한국 CAD/CAM 학회 논문집
제5권 제4호 2000년 12월 pp. 301-311

스위핑과 불리언 연산에 대한 형상 구속조건 관리

김웅주*, 정채봉**, 김재정***

Geometric Constraint Management for Sweeping and Boolean Operations

Kim, W. J.*, Jeong, C. B.** and Kim, J. J.***

ABSTRACT

For effective part modifications which is necessary in the design process frequently, variational geometric modeling with constraint management being used in a wide. Most variational geometric modeling methods, however, manage just the constraints about sketch elements used for generation of primitives. Thus, not only constraint propagation but also re-build of various modeling operations stored in the modeling history is necessary for part geometry modifications. Especially, re-build of high-cost Boolean operations is apt to deteriorate overall modeling efficiency abruptly. Therefore, in this paper we proposed an algorithm that can handle all geometric entities of the part directly. For this purpose, we introduced eight type geometric constraints to the various geometric calculations about all geometric entities in sweepings and Boolean operations as well as the existing constraints of the sketch elements. The algorithm has a merit of rapid part geometric modifications through only constraint propagation without re-build of modeling operations which are necessary in the existing variational geometric modeling method.

Key words : Variational geometric modeling, Constraint propagation, Sweeping, Boolean operations

1. 서 론

설계과정에서 부품의 형상 수정 작업은 빈번히 요구되는 작업이며, 따라서 부품 형상의 효율적인 수정은 전체 설계과정의 효율을 향상시키는 데 있어 매우 중요한 역할을 한다. 최근에는, 부품의 형상 수정을 효율적으로 수행하기 위해 가변형상(variational geometry)기법이 널리 사용되고 있다. 가변형상 기법은 치수와 형상 요소들 간의 기하학적 상관관계를 형상 구속조건(geometric constraints, 이하 구속조건이라 함)으로 표현 및 관리하여 그 형상 요소들의 변경된 치수나 위치를 신속히 재산해 내는 방법이다[1]. 최근, 많은 상용 CAD 시스템 개발자들은 그들의 상용 제품인 솔리드 모델링, 파라메트릭 모델링, 또는 특징형상 모델링 시스템 등에 가변형상 기법을 사용하고 있다. 예를 들면, 산업체에서 주로 많이 사용되는 CATIA™, Pro/ENGINE ER™, SoildWorks™ 등과 같은 상용 시스템들이 여기에 해당된다. 본 논문에서는 이러한 상용 CAD시스템들을 구분 없이 '가변형상 모델링 시스템' 또는 간략히 '시스템'이라 부르기로 한다. 가변형상 모델링 시스템을 이용하여 사용자가 원하는 '부품(part)'을 생성하는 방법은 그 종류에 따라 약간의 차이는 있지만, 여러 개의 3차원 '기본입체(primitive or feature)'를 생성하고, 그 기본입체에 시스템이 지원하는 다양한 '모델링 연산(modeling operations)'을 적용시킴으로써, 사용자가 원하는 복잡한 형태의 부품으로 변형시켜 나간다는 점에서 개념적으로 매우 유사하다. 대부분의 시스템들은 3D 공간 상의 임의의 평면(이하 기준평면; datum plane) 위에 필요한 형상 요소들을 정의하고, 그 요소들을 기준평면의 법선방향으로 '스위핑(sweeping or ex-truding)'하여 기본입체를 생성하는 방식을 지원한다. 또한 시스템이 지원하는 모델링 연산들을 살

*정회원, (주)한국타이어 중앙 연구소 REP개발팀
**학생회원, 한양대학교 기계설계학과
***종신회원, 한양대학교 기계공학부
- 논문투고일: 1999. 12. 7
- 심사완료일: 2000. 7. 28

펴보면, 기본입체의 특정 모서리나 면을 이용하여 Fillet, Draft, Shell 등의 작업을 수행하거나, 또 다른 기본입체와의 불리언 연산을 수행하여 사용자가 원하는 복잡한 부품의 형상을 표현할 수 있게 한다[2,3,4]. 본 논문에서는 기준평면상의 개별적인 형상 요소들을 '스케치 요소(sketch element)'라 하며, 하나의 기준 평면 상에 정의된 모든 스케치 요소들을 '스케치 세트(sketch set)'라고 부르기로 한다.

일반적으로 시스템은 스케치 요소들 간의 기하학적 관계들을 수학적으로 표현하여 관리하는데, 각각의 수식 하나를 '구속조건(constraint)'이라 부르며, 스케치 요소들 간의 모든 구속조건들을 한데 묶어 그 스케치 세트의 '구속조건 세트(constraint set)'라 부르기로 한다.

일반적으로 사용자는 하나의 부품을 생성하기 위해, 여러 개의 기본입체를 생성하고 다양한 종류의 모델링 연산을 수행하게 된다. 이 때, 시스템은 사용자가 수행한 모든 종류의 작업 및 그 속성들을 수행된 작업 순서에 따라 '모델링 히스토리(modelling history)'란 곳에 부품의 형상 정보와는 별개로 따로 저장하여 관리한다. 그러므로 모델링 히스토리 내에는 다수의 기준평면, 스케치 세트, 기본입체, 그리고 각종 모델링 연산이 그들의 생성 및 적용 순서에 따라 계층 구조로 저장된다. 각 기본입체 생성에 사용된 스케치 세트는 그 기본입체의 'parents'로 관리되며, 같은 맥락에서 스케치 세트는 해당 기준평면의 'child'로 관리되는 종속관계를 갖게 된다. 특히, 스케치 세트는 관련된 구속조건 세트를 포함하고 있으며, 사용자는 언제든지 특정 구속조건의 변수를 조정하여 스케치 세트를 변경할 수 있다.

대부분의 가변형상 모델링 시스템들은 특정 기본입체의 스케치 세트에 대한 구속조건을 변경하여 부품의 형상을 수정하는 방식을 지원한다. 부품을 수정하기 위해 사용자가 특정 구속조건의 변수를 조정하면, 시스템은 모델링 히스토리를 이용하여 다음과 같은 두 단계의 작업을 통해 부품의 형상을 수정한다. 첫 단계로, 변경된 구속조건이 포함된 구속조건 세트를 다시 풀이하여 해당 스케치 세트를 수정한다. 두 번째 단계로, 수정된 스케치 세트를 이용하여 모델링 히스토리 내에 저장된 모델링 연산들을 그 순서에 따라 재수행(re-build) 하여 부품의 형상을 수정한다[2,3,4]. 첫 단계와 관련하여 효율적인 구속조건 해결자(constraint solver)를 개발하기 위한 연구들[7-13]이 진행되고 있으며, 특히 두 번째 단계와 관련해서는 특징형상 모델의 수정을 효율적으로 돕기 위해 모델의 타폴로지 요소들에 일관된 꼬리표를 붙여 관리하는 topology naming에 관한 연구들[14,15]이 제안되어, 파라미터의 변경이 부품의 최종 타폴로지의 변경을 초래하는 경우에 효과적으로 대처할 수 있는 방법을 제시하고 있다. 그러나, 이러한 방법들은 모두 모델링 히스토리 내의 여러 모델링 연산들을 재수행 해야 하는 번거러움이 있다. 이는 특징 형상 라이브러리를 주로 사용하는 대개의 경우(라이브러리에 저장된 특징 형상의 다양한 variant를 만들어 사용하지만 그 타폴로지가 변하지 않는 경우)에는 계산상의 낭비가 매우 심하다. 특히, 상당히 많은 계산량이 요구되는 고비용의 불리언 연산의 재수행은 모델링 효율을 급격히 저하시킨다[6]. 이러한 문제는 모델링 연산 시 시스템이 단지 형상 요소들 만을 처리할 뿐, 해당 연산을 거친 결과 모델의 형상을 직접 관리하는 구속조건은 생성하지 못하기 때문이다.

따라서 본 논문에서는 모델링 연산 시 형상 요소들 사이의 기하학적 상관관계를 적절한 구속조건으로 표현하고, 이들을 기존의 스케치 세트에 대한 구속조건 세트들과 병합함으로써 부품의 모든 형상 요소들에 대한 확장된 구속조건 세트를 구축하는 방법을 제안 하였다. 이를 통해 부품의 타폴로지가 변하지 않는 파라미터 수정에 대하여 단지 구속조건의 파급만으로 수정된 모델을 효과적으로 생성할 수 있는 시스템을 개발하였다. 본 시스템에서는, 스케치 요소로는 점, 직선, 원호를 사용하였고, 기본입체를 생성하는 기초 작업인 이동 스위핑과 부품 수정 시 모델링 효율을 가장 급격히 저하시키는 불리언 연산만을 고려하였다. 본 논문의 나머지 부분은 다음과 같이 구성되어 있다. 다음 절은 구속조건 관리 방법을 설명하는 절이며, 본 논문의 핵심 사항인 이동 스위핑과 불리언 연산을 통해 모델의 구속조건이 병합되는 과정을 설명하는 절이 뒤따른다. 그 다음 절에서는 구현된 시스템의 적용 예를 통한 검증이 제시되며, 마지막 절은 본 논문의 결론이다.

2. 구속조건 그래프 관리

성공적인 가변형상 모델링 시스템을 구현하기 위해서는 형상 요소들 간의 구속조건을 추가, 삭제, 편집, 풀이 등의 효과적인 구속조건 관리 방법의 개발이 선결 과제이다. 다행스럽게도 이미 많은 연구자들에 의해 다양한 구속조건 관리 방법이 제안되었다. 일반적으로 구속조건을 관리하는 기존 연구들은 numerical approach[7-8], symbolic approach[9-11],

construction approach[12,13]로 구분할 수 있다. 이 중에서, symbolic approach는 구속조건을 대수식으로 표현하고 일반적인 symbolic method를 이용하여 구속조건을 관리하는 방법이다. 특히, Kondo[9,10]는 대수식으로 표현된 구속조건 세트를 각 스케치 요소 간의 종속관계에 기초하여 방향그래프로 관리하였다. 이 방법은 새롭게 생성되는 형상 요소가 기존 형상 요소와의 기하학적 상관관계를 가질 때 그들 간의 구속관계를 관리하기에 적합한 방식이다. 따라서 본 논문에서는 이동 스위핑 및 불리언 연산 과정에서 새롭게 생성되는 형상 요소와 기존 요소들 간의 기하학적 상관관계를 적절한 구속조건으로 표현하기에 적합하도록 Kondo의 방법을 확장하였다.

스케치 세트에 대한 구속조건 세트를 방향그래프로 관리한 Kondo의 방법에서는 하나의 스케치 요소가 생성될 때마다 이에 상응하는 하나의 구속조건이 생성된다. 이는 새로운 스케치 요소가 생성될 때, 기존 스케치 요소들과의 기하학적 상관관계인 구속조건이 사용자에 의해 주어짐을 전제로 한다. 이를 통해 새로운 스케치 요소는 일부 기존 스케치 요소들의 '종속요소(subordination element)'가 되며, 또한 이 요소들은 이를 이용하여 생성될 새로운 스케치 요소의 '기준요소(reference elements)'가 될 수 있다. 구속조건이 갖는 이러한 기준, 종속 관계는 방향그래프를 이용하여 Fig. 1과 같이 나타낼 수 있다. 그래프에서 각 절점(node)에는 형상 요소의 포인터가 저장되며, 간선(arc)에는 기준, 종속 관계에 있는 절점들과 구속조건의 속성(종류, 변수 등)이 저장된다.

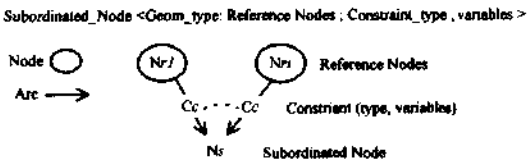

Fig. 1. Constraint graph.

Table 1. The modeling processes of Fig. 2.

Modeling processes
1. Define a datum plane with equation
2. Create a point P1 coordinate values on a datum plane
3. Create a point which has relative x-distance a from P1
4. Create a connecting line L1 through and P2
5. Create a line L2 which is parallel to L1 and has relative upper distance c from L1
6. Vreate a circle C1 whose center is P2 and radius is b
7. Create a intersecting point P2 with line L2 and circle C1
8. Create a connecting line L4 through P3 and P1
9. Create a connecting line L3 through P3 and P2

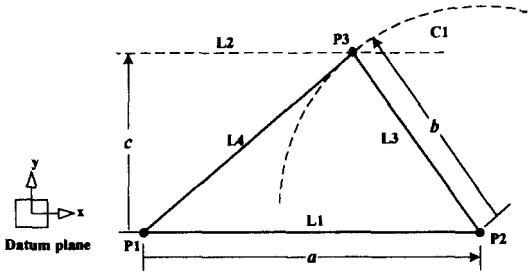

Fig. 2. A sketch set.

각 형상 요소들은 특정 기준평면 위에 정의되므로 최초 기준 평면을 정의하는 구속조건이 평면의 방정식 형태로 부여된다. 그런 다음 기준 평면 상에서 스케치 요소들이 정의되면, 각 요소에 대응되는 절점이 생성되며 해당 구속조건에 의해 간선으로 연결된다. 예를 들어 사용자가 테이블 1과 같은 작업 과정(work processes)을 통해 Fig. 2와 같은 스케치 세트를 생성한 경우, 형상 요소들 간의 구속조건은 테이블 2와 같고 그래프로 표현하면 Fig. 3과 같다. 그래프 생성에 대한 좀더 자세한 내용은 참고문헌[9,10]을 살펴보기 바란다.

구속조건 그래프는 다음과 같은 세 가지 특징이 있다. 첫째, 이 그래프는 비순환적(acyclic)이다. 즉, 한 절점에서 출발하는 간선은 어떠한 경로를 통해서도 자기 자신으로 되돌아오지 못한다. 만약 구속조건 그래프가 순환적이라면, 자기 자신을 구속하는 과잉 구속조건 상태임을 의미한다. 둘째, 한 절점을 향하는 모든 간선은 같은 종류의 속성을 지닌다. 만일 그렇지 않다면, 이 절점에는 두 가지 이상의 구속조건이 부여되는 결과이며, 이 역시 과잉 구속조건 상태임을 의미한다. 세째, 최초 절점(일반적으로 기준평면)을 제외한 나머지 절점들은 자신을 지시하는 간선이 반드시 있다. 만일 그렇지 않다면, 그 절점은 다른 절점에 의해 종속되지 않아 구속조건이 부족함을 의미한다. 위와 같은 세 가지 특성을 이용하여 구속조

Table 2. The corresponding constraint set of Table 1.

Constraints representations
1. PLN1<PLANE; NULL; equation, 0, 0, 1, 0>
2. P1<POINT: PLN1; coordinate; x1, y1>
3. P2<POINT: P1; distance, a, 0>
4. L1<LINE: P1, P2; line-connection>
5. L2<LINE: L1, u-paralle, c>
6. C1<CIRCLE: P2; radius-circle, b>
7. P3<POINT: L2, C1; intersection>
8. L4<LINE: P3, P2; line-connection>
9. L3<LINE: P3, P1; line-connection>

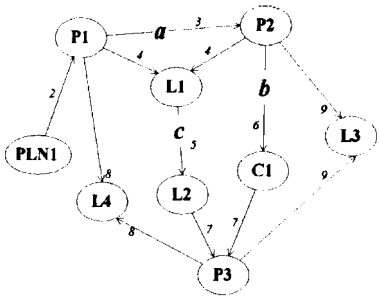

Fig. 3. The constrint graph of a sketch set in Fig. 2.

건의 과잉, 부족 여부를 언제든지 점검할 수 있다[9].

스케치 세트의 수정은 특정 구속조건의 변수를 조작하여 수행된다. 즉, 사용자가 특정 구속조건의 변수 값을 변경하면, 시스템은 변경된 변수를 갖는 간선을 찾아 그 구속조건(대수식)을 변경된 변수를 이용하여 다시 풀이하여, 그 간선이 지시하는 절점의 위치를 변경한다. 이 과정을 구속조건의 '갱신(update)'이라 하는데, 변경된 절점에 의해 종속되는 나머지 절점들(수정된 절점의 간선을 따라 도달 가능한 모든 절점)도 같은 방법으로 갱신하여 최종적인 형상 변경을 완료한다. 이러한 일련의 간선(구속조건) 갱신 과정을 '구속조건 파급(propagation)'이라 한다. 예를 들어, Fig. 2의 변수 b가 b'으로 수정되면 시스템은 구속조건 그래프를 탐색하여 Fig. 4(a)와 같이 6번 구속조건을 찾아 C1절점을 수정하고, 절점 C1

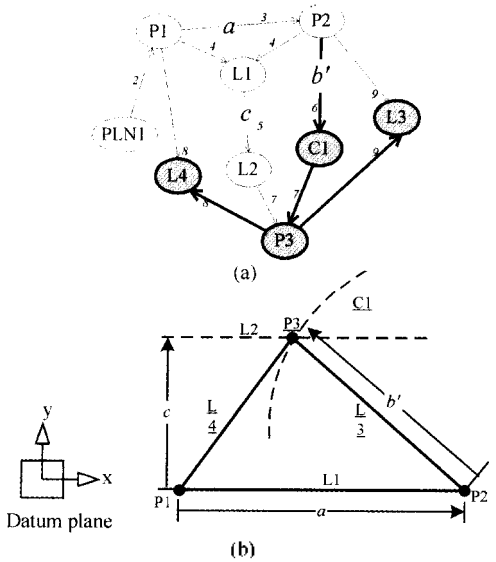

Fig. 4. A modification; (a) constrint propagation, (b) modified sketch set in Fig. 2.

으로부터 구속되는 절점들인(C1 절점의 간선을 따라 도달 가능한 모든 절점) P3, L3, L4를 차례로 수정하는 구속조건 파급과정을 절차를 거쳐 최종적으로 Fig. 4(b)와 같이 형상을 수정한다.

3. 모델링 연산에 대한 구속조건 관리

본 절에서는 앞서 설명한 스케치 세트에 대한 구속조건 관리 외에 본 논문에서 제안한 스위핑 및 불리언 연산에 대한 구속조건 관리 방법을 살펴본다. 각 모델링 연산 시, 해당 알고리즘에 의해 새롭게 생성되는 형상 요소와 기존 요소들과의 기하학적 관계를 이용하여 기존 요소들에 대한 구속조건 그래프를 순차적으로 확장하는 방법을 설명한다.

3.1 이동 스위핑에서 구속조건 그래프의 확장

이동 스위핑은 기준평면상의 스케치 세트를 그 법선 방향으로 일정 거리 만큼 이동 시킨 궤적의 부피를 생성하는 작업이다. 이때, 스케치 세트는 이미 구속조건 그래프를 가지고 있으므로 스위핑을 통해 새롭게 생성되는 형상 요소와 기존 형상 요소(datume plane 또는 스케치 세트) 간의 기하학적 관계를 적절한 구속조건으로 표현하여 스케치 세트의 구속조건 그래프를 기본입체의 구속조건 그래프로 확장하여야 한다.

본 논문에서 이동 스위핑은 다음과 같은 과정을 통해 생성된다. 먼저 기준평면과 d 만큼 떨어진 곳에 새로운 평면(이하 한계 평면이라 함) 하나를 생성하고(A유형), 스케치 요소인 점, 직선, 원호와 한계 평면을 이용하여 투영된 점들과 룰드곡면을 생성한다(B유형, C유형). 이어서 각 룰드곡면들과 한계 평면과의 교차 계산에 의해 기본입체의 경계 곡선 일부를 생성한다(D유형). 그런 다음 기준평면상의 점과 한계 평면 상에 투영된 점을 연결하여 나머지 경계 곡선을 생성한다(E유형). 그러므로 이들 각 과정에 대해 다음과 같은 종류의 구속조건을 정의할 수 있다.

A: PLN*i*<PLANE: PLN*j*; parallel, *d*>
B: PT*i*<POINT: PLN*j*, PT*k*; normal-projection>
C: SUR*i*<SURFACE: PLN*j*, CRV*k*; ruled-surface, d>
D: CRV*i*<CURVE: PLN*j*, SUR*k*; intersection>
E: LN*i*<LINE: PT*j*, PT*k*; line-connection>

각 구속조건의 기준 요소들은 **A**유형에서는 기준평면, **B**유형에서는 한계평면과 스케치 세트에 포함되는 점들, **C**유형에서는 한계평면과 스케치 세트에 포함되는 곡선들, **D**유형에서는 한계평면과 룰드곡면

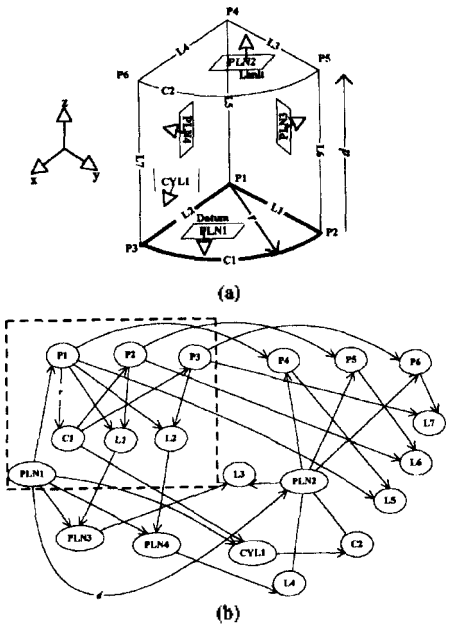

Fig. 5. An extruding; (a) a swept quarter cylinder, (b) the corresponding constraint graph.

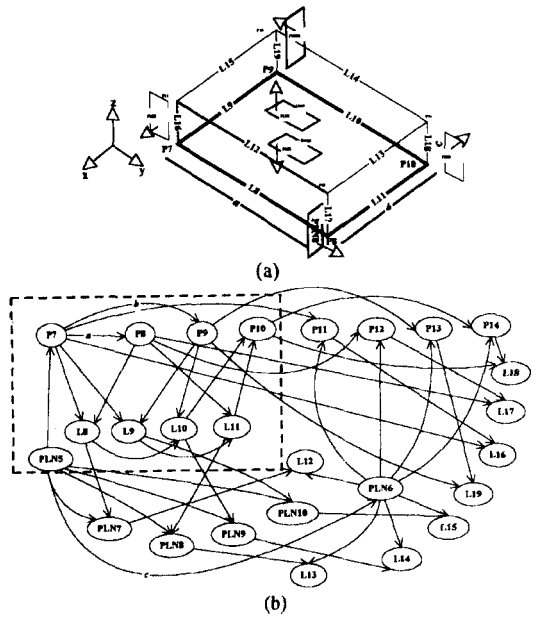

Fig. 6. An extruding; (a) a swept cuboid, (b) the corresponding constraint graph.

들, E유형에서는 스케치 세트에 포함되는 점들과 한계평면 상의 점들이다. D유형을 제외한 모든 구속조건의 기준요소는 모두 스케치 요소들을 직접 포함하고 있다. 그러나 D유형의 기준요소들이 스케치 세트에 의해 종속되므로, 결국 새롭게 생성되는 모든 형상 요소들은 기존 요소들에 의해 구속된다. 따라서 스위핑된 기본입체의 구속조건 그래프는 기준 평면 상의 스케치 세트의 구속조건 세트로부터 확장될 수 있다.

이 과정을 Fig. 5의 (a)와 같이 xy평면(PLN1; datum) 위에서 부채꼴 모양의 스케치 세트를 생성한 후, z축 방향으로 d만큼 스위핑한 사분 원기둥의 경우에 적용시켜 보자. 이 때, PLN1(datum)에서 생성된 부채꼴 모양의 스케치 세트의 구속조건 그래프는 점선으로 둘러싸인 사각형 내부와 같다고 가정하자. 우선, PLN1을 기준요소로 하고 구속조건 parallel(*d*)를 아크로 하는 한계 평면 PLN2가 생성되고, 이어서 한계 평면과 P1, P2, P3를 기준 요소로 하는 투영점(normal-projection point) P4, P5, P6가 차례로 생성된다. 그런 다음 스케치 세트의 L1, L2, C1과 한계평면을 기준요소로 하는 룰드곡면(ruled-surface) PLN3, PLN4, CYL1이 생성된다. 또한 룰드곡면들과 한계곡면을 기준 요소로 하는 교차곡선(intersection curve) L3, L4, C2 생성되며, 마지막으로 기준 평면 상의 스케치 요소 P1, P2, P3들과 투영점 P4, P5, P6들을 연결하는 직선 L5, L6, L7이 생성된다. 이 때, 기준 평면상의 스케치 세트에 대한 구속조건 그래프는 새롭게 생성되는 요소들을 새로운 절점으로 추가하고, 그들의 각 기준요소 절점으로부터 해당 구속조건을 속성으로 갖는 아크로 연결하여, 스위핑된 기본입체의 형상 요소를 모두 포함하는 새로운 구속조건 그래프로 확장된다(Fig. 5의 (b)참조).

스위핑에 의해 생성된 기본입체는 사용된 스케치 세트와 기준평면이 다를 뿐, 스위핑 과정을 통해 A~E까지의 일정한 유형의 구속조건에 의해 생성되므로, 확장된 구속조건 그래프는 그 스케치 세트를 어떻게 구성 하였는지에 따라 차이가 일을 뿐, 그 유형은 일정하다. Fig. 6은 또 다른 기본입체를 스위핑으로 생성한 경우의 확장된 구속조건 그래프를 보여주고 있는데, 두 구속조건 그래프를 비교하면 각 스케치 세트를 구성한 부분(점선 사각형으로 표시된 부분)만 차이가 있을 뿐, 그 형태가 Fig. 5의 경우에 비해 매우 흡사함을 알 수 있다.

3.2 불리언 연산에서 구속조건 그래프의 병합

불리언 연산은 두 입체 1에 부피 단위의 집합연산을 수행하는 작업이며, 경계도출(boundary evaluation)과 set membership classification을 통한 결합(combination)

단계로 나누어 진다[14]. 이 작업은 두 입체를 이용하여 새로운 입체를 생성하는 작업이므로, 스위핑과는 다르게 두 입체가 가지는 각각의 확장된 구속조건 세트를 병합하여 새로운 입체에 대한 병합된 구속조건 세트를 구성하여야 한다. 하지만 그 방법에 있어서는 스위핑과 마찬가지로 불리언 연산 과정에서 새롭게 생성되는 형상 요소들과 기존 두 입체의 형상 요소들과의 기하학적 관계를 적절한 구속조건으로 표현하여 두 구속조건 그래프를 하나의 구속조건 그래프로 병합한다.

불리언 연산의 첫 번째 단계인 경계 도출(boundary evaluation)에서는 두 입체의 교차 영역을 계산하여 새로운 형상 요소들을 생성한다[16]. 본 논문에서는 경계 도출 단계에서 새롭게 생성되는 형상 요소들을 그들이 생성되는 방법에 따라 크게 세가지 유형으로 분류하였다.

• 경계 교차 곡선(Boundary Intersecting Curves) : 두 입체의 경계 곡면 간의 교차 곡선 또는 경계 곡면과 경계 곡선의 교차 곡선(경계곡면이 서로 중첩(overlap)된 경우) (F유형).

• 경계 교차 점(Boundary Intersecting Points): 경계 교차 곡선 각 입체의 기존 경계 곡선과의 교차점 2 (G유형).

• 추가 분할 곡선(Additional Divided Curves): 경계 교차 점과 입체의 꼭지점에 의해 기존 경계곡선이 여러 부분으로 분할된 곡선(H유형).

경계 도출 단계에서 새롭게 생성되는 위와 같은 세 가지 유형의 형상 요소들에 대한 구속조건은 각각 다음과 같이 정의할 수 있다.

F: BIC*i*<CURVE: SUR*j*, SUR*k*; intersection, parameter directions>or<CURVE: SUR*j*, CUR*k*; intersection, parameter directions>

G: BIP*i*<POINT: CRV*j*, CRV*k*; intersection, parameter directions>

H: ADC*i*<CURVE: CRV*j*, BIP*k*, PTI; curve-dividing>

형상 요소간의 교차계산에 의해 생성되는 F유형과 G유형은 교차 구속조건을 갖는다. F유형의 기본요소는 서로 다른 두 입체의 경계 곡면이거나 그들이 서로 중첩되어 있는 경우에는 경계 곡면과 경계 곡선이 된다. G유형의 기준요소는 경계 교차 곡선 또는 기존 입체의 경계 곡선이다. 이 때, 경계 곡면이 중첩되어 생성된 경계 교차 곡선은 배제한다. 왜냐하면, 경계 교차 곡선 자체가 기존 경계 곡선이거나 그 일부 이므로 추가 분할 곡선 생성 단계에 아무런 영향을 미치지 못하기 때문이다(Fig. 7설명 참조). H유형은 입체의 경계 곡선이 경계 교차 점과 자신의 꼭지점에 의해 여러 부분으로 분할된 곡선이므로 구속조건 종류는 곡선 분할(curve-dividing)이다. 세 가지 경우 모두 스위핑 시와 마찬가지로 새롭게 생성되는 형상 요소는 두 입체의 기존 형상 요소들(기준요소)에 의해 종속된다. 이 종속관계에 따라 새로운 입체 대한 구속조건 그래프는 기존 두 입체의 구속조건 그래프로부터 자연스럽게 병합 될 수 있다.

이 과정을 Fig. 7과 같이 구속조건을 갖는 육면체 A와 사분 원기둥 B 간의 불리언 차집합의 경우에 적용시켜 보기로 하자. 먼저, 불리언 연산의 대상이 되는 두 입체 A, B는 이미 스위핑 과정을 통해 Fig. 5와 6과 같이 확장된 구속조건 세트를 가지고 있다. 두 입체의 경계 도출 과정에서, Fig. 7의 (a)와 같이 경계 교차 곡선이 계산된다. 이때 BIC1~BIC6은 입체 A, B의 경계 곡면 간의 교차 계산에 의해 생성된 것이며, BIC2, BIC4, BIC6와 BIC7은 경계 곡면의 중첩으로 인해 입체의 경계 곡면(PLN10, PLN4)과 경계 곡선(L7, L9, L15, L19)의 교차 계산에 의해 생성된 것이다. BIC4는 경계곡선의 전부가 경계 교차 곡선이 된 경우이고, 나머지는 곡선의 일부가 교차하는 경우이다. 모든 경계 교차 곡선이 생성될 때, 두 입체의 구속조건 그래프는 하나의 절점과 두 개의 아크(예를 들면, BIC1절점은 CYL1과 PLN6절점으로부터 연결되는 두 아크에 의해)가 추가되어 병합된다. 경계 교차 곡선들이 모두 계산된 후에는 이들 경계 교차 곡선과 기존 경계 곡선 과의 교차 검사에 의해 경계 교차 점 BIP1~BIP4가 생성된다. 이때, Fig. 7의 (b)와 같이 중첩된 경계 곡면에 의해 생성된 경계 교차 곡선(BIC2, BIC4, BIC6, BIC7)은 이 교차 계산에서 배제된다. 이들이 생성하는 경계 교차 점은 다른 경계 교차 곡선들이 생성하는 교차 점과 중복되거나, 아니면 기존 입체의 꼭지점이기 때문이다. 마지막으로, 입체의 경계 곡선 중 경계 교차 점이 위치하는 곡선을 그 경계 교차 점으로 분할하여 Fig. 7의 (c)와 같이 추가 분할 곡선을(ADC1~ADC6) 생성한다. 이러한 과정을 통해 두 입체의 구속조건 그래프는 Fig. 7의 (d)와 같이 병합된다.

불리언 연산의 두 번째 단계인combination에서는

[1] 불리언 연산의 대상은 하나의 기본입체(primitives), 또는 이미 여러 모델링 연산이 수행된 복잡한 모양의 입체일 수도 있다. 그러므로 본 논문에서 불리언 연산의 대상은 구속조건을 가지는 입체이다.

[2] 두 입체가 한 꼭지점(vertex)에서 만나는 경우, 경계 곡면과 경계 곡선의 교차에 의해 생성될 수 있으나, 이는 regularized Boolean operation에 의해 배제된다[16].

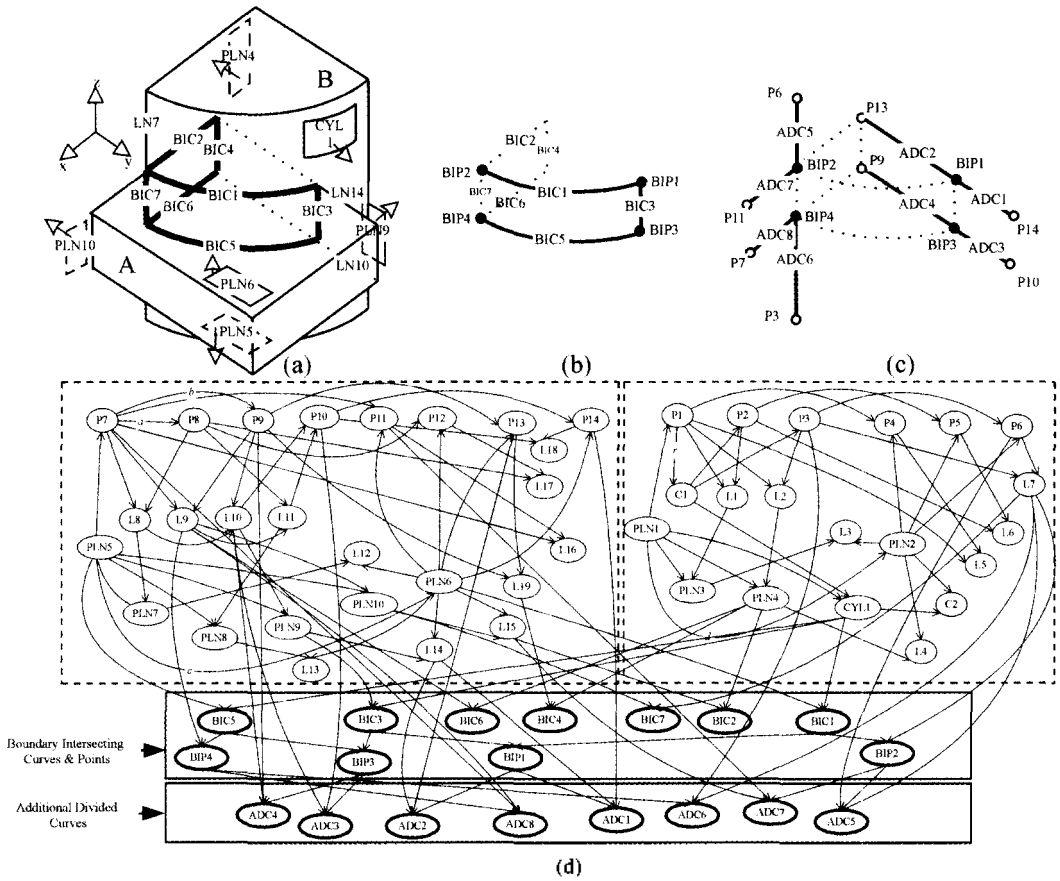

Fig. 7. Boundary evaluation process in a Boolean set operation; (a) Boundary intersecting curves; (b) Boundary intersecting points; (c) Additional divided curves; (d) A merged constraint graph in the middle of Boolean set operation.

형상 요소들에 대한 set membership classification 결과를 이용하여 새로운 입체의 형상을 구성하는데 필요한 요소들 만을 선택하여 취합한다. 선택 방법은 불리언 연산자의 종류에 따라 이미 테이블화 되어 있다[16]. 따라서 병합된 그래프 내에서 연산자의 종류에 따라 불필요한 절점들을 삭제하는 과정이 필요하다. 삭제 과정은 두 단계로 진행된다. 첫번째 단계에서는, 병합된 그래프 내에서 새로운 입체에 포함되는 절점들은 '삭제 불가 절점'으로, 그 나머지는 '삭제 후보 절점'으로 정의한다. 두 번째 단계에서는, 삭제 후보 절점들 가운데 삭제 불가 절점의 기준 요소가 아닌(아크의 방향 추적을 통해 삭제 불가 절점을 만나지 못하는) 절점을 삭제한다.

이 과정 역시 Fig. 7을 예로 들어 살펴보기로 하자. Fig. 7의 (d)와 같이 병합된 그래프의 모든 절점의 형상 요소들이 두 입체 A, B간의 불리언 차집합 연산의 결과 형상 내에 존재 하는 지를 검사 하면, Fig. 8에서보는 바와 같이 P13, C1, PLN2 등과 같이 굵은 실선으로 표시된 절점들은 삭제 후보 절점이 됨을 알 수 있다. 하지만 이들 모두를 삭제 할 수는 없다. 만일 삭제 후보 절점인 C1을 그래프에서 제거한다면, 삭제 불가 절점인 CYN1은 기준 요소를 잃어 버리게 된다. 따라서 삭제 후보 절점들 중 삭제 불가 절점의 기준요소가 될 수 없는 절점들만 삭제하여야 한다. 그래프 탐색을 통해 회색으로 채워진 L1, PLN3과 같이 삭제 불가 점절에 닿을 수 없는 삭제 후보 절점들만 삭제된다(Fig. 8참조). 위와 같은 방법으로 두 입체 A, B에 대한 불리언 합집합 및 교집합의 경우에 병합된 그래프를 Fig. 9와 Fig. 10에서 확인 할 수 있다. 이들은 불리언 연산의 작업 과정의 특성상 경계 도출 과정이 같으므로 두 번째 단계인 삭제 후보 절점을 찾아 그들 중 일부를 삭제한 결과 만이 다를 뿐이다. 이러한 방식을 통해, 여러 번의 불리언 연산이 수행 되어 비교적 복잡해진

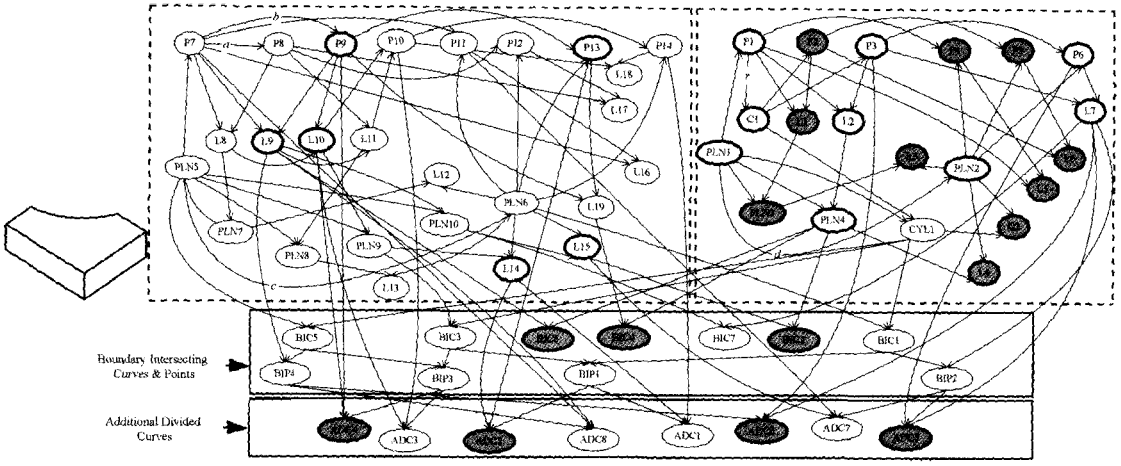

Fig. 8. The node Elimination in A−*B case (white bold: elimination condidate node, shaded bold: elimination node).

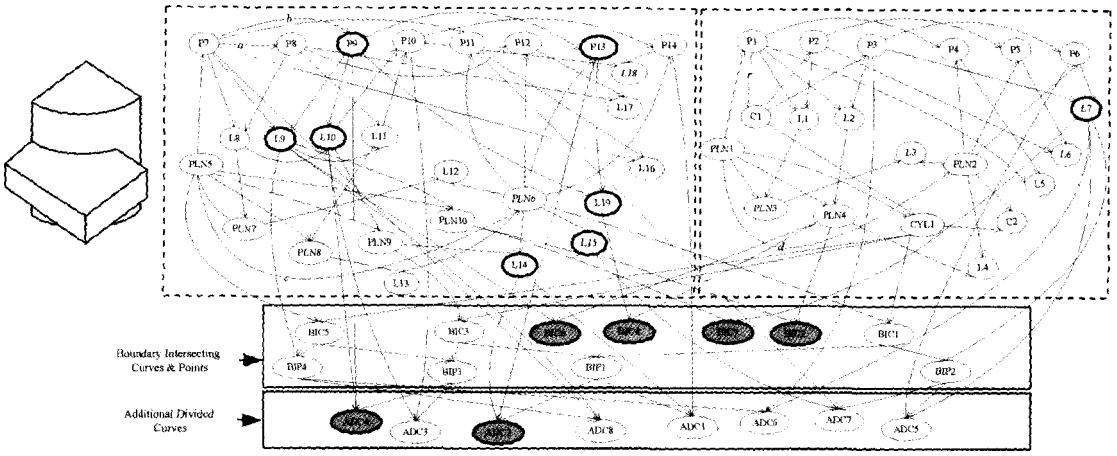

Fig. 9. Boolean Operation result (A ∪* B case).

부품이라 할지라도 그 형상 요소들은 결국 하나의 구속조건 그래프에 의해 관리될 수 있다.

구속조건 파급을 통한 모델 수정은 사용자가 구속조건의 변수를 조작하여 수행된다. 이 때, 병합된 구속조건 그래프에서 갱신 작업의 파급만으로 부품의 형상을 신속히 수정할 수 있다(2장 후반부 참조). 변수 하나가 변경되면, 그 변수를 포함하는 구속조건의 갱신을 통해 해당 절점의 형상 요소가 변경되고, 파급 과정을 통해 아크 방향을 따라 모든 형상 요소들이 변경되어 최종적으로 수정된 모델을 얻게 된다. 예를 들어 Fig. 7의 변수 r이 r'으로 변경된 경우 구속조건의 파급 과정은 Fig. 11과 같다. 먼저 변수 r에 의해 직접 구속되는 절점 C1이 변경되고 C1으로부터 구속되는 모든(굵은 실선으로 표시된) 절점들이 아크 방향을 따라 순차적으로 갱신되어 모델이 신속히 수정된다.

4. 적용사례

본 시스템을 확장성 및 상속성이 우수한 C++ 언어[16]와 그래픽 표현을 위한 OpenGL[17]을 이용하여 pentium MMX 200Hz PC상에서 구현 하였다. 솔리드 모델의 자료구조는 Fig. 12와 같이, B-rep을 위한 반 모서리 구조[18]와 구속조건 그래프를 위한 네트워크 구조로 나누어져 있으며, 구속조건 그래프의 각 절점은 B-rep 구조내의 해당 형상 요소를 지시하도록 하였다. 또한 Fig. 13은 구현된 시스템의 전체 구조와 초기 화면을 보여준다. 사용자가 부품을 정의

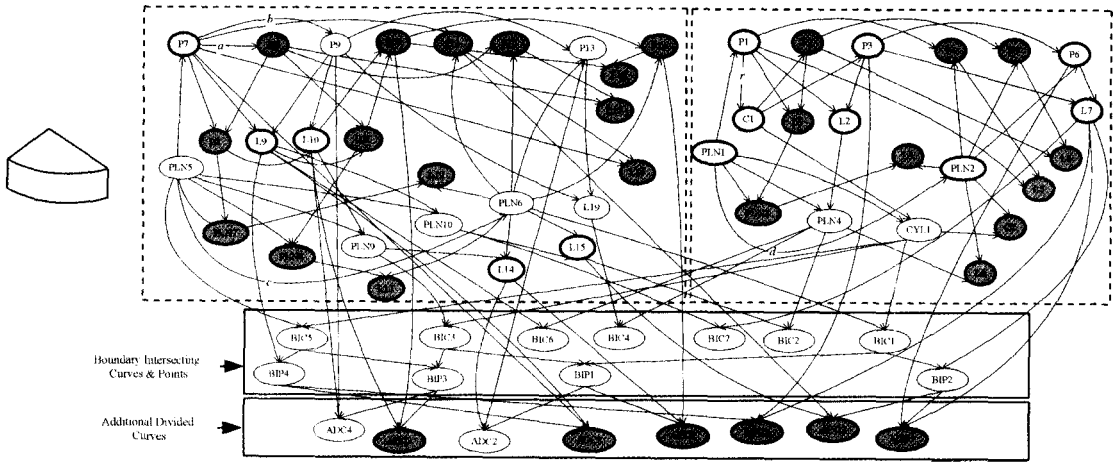

Fig. 10. Boolean Operation result (A∩* B case).

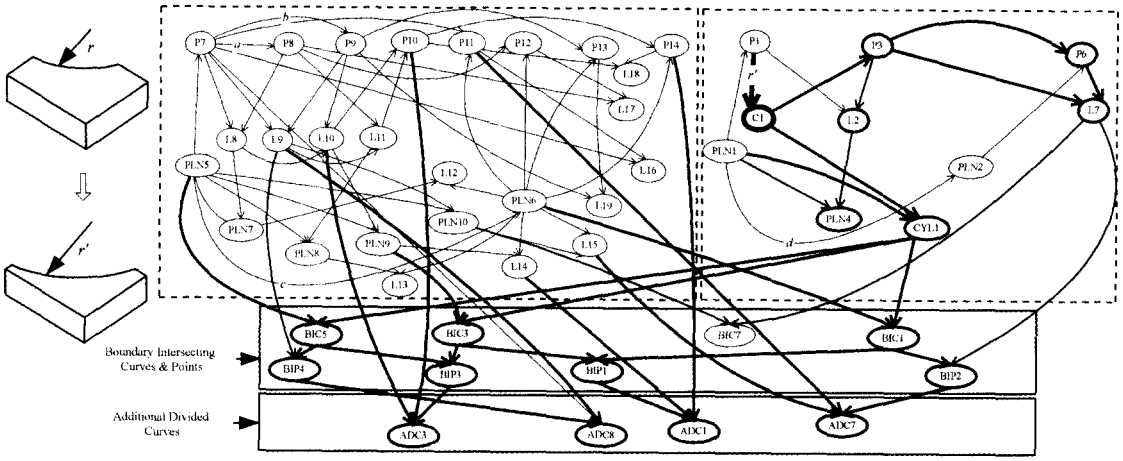

Fig. 11. Constraint propagation for a parameter change in Fig. 8.

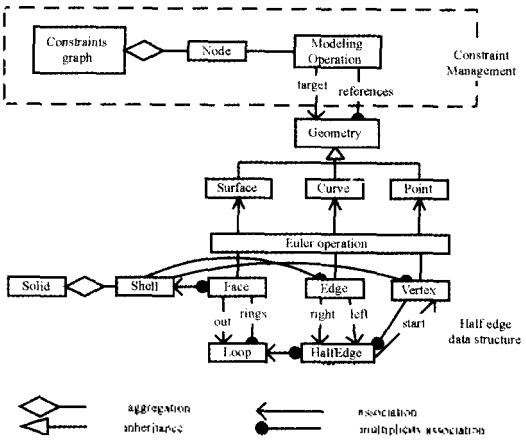

Fig. 12. Data structure.

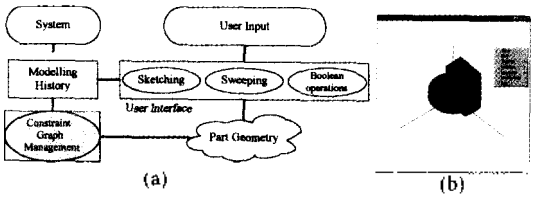

Fig. 13. System structure (a) and using example (b).

하기 위해 스케치 세트 및 모델링 연산을 수행하도록 user interface를 구성하였으며, 제안된 구속조건 관리 시스템이 모델의 형상 및 파라미터를 관리한다. 이 때, 기존 시스템들과 유사하게 모델링 히스토리를 관리하도록 하였다. 3장에서 설명한 구속조건의

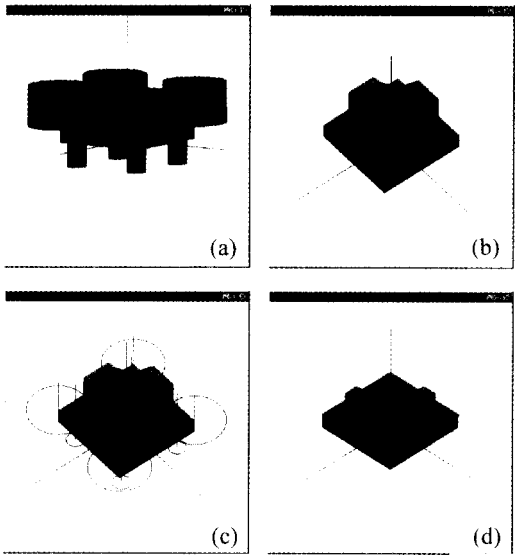

Fig. 14. System using example.

파급을 통한 모델 변경에는 필요하지 않으나, 추후 기존의 모델링 연산의 재수행 방법과 병행하여 사용하기 위해 존재한다. 개발된 구속조건 관리 시스템은 생성된 3차원 모델 형상의 모든 구속조건을 자동 생성하여 관리하며, 모델 수정을 위한 인터페이스를 위해 사용된 구속조건을 언제라도 가시화 할 수 있다. 이를 통해 사용자는 구속조건의 변수 값을 수정할 수 있으며, 히스토리 내의 다양한 모델링 작업을 재 수행 하지 않고, 구속조건의 파급만으로 모델의 형상을 신속히 변경할 수 있다.

일반적인 기계부품을 이용하여 본 시스템을 검증하였다. Fig. 14에 보이는 부품은 1개의 직육면체와 8개의 원기둥 기본입체를 사용하여 모델링 된 것이다. Fig. 14는 스위핑을 통해 생성된 기본입체, 모델링 된 결과, 수정 작업을 위한 파라미터(구속조건) 가시화, 모델의 수정 예를 차례로 보여주고 있다. 이때 부품의 구속조건 그래프 파급만으로 스위핑이나, 불리언 연산의 재 수행 없이 신속하게 부품의 형상을 갱신하였다. Fig. 15는 커넥팅 로드의 간략 모델에 본 시스템을 적용한 예이며, Fig. 16은 기계가공용 공구 홀더의 모델링과 수정 예를 보여주고 있다.

Fig. 15. System using example 2.

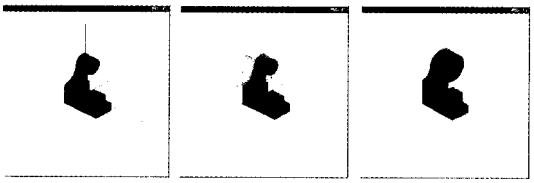

Fig. 16. System using example 3.

5. 결 론

본 논문에서는 가변 형상 시스템에서 구속조건을 갖는 스케치 세트의 스위핑 작업과 불리언 연산에서 발생되는 기하학적 관계를 8가지 유형의 구속조건으로 표현하여 기본입체와 부품의 구속조건 그래프를 자동으로 확장하는 방법을 제안하였다. 이를 통해 파라미터의 변경이 부품의 최종 타폴로지를 변화시키지 않는 변경인 경우에, 단지 구속조건의 파급만으로 부품의 형상을 신속히 수정할 수 있는 가변 형상 시스템을 구현하였다. 이는 기존 가변 형상 시스템에서 부품 수정을 위해 반드시 거쳐야 했던 모델링 연산을 재 수행할 필요가 없어 보다 신속하며 효율적인 모델 수정 방법이다. 또한 본 논문에서 제안된 구속조건 관리 방법은 일반적인 솔리드 모델링뿐 아니라, 파라메트릭 모델링 등에 이용될 수 있으며, 특히 특징형상 모델링에서 사용자 정의 특징형상 라이브러리를 구축하고 이를 이용하여 타폴로지 변경 없이 사용되는 다양한 variant 생성에 매우 효과적으로 사용될 수 있다고 판단된다.

그러나 본 논문에서 제안된 알고리즘은 이미 언급한 데로 구속조건의 변화가 모델의 최종 타폴로지를 변화 시키는 경우를 배제하고 있다. 향후, 이러한 한계를 극복하기 위해 불리언 연산 시, 경계 교차 요소들과 기존 요소들 간의 공간 관계 추론(spatial reasoning)을 통한 파라미터 변화가능 범위를 결정하는 연구를 통해, 기존의 불리언 연산 재수행 방법과 병행하는 방법에 대한 연구가 수행될 필요가 있다.

참고문헌

1. Hoffmann, C., "Geometric Constraints Solving in R^2 and R^3", *Computing in Euclidean Geometry*, (eds Du, Z. and Huang, F.), World Scientific, pp. 266-298, 1995.
2. *CATIA Part Design Users Guide version 4*, Dassault System, 1999.
3. *SolidWorks Users Guide*, SolidWorks Corporation, 1998.

4. *ProENGINEER, Users Guide*, Parametric Technology Corporation, 1998.
5. Gao, X. and Chou, S., "Solving geometric constraint systems. II. A symbolic approach and decision of Reconstructibility", *Computer-Aided Design*, Vol. 30, No. 2, pp. 115-122, 1998.
6. Hui, K. and Kan, Y., "Distributed boundary representation and Boolean operations on a massively parallel computer", *Computer-Aided Design*, Vol. 8, No. 28, pp. 115-122, 1998.
7. Sutherland, I., "Sketchpad, A Man-Machine Graphical Communication System", *proc. of the Spring Joint Comp. Conference*, NorthHalland, pp. 329-345, 1963.
8. Gossard, L. and Light, D., "Variational Geometry in Computer-Aided Design", *Computer-Aided Design*, Vol. 15, No. 3, pp. 171-177, 1981.
9. Kondo, K., "PIGMOD: parametric and interactive geometric modeler for mechanical design", *Computer-Aided Design*, Vol. 22, No. 10, pp. 221-233, 1990.
10. Kondo, K., "Algebraic Method for manipulation of dimensional relationship in geometric models", *Computer-Aided Design*, Vol. 24, No. 3, pp. 141-147, 1992.
11. Wu, W., *Basic Principles of Mechanical Theorem Proving in Geometries*, Springer-Verlag, 1993.
12. Kramer, G., *Solving Geometric Constraints Systems*, MIT Press, 1992.
13. Owen, J., "Algebraic Solution for Geometry from Dimensional Constraints", *ACM symp. Found of Solid Modelling*, ACM press, Austin TX, pp. 397-407, 1991.
14. Capoyleas, V., Chen, X. and Hoffmann, C., "Generic naming in generative constraint-based design", *Computer-Aided Design*, Vol. 26, No. 1, pp. 17-26, 1996.
15. Kripac, J., "A mechanism for persistentialy naming topological entities in history-based parametric solid models", *Computer-Aided Design*, Vol. 29, No. 2, pp. 113-122, 1997.
16. Requicha, A. and Voelcker, H., "Boolean Operation in Solid Modeling: Boundary Evaluation and Merging Algorithm", *proc. IEEE*, Oct., pp. 30-44, 1983.
17. Prata, S., *C++ PRIMER PLUS*, WAITE GROUP PRESS, 1991.
18. Neider, J., Davis, T. and Woo, M., *OpenGL Programming Guide*, Addison Wesley, 1995.
19. Toriya, H. and Chiyokura, H., *3D CAD principles and applications*, Springer-Verlag, 1991.

김 웅 주

1997년 한양대학교 기계설계학과 학사
1999년 한양대학교 기계설계학과 석사
1999년~현재 (주)한국타이어 REP 개발팀
관심분야: Geometric Constraints Solving, Solid Modeling

정 채 봉

1996년 한양대학교 기계공학과 학사
1998년 한양대학교 기계공학과 석사
1998년~현재 한양대학교 기계설계학과 박사과정
관심분야: Geometric Modeling, Feature Recognition, Parametric modeling

김 재 정

1981년 한양대학교 정밀기계학과 학사
1983년 George Washington 공학석사
1983년~1984년 미국 national Food Processors Association 연구원
1989년 미국 MIT 공학박사
1989년~1991년 미국 IBM T.J. Waston 연구소 연구원
1991년~1993년 한국 IBM 소프트웨어 연구소 연구원
1993년~현재 한양대학교 기계공학부 부교수
관심분야: Geometric Modeling, CAD/CAM응용