Intelligent Control Technology of Tire Tread Pattern Draft Angle for Intelligent Manufacturing

Heng Li1,*, Yazhou Chen1, Zhengxiong Huang1 and Quan Chen2

1 College of Mechanical and Energy Engineering, Xiamen 361021, China
201811802006@jmu.edu.cn
2 Xiamen Shengzheng Machinery Co., Ltd., Xiamen 361021, China

Abstract In view of the problem that the tire pattern design may not meet the production requirements, the designer needs to modify the tire pattern draft angle, and the existing pattern draft angle modification process is cumbersome, and the designer's experience requirements are relatively high, based on the tire pattern draft angle adjustment experience has developed a tire pattern draft angle intelligent control system for intelligent manufacturing, and developed a corresponding program based on UG10.0 based on this system. The program can automatically determine the type of pattern contour surface, can judge the contour surface unevenness based on the projection method, can automatically recommend the appropriate draft angle, and realize the rapid modification of the draft angle in batches. The actual application of the enterprise shows that this program can greatly improve the efficiency of tire pattern design.

1. Introduction
The tire needs to be molded through a tire mold, and the pattern on the tire corresponds to the pattern on the tire mold one-to-one. Electrical discharge machining (EDM) is a common method of tire mold processing. Due to the lack of understanding of tire mold processing method by tire designers, mold manufacturers need to make certain adjustments to the initial design pattern from a manufacturable perspective. Among them, adjusting the draft angle of the tire pattern is one of the main tasks.

Using the existing commands in the CAD software (e.g. UG10.0), adjusting the draft angle of the tire pattern has the following shortcomings: (1) UG generally uses pocket commands to generate blocks (or grooves). The designer cannot directly locate a certain side of the block, but needs to locate the block feature first, and then locate the block contour, then locate to a certain side by the outline. (2) When positioning from the cavity feature to an edge in the cavity contour line, the user needs to click the mouse several times. This is because the UG software implicitly selected the first side of the cavity outline (the first side is determined by the cross-section drawing). If a cavity contour line contains 6 sides, the user needs to click the mouse 5 times to select the fifth side, and the user needs to click 6 times to select the sixth side, which makes operation process very cumbersome.

To solve this problem, a method to rapidly extract the draft angle of each side of the pattern through UG API is proposed. Designers only need to operate in the table to modify the draft angle of each surface in batches. It reduces the number of mouse clicks, speeds up the design process, and improves the design efficiency.
2. Tire patterns
The friction between the tire pattern and the road surface is the power source for the vehicle. The performance of the tire is not only related to its internal structure, but also directly related to the tread pattern. The quality of pattern design directly affects tire traction, lateral force, wear resistance, drainage, and noise, etc. The tire geometry design includes two parts: tire cross-sectional shape design and tire tread shape design. The tire cross-sectional shape design produces the tire surface profile, and the tire tread geometry is derived from the tire tread general map. The pattern groove is composed of the side wall surface, the groove bottom surface and the transition surface between the side wall and the groove bottom. According to the number of layers of groove section, it can be divided into 1-layer pattern, 2-layer pattern and 3-layer pattern: (1) The 1-layer pattern is a common pattern, and its side wall is formed by a cavity draft, and its cross-section can be further divided into U-shaped and V-shaped; (2) The 2-layer pattern is formed by a transition surface and two cavity drafts; (3) The 3-layer pattern is formed by 2 transitional curved surfaces and 3 times of cavity drafting. The inclination angle $\alpha$ formed by the straight line perpendicular to the tread and the sidewall is called the draft angle. In the multi-layer pattern design, the grooves at different levels are offset. When the grooves are not offset, the 2-layer groove surfaces are directly connected. When there is an offset between the patterns, the pattern layers are connected by a small offset curve.

![Tire pattern 3D model](image1)

Figure 1 Tire pattern

The draft angle of each side of the pattern is changed along the axial section of the tire. Figure 1 shows an example of a block tire pattern and its cross-section.

3. Key technology

3.1 Judgement of contour surface type
The pattern draft angle is rapidly adjusted through the prototype, and the prototype needs to be able to automatically determine the type of pattern contour surface. Taking a pattern block as an example, the prototype needs to be able to automatically determine which surface is the top surface of the block, which surface is the bottom surface of the block, and which surface is the side surface of the block. The top and bottom surfaces do not need to adjust the draft angle, while the side surfaces need. UG uses boundary representation (B-Rep), that is, using points, lines, surfaces, bodies and other forms to represent 3D models. It is found that the type of contour surface can be determined by the angle relationship between the contour surface and the top surface of the light tire. For example, the angle between the top surface of the pattern, the bottom surface of the pattern and the top surface of the light tire is 0, and the angle between each side of the pattern and the top surface of the light tire is not 0. For this reason, it is proposed to build Pattern Attributed Adjacency Graphs (PAAG) to let the prototype automatically determine the type of pattern contour surface: $G=<F,E,A>$. The node $F$ represents the collection of the pattern contour surface $n_i$, and each pattern contour surface $f_i$ has a unique node $n_i$ corresponding to it; $A$ represents the set of pattern edges $e_k$. For adjacent pattern surfaces $f_i$ and $f_j$, there is a unique pattern edge $e_k$ corresponding to it; $A$ is the attribute of $E$. For each $a_k$, there is an attribute value corresponding to it. If the edge corresponding to two faces


forms a convex edge, the attribute value is 1 (shown with a solid green line in Figure 2); If the two faces corresponding to the edge form a concave edge, the attribute value is -1 (indicated by the red dashed line in Figure 2); if the two faces are coplanar (located on the same face and have the same normal vector), their attribute value is 0. 

Through the UG API function, the pattern contour surface information and pattern contour line information can be automatically extracted. Number the extracted pattern contour faces in the order of 1, 2, 3···i, fill in the corresponding nodes in PAAG in turn, and fill in the contour lines between adjacent contour faces in the corresponding edges in PAAG in turn. Further, the dihedral angle \( \theta \) between each node and the top surface of the light tire is automatically calculated by the prototype.

As is shown in Figure 3, obtain the target edge \( e_k \), take the midpoint \( O \) of \( e_k \), and note the coordinate of \( O \) as \((x_0, y_0, z_0)\); pass the \( O \) point as the reference plane \( \alpha \) with the target edge as the normal vector; the reference plane \( \alpha \) is associated with the two vertices of the intersection of the faces \( f_i \) and \( f_j \) are \( A(x_1, y_1, z_1) \) and \( B(x_2, y_2, z_2) \), respectively. Then the dihedral angle can be calculated according to formula (1):

\[
\theta = \arccos \left( \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \right) 
\]

\( \vec{a} = (x_1 - x_0, y_1 - y_0, z_1 - z_0) \)

\( \vec{b} = (x_2 - x_0, y_2 - y_0, z_2 - z_0) \)

When \( \theta \) is not 0, the surface represented by the corresponding node is set as the first type of surface (the pattern side surface). On the contrary, when \( \theta \) is 0, the surface represented by the corresponding node is set as the second type of surface. According to whether the second type surface is coplanar with the top surface of the light tire, the top surface of the pattern and the bottom surface of the pattern can be further judged.

3.2 Judgement of unevenness of pattern surface
There are three situations for the initial draft angle set on each side of the tire tread: (1) The draft angle
is set reasonably; (2) The draft angle is set too small; (3) There is a negative draft angle. In this paper, the surface with a negative draft angle is defined as a concave surface, and the surface with a positive draft angle is defined as a convex surface. To make the prototype automatically perform pattern drafting analysis, it is necessary to allow the prototype to automatically recognize the unevenness of the contour surface after identifying the contour surface type.

The pattern on the tire mold is generally processed by EDM, and the pattern with a negative draft angle cannot be designed with a reasonable electrode to penetrate the inner wall of the cavity for processing. It is found that the contour surface can be projected along the normal direction of the electrode line to the cut surface of the bare tire, and the problem of determining the unevenness of the contour surface can be transformed into the problem of determining the intersection of the contour lines in a two-dimensional plane.

The process of conversion from 3D to 2D conversion is as follows: (1) Extract the intersection point \( O \) between the electrode line and the surface of the light tire; (2) Construct a plane \( UOV \) tangent to the light tire at the intersection point (for clarity, the light tire, Other features such as blocks); (3) Project the upper contour surface extracted in section 4.1 into the \( UOV \) plane against the normal \( Z \) of the electrode line to obtain the upper contour line projection (marked in red in Figure 4); (4) Project the lower contour surface extracted in section 4.1 into the \( UOV \) plane against the normal direction of the electrode line to obtain the lower contour line projection (marked in green in Figure 4); (5) If the polygon formed by the upper contour line is completely If the polygonal envelope formed by the lower contour line does not have a negative draft angle on the side wall of the pattern; (6) If the polygon formed by the upper contour line and the polygon formed by the lower contour line have a cross relationship, there is a negative draft angle surface; (7) Discrete the upper contour line into a series of points; (8) Judge one by one whether the discrete points are inside or outside the polygon formed by the lower contour line to identify the specific negative angle surface. In this paper, this process is called the contour surface roughness judgment based on contour line projection.

To solve the problem of the intersection of plane contours, the ray method\(^9\), the corner method\(^10\), and the area sum method\(^11\) can be used. Their advantages and disadvantages are shown in Table 1.

| method       | advantage                        | disadvantage                              |
|--------------|----------------------------------|-------------------------------------------|
| Angle method | Just calculate the angle         | Involving inverse trigonometric calculations |
| Area method  | Just calculate the area          | Calculations involving multiple areas      |
| Ray method   | Only need to perform edge intersection operation | More calculation points                  |

![Figure 4 Pattern block and projection](image-url)
Because the intersection of edges in the ray method can be directly eliminated through coordinate comparison, and the calculation amount is relatively small, so the ray method is selected in this paper. The principle is: passing a point \( P_i \) to make a ray \( P_iX_i \) along a certain direction. If the number of intersection points between the ray and the polygon is an even number, the point is outside the polygon; if the number is an odd number, the point is inside the polygon. As shown in Figure 5: (1) \( P_1X_1 \) has 4 intersections with the polygon, then \( P_1 \) is outside the polygon; (2) \( P_3X_3 \) has 1 intersection with the polygon, then \( P_3 \) is inside the polygon. It needs to be emphasized that the above only applies to general situations and not to some special situations. For example, (3) \( P_2X_2 \) has 3 intersections with the polygon, and \( P_2 \) is outside the polygon; (4) \( P_4X_4 \) has 2 intersections with the polygon, and \( P_4 \) is inside the polygon. In order to determine the intersection of the upper contour projection and the lower contour projection of the pattern, this paper improves the traditional ray method, and the process is as follows:\[12]\:

![Figure 5 Topological relationship between points and polygons](image)

- **Step1:** Extract the vertices of the upper contour line projection polygon, arrange them in a counterclockwise direction, and store them in the array Upper\([n]\), where \( n \) is the number of upper contour line projection vertices;
- **Step2:** Extract a certain point in Upper\([n]\) and mark it as \( P_i \), and the initial value of \( i \) is 1;
- **Step3:** Judge whether the \( P_i \) point is located on the boundary of the projection polygon of the lower contour line, if it is to Step 8, otherwise, it will execute Step 4;
- **Step4:** Pass the point \( P_i \) as the horizontal right ray \( P_iX_i \), find the intersection point of the ray and the projection polygon of the lower contour line, and record the total as \( num1 \);
- **Step5:** Judge each intersection point in \( num1 \) in turn. If a certain intersection point is an end point of the lower contour line projection, then \( num2 = num2 + 1 \), \( num2 \) represents the number of intersection points between the ray \( P_iX_i \) and the lower contour line projection polygon at the end of the line segment. The initial value is 0;
- **Step6:** Calculate the total number of effective intersection points between the ray passing through the point \( P_i \) and the projection polygon of the lower contour line, denoted as inter_point_total, \( \text{inter\_point\_total} = num1 - num2 \);
- **Step7:** If \( \text{inter\_point\_total} \) is odd, return \( P_i \), point is inside the lower contour polygon; if \( \text{inter\_point\_total} \) is even, return \( P_i \) point is outside the lower contour polygon;
- **Step8:** Is \( i \) less than or equal to \( n \)? If not, go to Step 2, if yes, go to Step 9; //Determine whether all vertices in the polygon on the upper contour line have looped completely.
- **Step9:** The prototype ends.

### 4. Prototype development

Based on UG10.0 API, the pattern draft angle quick adjustment prototype is implemented in VS2010 C++ environment, and the realized functions include: (1) Quickly select pattern features; (2) Import pattern parameters into the dialog box tree list; (3) Perform pattern contour surface type judgment; (4) Perform pattern contour line projection; (5) Judge the intersection of upper and lower contour lines; (6) Modify the unreasonable draft angle. Among them, (3), (4) and (5) have been explained in 3.1 and 3.2, and (1) (2) (6) are introduced below.
4.1 Select pattern features
There are many types of UG design objects, such as Feature, Object and Body[13]. In the development process, it is needed to enter different data types to call different API functions, such as:

UF_MODL_ask_general_pocket() needs to input feature type; UF_CURVE_ask_line_data() needs to input object type. After the user selects the pattern feature, the type conversion is performed by calling the API function. Part of the functions used are shown in Table 2.

| Function                        | Application                        |
|---------------------------------|------------------------------------|
| UF_MODL_ask_feat_object()       | Find object by feature             |
| UF_MODL_ask_feat_body()         | Find body by feature               |
| UF_MODL_ask_body_feats()        | Find features through body         |
| UF_MODL_ask_object_feat()       | Find features through objects      |

4.2 Import pattern parameter tree list
Convert the pattern features obtained in 4.1 into tag_t type, and the code is as follows:

```cpp
NXOpen::BlockStyler::PropertyList* SectionPocket = select_feature0->GetProperties();
tag_t pocket_tag = Select_pocket;
```

Bring Select_pocket into the UF_MODL_ask_general_pocket function, get the number of sides of the pattern block by number_of_curves, and determine the number of rows in the tree list in the pop-up dialog box. Obtain the draft angle of each side from linear_cubic.start_value and linear_cubic.end_value, and fill in the corresponding column of the tree list.

4.3 Modification of unreasonable draft angle
By judging the intersection of contour lines, screen out the side with inappropriate draft angle. In the pop-up dialog box, when clicking a row in the tree list, the side of the pattern block will be highlighted. The functions implemented during the click process are as follows:

```cpp
std::vector<NXOpen::BlockStyler::Node*> nodes;
NXOpen::BlockStyler::Node* node = tree_control0->RootNode();
```

The updated draft angle in the dialog box can be stored in the parameters linear_cubic.start_value and linear_cubic.end_value. Through the UF_MODL_edit_general_pocket function, the pattern block model can be updated and the draft angle can be regenerated.

Figure 6 Example diagram
4.4 prototype operation
The main interface of the prototype is shown in Figure 6. It can be seen from Figure 6 that by using the prototype, it is easy to quickly select the corresponding pattern feature (other tire features have been hidden), and import the draft angle parameters of each side of the pattern feature into the dialog box tree list in batches. The prototype judges the unreasonable side, and user can modify it directly in the tree list. What’s more, the appropriate draft angle can be recommended through the prototype.

Table 3 Program application results

|                  | 1 edge | 2 edges | 3 edges | 4 edges | 5 edges |
|------------------|--------|---------|---------|---------|---------|
| Traditional method | 15s    | 21s     | 25s     | 29s     | 36s     |
| Pattern adjustment program | 8s     | 10s     | 13s     | 15s     | 18s     |
| Time saved       | 7s     | 11s     | 12s     | 14s     | 18s     |

As shown in Table 3, the application of the tire pattern quick adjustment program is compared with the traditional method for adjusting the number of different sides of the pentagonal block. It can be found that the application of the tire pattern adjustment program can significantly shorten the adjustment time of the pattern edge data, and the pattern edge is wrong. The more, the better the adjustment efficiency of the program. The program has been applied in the enterprise and the operation effect is good.

5. Conclusion
Aiming at the problem that the tire pattern draft angle adjustment process is cumbersome and thus the design efficiency is low, this paper proposes a approach for rapid adjustment of the tire pattern draft angle. The approach can automatically determine the type of pattern contour surface, automatically judge the unevenness of the pattern contour surface based on projection and modify the draft angle of multiple sides of the pattern in batches. Practical application shows that this prototype greatly improves the design efficiency of pattern draft angle. Further research will focus on how to incorporate more design experience and design knowledge into the pattern draft angle adjustment procedure.

References
[1] A N Gent, J D Walter. Tire Theory and Technology [M]. Tsinghua University Press. 2013.
[2] Cho J H, Cho J R, Woo J S, etal. Numerical Investigation of Snow Traction Characteristics of 3D Patterned Tire[J]. Journal of Terramechanics, 2012, 49:81-93.
[3] Cho J R, Lee H W, Jeong W B, etal. Numerical Estimation of Rolling Resistance and Temperature Distribution 3D Periodic Patterned Tire[J]. International Journal of Solids and Structures, 2013, 50(1):86-96.
[4] Li X F, Lei Y, Wang X F. Key technology of tire mold pattern modeling design[J]. Journal of Engineering Graphics, 2005(04):41-46.
[5] Dong Y D, Song Z H, Zhang R T. Semantic modeling method for tire 3D pattern[J]. China Mechanical Engineering, 2016(19): 2614-2623.
[6] Bai S S. Semantic-based CAD system design for tire 3D pattern[D]. Hefei: Hefei University of Technology, 2015.
[7] Hu Q W, Zhang X L, Hui H C. 3D Marking Information Management Method Based on Pro/E[J]. Standardization and quality of machinery industry. 2018(01):40-44.
[8] Liu W J, Gu L, Chang W. Manufacturing feature recognition method based on attribute adjacency graph[J]. Department of Computer Integrated Manufacturing. 2001(02):53-58.
[9] Chen R Q, Zhou J, Yu L A. A fast algorithm for judging the relationship between points and polygons[J]. Journal of Xi’an JiaoTong University. 2017(01):59-63.
[10] Luo W, Shun Y M, Chen Z W. Improved algorithm for judging the relative relationship between points and polygons [J]. *Machinery*. 1999(03):3-5.

[11] Liu Y S. Automatic Output of Judging Process Coordinate Results [J]. *Beijing Surveying and Mapping*. 2012(01):56-59.

[12] Zhai Y, Xu W D, Zhang Q. Judgment of Topological Relationship Between Points and Polygons or Polyhedrons [J]. *Computer Engineering and Design*. 2015(04):972-976.

[13] Mo R, Chang Z Y, Liu J H. *Detailed explanation of UG NX secondary development* [M]. Beijing: Electronic Industry Press. 2008.