Automatic design of **conformal cooling channels** in injection molding tooling

**Yingming Zhang¹, Binkui Hou¹, Qian Wang¹, Yang Li¹* and Zhigao Huang¹**

1 State Key Laboratory of Material Processing and Die & Mold Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China
E-mail: liyang@hust.edu.cn

**Abstract.** The generation of cooling system plays an important role in injection molding design. A conformal cooling system can effectively improve molding efficiency and product quality. This paper provides a generic approach for building conformal cooling channels. The centrelines of these channels are generated in two steps. First, we extract conformal loops based on geometric information of product. Second, centrelines in spiral shape are built by blending these loops. We devise algorithms to implement the entire design process. A case study verifies the feasibility of this approach.

1. **Introduction**

Injection molding is widely used to produce plastic product. The plastic product needs to be cooled before mold opening. This cooling stage takes up most of time in injection molding process. Cooling channels are devised for reducing cooling time and distributing temperature uniformly in injection molding design. Effective cooling has been proved by using conformal cooling system [1, 2]. However, it’s a complex process to build conformal cooling channels for product with complicated shapes.

In order to improve the design efficiency, researchers have carried out a series of studies on automated design of conformal cooling channels. Li [3] decomposed product into a lot of simple shape features based on their characteristics and built cooling system by connecting simple cooling channels established for each simple feature. However, it’s difficult to decompose complex products and identify characteristics of complex shapes. Yu et al. [4, 5] offset product surface to get conformal surface and computed cooling circuit with the information of this conformal surface to get conformal cooling channels. However, offsetting free-formed surface needs complex calculations and is error-prone in actual design.

The objective of this paper is to put forward a more general approach for rapid design of conformal cooling channels. Firstly, we create conformal loops to represent the geometric information of product. Secondly, channel centerlines are built by converting these conformal loops into spiral curves. Finally, cooling system is generated by sweeping cross-section along channel centerlines. The whole design process is achieved by algorithms proposed in this study. This method can effectively avoid problems of identifying or offsetting complex surface. A case study with complex shape verifies the feasibility of this approach.

2. **System process**
Conformal cooling system consists of a series of channels whose centerlines maintain a certain distance with product. Typical structure of conformal cooling channels is shown in figure 1, where \( W \) is the pitch distance between adjacent channels and \( D \) is the offset distance between centerline and mold surface.

![Figure 1. Conformal cooling channels](image)

The main challenge of building conformal channels lies in creating channel centerlines. These centerlines are generated by extracting conformal loops from product and blending these loops into spiral curves. In order to achieve the automatic design of cooling channels, all the data needed in design process are stored in a knowledge base. These data include pitch distance, offset distance, spiral type and cross-section of cooling channels. When the cooling area is specified, conformal loops can be generated by extracting section loops with pitch distance and offsetting loops with offset distance. We convert conformal loops into spiral curves according to the circuit type. Cooling channels are generated by sweeping cross-section along these spiral curves. The whole process can be illustrated in figure 2.

![Figure 2. The process of cooling channels generation](image)

3. Main algorithms

3.1. Conformal loops generation

Conformal loops are a set of contours which maintain certain distance with product. We create a set of reference planes and intersect them with cooling area to get intersection curves. The normal \( \mathbf{n} \) of each reference plane and section distance \( d_1 \) between inference planes can be read from constructed knowledge base. These curves contain geometric information of the product surface, and we can use them to create section loops. Conformal loops are built by offsetting these section loops with offset distance \( d_2 \). The input of this algorithm is cooling area \( S_c \), section distance \( d_1 \), offset distance \( d_2 \) and parting direction \( \mathbf{n} \). We can create conformal loops \( \{L\} \) by following steps:

1. Calculate anchor points of reference planes in a range created by projecting \( S_c \) onto \( \mathbf{n} \). All the anchor points are stored in a point set \( \{P_i\} \).
2. FOR each i DO
3. Create a reference plane \( RE \) whose normal is \( \mathbf{n} \) at point \( P_i \)
4. Calculate a curve set \( \{C_k\} \) by intersecting \( RE \) with \( S_c \)


5. Create loop collection \{LC\} and simple curve collection \{SC\}
6. FOR each k DO
   7. IF \(C_k\) is a loop, add \(C_k\) into \{LC\}
   8. Else, add \(C_k\) into \{SC\}
9. Create a loop \(L_{SC}\) by connecting curves in \{SC\} and add \(L_{SC}\) into \{LC\}
10. Create a total loop \(TL\) by combining loops in \{LC\}
11. Create a conformal loop \(CL\) by offsetting \(TL\) on RE with a method proposed in [6] and add \(CL\) into \{L\}

A loop \(L_{SC}\) can be generated by connecting adjacent curves whose endpoints are adjoining, and a total loop is created by combining discrete loops. The combining process is achieved by splitting loops into curves and connecting curves to be a loop. The concrete illustration is shown in figure 3.

![Figure 3. Conformal loops generation](image)

### 3.2. Spiral curves generation

Spiral curves are generated by blending conformal loops. There are three types of cooling circuit in injection mold design. They are single spiral in series connection, double spirals in series connection and spirals in parallel connection (see the illustration in figure 4). Their generation processes are explained in following sections.

![Figure 4. Spiral curves generation](image)
3.2.1. Single spiral in series connection. Single spiral is generated by creating a curve between adjacent loops. We use an algorithm to convert two loops into a spiral curve. The input of this algorithm is two adjacent loops \( L_i \) and \( L_{i+1} \). The concrete generation algorithm consists of following steps (see the illustration in figure 4(a)).

1. Set a point \( P_0^i \) in \( L_i \) as the start point of spiral curve in this section. Find a point \( P_0^{i+1} \) in \( L_{i+1} \), which is the closest point from \( P_0^i \). \( P_0^{i+1} \) is viewed as the end point of spiral curve in this section and the start point of spiral curve in next section.
2. Discrete curves \( L_i \) and \( L_{i+1} \) into \( n \) points separately. These points are separately stored in \( \{P_j^i\} \) and \( \{P_j^{i+1}\} \), where \( j=0, 1, ..., n-1 \).
3. Create a point \( P_j^* \) between \( P_j^i \) and \( P_j^{i+1} \) with a function \( P^{*}_j=(1-j/n)P_j^i +(j/n)P_j^{i+1} \). All these new points are stored in \( \{P^*_j\} \).
4. Connect points consecutively in \( \{P^*_j\} \) to form a spiral curve \( S_i \) between \( L_i \) and \( L_{i+1} \).

3.2.2. Double spirals in series connection. Double spirals are generated by creating two curves between adjacent loops. We use an algorithm to convert two loops into a pair of spiral curves. The input of this algorithm is two adjacent loops \( L_i \) and \( L_{i+1} \). The concrete generation algorithm consists of following steps (see the illustration in figure 4(b)).

1. Set a point \( P'_0 \) in \( L_i \) as start point of first spiral curve in this section. Discrete curves \( L_i \) into \( 2n \) points. These points are stored in \( \{P'_j\} \), where \( j=0, 1, ..., 2n-1 \). \( P'_0 \) is viewed as the start point of second spiral curve in this section.
2. Find a point \( P_{0}^{i+1} \) in \( L_{i+1} \), which is the closest point from \( P'_0 \). Discrete curves \( L_{i+1} \) into \( 2n \) points. These points are stored in \( \{P_{j}^{i+1}\} \), where \( j=0, 1, ..., 2n-1 \). \( P_{0}^{i+1} \) is viewed as end point of first spiral curve in this section and start point in next section. \( P_{0}^{i+1} \) is viewed as end point of second spiral curve in this section and start point in next section.
3. Create a point \( P_j^* \) between \( P'_j \) and \( P_{j}^{i+1} \). If \( j<n \), \( P_j^*=(1-j/n)P'_j +(j/n)P_{j}^{i+1} \). Else, \( P_j^*=(1-(j-n)/n)P'_j +(j-n)/n)P_{j}^{i+1} \). All these new points are stored in \( \{P^*_j\} \).
4. Connect points consecutively in \( \{P^*_j\} \) to form two spiral curves between \( L_i \) and \( L_{i+1} \).

3.2.3. Spirals in parallel connection. Parallel spirals are generated by connecting connection between adjacent loops. We use an algorithm to connect two loops with a straight line. The input of this algorithm is two adjacent loops \( L_i \) and \( L_{i+1} \). The concrete generation algorithm consists of following steps (see the illustration in figure 4(c)).

1. Set a point \( P_0^i \) in \( L_i \) as start point. Find a point \( P_0^{i+1} \) in \( L_{i+1} \), which is the closest point from \( P_0^i \). Discrete curves \( L_i \) and \( L_{i+1} \) into \( 2n \) points separately. These points are separately stored in \( \{P_j^i\} \) and \( \{P_{j}^{i+1}\} \), where \( j=0, 1, ..., 2n-1 \).
2. If \( i \) is an even number, Connect \( P_0^i \) and \( P_0^{i+1} \) with a straight line.
3. If \( i \) is an odd number, Connect \( P_0^i \) and \( P_0^{i+1} \) with a straight line.

All loops will be connected one by one with this algorithm. Figure 4(c) shows a concrete example. The straight line between \( P_0^i \) and \( P_0^{i+1} \) is an outlet for \( L_i \) and an inlet for \( L_{i+1} \). The straight line between \( P_n^{i+1} \) and \( P_0^{i+1} \) is an outlet for \( L_{i+1} \) and an inlet for \( L_{i+2} \). Water can be effectively conveyed in these parallel spirals.
4. Case study
The algorithms proposed in this paper have been developed with C++ and interfaced to the NX system. In this section, a case study of mouse is presented to verify the feasibility of the proposed cooling channels design process. The major steps of the design process are illustrated in figure 5.

Figure 5. The case of conformal cooling channel generation (a) cooling area (b) generate intersections (c) generate section loops (d) conformal loops generation (e) spiral curve generation (f) cooling system generation

In this case, there are many defects on the surface of cooling area. The proposed algorithm of conformal loops generation can effectively convert discrete curves into loops. Cooling circuit can be created by the method of spiral curve generation. This approach can be used for rapid design of cooling channels.

5. Conclusion
In this paper, an automatic method is proposed to design cooling channels. By using algorithm of generating conformal loops, simplified information of product shape will be obtained. We convert these loops into spiral curves by an algorithm proposed in this study. These spiral curves are used for generating the final cooling system. A case study verifies the reliability of this approach.

References
[1] Xu R X and Emanuel S 2009 Rapid thermal cycling with low thermal inertia tools Polymer Engineering & Science 49 305-16
[2] Park H S and Pham N H 2009 Design of conformal cooling channels for an automotive part International Journal of Automotive Technology 10 87-93
[3] Li C L 2007 Part segmentation by superquadric fitting-a new approach towards automatic design of cooling system for plastic injection mould International Journal of Advanced Manufacturing Technology 35 102-14
[4] Wang Y, Yu K M, Wang C C L and Zhang Y 2011 Automatic design of conformal cooling circuits for rapid tooling Computer-Aided Design 43 1001-10
[5] Wang Y, Yu K M and Wang C C L 2015 Corrigendum to “Spiral and conformal cooling in plastic injection molding” [I. Comput. Aided Des. 63C (2015) 1–11] Computer-Aided Design 63 1-11
[6] Farouki R and Srinathu J 2017 A real-time CNC interpolator algorithm for trimming and filling planar offset curves ☆ Computer-Aided Design