

STEP-NC의 피쳐 기반 공구경로 생성 및 갱신

조정훈*, 서석환**

Feature Based Tool Path Planning and Modification for STEP-NC

Jung-Hoon Cho* and Suk-Hwan Suh**

ABSTRACT

An increasing attention is paid to STEP-NC, the next generation CNC controller interfacing STEP-compatible data. In this paper, we first propose an architecture for the STEP-NC (called FBCC: Feature Based CNC Controller) accepting feature code compatible with STEP-data, and executing NC motion feature by feature while monitoring the execution status. The main thrust of the paper has been to develop an automatic on-line tool path generation and modification scheme for milling operation. The tool path is generated for each feature by decomposing into a finite number of primitive features. The key function in the new scheme is how to accommodate unexpected execution results. In our scheme, the tool path planner is designed to have a tracing capability for the tool path execution so that a new path can be generated from the point where the operation is stopped. An illustrative example is included to show the capability of the developed algorithm.

Key words : STEP-NC, Feature based CNC control, On-line tool path generation, On-line error recovery

1. 서 론

현재의 CNC 프로그래밍은 G-code(ISO 6983) 방식에 근거하여 공구의 궤적과 각종 기기의 조작을 직접 입력하는 형식이다. G-code를 기반으로 한 현재의 CNC 동작기구는 파트프로그램의 수정, 가공재개, 부분가공 등 운영상에 많은 문제점을 가진다(Table 1).

이러한 문제점을 극복할 수 있는 차세대 CNC제어기에 관한 연구는 G-code를 대체할 수 있는 새로운 코드의 개발^[1], 가공피쳐를 입력으로 하는 HNC(Hotonic Numerical Controller)^[2], FBC(Feature-based CNC)제어기^[3,4], 전체 가공과정을 통합하는 TRUE-CNC^[5,6] 등의 연구가 있으나 현재까지는 스킴의 제시에 머무르는 실정이다.

STEP-NC는 Product Model 및 공정계획의 호환 목적으로 제정된 STEP(ISO10303) 정보를 CNC 제어기가 관독할 수 있는 STEP 호환 NC 제어기로서,

CAD, CAPP, CAM, CNC의 일관 시스템 구축이 가능하다는 측면에서 큰 의미를 갖는다. STEP-NC에 관한 연구는 최근에 IMS project의 국제공동 연구팀 마로서 proposal 이 제출된 상태이다^[7]. 본 연구에서 제시하는 FBCC(Feature Based CNC Controller)는 1) STEP(ISO10303) 호환 피쳐 및 가공계획 정보를 입력으로 취하며, 2) STEP-NC의 자율적 NC코드 생성 및 모니터링 기능과, 3) HNC의 협제어 기능을 보유한 차세대 CNC 제어기이다.

본 연구에서는 실제적으로 현재의 CNC제어기의 성능을 개선할 수 있는 방안으로서 1) 가공피쳐(machining feature)와 피쳐그래프를 이용한 가공계획의 표현, 2) 공구경로의 자동 생성 및 가공상태의 파악을 위하여 단위가공피쳐(Primitive machining feature, 이하 primitive feature)의 개념 도입 및 자동 공구경로의 생성, 3) 가공의 진행을 판단하기 위하여 가공성 평가에 의한 공구의 선정 및 피쳐그래프의 수정 등의 스킴을 도입하였으며, 이와 같은 기능들을 수행할 수 있는 feature-based on-line 공구경로 생성 및 모니터링 알고리즘을 개발하였다. 논문의 구성은 2장에서는 FBCC의 입출력 및 구조, 3장

*학생회원, 포항공과대학교 산업공학과

**중신회원, 포항공과대학교 산업공학과

Table 1. Problems with G-code based part program

머신코드 형태의 파트프로그램	공정계획 정보 유실
-코드의 판독이 어려움 -작업 흐름 판독이 어려움 -부분 수정이 어려움	-공작물 셋업 및 공구 형상에 고정된 축동작 -현장이상상황 대처에 어려움
CNC 운영	Shop floor와의 인터페이스
-가공중 정지 및 가공재개가 어려움 -공구수명완료 및 공구파손시의 가공지령 재구성, 부분가공이 어려움	-DNC에서 지원하는 기능에 제한 -G-code정보로 인터페이스

에서는 on-line 공구경로 생성 및 모니터링 스킴, 4 장에서는 적용 예를 제시한다.

2. FBCC의 구조

2.1 FBCC의 입력과 처리능력

FBCC는 파트를 가공하기 위한 가공계획 정보를 STEP의 정보교환 방식을 이용한 FBCC코드의 형태로 입력 받는다. 본 논문에서는 FBCC 코드 자체는 다루지 않으며, 다만 Table 2(Input)와 같은 정보를 포함한다고 가정한다. 주지할 것은 Input 정보에는 공구경로가 포함되지 않으며, 이는 FBCC가 on-line 상에서 자율적으로 생성하는 것이다. STEP-NC의 입력정보는 ISO 14649로서 제정 중에 있으며 Table 2의 입력정보들은 ISO 14649로 표현될 수 있다. FBCC는 이외에도 생성된 공구경로의 실행상황 모니터링 및 이상상황 대처능력, 대외 CNC 협조제어 처리능력(Table 2의 처리능력 참조)을 갖는다.

2.2 FBCC의 구조

FBCC는 FBCC코드 입력, 가공계획 준비, 피쳐 선정 및 공구경로 생성을 위한 Intra-scheduling and planning, 공구경로의 실행을 위한 Intra-execution, 가공상태 감시를 위한 Intra-monitoring, 이상상황 분석 및 대처를 위한 Status analysis and emergency/

erroneous state solver, 상위 planner(Init Jobmanager & coordinator holon)와 외부 FBCC와의 협상을 담당하는 Inter-negotiation/bidding 모듈로 구성된다(Fig. 1).

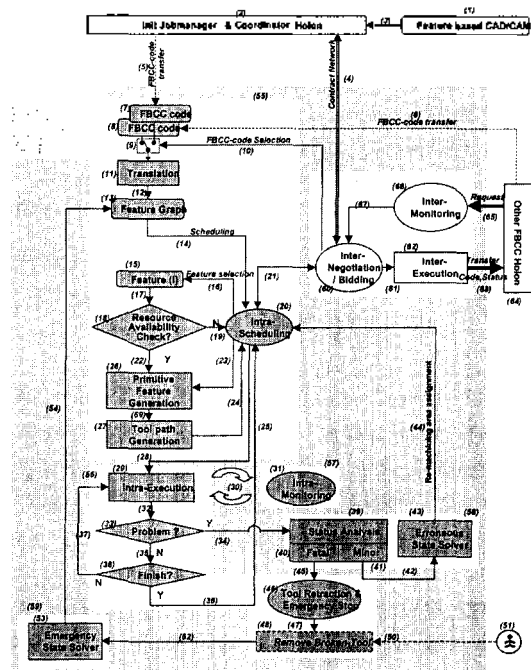


Fig. 1. FBCC architecture.

Table 2. Input and capability of FBCC

Input		처리능력
가공피쳐	피쳐타입, 형상, 가공조건, 공구	공구경로(G00, G01, G02, G03)
가공순서	가공선후관계	기계조작명령(M.F.S)
가공대안	동일 부위의 다른 가공계획	가공부위형상(primitive feature)
피쳐타입	공구경로생성방식 결정	가공순서판단(machining sequence)
가공형상	윤곽, 길이, 바닥형상 등	가공진행상황(feature graph update)
가공조건	피스타입, 절삭폭, 절삭깊이, 이송속도, 주축회전속도, 가공방향	대외 CNC 처리업무
공 구	타입, 치수	

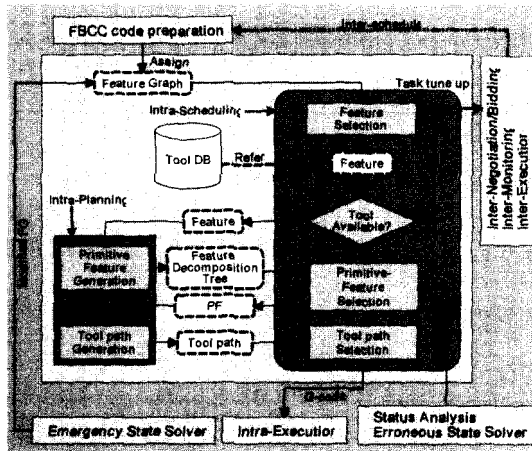


Fig. 2. Tool path generation module.

상위 planner로부터 계획되거나 다른 FBCC에서 요청된 FBCC 코드를 입력받아 Inter-negotiation and bidding 모듈의 판단 아래 가공할 파트를 선정하고, STEP-physical 파일 변환 과정을 거쳐 피쳐그래프 형태의 가공계획이 생성된다. Resource availability check 및 Intra-scheduling 모듈의 판단에 따라서 다음 가공피쳐를 선정한다. Intra-planning 모듈에서 단위가공피쳐 생성과정을 거쳐서 공구경로를 생성하고, Intra-execution 모듈에서 가공이 수행되며, Intra-monitoring 모듈에서 가공상황을 감시한다. 문제가 발생했을 때 상태분석 및 이상상황 대처를 실행하고, 피쳐그래프 갱신 과정을 거쳐 가공을 진행한다.

2.3 공구경로 생성 모듈의 구조

FBCC의 전체구조를 공구경로 생성부를 중심으로 재구성하면 Fig. 2와 같으며 세부 구조는 planning에 관련된 파트와 scheduling에 관한 파트로 나뉜다. Intra-Planning 파트는 주어진 피쳐의 공구경로 생성을 위하여 단위피쳐, 공구경로 정보들을 재산한다. Intra-Scheduling 파트는 피쳐그래프, 단위피쳐분해그래프, 공구경로 정보들로부터 피쳐, 단위피쳐, G-code 정보를 가공진도를 고려하여 순서대로 선정한다. 이상상황의 가공대책은 수정된 피쳐그래프의 형태로 전달되며 부분가공은 단위피쳐선택 파트에서 담당하도록 구성되어 있다.

3. On-line 공구경로 생성 방식

이와 같은 공구경로생성 모듈에서는 피쳐그래프로부터 공구경로(G-code)가 자동으로 생성되어 Intra-

Execution 모듈에서 가공이 수행되는데 3장에서는 공구경로를 생성하는 구체적인 방안을 제시한다. 그림 2의 공구경로 생성과정은 여러 단계의 과정을 거치게 되는데 주요한 정보의 변환과정은 피쳐그래프, 가공피쳐, 피쳐세분화그래프, 단위피쳐, 공구경로, G-code의 순서이다. 피쳐그래프는 파트형상 전체의 가공방법을 정의하며 가공피쳐는 피쳐그래프 중 한 스텝인 어떤 특징형상에 제한된 가공방법을 정의한다.

3.1 피쳐그래프의 구성 및 가공피쳐의 선정

3.1.1 피쳐그래프의 구성

피쳐그래프는 공정계획표현 모델의 일종으로서 AND/OR그래프에 기반을 두며 조립계획^[10] 또는 가공계획^[11] 등에 활용되었다. 피쳐그래프는 노드와 아크로 구성되며 노드는 Split-And, Split-Or, Joint-And, Joint-Or, Start, End, 가공피쳐 등의 7가지 타입으로 구성된다. Split-And와 Joint-And 노드는 한 쌍을 이루어 중간에 포함된 노드들이 모두 가공되어야 함을 표시한다(And관계). Split-Or와 Joint-Or 노드는 중간에 포함된 노드들 중에서 한가지 노드만 가공되어도 될 경우를 표시한다(Or관계).

피쳐그래프에 기술할 가공계획은 1) 특별한 제약이 필요 없는 가공계획과 2) 형상의 선후관계, 가공방법의 선후관계(황/정삭, 드릴/보링 등), 3) 사용할 공구군 관리 등 기술적으로 제약을 두어야 할 가공계획을 모두 표현할 수 있다. 피쳐그래프의 기본 구조는 순차, And, Or관계를 조합하여 Fig. 3과 같이 네 가지로 분류할 수 있다. 가공형상의 특성에 따라서 한가지 유형이 선택된다.

A, B와 A', B'는 각각 가공계획 1과 2의 가공피쳐들이다. "AB고정"은 A, B 피쳐간에 선후관계로 제약됨을 말하며 "AB자유"는 순서 상에 제약이 없다. "PS고정"은 가공계획 1, 2 사이에 교류가 없음을 말하며 "PS자유"는 가공계획을 공유함을 나타낸다. 가공순서의 종류는 Table 3과 같이 고정관계가 많을수록 줄어들며 자유관계가 많을수록 많아진다. 이상상

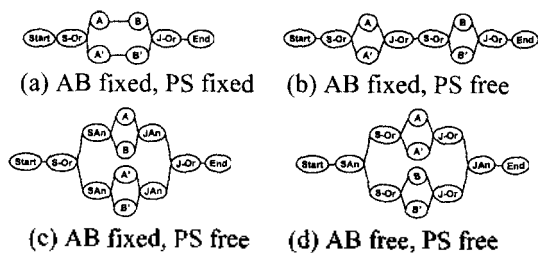


Fig. 3. Basic structure of feature graph.

Table 3. Feature graph and machining sequence

Plan	Description	Machining sequence
(a)	A, B간에 가공순서 고정 표준가공, 가공대안에서 dependency 존재	$A \rightarrow B$ or $A' \rightarrow B'$
(b)	A, B간에 가공순서 고정 표준가공, 가공대안에서 dependency 없음	$A \rightarrow B$ or $A' \rightarrow B'$ or $A' \rightarrow B$ or $A \rightarrow B'$
(c)	A, B간에 가공순서 자율 표준가공, 가공대안에서 dependency 고정	$A \rightarrow B$ or $B \rightarrow A$ or $A' \rightarrow B'$ or $B' \rightarrow A'$
(d)	A, B간에 가공순서 자율 표준가공, 가공대안에서 dependency 없음	$A \rightarrow B$ or $A \rightarrow B'$ or $A' \rightarrow B$ or $A' \rightarrow B'$ or $B \rightarrow A$ or $B \rightarrow A'$ or $B' \rightarrow A$ or $B' \rightarrow A'$

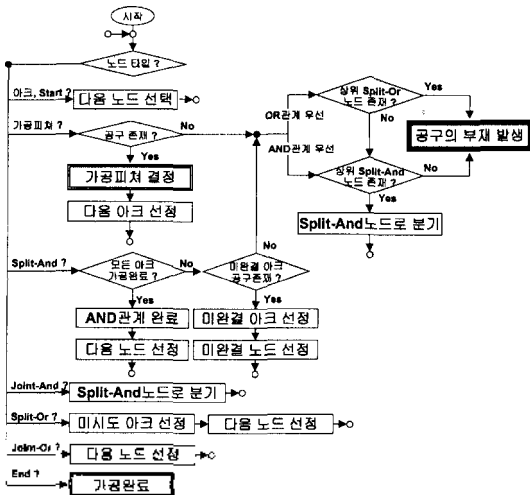


Fig. 4. Selection of machining feature.

황에서 가공순서가 많을수록 가공상황에 유연하게 대처할 수 있다.

1.1.2 On-line 가공을 위한 가공피쳐의 선정

FBCC의 Feature selection 파트는 피쳐그래프 상의 가공피쳐를 순서대로 선택한다(Fig. 4). AND관계에서는 세부그래프를 모두 완성해야 하며 공구의 부재시에 다른 세부그래프를 먼저 가공할 수 있다. 세부그래프에서 더 이상 가공할 수 있는 피쳐가 없을 경우에 가공이 중단되고 알람(alarm)이 발생한다.

OR관계에서는 우선순위가 높은 세부그래프를 먼저 선정하는데 가공대안을 선정할 경우에는 차선의 세부그래프를 선택한다. 가공대안의 선정 시에 알람을 발생시켜 피쳐그래프를 수정한다. AND 관계와 OR관계가 복합된 경우에는 공구부재시 가공할 수 있는 피쳐를 먼저 가공하는 AND관계 우선, 가공대안을 즉시 선택하는 OR관계 우선의 경우로 나뉜다.

3.2 가공피쳐(Machining feature)

피쳐그래프의 주요 구성요소는 가공피쳐노드이다.

여기서 가공피쳐는 유사한 모양의 특징형상을 가지는 개개의 작업단위로서 각각 사용되는 공구의 타입과 가공방법이 다른 평면가공, 측면가공, 포켓가공, 홈가공, 구멍가공 등을 말한다. 각각의 가공피쳐가 가지는 정보는 Table 2에서 가공순서와 가공대안을 제외한 나머지 정보들이며 on-line 공구경로 생성시에는 이 정보들로부터 단위피쳐를 생성하게 된다.

3.3 단위피쳐(Primitive feature)

가공피쳐가 가공해야 할 특징형상은 큰 덩어리로서 공구로 조금씩 나누어서 가공해야 한다. 단위피쳐는 가공피쳐 내의 일부 가공부위로서 가공면, 제거형상, 가공조건, 공구 정보를 가지며 이들로부터

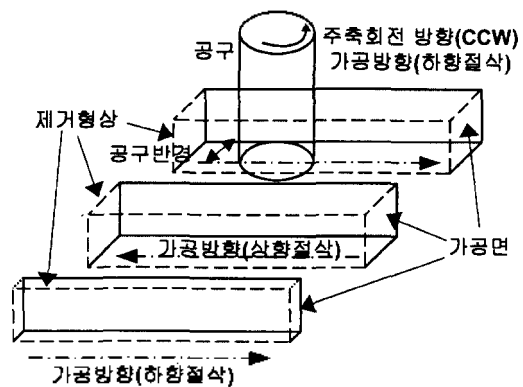
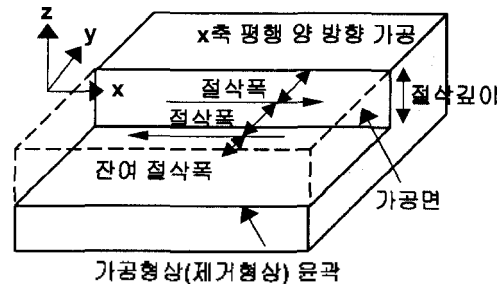


Fig. 5. SIDE-feature and its primitive features.

자동으로 생성되는 이동경로, 접근경로, 절삭경로, 빠짐경로의 네 가지 종류의 공구경로 한 묶음을 이용하여 제거형상을 모두 가공하고 가공면을 생성할 수 있는 특성을 가지는 작은 크기의 가공피처를 말한다. Primitive feature는 가공피처와 동일한 정보를 가지며, 단일 공구경로로서 제거되는 볼륨을 의미한다.

1.3.1 SIDE피처를 위한 단위피처의 생성

SIDE피처의 단위피처는 동일한 공구와 가공방식을 사용하는 포켓피처의 단위피처와 동일하게 취급된다. 단위피처의 형상은 Fig. 5와 같이 가공형상을 절삭폭, 절삭깊이, 경로 타입에 따라서 나눈 형상이며 생성과정은 포켓피처의 경우(3.3.3절)와 동일하다.

1.3.2 구멍피처를 위한 단위피처의 생성

구멍피처는 드릴공구의 1회 절입깊이가 단위피처의 형상을 결정하므로 전체 구멍형상을 분할하는 가장 단순한 형태로 정의된다(Fig. 6참조).

1.3.3 포켓피처를 위한 단위피처의 생성

(1) 포켓피처의 정보

포켓피처 정보를 구성하는 가공형상, 가공조건, 공구정보와 단위피처 분해를 위한 트리 정보는 표 4와 같다.

본 논문에서 가정하는 포켓피처의 형상은 그림 7과 같이 상면과 바닥면이 동일한 높이의 평면이며 윤곽과 아일랜드의 2차원 윤곽으로 범위가 결정된다(높이가 달라질 경우에는 각각의 높이에 대하여 포켓을 나누어 정의하는 방식을 취한다). A_T 는 가공형상영역으로서 포켓윤곽과 아일랜드 윤곽 정보를 가지며

A_R 은 소재형상영역의 윤곽정보를 가진다. 포켓피처에서는 초기에는 $A^R=A_T$ (가공형상영역이 소재로 채워져 있음)이다. d_z, d_{xy} 는 포켓의 현재 깊이와 가공폭의 현재 두께를 의미하는데 단위피처가 되기 위해서는 $d_z=d_A, d_{xy}=d_R$ 이 되어야 한다. b_C 는 평면으로 가정한다. p_C 는 "axis parallel" 또는 "contour parallel" 타입을 지정한다. v_D 는 axis parallel 타입의 가공영역 절단방향을 지정한다.

(2) 단위피처의 기하학적 연산

단위피처의 가공형상은 2차원의 가공영역과 높이로 표시된다. 어떤 영역은 닫힌 경계와 그 내부를 말하며 "A+아래첨자"로 표시하거나 <경계, 경계, ...>와 같이 표시한다. 경계는 영역의 가장 바깥쪽의 윤곽을 말하며 "C+아래첨자"로 표시한다.

어떤 윤곽의 오프셋(offset)은 윤곽과의 최소거리가 일정한 점들의 집합이며, 불리안 작업은 경계 또는 영역끼리의 합, 차, 교차의 작업이다. Offset작업과 불리안 작업은 경계와 영역에 둘 다 적용될 수 있으며 결과는 각각 새로운 경계와 영역이 된다. 즉,

- 경계=경계 (U, ∩, -) 경계
- 영역=영역 (U, ∩, -) 영역
- 경계=Offset(경계, 오프셋거리)
- 영역=Offset(영역, 오프셋거리) 와 같다.

단위피처의 형상정보는 가공형상영역(A_T)과 소재형상영역(A_R)으로 나뉜다(Fig. 8). 가공형상영역은 가공을 통해서 만들고자 하는 영역이며 소재형상영역은 제거해야할 영역이다.

가공형상영역(A_T)의 경계를 이용하여 공구경로를 구하고 공구경로를 이용하여 공구통과영역을 구하며 이를 이용하여 실가공영역과 미가공영역을 구한다. 실가공영역은 소재형상영역(A_R)에서 공구와 절삭경로에 의해서 실제로 가공이 되는 영역을 말한다. 실가공영역의 계산은 다음 식에 의한다.

$$C_{Tool} = \text{Offset}(C_T, -d_t/2) \tag{1}$$

$$A_{Tool} = \langle \text{Offset}(C_{Tool}, +d_D/2), \text{Offset}(C_{Tool}, -d_D/2) \rangle \tag{2}$$

$$A_{Cut} = A_{Tool} \cap A_R \tag{3}$$

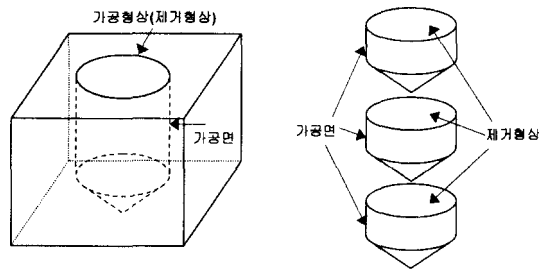


Fig. 6. HOLE-feature and its primitive features.

Table 4. POCKET-feature information

가공형상	가공조건	공구정보
origin_placement: o_n	axial depth of cut: d_A	tool type: t_c
origin_axis: o_x, o_y, o_z	radial depth of cut: d_R	tool diameter: d_D
maximum_depth: d_z	path type: p_C	트리구조
maximum_width: d_{xy}	cut direction: v_D	
bottom_condition: b_C	cutmode_type: c_m	left arc
target_area: A_T		right arc
removal_area: A_R		parent arc

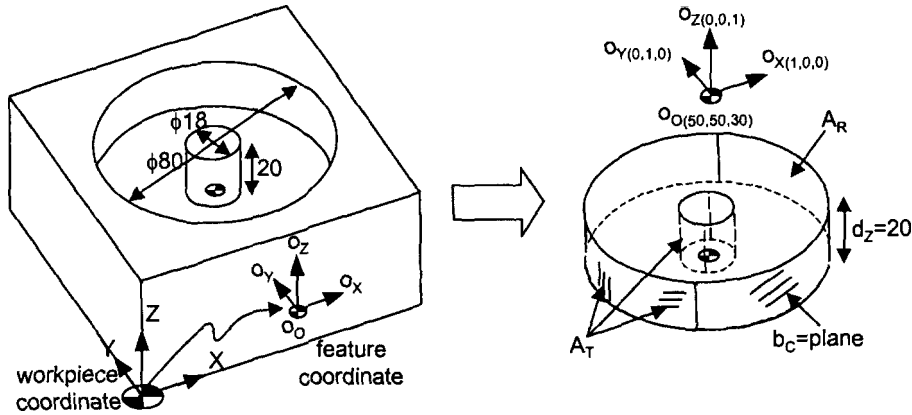


Fig. 7. Pictorial view of pocket feature information.

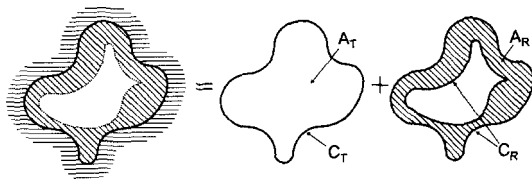


Fig. 8. Decomposition of primitive feature.

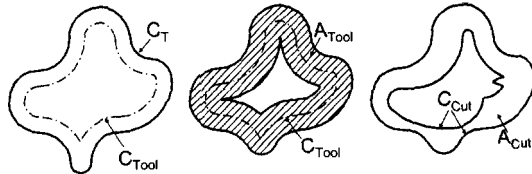


Fig. 9. Tool sweep area and cut area.

$d_p/2$ 은 공구의 반경, C_{Tool} 는 절삭경로, A_{Tool} 은 공구 통과 영역, A_{Cut} 은 실가공영역이다.

미가공영역은 가공 후 잔여영역이며 다음 식에 의해 구해지고 이는 다시 공구간섭영역과 절삭폭과다 영역으로 분리된다.

$$A_{Uncut} = A_R - A_{Tool} \quad (4)$$

공구간섭영역은 그 경계가 가공형상영역의 경계의 일부인 영역을 말하며 절삭폭과다 영역은 가공형상영역의 경계 내부에 존재한다.

미가공영역은 본래의 단위피처에서 분리되어 독립적인 가공형상으로 변환된다. 이때 미가공영역의 가공형상영역은 본래의 단위피처의 가공형상영역과 일치하게 되고 소재형상영역이 (4) 식에 의해서 계산된 영역이 된다. 모체가 되는 단위피처의 가공형상영역(A_T)은 비절삭영역을 제외한 영역으로 수정되며 (식 (5)) 소재형상영역은 식 (3)의 실가공영역이 된다.

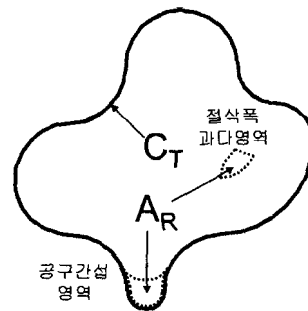


Fig. 10. Uncut area.

$$A_T' = A_T - A_{Uncut} \quad (5)$$

이상의 과정은 Pocket_PF_Division에서 수행된다. 단위피처 P 와 공구직경 d_p 가 입력이며 결과로 가공영역 P_{Cut} , 미가공영역 P_{Uncut} 이 구해진다.

PROCEDURE Pocket_PF_Division()

INPUT: P, d_p ;

OUTPUT: P_{Cut}, P_{Uncut} ;

PROCEDURE:

$P_{Cut} = P_{Uncut} = P$;

$C_{Tool} = \text{Offset}(P.C_T, -d_p/2)$;

$A_{Tool} = \langle \text{Offset}(C_{Tool}, d_p/2), \text{Offset}(C_{Tool}, -d_p/2) \rangle$;

$P_{Uncut}.A_T = P.A_T$;

$P_{Uncut}.A_R = P.A_R - A_{Tool}$;

$P_{Cut}.A_T = P_{Cut}.A_T - A_{Uncut}$;

$P_{Cut}.A_R = A_{Tool} \cap P.A_R$;

END_PROCEDURE;

(3) 포켓피처의 단위피처 분해

포켓피처의 전체가공영역은 절삭깊이, 경로타입, 절삭폭에 의해서 단위가공영역으로 나누고 각 단위

가공영역에 대하여 단위피처의 기하학적 연산에 의해 가공영역을 분할한다. 공구간섭영역은 다음 가공영역에서 함께 가공하고, 절삭폭초과영역은 다른 형태의 경로로 가공한다(3.4.2절 참조). 이 과정을 모든 단위가공영역에 대하여 적용한다. 분해과정은 절삭깊이 방향 분할, 절삭폭 방향 분할, 단위피처 분할, 단일 가공영역 분할의 네 가지로 나뉜다.

STEP-1) 절삭깊이 방향 분할: 주어진 가공피처(P)를 바닥면을 기준으로 가공깊이(d_A) 만큼의 영역(P_{JA})과 나머지 영역(P_{JR})으로 분할한다. 분할 시의 정보 수정은 다음과 같다.

- 정보 초기화: $P_{JA}=P_{JR}=P$
- 분할영역 가공깊이 수정: $P_{JA}.d_z=\min(P.d_A, P.d_z)$
- 잔여영역 원점 수정: $P_{JR}.O_0=P.O_0+(P_{JA}.d_z) P.O_z$
- 잔여영역 가공깊이 수정: $P_{JR}.d_z=\max(P.d_z- P_{JA}.d_z, 0)$

STEP-2) 절삭폭 방향으로의 분할: 절삭깊이로 분할된 영역(P_{JA})을 절삭폭 만큼의 영역(P_{JW})과 나머지 영역(P_{JR})으로 분할 한다. 분할방법은 윤곽방향분할과 평행축방향분할로 나뉜다(Fig. 11).

- 정보 초기화: $P_{JW}=P_{JR}=P$
- 분할영역 소재형상영역 수정:
 - For $p_c=$ contour parallel
 - $P_{JW}.A_R=\langle P.C_T, \text{Offset}(P.C_T^T, -d_R) \rangle$
 - For $p_c=$ axis parallel
 - $P_{JW}.A_R=P.A_R$ 을 수직평면으로 분할한 영역의 $-v_D$ 방향 영역(Fig. 11b)
- 분할영역 잔여가공폭 수정: $P_{JW}.d_{xy}=d_R$
- 잔여영역 가공형상영역, 소재형상영역 수정:
 - $P_{JR}.A_T=P_{JR}.A_R=P.A_T-P_{JW}.A_R$
- 잔여영역 잔여가공폭 수정: $P_{JR}.d_{xy}=\text{MAX}, (\text{MAX는 충분히 큰 양수}).$

STEP-3) 단위피처의 분할
 절삭폭 방향으로 분할된 가공영역(P_{JW})을 (2)의 Pocket_PF_Division 함수로 분해하여 실가공영역과

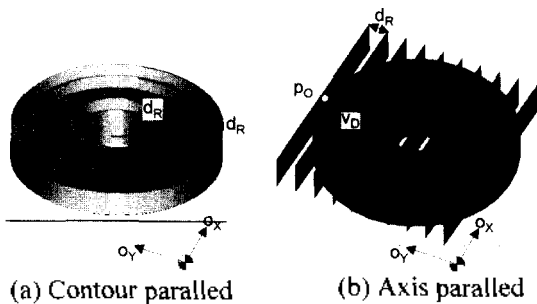


Fig. 11. Division by radial depth of cut.

미가공영역을 계산한다. 공구간섭영역은 다음 가공 피처에서 가공되도록 할당한다.

STEP-4) 단일가공영역으로 분할

실가공영역은 분리된 몇 개의 절삭경로를 가질 수 있으므로 각각의 절삭경로마다 단일가공영역으로 분해한다. 절삭경로는 순수하게 소재형상영역의 제거에 관여하는 경로이다. 전체 절삭경로는 가공형상영역의 전체 경계의 읍셋으로 구한다. 이 중에서 미절삭경로는 가공형상영역에서 소재형상영역을 뺀 영역의 경계를 읍셋한 윤곽으로 볼 수 있다. 전체 절삭경로에서 이를 제거하면 실제 절삭경로를 식 (6)과 같이 구할 수 있다. 여기서 C_{Tool} 은 실질삭경로, A_{Free} 는 미절삭영역을 말한다.

$$C_{Tool}=\text{Offset}(C_T, -d_T/2)-\text{Offset}(C_{Free}, -d_T/2) \quad (6)$$

where $A_{Free}=A_T-A_R$ and $C_{Free}=\text{boundary of } A_{Free}$

실절삭경로 C_{Tool} 은 여러 개로 나뉘는데 각각의 분리된 경로(C_{Tool}^*)에 대하여 임의의 순서로 가공영역 분할이 발생한다. 임의의 가공경로 C_{Tool}^* 에 대하여 다음과 같이 가공영역 정보를 갱신한다.

- 정보 초기화: $P_{JS}=P_{JR}=P$
 - 소재형상영역 정보: $P_{JS}.A_R=P.A_R \cap A_{Tool}$
 where $A_{Tool}=\text{Offset}(\langle C_{Tool}^* \rangle, d_T/2)$
 - 가공형상영역 정보: $P_{JS}.A_T=P.A_T-P.A_R+P_{JS}.A_R$
 - P_{JR} 의 가공형상영역 정보: $P_{JR}.A_R=P.A_R-A_{Tool}$
- 이 과정은 Test_Single_Area()함수에서 수행된다.

PROCEDURE Test_Single_Area()

```

INPUT: 단위피처 P;
OUTPUT: 단위영역 분해 결과 P.left, P.right;
PROCEDURE:
    If (P.A_R=∅) return;
    P_JS=P_JR=P;
    A_Free=P.A_T-P.A_R;
    C_Tool=Offset(P.C_T, -d_T/2)-Offset(P.C_Free, -d_T/2);
    C_Tool*=Select a tool path from C_Tool;
    A_Tool=Offset(⟨C_Tool*⟩, d_T/2);
    (⟨C_Tool*⟩은 경계가 C_Tool*, 면적이 0인 영역)
    P_JS.A_R=P.A_R ∩ A_Tool;
    P_JS.A_T=P.A_T-P.A_R+P_JS.A_R;
    P_JR.A_R=P.A_R-A_Tool;
    P.left=P_JS;
    P.right=P_JR;
    Test_Single_Area(P.right);
END_PROCEDURE;
STEP1~3의 과정으로 생성되는 단위피처는 실가공
    
```

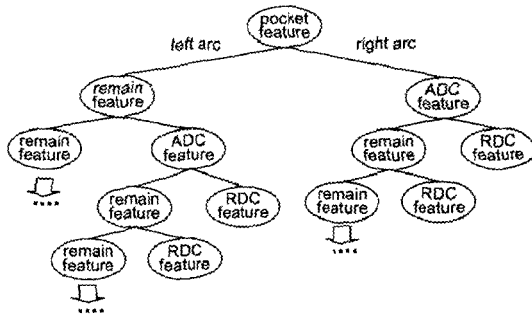


Fig. 12. Binary tree.

영역(단위피처), 공구간섭영역, 절삭폭초과영역으로 분해된다.

(4) 단위피처의 저장

단위피처의 생성과정을 저장하는 이진트리(binary tree)는 포켓피처를 루트노드(root node)로 하며 자식노드를 가리키는 left arc와 right arc를 가진다. right arc는 단위피처를 생성할 가공영역을 가리키며 left arc는 자신의 가공영역에서 right arc의 가공영역을 제외할 나머지 가공영역의 정보를 가리킨다. 최종적으로 leaf node는 단위피처 또는 미가공영역으로 세분화된다(Fig. 12). 트리구조에서 어떤 노드의 sibling 노드는 parent 노드의 다른 하위 노드를 말하며 uncle노드는 parent 노드의 sibling 노드를 말한다.

(5) 단위피처 분해 함수

포켓정보는 P로 주어지며 출력은 이진트리의 단위피처세분화 결과이다. 알고리즘의 전체과정은 Pocket_To_PF에서 수행된다.

```

PROCEDURE Pocket_To_PF(i)
INPUT: Pocket data (P);
OUTPUT: Pocket decomposition (Pleft, Pright);
PROCEDURE:
  If (P.AR=∅) return;
  Else If (dY>dX) { //STEP-1
    PDA=Divide P with dX;
    Pocket_To_PF(P.right (≠PDA);
    Pocket_To_PF(P.left (≠P-PDA);
  }
  Else If (dXV>dR) { //STEP-2
    PDR=Divide P with dR;
    Pocket_To_PF(P.right (≠PDR);
    Pocket_To_PF(P.left (≠P-PDR);
  }
  Else { // STEP-3,4

```

```

PUncut, PCut=PF_Division(P);
If (PUncut=∅) Test_Single_Area(P≠PCut);
If (PCut≠∅) {
  If (PUncut="exceeds RDC") {
    Test_Single_Area(P.left (≠PCut);
    P.right (≠PUncut);
  }
  Else If (PUncut="interference") {
    Pparent=Pparent-PUncut;
    Pocket_To_PF(P.uncle≠P.uncle ∪ PUncut);
    Test_Single_Area(P≠PCut);
  }
}
}
END_PROCEDURE;

```

3.4 공구경로 생성

공구경로는 가공피처의 단위피처세분화트리를 이용하여 생성한다. 먼저 트리의 leaf노드를 좌측에서부터 차례로 선정하며 선정된 leaf노드의 특성에 따라서 단위피처와 미가공영역의 두 가지로 나누어 절삭경로를 생성하고 현재 공구의 위치에 따라서 빠짐경로, 이동경로, 접근경로를 생성한다. 각각의 경로는 G-code로 구성되며 경로에서 G-code로의 변환은 미리 정의된 변환표에 의한다.

3.4.1 가공영역의 절삭경로의 생성

3.3절로부터 단위피처는 단일절삭경로로 가공할 수 있는 크기의 소재형상을 가지도록 분해되었다. 절삭경로(C_{tool})는 식 (6)으로 구한다. C_{tool}에서 절삭방향(c_m, 상향/하향)으로 시작점과 끝점을 결정하여 절삭경로로 변환한다. 절삭경로타입과 위치좌표는 CI정보의 형태로 저장되며 가공직전에 G-code로 변환된다.

3.4.2 미가공영역의 절삭경로의 생성

미가공영역은 공구간섭영역과 절삭폭초과영역으로 나뉜다. 공구간섭영역은 가공피처를 완성할 수 없는 경우이므로 이상상황으로 처리한다(3.5절 참조). 절삭폭초과영역은 Axis parallel타입과 절삭폭 d_R로 영역을 분할하여 가공한다(Fig. 13).

가공영역분할은 Divide_Uncut_Area 함수에 의해서 수행된다. A_n는 수행순에 의해서 분해된 영역, N_n는 영역의 개수를 말한다. A₁는 -x방향에서부터 정렬되어 있으며 A₁는 처음 영역을 말한다. 분할된 결과로서 Fig. 13의 A₁₀~A₁의 순서로 가공된다.

```

PROCEDURE Divide_Uncut_Area(i)

```

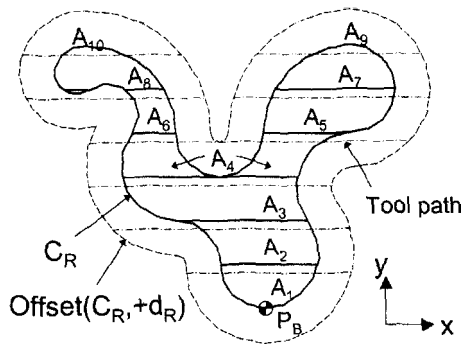



Fig. 13. Toolpath for uncut area.

INPUT: Uncut area P;
 OUTPUT: 가공영역 분해 트리 P.left, P.right;
 PROCEDURE:

```

    If (PAR=φ) return;
    PDR=PDR=P;
    PB=Find bottom point of P;
    Ai, Ni←Divide P with line of y=PB and y=PB+dR;
    PDR.Ai=P.Ai∩P.AR+Ai;
    PDR.AR=Ai;
    PDR.AR=P.AR-Ai;
    P.right=PDR;
    Divide_Uncut_Area(P.left←PDR);
    END_PROCEDURE;
    
```

영역 A_i의 실질삭경로는 $y_{i11}=P_B(i)+d_R/2$ 를 지나는 수평선이 Offset(C₁, +d_D)을 좌우로 절단된 선분(Fig. 13참조)이다. 실질삭경로는 직선경로타입의 CL정보로 저장되며 가공작전에 G-code로 변환된다.

3.4.3 보조경로의 생성

이동경로는 Z축을 고정시키고 단위피치의 시작점의 XY좌표로 이동한다. 접근경로는 Z축을 따라서 단위피치의 가공면으로 이동한다. 빠짐경로는 다음 단위피치의 가공시작점의 좌표에 따라서 Z축 빠짐량을 결정한다. 단위피치에서 사용되는 공구와 절삭조건이 변경될 경우에 공구교환경로와 절삭조건 설정 코드가 추가된다.

3.5 이상상황의 복구

이상상황은 크게 세 가지로 나뉘는데 Feature selection 파트에서 더 이상 선정할 가공피치가 없는 가공전 이상상황, 공구의 파손이나 과도한 마모의 문제로 인한 가공 중 이상상황, 가공 후에 가공결과에 의해 요구되는 가공후 이상상황이 있다. 가공대안의

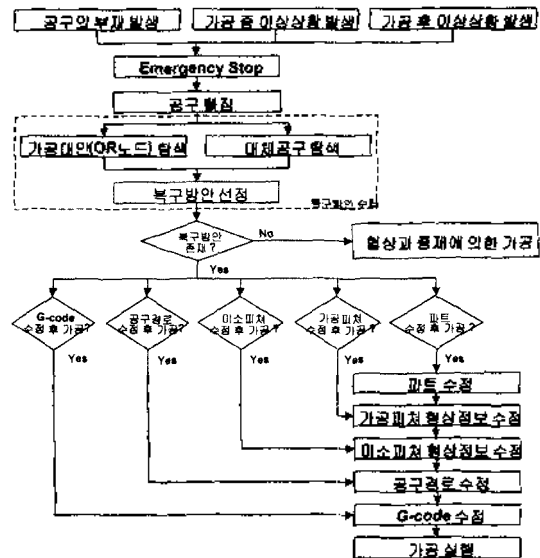


Fig. 14. Emergency state recovery.

탐색은 피쳐그래프에서 상위 OR관계를 찾는 과정이며 Split-Or노드의 존재여부로 대안을 결정한다. 대체공구의 탐색 과정에서는 G-code, 공구경로, 단위 피치, 가공피치, 가공되쳐, 파트의 단계로 모든 가능한 대체공구를 탐색한다.

이상상황의 복구는 현재까지의 가공정보를 수정하고 새로운 가공정보를 생성해 내는 과정으로서 가공정보의 수정이 주요 과정이다. 가공정보의 수정은 복구방안에 따라서 G-code, 가공경로, 단위피치, 가공 피치, 피쳐그래프의 단계별로 발생한다. 이러한 과정은 복구과정의 전체적인 흐름(Fig. 14)에서 설명되어 있다.

3.5.1 G-code의 수정

동일한 공구로 가공을 재개할 경우에 G-code의 수정으로 가공재개가 가능하다. G-code는 경로의 종류, 좌표, 이송속도 정보를 가진다. 문제 발생 지점에서 경로가 분할되어 나머지 경로가 절삭경로로 사용되며 절삭경로 실행 전에 공구의 교환경로, 이동경로, 접근경로가 삽입된다.

3.5.2 가공경로의 수정

가공재개를 다른 직경의 공구로 수행하며 단위피치의 수정 없이 가공이 가능한 경우에 해당한다. 각각의 단위피치마다 가공경로가 재계산되며 가공중단 지점의 단위피치는 잔여형상 계산 후에 가공경로가 재계산된다.

가공이 중단된 단위피치(P, Fig. 16(a))는 이진트리 상에서 가공된 영역(P_{Cut})과 잔여영역(P_{Uncut})으로 분

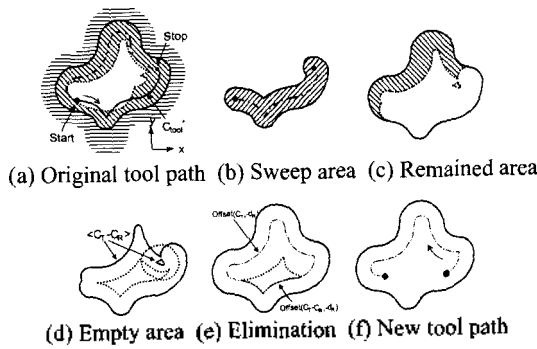


Fig. 15. Tool path modification.

해한다. 분해과정은 다음과 같다.

- 1) 정보초기화: $P_{Cut} = P_{Uncut} = P$
- 2) 수행된 경로 계산 (Fig. 15(a)): C_{Tool}^*
- 3) 공구통과영역 계산 (Fig. 15(b)):
 $A_{Tool} = \text{Offset}(\langle C_{Tool}^* \rangle, d_D/2)$
- 4) 가공영역의 소재형상 계산:
 $P_{Cut, AR} = P_{AR} \cap A_{Tool}$
- 5) 가공영역의 가공형상 계산:
 $P_{Cut, AT} = P_{AT} - P_{AR} + P_{Cut, AR}$
- 6) 잔여영역의 가공형상 산출: $P_{Uncut, AT} = P_{AT}$
- 7) 잔여영역의 소재형상 계산 (Fig. 15(c)):
 $P_{Uncut, AR} = P_{AR} - A_{Tool}$

잔여영역은 단위피치의 특징을 지니므로 3.4.1의 절삭경로 생성과정으로 가공경로를 계산한다. 소재형상 제거에 필요한 실가공경로는 Fig. 15(f)와 같으며 이는 전체 공구경로(Fig. 15(e))에서 자유영역 음셋윤곽(Fig. 15(d))을 제외한 윤곽이다. 계산된 C_{Tool} 의 공구통과영역은 소재형상을 모두 제거하며 가공시작점에서 소재형상과 접하도록 공구위치를 결정해주는 특징을 가진다(Fig. 16).

3.5.3 단위피치의 수정

이 경우는 단위피치를 수정하여 가공피치를 재가공할 경우이다. 단위피치의 수정은 현재의 단위피치

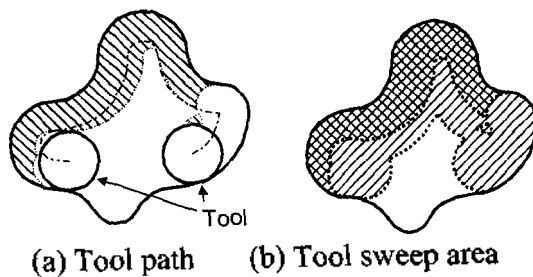


Fig. 16. Modified tool path.

의 분할 상태를 초기 가공영역 분할 상태로 보고 단위피치 영역분할 방식(STEP-3,4)을 이상상황 발생 피치부터 마지막 단위피치까지 적용하여 수행한다. 가공이 중단된 단위피치를 가공영역과 잔여영역으로 분리한 후에 단위피치 수정함수(Modify_Pocket_PF)를 적용한다. 함수의 주요 과정은 이진트리의 좌측노드부터 leaf노드를 찾고, 새 공구로 단위피치분할(PF_Division)을 수행한 후에, 절삭폭초과영역은 이진트리로 분해하며 공구간섭영역은 return하여 다음 영역에 추가하는 방식으로 우측 leaf노드까지 진행된다.

PROCEDURE Modify_Pocket_PF()

INPUT: Primitive feature (P);

OUTPUT: Modified primitive feature of P;

PROCEDURE:

```

If (P.AR=φ) return;
If (P.left≠φOR P.right≠φ) {
  If (P.left≠φ) dPF=Modify_Pocket_PF(P.left);
  If (P.right≠φ) {
    dPF=Modify_Pocket_PF(P.right+dPF);
    return dPF;
  }
}
Else {
  P_Uncut, P_Cut=PF_Division(P);
  If (P_Uncut=φ) Test_Single_Area(P);
  If (P_Uncut≠φ) {
    If (P_Uncut="exceeds RDC") {
      Test_Single_Area(P.left←P_Cut);
      P.right←P_Uncut;
      return φ;
    }
    Else If (P_Uncut="interference") {
      Test_Single_Area(P←P_Cut);
      return P_Uncut;
    }
  }
}
}

```

END_PROCEDURE;

Fig. 7의 포켓형상을 직경 12 mm의 엔드밀과 contour parallel 경로로 가공한다면 Fig. 17(a)와 같은 단위피치들이 얻어진다. 초기 이진트리의 첫번째 leaf 노드를 가공하는 도중에 직경 16 mm의 엔드밀로 교환할 경우에 이진트리를 수정하여 16 mm 공구에 대한 단위피치들을 얻는 과정을 Fig. 17(b)에 나

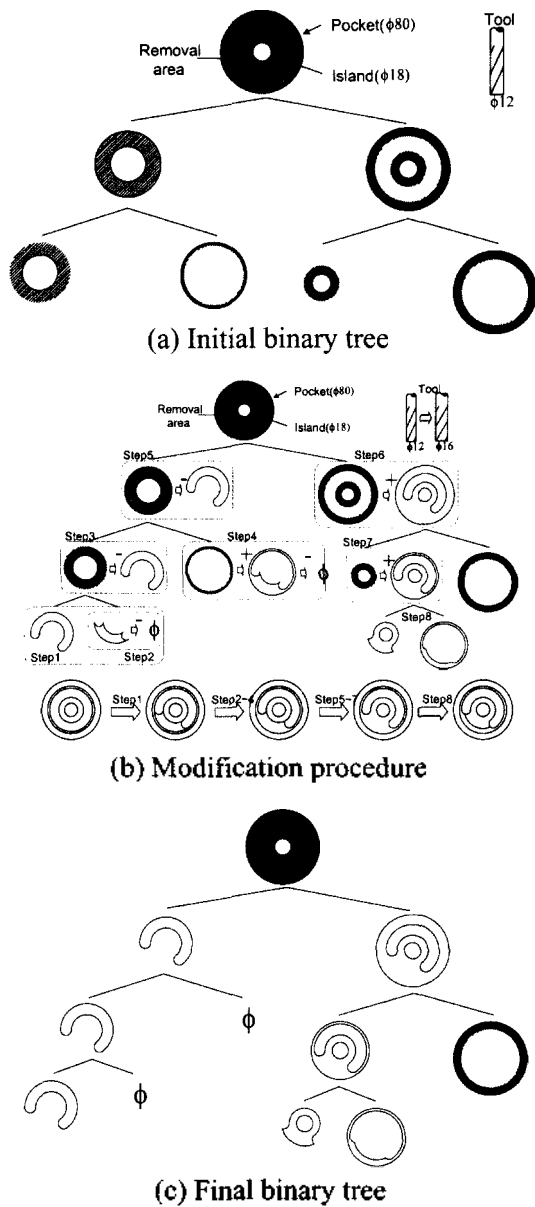


Fig. 17. Modification of primitive feature.

타내었다. Step1에서 첫번째 단위피쳐의 형상이 가공된 영역과 잔여영역으로 분할되었다. 이때 전체 포켓형상에서의 영역 분할 상태는 Fig. 17(b)의 하단에 표시하였다. Modify_Pocket_PF() 함수를 Step1까지 적용된 이진트리에 적용하면 Step2~Step8의 과정을 거쳐서 Fig. 17(c)의 이진트리로 수정된다. 각각의 단계에서 음수(-)로 표시된 단위피쳐는 직경 16 mm의 엔드밀로 가공이 되지 않아서 이진트리에서 제거되

는 과정을 나타내며, 양수(+)로 표시된 단위피쳐는 sibling노드에서 넘어온 가공되지 않은 단위피쳐를 받아서 자신에게 합쳐주는 과정을 나타낸다.

3.5.4 가공피쳐의 수정

가공피쳐를 수정한다는 것은 어떤 가공형상을 가공할 수 있는 다른 set의 가공피쳐의 조합을 사용하여 가공한다는 것으로서 1) 현재의 가공피쳐의 조합을 사용하여 자연스럽게 수정된 가공피쳐의 조합을 유도하는 방식, 2) 가공대안 등에 의하여 강제적으로 다른 가공피쳐의 조합을 적용하는 방식으로 나뉜다. 두번째 방식은 피쳐그래프의 가공대안에 의해서 수행된다.

첫번째 방식은 Modify_Pocket_PF함수의 수행결과에 의해서 수행된다. 만일 $dPF = \text{Modify_Pocket_PF}(\text{pocket})$ 에서 $dPF \neq \phi$ 일 때는 새 공구로 가공할 수 없는 경우이다. dPF의 처리에는 1) 다른 공구로 즉시 가공하는 방안, 2) 다른 가공피쳐에서 처리하는 방안의 두 가지가 존재한다.

다른 가공피쳐에서 처리하는 방안은 dPF가 다른 가공피쳐의 제거형상으로 사용되어 공구경로 생성시에 반영되도록 한다. 이는 dPF의 가공형상이 다른 가공피쳐의 가공형상에 포함될 경우인 $dPF.C_T \subset F.C_T$ 에 해당한다. dPF를 포함하는 F의 소재형상은 $F.A'_R = F.A_R + dPF.A_R$ 와 같이 수정되면서 dPF를 흡수한다.

두번째 방식인 가공대안에 의한 수정은 가공형상 내의 동일한 가공영역에 대하여 다른 가공피쳐의 조합으로 가공하는 것으로서 가공대안의 가공피쳐들에 대하여 소재형상의 전면적인 수정이 요구된다. 이것은 초기가공계획의 가공결과에 의해서 선가공된 영역이 발생했기 때문인데 가공대안의 가공피쳐들에서는 다음과 같은 수정이 요구된다.

1) 소재형상 변경: 가공피쳐의 소재형상에서 선가공된 소재형상을 제거한다. 소재형상의 제거에는 슬리드형상의 볼리안작업이 요구된다.

2) 가공피쳐 수정: 소재형상이 변경됨으로써 원래 가공피쳐를 그대로 처리할 수 없는 경우에는 처리될 수 있는 형태로 수정된다.

소재형상 변경 후에 포켓피쳐는 소재형상의 높이가 일정해야 하는 제약조건이 위배될 수 있다. 이를 만족시키기 위하여 동일한 높이의 포켓피쳐의 집합으로 분할하며 Update_F()에 의해서 수행된다. Fig. 18의 경우에는 B₃, B₄, B₂, B₇, B₈, B₆, B₁₀, B₁₁, B₉, B₅, B₁의 순서로 분할된다.

PROCEDURE Update_F()

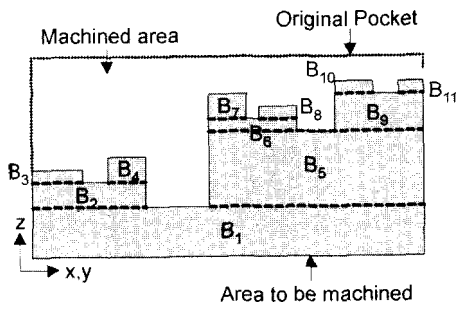


Fig. 18. Division of machining feature.

INPUT: 가공피쳐 P;
 OUTPUT: 분할된 가공피쳐 리스트 P*;
 PROCEDURE:

```

    If (P.Ar=∅) return;
    PB=Find base pocket;
    P←P-PB;
    PB*=∅;
    While (Standing pocket exists) {
        PS=Standing pocket in P;
        P=P-PS;
        PB*←PB*+Update_F(PS);
    }
    return {PB*, PB};
    End_Procedure;
    
```

3.5.5 파트형상의 수정

이 경우는 가공피쳐의 가공대안이나 대체공구가 없는 경우로서 파트형상의 수정이 요구된다. 파트형상의 수정은 FBCC에서 판단할 수 없으므로 파트 설계자와의 협상에 의해서 가공진행여부를 결정한다. 미가공영역이 수정될 부위에 해당한다.

3.5.6 복구방안의 수립

가공대안에 의한 복구는 피쳐그래프 상에서 사전에 계획된 동일한 부위에 대한 다른 가공계획으로서 공정계획자가 미리 작성해 놓으며 그래프의 탐색으로 쉽게 이용할 수 있다. 탐색은 피쳐그래프에서 현재 가공피쳐의 위치에서 역순으로 아크와 노드를 거슬러 올라가다가 Split-OR노드를 발견했을 때 OR관계에서 현재까지 시도되지 않은 아크를 선정하는 과정이다. 가공대안의 가공피쳐들은 3.5.4의 가공피쳐의 수정의 과정을 거쳐서 각각의 가공피쳐의 구성과 소재형상을 수정한다.

대체공구에 의한 복구는 공구메거진에 존재하는 동종의 공구를 이용하여 가공을 재개하는 것으로서 공구의 스펙에 따라서 자체적으로 공구경로를 생성

하기 위하여 가공정보의 수정 단계를 공구스펙과 가공형상에 따라서 결정한다. 단계별 가공정보의 수정은 3.5.1-3.5.5와 같으며 공구와 수정단계의 결정은 Find_Replaceable_Tool() 알고리즘에 의한다.

PROCEDURE Find_Replaceable_Tool()

INPUT: 문제 발생 가공피쳐 P, 공구집합 T, 공구 개수 N_T;
 OUTPUT: 수정 단계별 공구집합 T_{G-code}, T_{Toolpath}, T_{PF}, T_F, T_P;

PROCEDURE:

```

    i=0;
    TG-code=TToolpath=TPF=TF=TP=∅;
    while (i<NT) {
        i=i+1;
        If (Diameter of Ti=P.dD) TG-code←Ti;
        Else{
            dPF=Test_Pocket_PF(P);
            If (dPF=∅) TToolpath←Ti;
            Else{
                dPF=Modify_Pocket_PF(P);
                If (dPF=∅) TPF (Ti;
                Else{
                    Find feature F that satisfy dP.P.Ar⊂F.Ar;
                    If (F exist) TF←Ti;
                    Else TP←Ti;
                }
            }
        }
    }
    return TG-code, TToolpath, TPF, TF, TP;
    End_Procedure;
    
```

Test_Pocket_PF()는 단위피쳐그래프를 새 공구로 가공할 때 미가공영역의 발생 여부를 테스트하며 최초의 미가공영역 발생시에 종료된다.

PROCEDURE Test_Pocket_PF()

INPUT: Pocket feature P;
 OUTPUT: First uncut area;
 PROCEDURE:
 If (P=∅) return;
 If (P.left≠∅OR P.right≠∅) {
 If (P.left≠∅) dPF=Modify_Pocket_PF(P.left);
 If (dPF≠∅) return dPF;
 If (P.right≠∅) dPF=Modify_Pocket_PF(P.right);
 If (dPF≠∅) return dPF;

```

}
Else {
    PUncut, PCut = PF_Division(P);
    return PUncut;
}
END_PROCEDURE;
    
```

4. 적용예

Fig. 19의 형상에 대하여 1) 가공대안을 포함하는 FBCC 입력정보 준비, 2) on-line 공구경로 생성, 3) 이상상황 발생시의 공구경로 재생성 과정을 보인다.

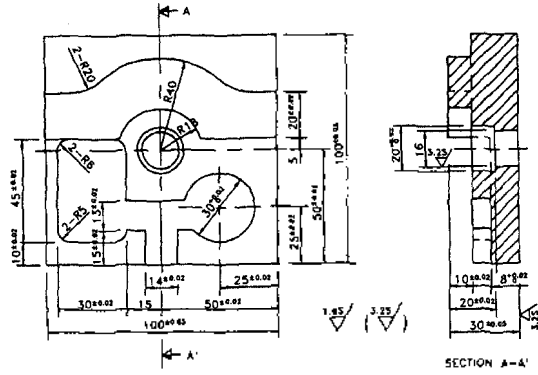
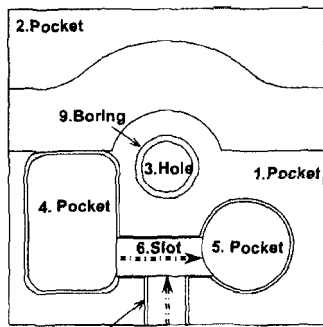


Fig. 19. An example part.

4.1 예제형상 입력정보

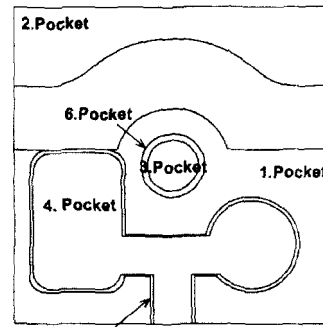
Fig. 19의 예제형상은 두가지 가공방식(Fig. 20(a)-(b))으로 가공될 수 있으며, 각각의 가공방식 및 순

서는 FBCC 코드의 스키마에 의해 STEP physical file 형태로 작성되어 FBCC에 입력된다고 가정한다 (주: FBCC 코드 스키마는 본 논문에서는 취급하지



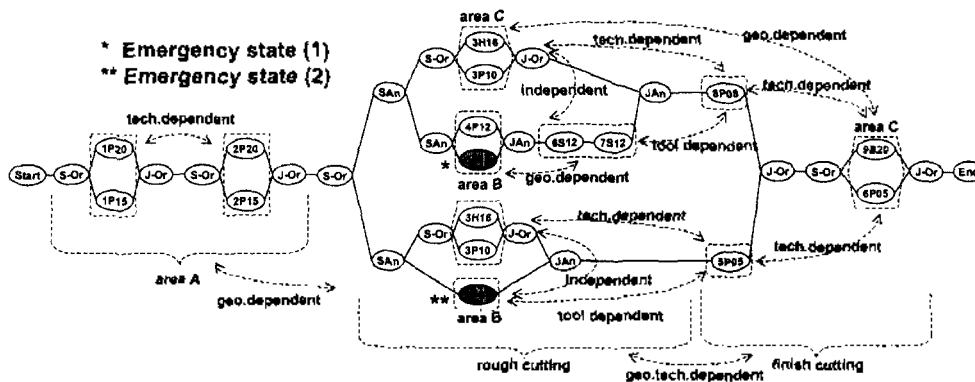
area A : 1.Pocket, 2.Pocket, 7.Slot, 8.Pocket
 area B : 4.Pocket, 5.Pocket, 6.Slot
 area C : 3.Hole, 9.Boring

(a) Process plan 1



area A : 1.Pocket, 2.Pocket
 area B : 4.Pocket, 5.Pocket
 area C : 3.Pocket, 6.Pocket

(b) Process plan 2



(c) Feature graph

Fig. 20. Process plan and feature graph.

않았음).

구체적으로, 예제 형상의 가공피처는 크게 상면의 포켓가공(영역A), 하단의 포켓가공(영역B), 중앙의 구멍가공(영역C)으로 구성되어 있다. 피처그래프의 각 노드의 이름은 {피처번호, 피처타입, 공구직경}으로 표시하였다(예: 1P20={1번피처, Pocket, 공구직경 20}, S=슬롯, H=구멍, B=보링). 다음과 같은 전략이 피처그래프에 반영되었다.

- 1) 영역 A는 직경 20 mm 또는 직경 15 mm 사용하고 Pocket.1이 Pocket.2에 우선한다.
- 2) 영역B는 황삭에 직경 12, 정삭에 직경8의 공구를 사용하거나 황삭에 직경 7, 정삭에 직경5의 공구를 사용한다.
- 3) 직경 12의 황삭에서는 가공영역을 네 구역으로 나누고 직경7에서는 나누지 않는다.
- 4) 영역C는 드릴과 보링 또는 포켓으로 가공하며 드릴형상을 먼저 가공한다.
- 5) 영역 B의 황삭과 영역 C의 드릴링(또는 포켓)은 정삭과 보링(또는 포켓)에 앞서 가공된다.

4.2 On-line 공구경로 생성

입력된 FBCC코드는 STEP-physical 파일의 해석 과정을 통하여 피처그래프의 형태로 변환되며 Intra-scheduling모듈은 피처그래프를 이용하여 가공을 진행한다. Intra-scheduling모듈 내의 Feature selection 모듈(Fig. 2참조)에서는 3.1.2절의 피처선정규칙에 따라서 피처그래프의 start노드에서 시작하여 피처를 선정한다. Pocket 1과 Pocket 2는 Table 3의 Plan(b)와 같은 가공순서 선정이 가능하다. 첫번째 OR관계에서 1P20피처를 선정하며 만일 선정할 순간에 직경 20의 공구가 없을 경우에는 1P15를 선정할 수 있다. 다음 OR관계에서 공구의 상황에 따라서 2P20과 2P15의 두 피처중 한가지를 선정한다. 영역B와 C의 황삭과 영역B의 정삭은 Table 3의 Plan(c)의 경우이다.

가공피처 4P12은 직경 12의 공구로 axis parallel의 경로타입으로 절삭폭 10으로 가공한다. 먼저 가공영역을 나누고 각각의 임시 가공영역을 CPF41~CPF45로 명명한다. CPF41에서 가공면을 공구반경 6 mm로 읍셋하여 공구경로를 만들고 공구통과영역을 계산하면 모든 가공영역을 제거되므로 단위피처가 된다. PF41의 실가공경로는 식 (6)에 의해서 구해진다. PF42는 가공면을 읍셋하여 공구경로를 만들면 Fig. 21과 같이 루프가 발생하고 이를 제거한 경로로 제거부위를 계산하면 미가공영역(Int-PF421,

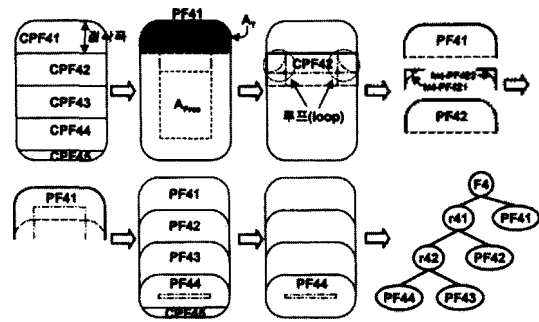
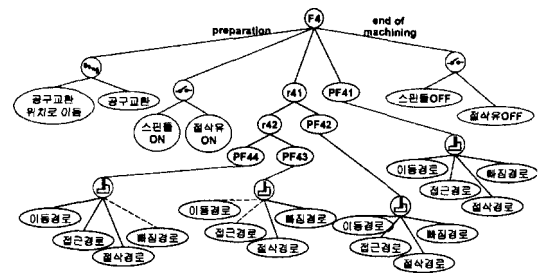


Fig. 21. Primitive feature generation for Pocket 4.

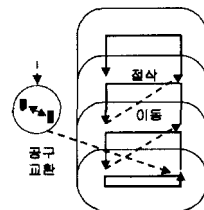
Int-PF422로 표시)이 발생한다. 이들의 가공면과 PF41의 제거형상이 경계를 공유하므로 PF41에 부착되고 가공경로가 수정된다.

마찬가지로 CPF43도 미가공영역을 PF42에 부착하여 PF42의 가공영역을 수정한다. CPF43과 CPF44 영역도 같은 과정으로 단위피처가 된다. CPF45는 전체 영역이 공구간섭을 일으키므로 가공이 되지 않고 이웃한 단위피처 PF44에 부착된다. 생성된 단위피처는 가공순서와 영역의 관계를 표시하는 아진트리로 표시된다.

가공피처세분화 그래프의 각각의 단위피처 노드마다 이동경로, 접근경로, 절삭경로, 빠짐경로를 삼입한다. 절삭경로는 하향절삭이 되도록 방향이 정해지며 이동경로와 접근경로는 이웃한 단위피처에 따라서 생략될 수 있다. 최종적으로 공구교환과 주축회전, 절삭유 ON/OFF 코드를 가공피처의 처음과 끝에



(a) Tool path on binary tree



(b) Tool path

Fig. 22. Tool path generation.

삽입함으로써 공구경로가 완성되며, 이는 G-code로 표현될 수 있다(Fig. 22). (지면관계로 G-code list는 생략하였음).

4.3 이상상황 대처

가공 중 이상상황의 대처방안은 1) 가공대안(OR 노드 탐색)에 의한 대처와 2) 대체공구탐색에 의한 대처의 두 가지가 있다. 이하에서는 피쳐그래프의 5P12의 피쳐를 가공하다가 첫번째 단위피쳐에서 공구에 이상이 발생한 경우: 1) 가공대안에 의해서 복구하는 과정과 2) 이 과정에서 다시 공구에 문제가 발생하여 대체공구에 의해 문제를 해결하는 과정을 보인다.

1) 가공대안에 의한 대처

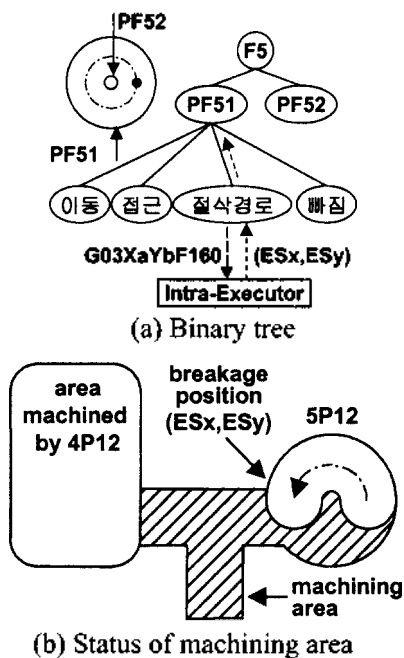


Fig. 23. Intermediate state of machining feature.

5P12피쳐의 가공은 Fig. 24(a)와 같은 두개의 단위 피쳐로 구성된다. 단위피쳐 PF51로부터 공구경로를 계산한 후에 Intra-executor에서 가공을 진행하다가 Fig. 23(b)에 표시된 지점에서 공구에 문제가 발생되었다. Fig. 14의 이상상황 복구과정에 따라서 비상정지와 공구빠짐이 일어나고 단위피쳐그래프를 갱신한다. 갱신과정은 가공이 정지된 위치(ESx, ESy)가 단위피쳐(PF51)에 전달되고 단위피쳐는 가공이 된 영역과 남아있는 영역을 공구파손지점을 경계로 분리하여 단위피쳐 구성을 바꾼다(Fig. 23 및 3.5.2절 참조).

이 상태에서 가공대안의 가공방법을 이용하여 복구한다. 가공대안은 현재 가공피쳐 위치의 상위 OR-관계로부터 발견하며 가공대안 내의 피쳐 중에서 가공이 완료된 3H16, 4P12는 제외되고 가공이 되어야 할 4P07과 5P05가 연결되는 형태의 피쳐그래프로 갱신된다(Fig. 24).

가공피쳐 형상정보 수정은 가공대안 내의 모든 피쳐를 대상으로 수행되며 가공계획1에서 반만 가공된 5P12와 아직 가공이 시작되지 않은 6S12와 7S12의 가공영역(Fig. 20(a)참조)이 합쳐짐으로써 Fig. 23(b)의 빗금친 가공영역이 남고 이것과 경계를 가지는 가공대안 내의 4P07의 가공윤곽만 수정된다. 갱신된 4P07의 포켓형상을 contour parallel방식의 경로타입, 직경7의 공구, 절삭폭 5로 가공할 때 단위피쳐의 생성은 Fig. 25와 같다. 생성된 단위피쳐는 PF1~PF5까지의 단위피쳐와 1개의 잔여형상으로 구성되며 가공개는 PF5부터 시작된다. 여기까지가 단위피쳐 형상정보의 수정과정이며 남은 과정은 공구경로 수정 및 가공실행이다. 공구파손 시 PF512 (Fig. 24 참조)까지 가공된 상태이므로 이후 단위피쳐의 공구경로는 취소되고 단위피쳐 PF512에서 공구빠짐, 가공피쳐 4P07의 PF5의 실행 전에 공구교환위치로 이동, 공구교환, 스피들ON, 절삭유 ON, PF5로의 이동 경로, 접근경로, 절삭경로, 빠짐경로의 순으로 가공이 재개된다.

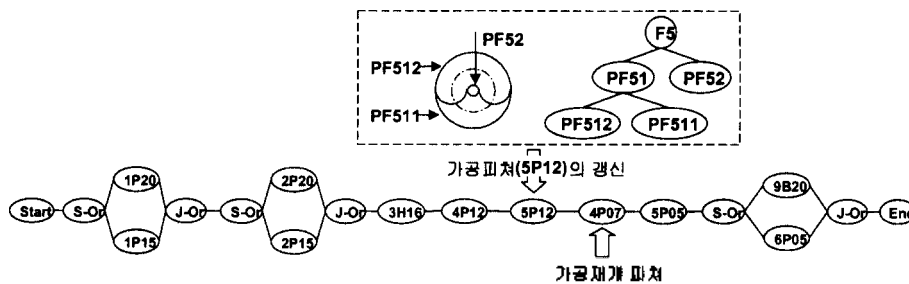


Fig. 24. Feature graph modification.

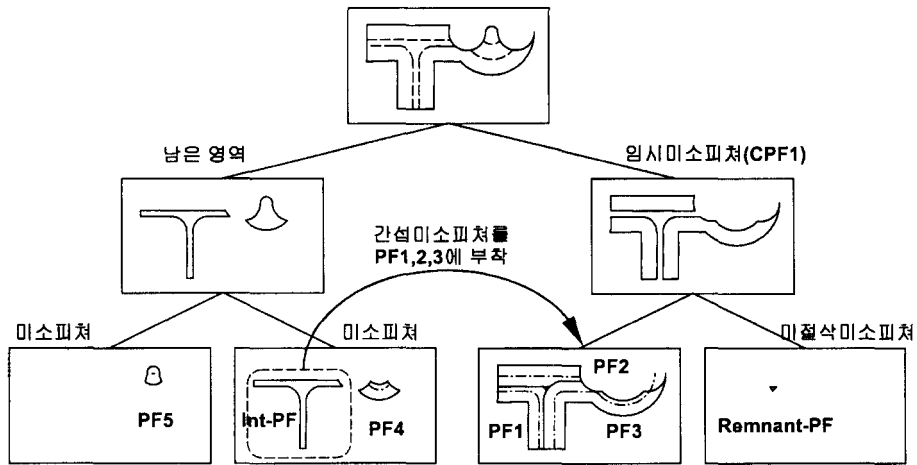


Fig. 25. Modification of primitive feature.

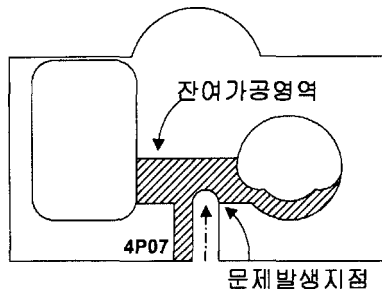


Fig. 26. Remained area of pocket feature (4P07).

2) 대체공구 탐색에 의한 대처

두 번째 시나리오는 가공 대안의 실행이 재개된 후에 PF5, PF4의 가공이 완료되고 PF3의 가공도중에 발생한다고 가정하며 대체공구와 갱신된 가공영역을 보인다. 가공중지 지점의 잔여형상은 Fig. 26과 같으며 가공피쳐 4P07의 단위피쳐그래프가 수정된다. 대체공구로서 직경 20.15의 공구는 간섭 때문에 가공피쳐 4P07의 일부분만 가공하고 다음 가공피쳐

인 5P05에서 나머지 영역을 포함하여 가공할 수 있는 경우이며(가공피쳐의 수정) 직경 8,10,5의 공구는 단위피쳐의 형상을 수정하여 가공할 수 있는 경우이고 이때의 각각의 단위피쳐의 재구성 결과는 Fig. 27과 같다.

5. 결 론

본 논문에서는 1) STEP 호환 피쳐 및 가공계획 정보를 입력으로 취하며 2) STEP-NC의 자율적 NC 코드 생성 및 모니터링 기능과 3) HNC의 협조제어 기능을 보유하는 FBCC(Feature Based CNC Controller)의 구조를 제시하고 온라인 공구경로 생성 및 갱신 알고리즘을 개발하였다. 연구된 알고리즘은 1) 가공피쳐와 피쳐그래프를 이용한 가공계획을 입력으로 받아서, 2) 피쳐단위별로 공구경로를 자동으로 생성하고, 3) 집행상황을 모니터링하여 이상상황에의 대처능력을 갖도록 단위피쳐 단위별 트레이싱 능력

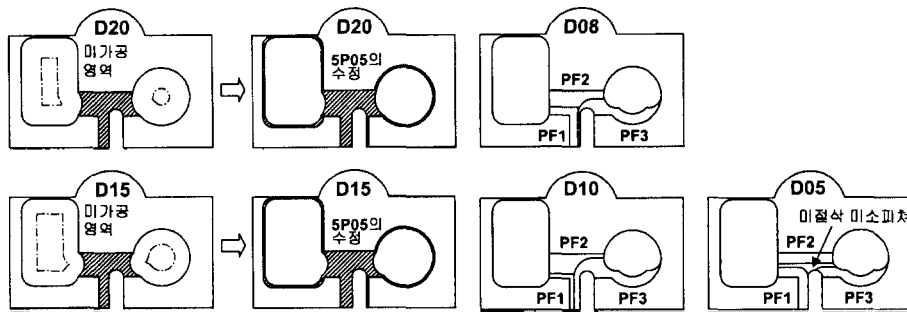


Fig. 27. Machining area with alternative cutting tools.

및 경로갱신 능력을 갖는 자율적 알고리즘이다. 온라인 자율공구경로 생성 및 갱신 알고리즘은 새로운 시도의 차세대 CNC가 갖추고 있어야 할 필수적인 기능으로서, 연구된 알고리즘은 예제를 통하여 유효성 및 효율성이 검증되었다.

참고문헌

1. Weck, M., "NC Programming Interface for Milling of Complex Surfaces", 6th CNC Controller Workshop, SNU, pp. 73-98, August 1995.
2. Brouer, N., Weck, M., "Feature-Oriented Programming Interface of an Autonomous Production Cell", 4th IFAC Workshop on Intelligent Manufacturing Systems, IMS'97, Seoul, pp. 223-228, July 1997.
3. Kruth, J.P., "A Prototype NC controller driven by feature-based part description", Proceedings of the 4th Pacific Conference on Manufacturing, Jakarta, Indonesia, pp. 535-544, Dec. 1994.
4. 조정훈, 서석환, "Feature-based CNC 제어기 연구: CNC용 가공피쳐 정보의 모델링을 중심으로", *Proc. of CAD/CAM conf.*, Seoul, pp. 124-129, 1997.
5. Hong, H.D., Suh, S.H., Cho, J.H., "Feature based control and message specification supporting HMS paradigm", Proc. of 31st CIRP International Seminar on Manufacturing Systems, Berkeley CA, USA, May 26-28, pp. 425-430, 1998.
6. Yamazaki, K., et al., "Autonomous Proficient CNC Controller for High-Performance Machine Tools Based on an Open Architecture Concept", *CIRP Annals*, Vol. 46, No. 1, pp. 275-278, 1997.
7. Liu, J., Yamazaki, K., "Agile Production Realization Based on Autonomously Proficient CNC Controller Infrastructure", *Annals of CIRP*, Vol. 47, No. 1, pp. 414-419, 1998.
8. "IMS Projects", IMS brochure, 1998.
9. "Data model for Computerized Numerical Controllers", ISO/TC184/SC1/WG7, March 1999.
10. Homen de Mello, L.S., Sanderson, A.C., "AND/OR Graph Representation of Assembly Plans", IEEE Trans. Robotics and Automation, Vol. 6, No. 2, pp. 188-199, 1990.
11. Lee, S., Wysk, R.A., Smith, J.S., "Process planning interface for shop floor control architecture for computer integrated manufacturing", Int. J. of Production Research, Vol. 33, No. 9, pp. 2415-2435, 1994.



조 정 훈

1992년 인하대학교 자동화공학과 학사
1994년 포항공과대학교 산업공학과 석사
1994년~현재 포항공과대학교 산업공학과 박사과정
관심분야: STEP-NC, 지능형 CAM 시스템, 제품정보교환(STEP), 개방형 수치제어기



서 석 환

1976년 고려대학교 산업공학과 학사
1978년 한국과학기술원 산업공학과 석사
1986년 Ohio State Univ. 공학박사
1986년~1987년 University of Michigan Center for Research on Integrated Manufacturing 연구원
1987년~현재 포항공과대학교 교수
관심분야: STEP-NC, 지능형 CAM 시스템, 자율가공시스템, 개방형 수치제어기, VMS(Virtual Manufacturing System) (URL: <http://camah.postech.ac.kr>)