Path Generation for Rapid Filling of Planar Areas with Optimized Parallel Curve

Shengfan Cao, Ze Wang, Chuxiong Hu, Yu Zhu

Abstract— **Path generation is an essential part of CNC manufacturing, and some manufacturing processes prefer efficiency to accuracy. Traditional ways of path generation often lack continuity or contain numerous sharp turns those both significantly reduce efficiency. In this article, the potential of trading off marginal accuracy for higher efficiency is explored, and a novel method is developed to optimize the parallel curve strategy regarding sharp turns and continuity with minimized waste of filling materials. First, two types of self-intersections, global self-intersection (GSI) and local self-intersection (LSI) are identified. Based on medial axis extraction, the geometric prerequisites for GSI are discovered, and an algorithm is proposed to eliminate GSI with area reconstruction. For LSI, the connection with the curvature is exploited, and a method is developed to effectively identify the sharp turns without extra calculations. Then, a corner smoothing method is applied to improve the smoothness of the trajectories. Furthermore, a way to transform the set of parallel curves into a single spiral curve with offset is proposed to further enhance the efficiency. The proposed strategy to generate a smooth and continuous trajectory is valuable to various planar shapes and can be applied in various manufacturing scenarios.**

I. INTRODUCTION

Layered manufacturing is a technique to manufacture workpieces with sophisticated shapes. 3D printing technology, as one of the most representative form of layered manufacturing, is prominent for its freedom on design and minimizing material wastes [1], [2]. In 3D printing, the 3-dimensional workpiece is first sliced using a series of parallel planes to derive the cross-sections on each plane [3]. Then, a path generation algorithm is implemented to derive a tool path for each cross-section for a cutter or an extrusion head to track and traverse the designated planar area. In recent years, there exist manufacturing systems where the additive manufacturing (AM) technology is integrated with traditional subtractive manufacturing systems, constituting the hybrid manufacturing (HM) systems [4], [5]. The AM process in the hybrid manufacturing systems deviates from the ordinary additive manufacturing systems, in that the primary purpose of the AM process is to provide a work blank which is machined to have better dimensional and surface accuracy. Therefore, the accuracy of the AM steps in HM systems can be reasonably relaxed without negatively affecting the accuracy of the final product. To maximize the efficiency on the level of the trajectories, a custom-made strategy for the HM systems should be developed.

The strategy for the tool path generation has a significant impact on the physical properties of the generated parts, such as density and stress in the part [6]. Also, the curvature of the tool path can affect the efficiency. Under restrictions on maximum velocity, acceleration, and jerk, a trajectory with a smaller curvature allows higher trajectory speed while reducing tracking error [7], [8]. The most traditional strategy is the zigzag pattern [9] where the area is filled with straight lines completed back and forth. Similarly, there exist strategies where grids [10] and polygons are used to fill the area. These strategies are easy to implement and mostly robust against irregular shapes. However, the excessive number of sharp turns significant reduces the trajectory efficiency, especially for irregularly shaped outlines. Also, using grids and polygons to fill the area causes the trajectory to intersect with itself, resulting in large overlap and unfilled areas.

To mitigate the loss of trajectory speed due to the sharp turns, parallel curves were introduced into the field of path generation. Strategies based on parallel curves stand out with their relatively lower curvature throughout the entire trajectory. Also, these strategies are capable of filling the plane with minimum overlap and unfilled area with much fewer sharp turns and is more efficient regarding the trajectory speed [11], [12]. With offsetting the preliminary curve iteratively [13] and eliminating the self-intersections [14], one can derive a complete set of curves that fills the area. The strategy is a common practice in cutter compensation for computer numerical controlled (CNC) machine tools. However, the performance of a strategy purely relying on the parallel curves is not robust to the given outline. According to the given outline, the area may break into multiple disconnected sub-domains during offsetting. To address this problem, Hilbert curves were developed based on parallel curve strategy [15], [16]. The offsetting procedure is refined into a tree structure according to the separation of the area during the process. The trajectory is then rescheduled in a bottom up manner to find a continuous path that can traverse the area with one touch. The Hilbert curve strategy significantly boosted the robustness of the strategy. Nevertheless, the improvement is at the cost of significantly more sharp turns in the trajectories. Also, as is discussed in more depth in this article, the parallel curve strategy will by nature cause sharp turns to appear along the generated trajectory. Completely smooth trajectories with little overlap and unfilled areas remain elusive.

This article explores the potential of trading off marginal

This work is supported in part by National Nature Science Foundation of China under Grant 51922059 & 51775305 and Beijing Natural Science Foundation JQ19010.

All authors are all with the State Key Lab of Tribology, Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China, and also with the Beijing Key Lab of Precision/Ultra-Precision Manufacture Equipment and Control, Tsinghua University, Beijing 100084, China.

Corresponding author: Chuxiong Hu (cxhu@tsinghua.edu.cn).

accuracy for higher efficiency, i.e., strategies for path generation if the restriction on the margin is relaxed. The proposed algorithm generates continuous and smooth trajectories that ultimately lead to higher trajectory speed by allowing the generated tool path to slightly overflow the original outline. More specifically, the goal is to generate trajectories with the following characteristics:

1) able to fill the area evenly, controlling the minimum overlapping or unfilled area;
2) continuous, and therefore can be completed without tool retraction;
3) smooth, without non-differentiable points;
4) cause minimum overflow while ensuring the features above.

The proposed method is entirely based on parallel curves where various useful properties are derived during the offset procedure. Section II gives an overview of the parallel curves derivation and self-intersection elimination. The self-intersections are further categorized into two types, i.e., GSI and LSI, which leads to Section III and Section V where optimizations are performed for both types. It is discussed in Section III that GSI can be eliminated by breaking the prerequisites. From a special case to more general scenarios, the geometric prerequisites for GSI are derived based on medial axis extraction and width analysis. An algorithm to eliminate GSI is then presented by area reconstruction from the medial axis with optimized width. In contrast, LSI cannot be eliminated. Nevertheless, Section V shows that the relationship between LSI and the curvature makes it possible to identify the sharp turns during offset procedure without explicit examinations. A corner smoothing algorithm can be directly implemented on segments producing LSI to control the maximum curvature. Furthermore, Section IV presents a trick to transform the set of curves into a single spiral curve, eliminating tool retractions and further enhancing the trajectory speed.

II. PARALLEL CURVES AND SELF-INTERSECTIONS

Parallel curves is a set of curves where the minimal distance between two adjacent curves are equivalent for any point on the curves. This property ensures uniform coating of the given area. The operation to derive the parallel curve is called offset. When offsetting a curve, one may derive the induced parallel curve as

$$l' = \{\vec{r}_P + d \times \vec{n} | P \in l\} \tag{1}$$

where d is the offset distance, and $\vec{n}$ is the perpendicular unit vector pointing to the inside of the enclosed area. However, offsetting a curve causes the curvature radius of all points to decrease by the same value, which inevitably causes sharp turns to appear when operating on the margin of a closed region.

A correct tool path should not be self-intersecting to maximize efficiency, as well as to avoid collision between the extruded material and the tool. To be non-self-intersecting, the induced parallel curve must satisfy

$$\min_{P' \in l'} \left(\min_{P \in l} D(P', P) \right) = d \tag{2}$$

where l is the original curve (or the preceding curve), l' is the induced parallel curve (or the succeeding curve), $D(P_1, P_2)$ is the Euclidean distance between P_1 and P_2, and d is the offset distance.

With an algorithm based on the winding number, it is possible to distinguish the self-intersections within $O((n+k)\log n)$ time [17]. After removing points not satisfying Eq. (2), one can derive the parallel curve of the given outline that has no self-intersections.

The self-intersecting segments can be further classified into the following two categories.

- **Local self-intersections (LSI):** after removal, the remaining curve remains one connected region.
- **Global self-intersections (GSI):** after removal, the remaining curve breaks into multiple separated enclosed areas.

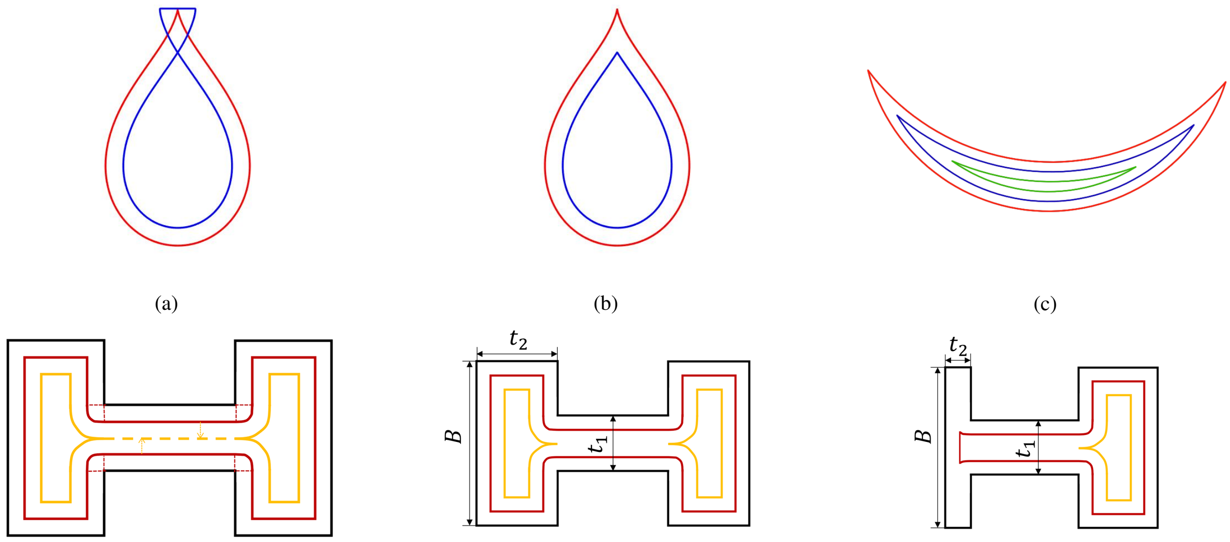
Fig. 1 provides examples of LSI and GSI. Note that the existence of self-intersections are not directly related to the convexity of the outline. More details about this note will be presented in Section III.

It should be pointed out that the tool path after the elimination of self-intersections is still not ideal. First, the trajectory is composed of a series of closed loops, making it impossible to complete the trajectory with one touch. Second, there are sharp turns on the trajectory that inevitably appear as the curves are offset further inwards. Third, the area may break into several non-connected domains for certain outlines. A typical outline that may lead to such problems is a dumbbell with a thin stick connecting two large balls, which is shown in Fig. 1(e).

III. GSI AND PREVENTION

Global self-intersections cause the remaining curve to become multiple separated enclosed areas after removal instead of one continuous curve. Curves with global self-intersections are impossible to complete without tool retractions, reducing the evenness of the upper surface of the manufactured part and severely impairing efficiency. Therefore, an ideal trajectory should be devoid of global self-intersections to allow completion with one touch.

One of the prerequisites for global self-intersection is a narrow passage whose width is below the offset distance d, a condition only satisfied in concave segments, allowing the passageway to widen to a distance to produce valid successors. In other words, global self-intersection does not exist in convex areas. On the other hand, an area does not have to be convex to prevent global self-intersections. For example, the non-convex banana-shaped area in Fig. 1(c) does not cause any global self-intersections. Therefore, it is necessary to find a tighter condition to prevent global self-intersections.



(a) (b) (c)

(d) (e) (f)

Fig. 1. Examples of LSI and GSI. (a) Local self-intersections (LSI). (b) The removal of LSI. (c) Banana-shaped outline and its parallel curves (d) Global self-intersections (GSI) (e) The removal of GSI (f) "H-shaped steel" and its parallel curves. Note that the existence of LSI and GSI are not directly related to the convexity, e.g., the non-convex outline (c) and (f) are both devoid of GSI.

A. GSI: Special Situation ("H-shaped steel")

Consider the cross-section of a "H-shaped steel". Figure 1(e) or Figure 1(f) shows the generated trajectory using the traditional parallel curve generation strategy.

In the case of the "H-shaped steel", the existence of global self-intersections indicates that the horizontal part would shrink down before the vertical parts disappear, like a gate cutting the area in halves. Therefore, one of the two following requirements should be satisfied to prevent GSI, i.e.,

$$B < t_1 \tag{3}$$

$$t_2 < t_1 \tag{4}$$

Eq. (3) suggests that on the "main road", the width must not have a local minimum. Eq. (4) indicates that if there exist "branches" along the "main road", all "branches" must be narrower than the "main road" so that they shut off before the "main road" does, preventing the area from splitting up. As shown in Fig. 1(f), a "H-shaped steel" satisfying the criteria does not generate global self-intersections. In this sense, for outlines that does not match the criteria, the area can be expanded by increasing t_1 such that

$$t_1 \geq \min(B, t_2) \tag{5}$$

B. GSI: General situation

Now consider a general situation and find the equivalence to the "H-shaped steel" situation above, so that any given curve can be optimized using similar methods. The key is to extract the features from any given area.

The area is first characterized with its medial axis, which can be defined by the offset process, and the width for each point on the medial axis [18], [19]. The medial axis and the width are sufficient to reconstruct the area. Deriving the medial axis by the offset process is similar to igniting the margin of the outline simultaneously, letting the fire spread in all directions at the same rate at any point. During the process, points on the original curve converge at various points on the surface, constituting the medial axis of the given area. Therefore, the medial axis can be derived by iterating the offset procedure until no point on the curve can generate valid successors. For instance, Fig. 2 provides a random curve and its medial axis. The black curve is the tool path to traverse the area generated using the parallel curve strategy. The red line represents the medial axis of the area.

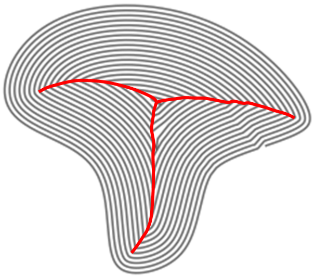


Fig. 2. The medial axis of a random curve.

Each point P on the medial axis corresponds to at least two points P_1 and P_2 on the original curve. The connecting lines between P and $P_1(P_2)$ are perpendicular to the tangent vector on the original curve at $P_1(P_2)$, and their lengths would be equivalent. The shared distance is defined as the width for each point P on the medial axis, i.e.,

$$w(P) = d(P, P_1) = d(P, P_2) = \min_{P_0 \in l} \left(d(P, P_0) \right), P \in M \tag{6}$$

Global self-intersections are closely related to the width. Offsetting a curve causes all points on the original curve to move by the same distance, indicating that the relationship between the width and the existence of GSI remains unchanged after offsetting. Therefore, with proper preprocessing, global self-intersections can be examined and eliminated before the main offsetting procedure begins.

The derived medial axis is typically a tree-like structure, with edges reaching out from multiple ends and converging in the central part of the domain. Tree nodes are the ends and the converging points on the medial axis, and edges are the segments of the medial axis that connect the nodes. Any two adjacent nodes and the edge connecting the two nodes constitute a minimal unit. Each minimal unit is analogical to the special case "H-shaped steel" in Section III-A. Parameter B is the width of the nodes, while t_1 and t_2 are the minimum width of the "main road" and the width of the "branches" respectively. The width of the "branch" refers to the maximum width from the node down to every leaf node.

Note that selecting different paths as the "main road" may lead to different results. To minimize the adjustment, a greedy policy can be adopted to pinpoint the point with the maximum width as the root of the tree. After selecting the root, one may proceed to traverse the entire graph and build a tree. Initially, each node records the width, and the maximum and minimum width in the edges are recorded.

To avoid global self-intersections, $t_2 < t_1$ must be satisfied at every intersection, ensuring the dominance of the "main road" regarding width. If not, the "main road" must be widened by increasing the width of the edge to meet the following criteria:

- the nodes closer to the root has larger widths;
- the width along the edge falls between the width of the two nodes.

Since the point with the maximum width has been selected as the root and the leaves of the tree are all endpoints on the graph, the criteria above can be met by passing the maximum width from bottom up to the root. Also, the width along the edges must be truncated to fall within the width of the two nodes. After all the edges are examined and adjusted, the area can be reconstructed according to the medial axis and the width. This procedure eliminates global self-intersections fundamentally.

Fig. 3 shows the complete process of eliminating global self-intersections for a random curve where the number refers to the width at each point. The example gives a bad case in GSI elimination, that is, the ends have much greater width than the middle parts. With the proposed method, by increasing the width along the segment on the medial axis, the property of non-decreasing width is ensured and therefore not causing GSI. Note that some short branches with small width may be overridden by the expanded 'main road'. In other words, the actual medial axis used in reconstruction may differ from the original curve.

IV. TRANSFORMATION INTO SPIRAL CURVES

Offsetting a curve causes the curvature radius to change by d without altering its curvature center and retains the tangent vector at each point. As part of the pre-processing process, this property is fully utilized by performing a special offset procedure on the original margin to transform the original margin, a loop, into an initial spiral curve with overflow allowed. More specifically, one can select a point with a sufficiently high curvature radius as the entry point, and offset each point on the curve with an offset distance that changes continuously from $\frac{d}{2}$ to $-\frac{d}{2}$. After eliminating self-intersections, the initial curve is transformed into a spiral curve. By continuously performing offsetting on the pre-processed outline, the ends smoothly connect, and the segments are integrated into one continuous trajectory. The pre-processing process should be performed before the optimization proposed in Section III-B to ensure that the correct medial axis and width are derived.

Fig. 4 shows the result of the proposed transformation. The example is to pre-process the points sampled from an oval before deriving the set of parallel curves with distance $d = 0.2$. The entry point is selected to be the point with the minimum curvature, and the offset distance in the pre-processing stage for each point varies uniformly from 0.1 to -0.1. Note that both ends share the same perpendicular vector, and therefore the induced parallel curve from the pre-processed preliminary curve would smoothly connect, leading to a perfect spiral curve.

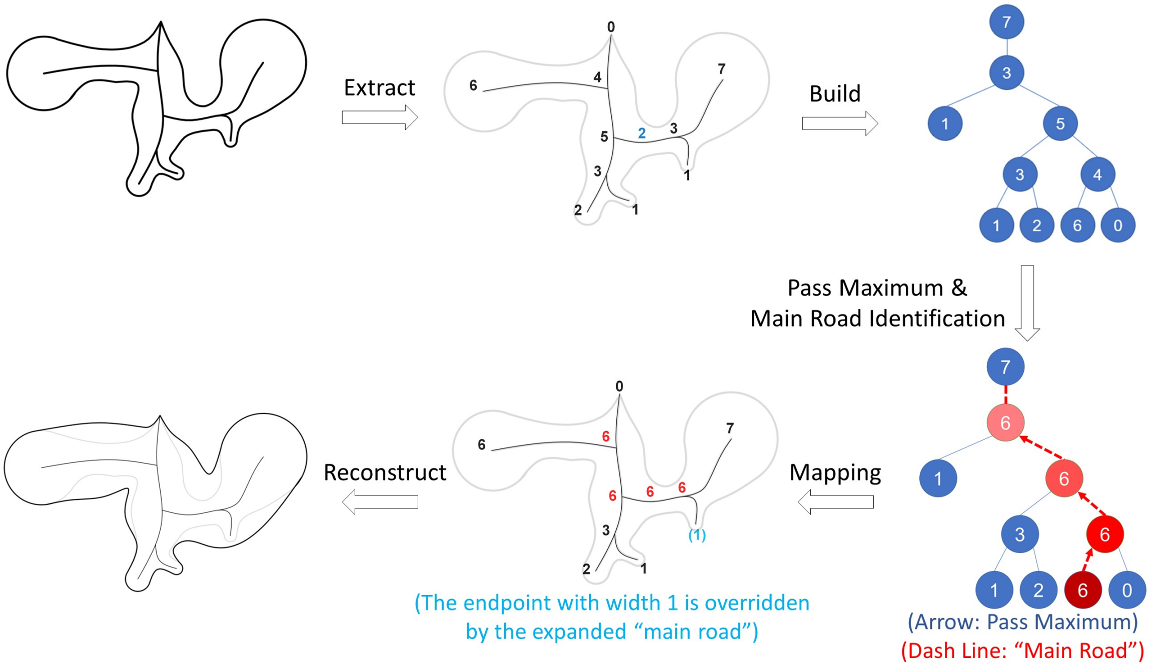
V. LSI AND OPTIMIZATION

Contrasting GSI, local self-intersections (LSI) are related to the curvature radii of individual points. This type of self-intersections is unavoidable, so instead this section focuses on ways to increase the smoothness, improving the trajectory speed without undermining the completeness and evenness of the trajectory.

With the pre-processing procedure given in Section III eliminating global self-intersections, the self-intersecting segments now contain all sharp turns. Therefore, the checking procedure could capture all the sharp turns without explicitly checking the curvature radius. Therefore, it is possible to adjust these segments with no successors to have better properties (e.g., curvature with lower value and gradient) without affecting the property of equidistance for the rest of the curve.

Theorem 1: In segments that causes local self-intersections, at least one point with a curvature radius below d exists. Vise versa, the segments on the original curve with a smaller curvature radius than the parallel distance lead to the local self-intersection.

Proof: Suppose the curvature radius is above d for any point on the segment causing local self-intersections. After the offset and before the checking procedure, all points have a curvature radius greater than 0. In this case, the direction of the curve remains the same for the new curve, which rules out the local self-intersections and leads to a contradiction. ■

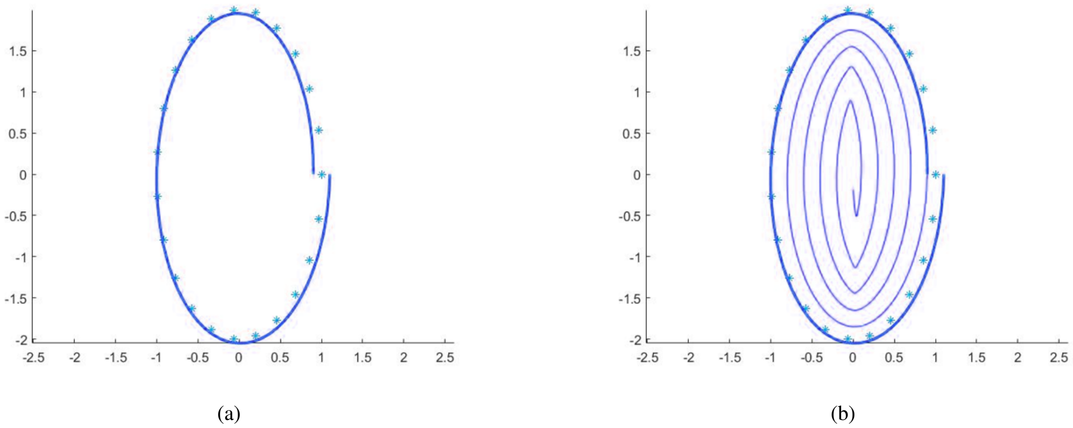


(The endpoint with width 1 is overridden by the expanded "main road")

(Arrow: Pass Maximum)

(Dash Line: "Main Road")

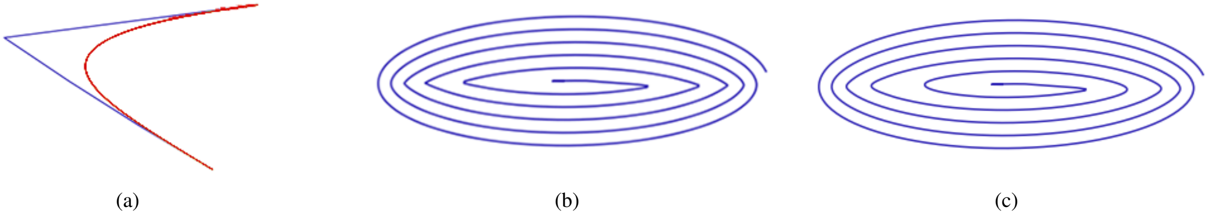
Fig. 3. Complete process of global self-intersection elimination



(a)

(b)

Fig. 4. Transformation from enclosed loop to spiral curve. (a) Initial spiral curve for an oval (Asterisk: Original Outline; Solid Line: Pre-processed Outline (b) Oval Filled with Spiral Curve



(a)

(b)

(c)

Fig. 5. Proposed LSI Optimization. (a) Corner Smoothing for LSI. (Blue curve: Original. Red curve: Optimized.) (b) Oval before proposed optimization. (c) Oval after proposed optimization.

Consider a segment only with local self-intersections with at least one sharp point with a curvature radius below d. The techniques for corner smoothing can be applied to control the maximum curvature throughout the segment. Corner smoothing is a separate field with numerous existing research, and we omit the in-depth analysis of its actualization here. Here, we perform an interpolation with a non-uniform rational B spline (NURBS) [20], [21], with the points describing the segment to be optimized as the control points. Fig. 5(a) shows a segment about to generate a local self-intersection before and after corner smoothing with blue and orange lines respectively. Regarding the code implementation, it is possible to put the optimizer on a different thread and process it simultaneously to improve efficiency, since optimization does not affect the offsetting and checking procedure.

Fig. 5(b) shows the spiral curve for an oval without the proposed optimization for local self-intersections, while Fig. 5(c) shows the optimized curve. It should be noticed that, the optimization procedure does not affect the curve where the curvature radius is large enough, and preserves the benefits of using parallel curves to fill an area evenly. Optimization is executed along the longer axis of the oval where the curvature radius is the smallest, using interpolation as the corner smoothing method.

VI. Conclusions

This article has presented a trajectory generation strategy that produces a continuous and smooth trajectory for the outline of any given closed planar region. The method is especially suitable for manufacturing steps using planar manufacturing where the marginal accuracy is not the primary concern. With the proposed method, one can derive a trajectory to fill a region no smaller than the given closed region with minimum overflow. The derived trajectory is significantly easier to track with higher speed and less error at the cost of a slightly longer trajectory than required to fill the region as is. The work includes theoretical and numerical fields which are of interest to several communities of scientists and engineers. This method is anticipated to help enhancing the efficiency in larger systems beyond state-of-the-art devices and techniques. The pseudo-code for the complete process of the proposed is given in the appendix. For future studies, the performance of the proposed method and its connection with the geometrical characteristics of the given outline may worth further analysis.

Appendix

A. The Pseudo-code for the proposed method

Algorithm 1 Offset, Check, and Optimization

```
Require: $Curve$, $d$
Initial Offset for $Curve$ to transform into a spiral curve
$CurrentLayer \leftarrow Curve$
while $CurrentLayer$ is not empty do
    $NextLayer \leftarrow$ Offset $Point$ for $Point \in Layer$
    Remove self-intersections and build medial axis
    $CurrentLayer \leftarrow NextLayer$
end while
for $Point \in$ medial axis do
    Derive $width$ for $Point$
end for
$Root \leftarrow argmax(Width)$
Traverse the medial axis from $Root$
for $Node \in tree$ from bottom up do
    $Width(Node) \leftarrow \max(Node, Successors(Node))$
end for
Reconstruct the area with optimized width based on the medial axis
$Point \leftarrow Head(curve)$
while $Point$ is not $End(curve)$ do
    Offset $Point$ by $d$
    Check validity for $Point$
    if $Point$ is valid then
        Append $Point$ to the end of $Curve$
        if Optimize Buffer is not empty then
            Corner Smoothing for Optimize Buffer
            Clear Optimize Buffer
        end if
    else
        Append $Point$ to Optimize Buffer
    end if
    $Point \leftarrow NextPoint$
end while
```

Acknowledgment

The first author thanks Suqin He, Yuchen Guo, Chuting Cai for the valuable discussions and comments that helped improving the quality of the article.

References

[1] I. Gibson, D. W. Rosen, and B. Stucker, *Additive Manufacturing Technologies*. Springer US, 2010.

[2] I. Hager, A. Golonka, and R. Putanowicz, "3d printing of buildings and building components as the future of sustainable construction?" *Procedia Engineering*, vol. 151, pp. 292–299, 2016, ecology and new building materials and products 2016.

[3] J. Zhao, R. Xia, W. Liu, and H. Wang, "A computing method for accurate slice contours based on an STL model," *Virtual and Physical Prototyping*, vol. 4, no. 1, pp. 29–37, mar 2009.

[4] K. Lorenz, J. Jones, D. Wimpenny, and M. Jackson, "A review of hybrid manufacturing," in *2014 International Solid Freeform Fabrication Symposium*, 2015.

[5] Z. Zhu, V. G. Dhokia, A. Nassehi, and S. T. Newman, "A review of hybrid manufacturing processes–state of the art and future perspectives," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 7, pp. 596–615, 2013.

[6] W. Aiyiti, L. Xiang, L. Z. Zhang, and R. M. Chen, "Study on the veritable parameters filling method of plasma arc welding based rapid prototyping," *Key Engineering Materials*, vol. 522, pp. 110–116, aug 2012.

[7] T. Ou, C. Hu, Y. Zhu, M. Zhang, and L. Zhu, "Intelligent feedforward compensation motion control of maglev planar motor with precise reference modification prediction," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 7768–7777, sep 2021.

[8] Z. Wang, R. Zhou, C. Hu, and Y. Zhu, "Online iterative learning compensation method based on model prediction for trajectory tracking control systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 415–425, jan 2022.

[9] V. Rajan, V. Srinivasan, and K. A. Tarabanis, "The optimal zigzag direction for filling a two-dimensional region," *Rapid Prototyping Journal*, vol. 7, no. 5, pp. 231–241, dec 2001.

[10] I. Buj-Corral, A. Bagheri, A. Domínguez-Fernández, and R. Casado-López, "Influence of infill and nozzle diameter on porosity of fdm printed parts with rectilinear grid pattern," *Procedia Manufacturing*, vol. 41, pp. 288–295, 2019, 8th Manufacturing Engineering Society International Conference, MESIC 2019, 19-21 June 2019, Madrid, Spain.

[11] S. Park and B. Choi, "Tool-path planning for direction-parallel area milling," *Computer-Aided Design*, vol. 32, no. 1, pp. 17–25, jan 2000.

[12] B. H. Kim and B. K. Choi, "Machining efficiency comparison direction-parallel tool path with contour-parallel tool path," *Computer-Aided Design*, vol. 34, no. 2, pp. 89–95, 2002.

[13] R. Farouki and C. Neff, "Analytic properties of plane offset curves," *Computer Aided Geometric Design*, vol. 7, no. 1-4, pp. 83–99, jun 1990.

[14] J.-K. Seong, G. Elber, and M.-S. Kim, "Trimming local and global self-intersections in offset curves/surfaces using distance maps," *Computer-Aided Design*, vol. 38, pp. 183–193, 03 2006.

[15] S. Aluru and F. Sevilgen, "Parallel domain decomposition and load balancing using space-filling curves," in *Proceedings Fourth International Conference on High-Performance Computing*, 1997, pp. 230–235.

[16] C. Pérez-Demydenko, I. Brito-Reyes, B. A. Fernández, and E. Estevez-Rams, "The complete set of homogeneous hilbert curves in two dimensions," *Applied Mathematics and Computation*, vol. 234, pp. 531–542, may 2014.

[17] X. Chen and S. McMains, "Polygon offsetting by computing winding numbers," 2005.

[18] R. Ogniewicz and O. Kübler, "Hierarchic voronoi skeletons," *Pattern Recognition*, vol. 28, no. 3, pp. 343–359, mar 1995.

[19] R. Dorado, "Medial axis of a planar region by offset self-intersections," *Computer-Aided Design*, vol. 41, no. 12, pp. 1050–1059, dec 2009.

[20] B. Sencer and E. Shamoto, "Curvature-continuous sharp corner smoothing scheme for cartesian motion systems," in *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*. IEEE, mar 2014.

[21] M. Duan and C. Okwudire, "Minimum-time cornering for CNC machines using an optimal control method with NURBS parameterization," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5-8, pp. 1405–1418, nov 2015.