, Conmilit Press Limited, Hong Kong. 1999.
- [33] Watkins, M. A.: Problems in geometric continuity, *Computer-Aided Design*, 20(8), 1988, 499-502. [https://doi.org/10.1016/0010-4485\(88\)90012-7](https://doi.org/10.1016/0010-4485(88)90012-7)