Automatic Tool Selection in V-bending Processes by Using an Intelligent Collision Detection Algorithm

A A Salem
Department of Mechanical Design and Production, Faculty of Engineering, Cairo University, Giza, Egypt.

amrabdel-alim@edu.cu.eg

Abstract. V-bending is widely used to produce the sheet metal components. There are global changes in the shape of the sheet metal component during progressive bending processes. Accordingly, collisions may be occurred between part and tool during bending. Collision-free is considered one of the feasibility conditions of V-bending process planning which the tool selection is verified by the absence of the collisions. This paper proposes an intelligent collision detection algorithm which has the ability to distinguish between 2D bent parts and the other bent parts. Due to this ability, 2D and 3D collision detection subroutines have been developed in the proposed algorithm. This division of algorithm's subroutines could reduce the computational operations during collisions detecting.

1. Introduction
Industrial companies require quick products with high quality levels and low costs to meet the high levels of competition in today's market demand. This need demands efficient product's process plans which reduce the production time by minimizing the non-added-value activities. Correspondingly, the subtle bending tool selection is one of the goals of the efficient V-bending process planning. Collisions may be occurred during the bending process due to the global changes of the shape of the sheet metal and these collisions are considered as one of the infeasibility conditions of the V-bending process planning. Consequently, to avoid these collisions, the collision detection algorithm must be followed. Much of the literature on the V-bending process planning have detected the collisions by using external collision detection library or by developing a collision detection algorithm. The collision detection algorithms are divided into three generic categories [1]. In the first one, collisions are detected in situations between numbers of convex polyhedral. Meanwhile, space occupancy method has been used in the second category. Besides, the sweep volume calculation method is used between the moving objects in last category. In [2] and [3], collisions are detected in certain situations (before and after the bending process) of the bent parts which have only parallel bend lines. Consequently, collisions-free bending process verifies the selection of the bending tool in a certain process plan. The Space occupancy method is used to detect the collisions in [4] by subtracting the volume of the bent part of the volume of the tools, and then, if the volume of tool is divided into two solid volumes, the bent part could collide the tool. Geometrical constraints have developed in [5] to avoid collisions between the part and tools and these constraints are considered a pre-selection stage of the tool selection algorithm to minimize the change of tools during bending processes of one part. On contrary to [5], geometrical constraints have used in [6] to design the collision-free bending tool for a group of bent parts. However, much collision detection algorithms have been developed, they don't pay much
attention to the relations between bend lines and the benefits of these relations on controlling the computational loads of the detecting process. The proposed intelligent collision detection algorithm doesn't detect only the collisions during bending the bent parts which have only parallel bend lines, but also detect the collisions of the bent parts which have bend lines in different directions. Moreover, it has ability to recognize the relations between the bend lines in sheet metal part. Accordingly, there are two subroutines in this algorithm which are 2D collision detection subroutine for the bent parts which have only parallel bend lines (2D bent parts) and 3D collision detection subroutine for the other bent parts (3D bent parts). This division could reduce the computational loads by detecting collisions of 2D bent parts without need of performing much computational operations as in 3D collision detection subroutine. Besides, the proposed algorithm uses simplified intersection checks which depend on the parametric equations.

2. Tool selection in V-bending

The V-bending process is carried out on the press brake by using its movable and static parts where the punch press down the sheet metal to the die with enough tonnage. There are different types of V-bending tools such as sash, gooseneck, and narrow relief punches and different V-dies. Accordingly, to select a punch and die to perform V-bending of certain bend line, the tool must satisfy the technological constraints as described in [5]. After satisfying the technological constraints, these tools must lead to collision-free bending process. Due to the interference between the bending sequence and the tool selection of V-bending process, a collision could be occurred by using a specific bending tool set and could not be occurred by using another tool. For instance, the collision detection process is performed in V-bending processes on bent part which shown in figure 1(a) by using two different bending punches which shown in figure 1(b). Even though there is a collision during bending b2 by using first punch according to the bending sequence {b1, b3, b2} as shown in figure 1(c), there aren't collisions by using the second punch according to the same sequence as shown in figure 1(d).

![Figure 1](image)

**Figure 1.** (a) sheet metal part, (b) two different bending punches, (c) collision in bending b2, (d) collision-free in bending b2.

3. The proposed collision detection algorithm

The proposed collision detection algorithm is developed using the C++ programming language under Windows operating system. This algorithm depends on detecting the collisions during static situations of the bending processes. The following flowchart which shown in figure 2 presents the operations of the proposed algorithm. As presented the first operation is the data extraction of the bent part from STEP-AP203 format file, and then the extracted geometrical data of the bent part are the input data to the feature recognition and reasoning system as described in [7]. This feature recognition system produces the central surface of the flat pattern of the part which is used in collision detection algorithm. The algorithm proposes bending sequence and tool list to perform bending of the part where the feasibility of proposed sequence and tool list is tested by the proposed collision detection algorithm. Bend lines are bent one by one due to the bending sequence and the algorithm detects the collisions of each bending process.
The bending of the bend line includes two operations, which are the transformation of bend line from the planar surface (flat pattern of the bend line) to the cylindrical surface and the rotation of the two portions of the bent part which adjacent to this bend line by a half of the bending angle. After bending operation is finished, the setup operation is started. The setup operation is the operation of the positioning of the bent part and tool set as in real bending. Then, the collisions are detected by 2D or 3D collision detection subroutine. As soon as the bent part is classified as 2D bent part, collisions are detected by 2D subroutine. Otherwise, collisions are detected by 3D subroutine.

3.1. 2D collision detection subroutine

As the 2D bent part could be represented by its projection in 2D plane, the collisions between 2D bent parts and bending tool could be presented in 2D projection of this bending process. On the contrary to 2D bent parts, the bending of 3D bent parts couldn't represented by the projection in 2D plane, because collision between faces which don't align to bending tool axis and bending tool couldn't be represented in 2D plane projection. Due to the projections of part and bending tools consist of line and arc edges, the collisions between tools and bent part are the intersections between any line of bent part projection and any line or arc of bending tool projection as shown in figure 1(c). Firstly, this subroutine extracts the projections of bending tools and part from the geometrical data. These projections consist of any edge of tool and part which doesn't align to the tool axis. After the projections extracting, the detection process for line-line intersection or line-circle (arc) intersection will start.

In line-line intersection, there are only two relations between lines in 2D plane, the first one is parallelism relation and the second one is intersection relation. If the slopes of two lines are equal to each other, these lines are parallel lines and couldn't intersect with each other. Unless the slopes of lines are equal, these lines are intersecting lines. The intersection point is the point which could satisfy the equation line of two intersecting lines. In line-circle (arc) intersection, the 2D collision detection subroutine uses the perpendicular distance between the center of the circle (the circle of this arc) and the line. If this distance is shorter than the radius of the circle, there would be intersection between the line and the circle.

![Figure 2. flowchart of the intelligent collision detection algorithm](image-url)
3.2. 3D collision detection subroutine

If the bent part collides the tool set on the press brake, at least one face (planar polygon) of this part intersects with one polygon (planar surface or cylindrical surface) of the tool set. Consequently, to detect the collisions, intersection between two polygons in 3D must be detected. As soon as there is a collision between two polygons in 3D, at least one edge of one polygon must intersect the other polygon and versa vice as shown in figure 3. Parametric equation of line edge, planar surface, and cylindrical surface, which are presented in equations (1), (2), (3) respectively, are used to determine the intersection point. Where \((x_a, y_a, z_a)\) and \((x_e, y_e, z_e)\) in equation (1) are the coordinates of the end points of line edge, and the parameter \(t\) represents the position of any point lies between the end points, and \((a, b, c, d)\) in equation (2) are determined from normal vector to the planar surface and one point lies on it. As the tool is aligned to x-axis, equation (3) presents the equation of cylindrical surface which is aligned to x-axis, and \((y_c, z_c)\) are the coordinates of the center of this surface and \(r\) is the radius of it. Just as the collision is occurring if the line edge (line segment) intersects the polygon (don't only intersect the surface of the polygon), checking process is performed to specify if the intersection point inside the line segment and inside the polygon.

To get an intersection point between line and plane, the coordinates of the equation (1) are substituted in equation (2), then the parameter \(t\) could be determined. The coordinates of the intersection point are determined after substituting by the parameter \(t\) in the equation (1). Similarly, the intersection points between line and cylindrical surface could be determined. There are two values of the parameter \(t\) are gotten by solving this equation because there are two intersection points between line edge and cylindrical surface. After getting the intersection point between line edge and planar surface, the location of intersection point must be specified which is inside or outside this polygon. Figure 4 presents the steps of the test of the intersection point location which inside or outside planar polygon.

First, the planar polygon which presented in figure 4(a) is divided into a triangle as shown in figure 4(b), and then the subroutine starts with one of these triangles with area A as shown in figure 4(c). The area of any triangle is determined by half of the cross product of any two vectors of the triangle's sides. There are three triangles which are consisted of intersection point and start point and end point of every edge of the triangle. The areas of these triangles are D, H, and G. As soon as the summation of these areas of these triangles \((D+H+G)\) equal to the area of the original triangle, the intersection point is inside the original triangle (inside the polygon) as shown in figure 4(d). On the other hand, if this summation is larger than the area of the original triangle, the intersection point is outside the original triangle as shown in figure 4(e). This examination is applied to the divided triangles one by one.

Finally, if the intersection point is outside all of the divided triangles from the original planar polygon, there isn't intersection between a line and planar polygon. In case of the one of the tool faces is a cylindrical surface polygon, there are two intersection points as mentioned above. If the coordinates of one of the intersection points are located within the coordinate between arc coordinates and line coordinates of the cylindrical polygon, this intersection point inside the cylindrical surface polygon.

\[
(x, y, z) = (x_a, y_a, z_a) + t (x_e - x_a, y_e - y_a, z_e - z_a)
\]

\[
ax + by + cz = d
\]

\[
(y - y_c)^2 + (z - z_c)^2 = r^2
\]
Figure 4. steps of the test of the intersection point location which inside or outside planar polygon: (a) planar polygon, (b) dividing the planar polygon into triangles, (c) one selected triangle with area A, (d) in case of intersection point is inside selected triangle, (e) in case of intersection point is outside selected triangle

4. Conclusions
The proposed intelligent collision detection algorithm in this paper is used to select the V-bending tools automatically by ratifying the feasibility condition of V-bending process planning. This algorithm uses the relations between bend lines to classify any bent part as a 2D or 3D bent part. The collisions between the 2D bent parts and the bending tools could be detected in 2D plane rather than in 3D. Accordingly, this classification of bent parts could reduce the computational loads of the collision detection processes. 2D and 3D collision detection subroutines have been developed in the proposed algorithm to detect the collisions during bending according to the type of bent part. In the future work, suitable optimization technique will link to this collision detection algorithm to search for an optimized bend sequence and bending tools.

5. References
[1] J. R. Duflou, J. Váncza, and R. Aerens, “Computer aided process planning for sheet metal bending: A state of the art,” Comput. Ind., vol. 56, no. 7, pp. 747–771, Sep. 2005.
[2] A. C. Lin and C. Chen, “Sequence Planning and Tool Selection for Bending Processes of 2.5D Sheet Metals,” Adv. Mech. Eng., vol. 2014, 2014.
[3] J. C. Rico, J. M. González, S. Mateos, E. Cuesta, and G. Valiño, “Automatic determination of bending sequences for sheet metal parts with parallel bends,” Int. J. Prod. Res., vol. 41, no. 14, pp. 3273–3299, 2003.
[4] M. Engineering, “Sheet Metal Bending Operation Planning : Using Virtual Node Generation to Improve Search Efficiency,” J. Manuf. Syst., vol. 18, no. 2, pp. 127–139, 1999.
[5] J. R. Duflou, T. H. M. Nguyen, and J.-P. Kruth, “Intelligent tool pre-selection: A contribution to automatic process planning for sheet metal bending,” in Proceedings of the TMCE 2004, April 13-17, 2004, pp. 671–682.
[6] U. Alva and S. K. Gupta, “Automated design of sheet metal punches for bending multiple parts in a single setup,” Robot. Comput. Integr. Manuf., vol. 17, no. 1–2, pp. 33–47, Feb. 2001.
[7] A. A. Salem, T. F. Abdelmaguid, A. S. Wifi, and A. Elmokadem, “Towards an efficient process planning of the V-bending process: an enhanced automated feature recognition system,” Int. J. Adv. Manuf. Technol., 2017.