

# 자유곡면 가공을 위한 NC 가공데이터 생성 S/W 개발

황 지 선

현대자동차(주) 금형부

## 요 약

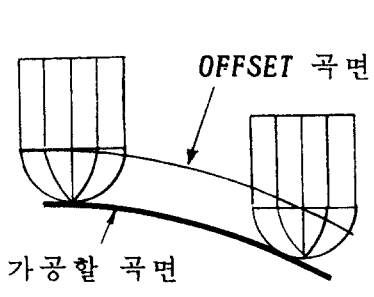
사각 *PATCH*로써 곡면을 표현하는 일반적인 *CAD/CAM* 시스템에서 *NC* 가공을 위한 가공데이터를 생성할 때에는 보통 두가지 가공방법을 사용한다. 그 하나는 *ISOPARAMETRIC* 가공이며 다른 하나는 *CARTESIAN* 가공이다. 금형부에서 사용하는 *CAD/CAM* 시스템인 *CATIA*에서도 위의 두가지 *NC* 가공방식을 모두 사용할 수 있다.

그런데 *ISOPARAMETRIC* 가공의 경우, 공구간섭처리를 위한 곡면모델의 수정에 많은 시간이 소요되며, *NC* 프로그래머의 실수로 인한 가공불량 발생의 가능성도 크고 황삭가공에도 매우 불편하다. 이러한 문제는 *CARTESIAN* 가공방법으로 해결이 가능하며, *CATIA*의 경우 *PARALLEL PLANE MACHINING* 기능이다. 그러나 이 기능을 *CATIA*에서 통상적인 방법으로 *MODELING*된 곡면에 적용할 경우에 그 신뢰성이 매우 낮다.

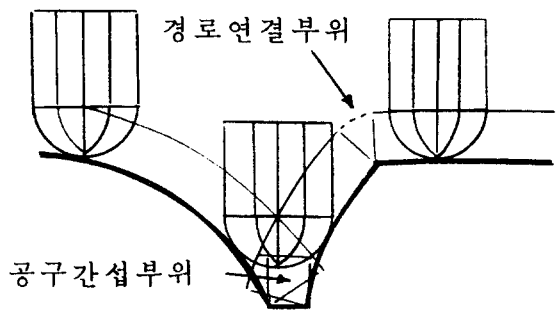
따라서 금형부에서는 그 방식은 *CARTESIAN*으로 하되 신뢰성이 높은 *NC* 가공 *S/W*를 개발하게 되었다. 이는 *CATIA* 시스템에서 *MODELING*된 곡면으로부터 *NC* 가공데이터를 생성하는 *S/W*로서 *VS FORTRAN*으로 작성되었다.

## 서 론

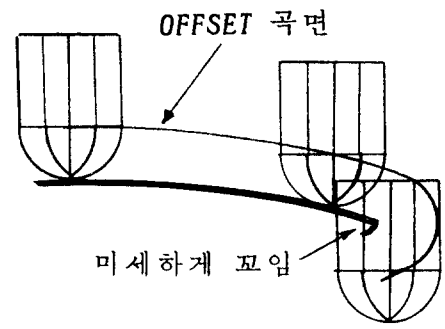
*BALL-END MILL*을 사용하여 자유곡면을 *NC* 가공하기 위해서는 이 곡면으로부터 공구 반경만큼 *OFFSET*된 *OFFSET* 곡면을 생성하고 이 *OFFSET* 곡면상에 공구의 중심을 위치시켜 *CL* 데이터(*CUTTER LOCATION* 데이터)를 계산하면 된다. <그림 1> 그러나 많은 수의 곡면으로 이루어진 복합곡면을 가공할 때에는 *OFFSET* 곡면들은 공간상에 복잡하게 위치하여 *CL* 데이터의 경로가 꼬이는 부위가 생긴다. 이러한 부분이 공구와 곡면이 간섭되는 부분으



<그림 1> 곡면의 NC 가공



<그림 2> 공구간섭 및 경로연결



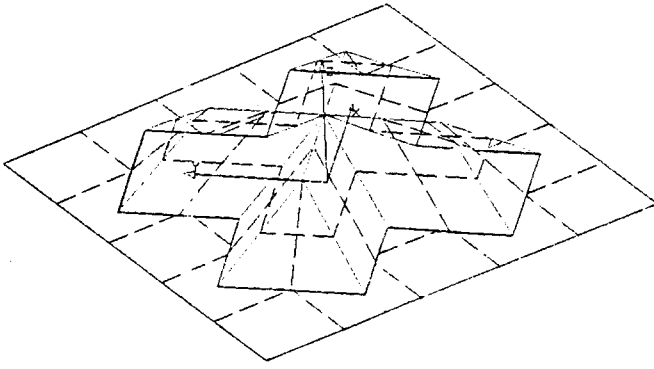
<그림 3> 이상곡면의 예

로서 이는 표입 부분의 CL 데이터를 제거하여 간섭을 피해야 한다. 또 곡면의 접선벡터가 연속하지 않는 곡면들의 가장자리 부분에서는 OFFSET 곡면이 연속하지 않으므로 경로를 적당히 연결해야 한다.<그림 2> 그리고 일반 CAD/CAM 시스템에서 가끔 볼 수 있는 이상곡면의 경우, 가공불량이 생길 수도 있다.<그림 3>

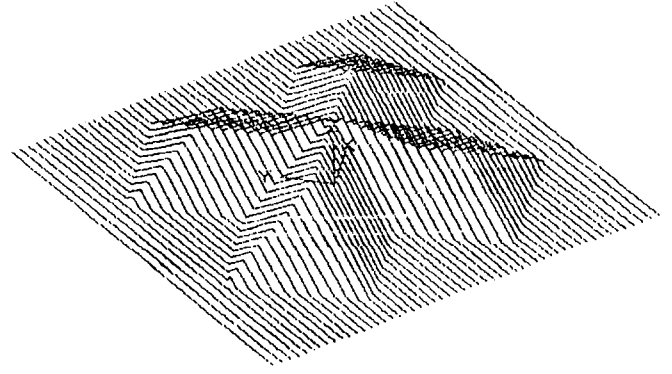
이렇게 OFFSET 곡면을 직접 이용하여 CL 데이터를 생성시킬 때에 해결해야 하는 문제를 보다 쉽게 처리하기 위해 본 연구에서는 복합곡면으로 부터 일정한 허용오차( $\alpha$ )내에 드는 일련의 점 좌표들을 얻어 이를 적당한 방법으로 삼각형으로 구성 (TRIANGULATION)하여 삼각형 GRID를 형성한 다음 이로부터 OFFSET 곡면을 만들어 CL 데이터를 계산하는 알고리즘을 개발하게 되었다.

곡면상에 점 데이터 생성

CATIA 시스템에서 가공해야 할 복합곡면을 MODELING한 다음, 점좌표 추출용 프로그램을 실행시키면 주어진 허용오차내에 드는 곡면상의 점데이터를 얻을 수 있다.<그림 4> 이 프로그램은 CATIA에서 제공하는 GEOMETRY INTERFACE SUBROUTINE을 사용하여 개발된 것이다.



a. CATIA 곡면모델



b. 점데이터 곡면

<그림 4> 점데이터의 생성

### CL-POINT의 계산

CL-POINT는 공구가 곡면에 접할 때의 공구 끝점의 위치이다. 이 좌표를 계산하기 위해 본 연구에서 사용하는 방법은 다음과 같다.

XY평면상에 임의의  $(x,y)$ 좌표값이 주어져 있다고 가정하면 이 점을 통과하고 Z 축에 평행한 직선을 설정할 수 있다. 이때 직선을 중심축으로 하여 공구를 수직으로 이동시켜, 공구가 곡면에 접하게 될 때의 공구 끝의 z 값을 계산하면 이 점이 CL-POINT가 된다. <그림 5> 그런데 앞에서 설명한 바와 같이 공구의 중심은 공구반경만큼 OFFSET된 OFFSET 곡면상에 위치하므로 이 OFFSET 곡면과 직선의 교점의 z 좌표값에서 공구반경만큼 빼면 같은 값의 CL-POINT를 얻게된다.

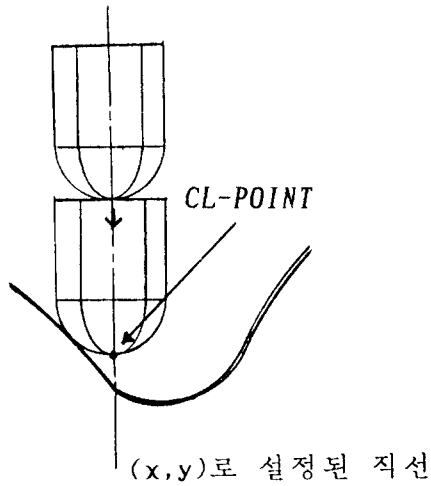
따라서 앞에서 얻어진 점데이터로 구성된 삼각형 GRID 곡면의 OFFSET 곡면과 직선과의 교차점을 계산한다면 CL-POINT는 쉽게 구할 수 있다.

#### 1) 계산영역 설정

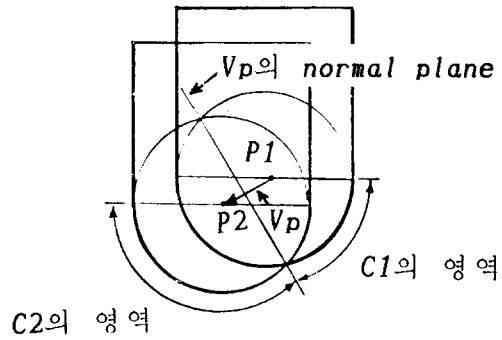
주어진  $(x,y)$ 좌표에 공구를 위치시켰을 때에 공구 z 값의 계산에 영향을 주는 영역을 먼저 설정해야 한다. 이는 주어진  $(x,y)$ 좌표로부터 공구 반경으로 원을 설정하여 원내의 점들과 원을 처음으로 벗어나는 점들로 구성되어 있다. 그런데 이 영역은 그전에 계산된 CC(CUTTER CONTACT)-POINT와 관계를 가지고 있어 영역을 줄일 수 있다.

정리 : 임의  $(x_1,y_1)$ 좌표에서 CL-POINT가 계산되었고 다음의  $(x_2,y_2)$ 좌표에서 CL-POINT가 계산되었을 때  $(x_2,y_2)$ 에서의 CC-point는  $(x_1,y_1)$ 에서의 CC-POINT보다 공구중심진행 벡터방향으로 항상 앞에 있다. <그림 6>

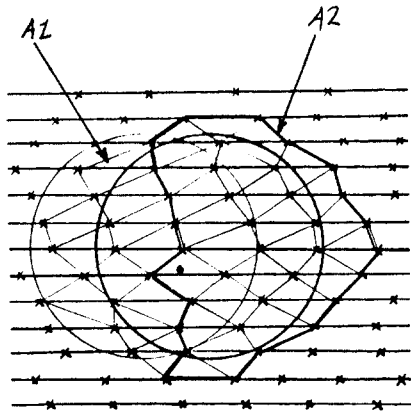
이 정리를 적용하면 계산대상영역은 <그림 7>과 같이 구성된다.



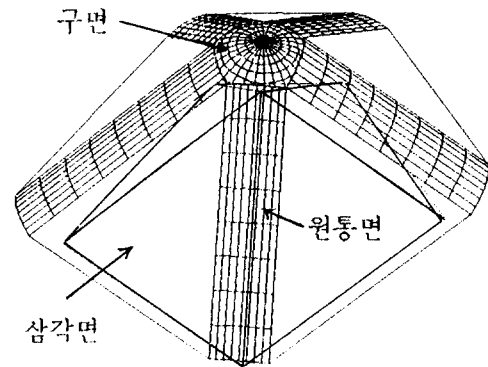
<그림 5> CL-POINT의 계산



<그림 6> CC-POINT에 관한 정리



<그림 7> 계산영역



<그림 8> 삼각형 GRID곡면의 OFFSET

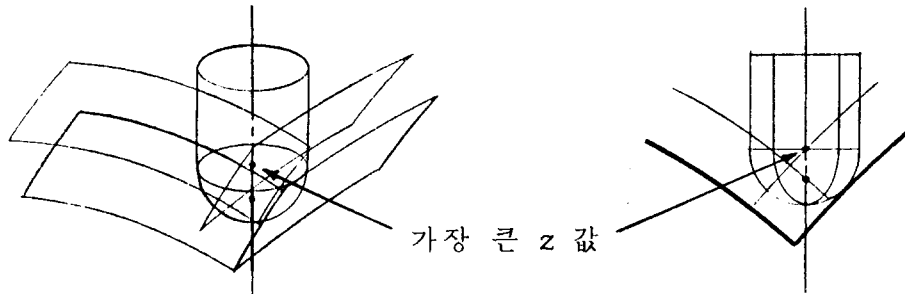
## 2) CL-POINT의 Z값 계산

계산영역내에 있는 점을 적당한 방법으로 TRIANGULATION하여 삼각형GRID를 형성하고 이를 OFFSET하여 OFFSET곡면을 생성한다. 그러면 OFFSET 곡면은 점, 직선 그리고 삼각면이 OFFSET된 것이므로 점의 경우 구면, 직선의 경우 원통면 그리고 삼각면의 경우 평면으로 구성된다.<그림 8> 이때 OFFSET 곡면과 주어진 직선이 만나는 점들 중 가장 높은 z 값을 선택하면 이 점이 바로 CL-POINT가 된다.<그림 9>

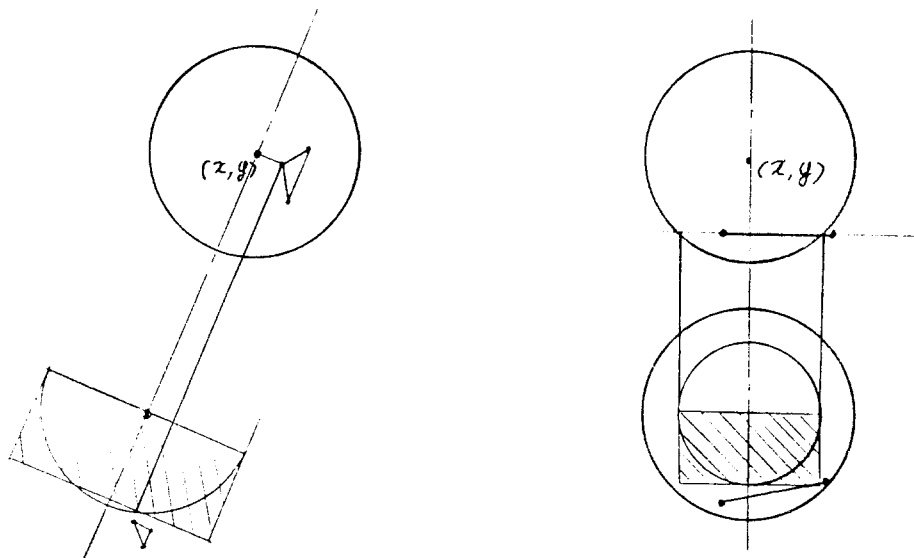
이는 계산속도를 빠르게 하기 위해 다음과 같은 방법을 사용한다. 먼저 계산영역내의 모든 점과 계산을 하여 초기 z 값을 구한다. 이것은 OFFSET면중 구면과의 교점중 가장 높은 값을 구하는 것이다. 초기 z 값이 계산되면 이 위치에서 공구를 포함하는 MAJORIZING VOLUME을 설정하여 삼각면과 원통면과의 z 값 계산을 할 것인가를 판정한다.<그림 10> 이 MAJORIZING VOLUME에 삼각면이나 직선이 포함되면 이들과 z 값 계산을 하여 최종 z 값을 결정한다. 이는 다음과 같은 알고리즘으로 설명될 수 있다.

(a) 모든 점과의 계산으로 초기 z 값 계산

(b) 계산영역내의 다음의 삼각면을 선택하여 MAJORIZING VOLUME에의 포함여부 테스트  
포함하지 않을 경우 (b)로 이동



<그림 9> 간섭부위의 CL-POINT 계산



a. 삼각면에 대한 MAJORIZING VOLUME    b. 직선에 대한 MAJORIZING VOLUME

<그림 10> MAJORIZING VOLUME

(c) 삼각면의 OFFSET면 생성 및 직선과의 교차여부 테스트  
교차할 경우: 현재의 z 값과 교차점의 z 값을 비교, 큰 값을 보관하고 (b)로 이동

(d) 직선의 MAJORIZING VOLUME에 포함여부 테스트  
포함하지 않을 경우 (b)로 이동

(e) 원통면과 직선과의 교차여부 테스트  
교차할 경우: 현재의 z 값과 교차점의 z 값을 비교, 큰 값을 보관하고 (b)로 이동

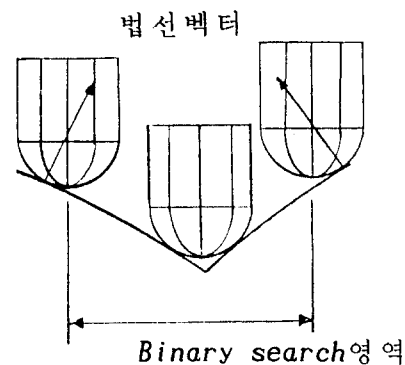
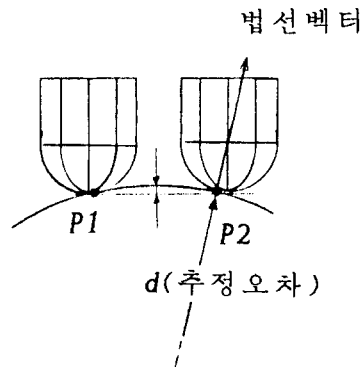
## 공구 이동거리의 예측

### 1) 허용오차( $\beta$ )를 고려한 이동거리 예측

한 *CL-POINT*에서 다음 *CL-POINT*까지 공구가 이동할 때에 가공되는 부분은 실제곡면과 비교하여 허용오차내에 들어야 한다. 따라서 다음점까지의 거리를 미리 예측하고 다음점을 계산한 후 오차를 추정하여 오차가 허용오차보다 클 때는 이동거리를 축소시켜야 한다. 본 연구에서는 오차를 추정하기 위해 다음과 같은 방법을 사용한다. 즉 현재의 *CL-POINT*와 이 점에서의 법선벡터 그리고 바로 전의 *CL-POINT*로 가상의 원을 형성하여 오차를 추정, 현재의 *CL-POINT*까지의 이동거리를 조정하고 다음 *CL-POINT*의 위치를 추정한다. <그림 11>

### 2) 오목한 간섭부분에서의 *CL-POINT* 추정

오목한 간섭부위에서는 공구가 놓이는 점을 추정하여 형상이 정확하게 깎이도록 해야 한다. 이를 위해서 본 연구에서는 두 *CC-POINT*의 법선벡터 방향을 보고 오목한 부분이라고 판단될 경우 *BINARY SEARCH*를 하여 이부분에 공구가 정확히 위치되도록 *CL-POINT*를 계산한다. <그림 12>

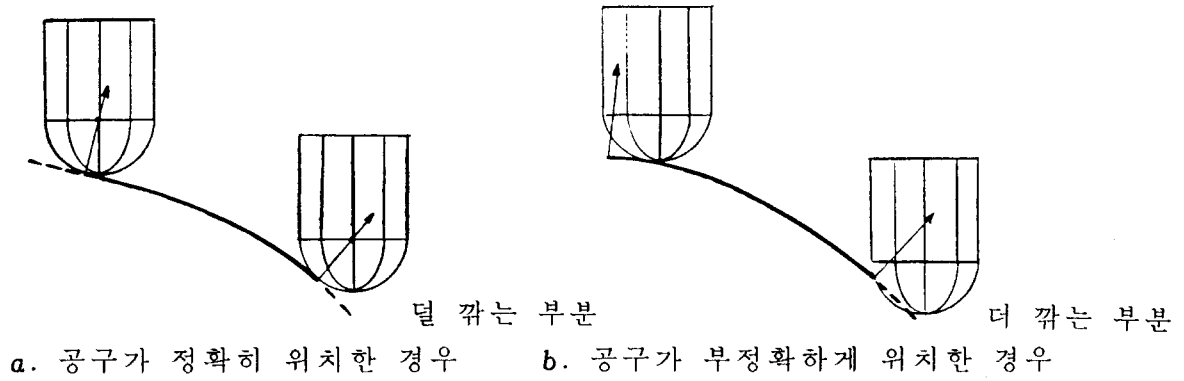


<그림 11> 오차의 추정    <그림 12> 오목한 부분의 *CL-POINT* 계산

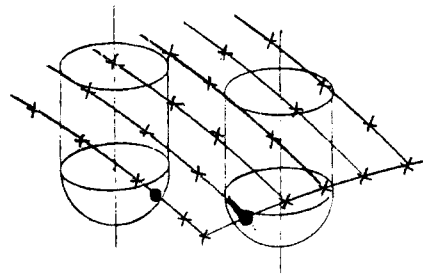
### 가공형상 가장자리에서의 공구위치 결정

본 연구의 방법으로 *CL* 데이터를 생성할 경우 곡면의 가장자리 부분의 *CL-POINT*를 정확히 계산해야 한다. 왜냐하면 곡면 끝부분에서 부정확하게 *CL* 데이터를 얻을 경우 주어진 곡면을 덜 깎거나 이 곡면이 다른 곡면과 연결되어 있을 경우 예상치 못한 가공불량을 초래할 수 있기 때문이다. <그림 13>

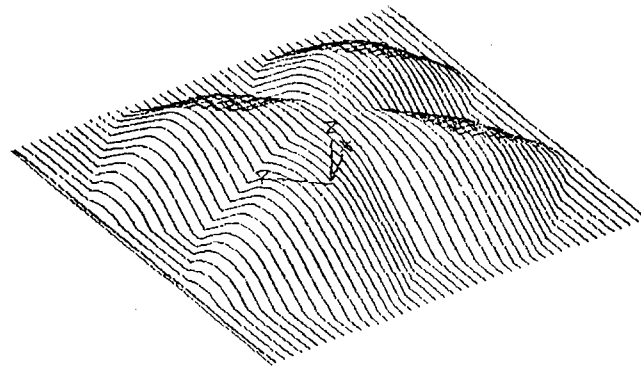
이러한 시작점과 끝점 예측을 위해 본 연구에서는 *BINARY SEARCH*를 행한다. 이때 한 점이 가공할 면을 벗어났다고 판단하는 기준은 최종 *CL-POINT*가 점 또는 직선과 접촉되어 계산된 경우에 접촉점 또는 접촉된 직선이 삼각형 *GRID* 곡면의 가장 외각부위일 경우이다. <그림 14>



<그림 13> 가장자리에서의 공구 위치



<그림 14> 가공면을 벗어난 공구 위치



<그림 15> CL 데이터 그림

결론

이 연구에서는 곡면으로부터 점 데이터를 얻을 때의 허용오차( $\alpha$ ), 그리고 이 점데이터로부터 CL 데이터를 계산할 때의 허용오차( $\beta$ )등 실제곡면으로부터 가공까지 두가지 오차를 가지고 있다. 따라서 기계가공 허용오차를  $\gamma$ 라하면 이론적으로

$$\alpha + \beta < r$$

의 관계로서  $\alpha$ 와  $\beta$ 를 결정해야 한다.

실제로 가공을 해본 결과 이론적인 오차보다 상당히 작은 오차가 발생하여 기계가공에 문제가 없다는 것이 확인되었다. 또 이러한 방법으로 NC 데이터를 생성할 때에 기존의 방법과 비교하여 NC 프로그램 시간을 상당히 단축하였고, NC 데이터의 신뢰성도 높아졌다. 본 연구에 이어 앞으로 해야 할 일은 FILLETED-END MILL을 사용, 황삭가공을 하기 위해 새로운 연구를 하는 일이다. <그림 15>는 본 연구의 방법으로 계산된 CL 데이터를 예시한 것이다.

#### 참고 문헌

1. Choi B.K, Jun C.S, *Ball-end mill cutter interference avoidance in NC machining of sculptured surface, CAD, Vol 21, No 6, Jul/Aug. 1989*
2. 황지선, 복합곡면의 modeling 및 가공에 관한 연구, KAIST MS Thesis 1987
3. Tsukasa K, Takeshi, K, Katsumasa S, *Machining System based on Inverse Offset Method, 日本 精密工學會誌, May. 1988*