

컨피규레이션 공간을 이용한 기구학적 공차 설계

경 만 호*

Kinematic Tolerance Synthesis Using Generalized Configuration Spaces

Kyung, M.-H.*

ABSTRACT

This paper presents a new framework of kinematic tolerance synthesis and describes the implemented algorithm for planar mechanical systems comprised of higher kinematic pairs. Input to the synthesis algorithm is a parametric model of the mechanical system with allowed parameter ranges (tolerance ranges). The model is specified as the part profiles consisting of line and arc segments and the motion axes along which each part moves. The algorithm analyzes tolerance in generalized configuration space, called contact zones bounding the worst-case variations, and identifies bad system variations. The bad system variations then are removed out of the parameter ranges by adjusting the nominal parameter values if possible and then shrinking the ranges otherwise. This cycle is repeated until no more bad variations are found. I show the effectiveness of the algorithm by case studies on several mechanisms.

Key words : higher pairs, configuration space, kinematic tolerance synthesis

1. 서 론

본 논문에서는 고차대우(higher-pair)들로 구성된 기구의 공차 설계(kinematic tolerance synthesis)를 자동적으로 수행하는 알고리즘을 처음으로 제안한다. 고차대우는 캠이나 기어와 같이 점 또는 선 접촉에 의해 웨어를 이루는 조인트를 말한다. 공차 설계는 기계 설계 과정에서 기계 요소들의 공차 내역(tolerance specification)을 정하는 작업으로, 기계 시스템의 올바른 작동을 보장하는 범위 안에서 최적의 공차 범위를 찾는 것이다. 공차 범위는 지나치게 넓은 경우 불량율을 높이는 문제가 있고, 반면에 작을 경우 높은 가공 및 조립 정밀도를 요구하여 급격한 생산 비용의 증가를 불러오는 문제가 있다. 따라서, 가능한 불량율 최소화하면서 필요 이상의 정밀도를 요구하지 않는 최적의 공차 범위를 찾는 것이 생산성 향상을 위한 매우 중요한 일이다.

일반적으로 최적 공차 범위를 찾는 것은 설정과 해

석을 반복해서 이루어 진다. 먼저 대강의 초기 공차 범위를 설정하고, 공차 해석(tolerance analysis)을 하여 공차 범위 안에서 변형된 기계 요소들이 올바르게 동작하는지 검증한다. 반일 결함이 있는 변형이 발견된 경우 두 가지 방향으로 수정을 하게 되는데, 부품을 재설계하는 것과 공차 범위를 축소하는 것이다. 부품을 재설계하는 경우는 공차 범위를 그대로 유지하므로 생산 비용을 증가시키지 않지만, 최선의 수정 방향을 결정하기 위한 노력이 필요하다. 공차 범위 축소는 가장 쉬운 선택이지만, 생산 정밀도를 높이기 위한 비용 증가가 발생한다. 일단 설계 또는 공차 범위에 대한 수정이 이루어지면 다시 공차 해석을 반복하여 결함 변형이 발생하는지 확인한다. 이러한 과정을 반복하면 점차적으로 최적의 공차 범위를 가지는 설계로 수렴해 가게 된다.

저자의 관련 연구 조사에 의하면 아직까지 고차 대우들로 구성된 기구에 대한 기구학적 공차 설계를 자동적으로 수행하는 방법은 알려진 것이 없다. 그 이유는 접촉면이 지속적으로 이동하는 고차 대우의 운동 변형을 일반적으로 모델링하기가 어렵기 때문이다. 본 논문에서는 Sacks 등^[1]에 의해 제안된 컨피규레이션

*강희원, 아주대학교 미디어학부
- 논문투고일: 2004. 12. 29
- 심사완료일: 2005. 05. 18

공간과 접촉역(contact zone)을 이용하여 고차대우의 운동 변형을 표현하고 이를 바탕으로 공차 설계의 자동화 알고리즘을 개발하였다. 컨피규레이션 공간은 기구를 구성하는 요소들의 공간적 상태를 나타내는 공간이다. 즉, 이 공간에서의 좌표축은 각 요소들의 회전각 또는 이동 변위를 나타낸다. 접촉역은 기구의 최대, 최소 운동 변형을 컨피규레이션 공간에서 나타낸 것이다. 본 논문에서 제안된 알고리즘은 접촉역을 이용하여 결합 변형을 찾아내고, 이러한 변형들을 자동적으로 제거해 줌으로서 기구 설계의 시간과 비용을 최소화할 수 있다.

본 알고리즘에 입력되는 기구 모델은 2차원 형상을 가지며, 기계 요소들의 기하학적인 형상과 배치가 파라미터로 제어되는 일반적인 파라메트릭 모델이다(그림 Fig. 2 참조). 따라서, 기하 공차에서의 형상과 관련된 모양 공차, 두 부품의 상대적인 위치와 관련된 자세 공차, 위치 공차, 흔들림 공차 등을 모두 파라미터의 허용 범위로 나타낼 수 있다. 기구를 구성하는 페어들은 모두 고차대우들로 구성되어 있다고 가정하고 있는데, 알고리즘의 확장을 통해 저차대우를 포함시키는 것도 가능하다. 본 연구에서 제안된 알고리즘은 직선과 원호로 디자인된 부품들로 구성된 기계 시스템에 대해서 구현을 하였다. 스플라인이나 베지어 곡선과 같은 자유 곡선으로 디자인 된 경우에는 연속된 직선과 원호로 근사시켜 입력하면 된다. Sacks 등¹⁸⁾의 연구에 의하면 이러한 고차 대우로 구성된 기구는 전체 엔지니어링 응용 분야에서 90% 가까이를 차지한다고 한다. 따라서, 본 알고리즘의 입력 모델은 충분히 넓은 커버리지를 가진다고 볼 수 있다.

개략적인 알고리즘은 먼저 접촉역을 이용해 설정된 공차 범위 안에 있는 변형들 중에서 운동 변형이 가장 크게 일어나는 것들을 찾아서, 검사 리스트에 저장한다. 다음은 검사 리스트에 저장된 각 변형들에 대해서 결합이 있다고 판단되는 변형들만을 찾아내어, 제거 리스트에 저장한다. 다음은 공차 범위로부터 제거 리스트에 있는 시스템 변형들을 제외시키도록 명목 설계의 파라미터 값들을 조정한다. 이 알고리즘은 제거 리스트가 공집합이 될 때까지 반복된다. 본 논문에서는 위 공차 설계 알고리즘에 관해 자세히 설명하고, 실제 몇 가지 기계 모델에 대한 적용을 통해 타당성을 보이고자 한다.

2. 관련 연구

공차 설계는 생산품의 품질과 비용에 밀접한 관련

이 있기 때문에, 그 동안 기계 설계 분야에서 중요하게 다루어졌고 많은 연구들이 되어 왔다. 전통적인 공차 설계 연구의 결과들은 Zhang¹¹⁾에 잘 정리되어 있다.

기구학적 공차 설계는 기구학적 해석(kinematic analysis)을 바탕으로 하는데, 고정된 접촉을 가지는 어셈블리의 기구학적 해석은 일반적으로 많은 연구가 되어 왔다. Clement 등¹²⁾과 Whitney 등¹³⁾은 중요한 치수 파라미터들을 찾고, 부품의 위치와 접촉으로부터 공차 체인을 만들고, 벡터, 토오소, 매트릭스 변환 등을 이용하여 파라미터 범위를 정하는 방법들을 제안하였다. 링크와 같은 저차대우(lower pair)로 구성된 기구들에 대한 기구학적 해석에 관한 연구는 [4]에 잘 정리되어 있다. 고정 접촉에 대한 기구학적 변형은 선형적 근사 또는 Monto Carlo 방법과 같은 통계적 시뮬레이션 등으로 구할 수 있는데, 부정확하거나 또는 시간이 많이 걸리는 문제가 있다. Glancy와 Chase¹⁵⁾는 설계 함수의 미분을 계산하여 기구학적 변형을 실험에서 얻은 통계 데이터에 맞추는 방법을 제안하였다. 고정 접촉을 가지는 어셈블리에 대한 위와 같은 연구 결과들은 기어, 캠 등과 같이 접촉면이 시간에 따라 변하는 고차 대우들에 대해서는 적용할 수 없는 한계를 가지고 있다.

고차 대우로 구성된 기구에 대한 일반화된 공차 해석 방법은 Sack 등^{16,17)}에서 처음으로 개발되었다. 이들은 컨피규레이션 공간 상에서 기구학적 변형을 계산하여 공차 범위 안의 결함을 예측하는 방법을 개발하였다. 이 결과들은 본 논문의 알고리즘 개발에 기반이 되었다.

3. 컨피규레이션 공간

3.1 정의

본 연구는 기구의 컨피규레이션 공간을 기반으로 하고 있다¹⁸⁾. 컨피규레이션 공간은 기구를 구성하는 요소들의 공간적 상태를 각 축으로 하는 공간이다. 예를 들어 Fig. 1(a)와 같이 휠과 드라이버로 구성되어 있는 제네바 페어에 대해서, 컨피규레이션 공간은 Fig. 1(b)와 같이 만들어진다. 가로 축은 드라이버의 회전각 θ 가 되고, 세로 축은 휠의 회전각 ω 가 된다. 따라서, 컨피규레이션 공간에서의 한 점은 제네바 페어의 특정한 시점의 공간적 상태를 나타내게 된다. 컨피규레이션 공간은 세 종류의 부공간으로 분할되는데, 두 부품이 서로 떨어져 있는 자유 공간(free space), 두 부품이 서로 교차하는 교차 공간(blocked space), 그

리고 두 공간의 경계가 되는 접촉 공간(contact space)이다. 접촉 공간은 자유도가 2인 고차 대우인 경우 접촉 곡선이 연속적으로 연결된 접촉 곡선 체인이다. 각각의 접촉 곡선은 접촉하는 두 면에 대한 접촉방정식으로 정의된다. 물리적으로 두 부품이 서로 교차하고 있는 것은 불가능하므로 교차 공간의 점은 불가능한 컨피규레이션이다. 따라서 접촉 곡선은 기구학적 배어를 구성하는 두 부품의 운동을 제한하는 일종의 경계 제한(boundary constraint)이라고 할 수 있다.

고차 대우에서 운동은 두 기계 요소의 접촉을 통해 이루어지므로, 접촉 곡선 상에서 변하는 컨피규레이션이 된다. 따라서, 접촉 곡선의 모양을 통해 고차 대우의 운동을 해석할 수 있다. Fig. 1(a)는 제네바 페어의 컨피규레이션 공간이 원점 (0,0)에 위치해 있는 경우다. 만일 드라이버를 시계방향으로 회전시키면 컨피규레이션 공간에서의 점은 왼쪽으로 이동하면서 기울어진 접촉 곡선을 만나게 된다. 이때 컨피규레이션 점은 접촉 곡선을 따라 올라가게 되어 ω 값이 증가하게 되고, 따라서 휠이 반시계 방향으로 회전하게 된다. 수평 통로에 이르면 접촉 곡선이 수평이 되어 컨피규레이션 점도 수평 이동으로 움직이게 된다. 따라서, ω 값이 일정하게 제한되므로 휠의 움직임은 고정된다. 이 모드는 드라이버의 바깥 면이 휠의 오른쪽 아래 모퉁한 원호 면과 접촉하면서 휠을 고정시키고 있는 상태이다. 이와 같이 고차 대우의 운동은 컨피규레이션 공간상에서 완전하게 해석될 수 있다.

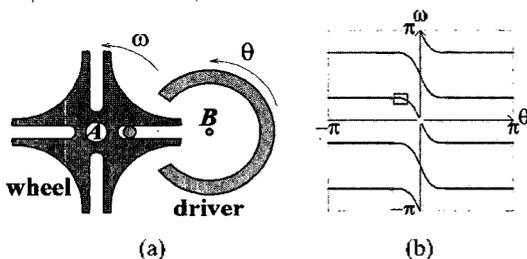


Fig. 1. (a) Geneva pair, (b) configuration space.

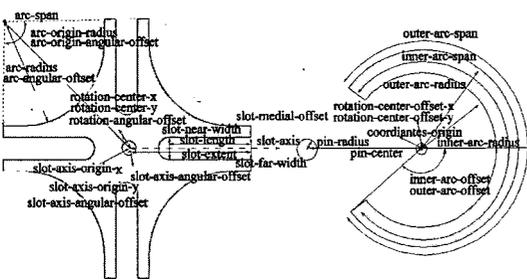


Fig. 2. Parametric model of Geneva pair.

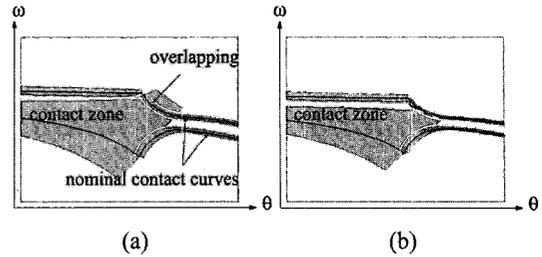


Fig. 3. (a) contact zone, (b) modified contact zone.

3.2 접촉역

전통적인 공차 해석에서는 기계 요소 형상의 변형 범위를 나타내는 허용역(tolerance zone)을 이용하여 공차 해석을 하였다. 이는 어셈블리 공차 해석에는 유용하지만, 운동 변형에 대한 직접적인 정보를 주지 않으므로 기구학적 공차 해석에는 부적합하다. 특히 접촉선이 시간에 따라 변하는 고차 대우의 경우에는 더욱 적용하기가 어렵다.

컨피규레이션 공간상에서 기계 요소의 변형으로 인한 고차 대우의 운동 변형은 접촉 곡선의 변형으로 나타난다. 이때 접촉 곡선 변형의 최대 최소 범위를 접촉역이라고 부른다. 따라서, 접촉역은 명목 접촉 곡선(검은색)을 따라 놓여진 밴드(초록색)로 나타난다. Fig. 2와 같이 정의된 제네바 페어의 파라미터 모델에서 각 파라미터의 공차 한계를 $\pm 0.018 \text{ mm}$ 로 설정할 경우, 접촉역은 Fig. 3과 같이 구해진다. 접촉역을 구하면 기구학적 민감도 분석을 쉽게 할 수가 있다. 즉, 밴드의 두꺼운 부분은 접촉 곡선의 변화가 큰 부분이고, 따라서 파라미터 변화에 민감한 운동 변형이 많이 일어나는 구간이 된다.

접촉역의 모양은 기구학적 결함에 대한 중요한 단서를 준다. 예를 들어 Fig. 3(a)를 보면 위쪽 아래의 접촉 곡선의 접촉역이 위 곡선의 접촉역과 중첩되는 것을 발견할 수 있다. 이 부분은 드라이버의 편이 휠 슬롯을 나오고 바깥면이 휠을 잠그면서 회전하는 부근의 컨피규레이션 공간이다. 접촉역의 중첩은 제네바 페어의 변형들 중에 아래와 위의 접촉 곡선이 서로 교차되는 변형이 존재한다는 것을 의미한다. 두 곡선이 서로 교차하는 점은 드라이버가 휠을 서로 다른 두 군데에서 동시에 반대 방향으로 미는 순간으로, 이러한 모순된 동작으로 제네바 페어는 여기서 멈추게 된다. 이와 같이, 접촉역의 중첩은 공차 범위 안에 결함 변형들이 존재하여 생산 과정에 불량품이 생산될 수 있다는 가능성을 암시한다. Fig. 3(b)는 파라미터의 명목 값을 약간 수정했을 경우 얻어지는 접촉역이다. 접

촉역의 중첩이 발견되지 않아 불량율이 낮은 설계로 판단할 수 있다.

4. 공차 설계 알고리즘

본 논문에서 제안한 공차 설계 알고리즘은 대략적으로 세 단계로 나누어 진다.

1. 주어진 명목 설계에 대한 접촉역을 생성하고, 이로부터 결함 검사를 위한 설계 변형을 구하여 검사 리스트에 저장한다. 검사할 설계 변형들은 운동 변형이 최대가 되는 변형들로, 접촉 곡선이 접촉역의 경계에 도달하는 변형들을 선택한다.
2. 다음은 검사 리스트에 있는 변형들 중에서 결함 변형들을 찾아 제거 리스트에 저장 한다. 이 과정은 컨퍼레이션 공간 배치 알고리즘과 기구학적 시뮬레이션을 병행 사용하여 수행한다.
3. 위에서 구해진 제거 리스트를 이용하여 설계의 명목 파라미터 값들을 수정한다. 명목 설계의 수정은 공차 범위를 이동시켜 제거 리스트의 변형들이 공차 범위 밖으로 나가도록 만든다.

위 알고리즘은 반복적으로 실행되어 제거 리스트가 공집합이 될 때 끝나게 된다. 제거 리스트는 대부분의 경우 알고리즘이 반복됨에 따라 지속적으로 감소하게 되는데, 최악의 경우 제거 리스트에 기존에 제거된 변형이 다시 들어오는 수가 있다. 이럴 경우 공차 설계 프로그램이 무한 루프에 빠지게 된다. 이미 제거된 변형이 다음 반복에 다시 등장하는 것은 공차 범위가 커서 모든 결함 변형을 완벽히 제거할 수 없는 경우이다. 이런 경우가 발생하면 불가피하게 설계 파라미터들 중에 하나를 선택하여 공차 범위를 줄이게 된다.

지금부터 각 단계의 자세한 설명을 하도록 하겠다.

4.1 검사 리스트 생성

검사 리스트는 공차 범위 안에서 발생하는 기구학적 결함들을 찾기 위해 선택된 기구 변형들의 집합이다. 기구 변형들은 설계 파라미터 공간에서의 벡터 $\mathbf{u} = (u_1, \dots, u_n)$ 으로 공차 구간 안에 있는 점들이다. 파라미터 공간에서 공차 구간은 명목 설계 \mathbf{u}_0 을 중심으로 하고 공차 한계 δ 를 한 변의 길이로 하는 n 차원 박스가 된다. 검사 리스트에 있는 기구 변형들은 결함 발생 확률이 높은 최대 운동 변형들로 선택한다. 최대 운동 변형들은 컨퍼레이션 공간에서 접촉 곡선이 접촉역의 경계에 이르는 변형이 되기 때문에, 접촉역

계산을 통해서 쉽게 얻을 수가 있다.

본 논문에서는 접촉역 계산을 위해 Kyung과 Sacks¹⁹⁾가 개발한 비선형 접촉역 알고리즘을 이용하였다. 접촉 곡선을 정의하는 접촉 방정식은 접촉하는 두 접촉면 f_1, f_2 에 대해 $C(\theta, \omega, f_1, f_2) = 0$ 로 나타낼 수 있고, 이 식을 만족하는 (θ, ω) 가 접촉 곡선이 된다. f_1 과 f_2 의 모양은 설계 파라미터 u_1, \dots, u_2 의 함수이기 때문에 $C(\theta, \omega, u_1, \dots, u_2) = 0$ 으로 쓸 수 있다. 접촉역 계산에서는 위 식을 만족하면서 접촉 곡선 위의 점 (θ, ω) 가 접촉 곡선의 노멀 방향으로 최대 변형되는 점을 구하게 된다. 이때 얻어진 파라미터 벡터 \mathbf{u} 가 검사 리스트에 저장될 기구 변형이 된다.

위에서 얻어진 기구 변형만으로는 결함 변형을 찾는 데 충분하지 않다. 그 이유는 접촉역 계산이 개별 접촉 곡선 단위로 이루어지기 때문에, 접촉역으로부터 구해진 변형에서는 해당 접촉 곡선만이 최대가 된다. 따라서, 여러 부분의 접촉이 상호작용하여 발생하는 대부분의 결함들은 위의 변형들에서는 발견되지 않을 확률이 높다. 이러한 예는 Fig. 4에 예시되어 있다. 컨퍼레이션 공간이 그림과 같고, 두 접촉 곡선 사이의 자유 공간을 지나면서 동작하도록 되어 있다고 하자. 만일 Fig. 4(b)와 같이 위 아래의 접촉 곡선이 동시에 최대로 변형되어 자유 공간의 중간이 막히면, 페어의 작동이 더 이상 진행되지 못하고 멈추게 된다. 이러한 결함은 접촉역으로부터 얻어진 변형에서는 Fig. 4(c)와 같이 일부 접촉 곡선만이 최대로 변형되기 때문에 발생하지 않게 된다.

이러한 문제를 해결하기 위해 본 논문에서는 파라미터 조합이라는 방법을 사용하였다. Fig. 4(b)와 같은 경우를 포함하기 위해서는 접촉 곡선들의 변형이 가능한 많은 부분에서 동시에 이루어 져야 한다. 이러한 파라미터 벡터를 찾는 것은 실제로 쉽지 않다. 왜냐하면 접촉 곡선들의 조합은 이론적으로 접촉 곡선 수에 대해 기하급수적으로 증가하기 때문이다. 대신에 본 논문에서는 서로 독립적인 접촉 곡선들 간의 조합에 대해서만 고려를 하였다. 이 경우 상호 독립적이지 않

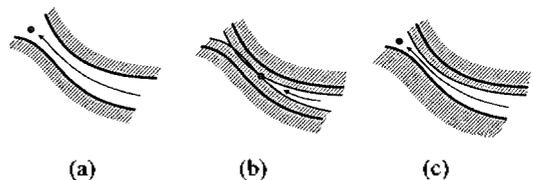


Fig. 4. (a) Nominal contact curves, (b) extreme variations of the upper and lower contact curves, (c) extreme variation of the upper curve.

은 접촉 곡선들이 동시에 최대 변형되는 경우는 제외 되게 되는데, 실험적으로 이러한 접촉 곡선들이 동시에 최대 변형되는 경우는 매우 드물다. 독립적인 접촉 곡선 조합은 다이나믹 알고리즘을 이용하였다. 이 알고리즘을 *파라미터 조합*이라 부르고 다음 장에서 자세히 설명한다.

4.1.1 파라미터 조합 알고리즘

파라미터 조합은 검사 리스트에서 선택된 두 개의 파라미터 벡터에 대해 수행된다. 두 벡터가 조합되기 위해서는 최대 변형 접촉 곡선들이 서로 독립적이어야 한다. 다시 말해 두 접촉 곡선을 정의하는 접촉 방정식들이 서로 다른 설계 파라미터를 사용하는 경우이다. 기계 요소의 접촉면은 점, 직선, 또는 원호로 모델링되기 때문에, 접촉 방정식에는 전체 n 개의 파라미터들 중에서 보통 5~6개 정도만이 사용된다. 사용되는 파라미터들이 공통적이지 않은 접촉 곡선들은 서로 독립적으로 움직이므로, 간단히 파라미터 값들을 조합함으로써 두 곡선의 최대 변형을 동시에 가지는 파라미터 벡터를 구할 수 있다.

Table 1은 조합 알고리즘을 의사코드로 보여주고 있다. V_0 는 입력 검사 리스트고, V_1 은 출력 검사 리스트다. 알고리즘은 V_0 에서 파라미터 벡터 u_i 를 꺼내어 이 벡터와 V_0 의 나머지 벡터들과의 가능한 조합을 찾는다. 조합 가능 여부는 최대 변형 접촉 곡선들에 사용된 파라미터들 중에 공통된 파라미터가 있는지로 판단한다. 만일 u_i 와 조합이 가능하다면, 그리고 u_i 와 u_j 에서 사용된 파라미터들이 각각 집합 X_1 과 X_2 라 면, 새로운 파라미터 벡터 u_{new} 은

$$u_{new}[k] = \begin{cases} u_i[k] & \text{if } u_k \in X_1 \\ u_j[k] & \text{if } u_k \in X_2, \quad k = 1, \dots, n \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

새로 만들어진 u_{new} 는 다음 조합에 계속 사용되기 위하여 검사 리스트에 다시 저장된다. u_i 를 이용한 조합이 모두 끝나면 마지막으로 u_i 는 V_1 에 저장된다.

이론적으로 V_1 의 크기는 파라미터 수에 따라 기하급수적으로 증가하게 된다. 이를 피하기 위해, 새로운 파라미터 벡터를 추가할 때마다 검사 리스트에 ϵ 범위 안에 다른 벡터가 이미 있는지를 찾아 보고, 없을 경우에만 추가를 한다. 파라미터 공간에서 가까운 거리에 있는 점들은 대부분 기구학적으로 유사한 운동을 하기 때문에 결함에 있는 경우 주변 점들도 같은 결함을 갖게 된다. 따라서, 검사 리스트의 벡터들로부터 일정한 거리 안에 있는 파라미터 벡터들은 새로 추가를 하지 않아도 큰 문제가 없다. 이러한 필터링을 실험에 적용한 결과 2800여개의 조합된 파라미터 벡터들을 675개의 벡터로 줄일 수가 있었고, 이때의 결과는 필터링을 하지 않았을 때와 차이가 없었다.

4.2 기구학적 결함 변형 인식

검사 리스트가 완성되면 다음 단계는 기구학적인 결함 변형을 찾아서 제거 리스트에 저장하는 것이다. 어떤 경우를 결함으로 판단할 것인가는 설계 목적에 따라 달라지게 된다. 본 연구에서는 결함 여부를 기구학적 변형이 명목 설계와 다른 정도를 보고서 결정하였다. 이러한 비교를 자동으로 하기 위해 컨피규레이션 매치와 기구학적 시뮬레이션을 동시에 사용하였다. 컨피규레이션 매치는 기구학적 변형의 질적인 차이를 판단하는데 사용하였고, 기구학적 시뮬레이션은 최종 출력 함수 값의 차이를 판단하는데 사용하였다. 두 가지 중 한가지 테스트라도 통과하지 못하면 결함으로 판단하였다.

기구학적으로 질적인 차이는 컨피규레이션 공간상에서 자유 공간의 위상(topology) 변화로 나타난다. 예를 들어 변형들 중에 두 개의 접촉 곡선이 교차하여 자유 공간이 중간에 분리되거나(Fig. 4b), 또는 연속된 두 개의 접촉 곡선이 떨어지면서 새로운 자유 공간이 나타나는 경우가 해당된다. 두 컨피규레이션 공간의 위상 변화는 Kyung과 Sacks^[10]가 개발한 컨피규레이션 공간 매치 알고리즘을 이용하여 자동적으로 찾을 수 있다. 컨피규레이션 공간 매치 알고리즘은 자유 공간을 둘러 싸고 있는 연속된 접촉 곡선들을 차례대로 따라가면서 비교한다. 만일 접촉면이 일치하지 않는

```

V0 = {ui | ui is a kinematic variation};
V1 = ∅;
for ui ∈ V0 do
    V0 = V0 - {ui};
    for uj ∈ V0 do
        if ui and uj is combinable then
            unew = Combine(ui, uj);
            V0 = V0 + {unew};
        end
    end
    V1 = V1 + {ui};
end
    
```

Table 1. 파라미터 조합 알고리즘.

접촉 곡선이 발견되면, 주변 접촉 곡선들을 분석하여 자유 공간의 위상 변화가 발생했는지를 판단하게 된다.

기구 변형의 출력 함수가 명목 설계로부터 일정 범위를 벗어나는지 여부를 판단하는 것은 기구학적 시뮬레이션을 이용하였다. 시뮬레이션에는 Sacks와 Joskowicz²¹가 개발한 알고리즘을 이용하였다. 기구학적 시뮬레이션을 하게 되면 시작 컨피규레이션에서 시작하여 시간에 따라 변화하는 각 요소들의 상태 값을 구할 수 있다. 이 결과에서 입력 요소의 상태 값 θ 와 출력 요소의 상태 값 ω 를 2차원 평면에 표현하면 입력에 대한 출력 함수 곡선 $\omega=f_0(\theta)$ 을 얻을 수 있다. 이 함수 곡선과 명목 설계의 함수 곡선 $\omega=f_1(\theta)$ 의 차이를 구하여 $|f_0(\theta)-f_1(\theta)|>\delta$ 인 경우를 결함으로 판단한다.

4.3 결함 변형의 제거

다음 단계는 결함이 있는 것으로 판단된 제거 리스트의 파라미터 벡터들을 제거하는 것이다. 공차 범위 안에 있는 특정 파라미터 벡터들은 Fig. 5에서와 같이 공차 범위를 이동하여 제거할 수 있다. 다른 방법은 공차 한계를 축소시켜 특정 파라미터 벡터들이 범위 밖으로 나가도록 하는 것이다. 본 연구에서는 우선적으로 공차 범위를 이동시켜 최대한 결함 변형들을 제거하고, 남은 결함 변형들은 이차적으로 공차 한계를 축소시켜 제거한다.

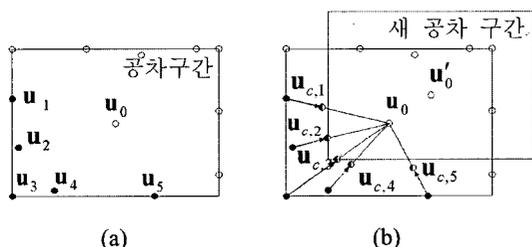


Fig. 5. (a) Tolerance interval where u_i 's are defective variations; (b) modified tolerance interval with critical points $u_{c,i}$.

공차 범위를 이동시키는 것은 명목 설계값 u_0 에 이동 벡터 Δu 를 더하는 것으로 표현할 수 있다:

$$u_0' = u_0 + \Delta u \quad (2)$$

이때 제거 리스트의 결함 변형들을 모두 제거하는 최소한의 Δu 를 결정하는 것이 문제다. 먼저 파라미터 벡터 u_w 하나를 제거하는 경우를 보자. u_w 를 제거하는 가장 쉬운 방법은 u_w 에서 공차 범위의 경계까지 가장

가까운 축을 이동 축으로 선택해서 그 축을 따라 공차 범위를 이동시키는 것이다. 공차 경계까지의 거리만큼 이동을 시키면 u_w 는 자연스럽게 범위 안에서 제거된다. 하지만 이 방법은 사실상 효율적이지 못하다. 왜냐하면 u_w 의 주위에도 같은 결함을 가지는 파라미터 벡터들이 존재할 가능성이 높기 때문이다. 이러한 파라미터 벡터들은 u_w 가 제거된 새로운 공차 범위 내부에 여전히 존재할 수 있다. 따라서 u_w 주위의 결함 있는 벡터들까지도 같이 제거하는 것이 훨씬 효율적이다. 이를 위해 u_w 대신에 u_0 와 u_w 를 연결하는 직선 상에서 결함이 처음으로 나타나기 시작하는 점 u_c 를 찾아서 이 점을 제거하는 방법을 택했다(Fig. 5b). 이 점을 전이점이라 부르겠다. 전이점을 찾는 데는 일반적인 이진 탐색법(binary search)을 사용하였다. 먼저 중간점 $u_{mid}=(u_w+u_0)/2$ 를 u_0 와 비교하여 결함이 있는지 확인하고, 결함이 없다면 구간 $[u_{mid}, u_w]$ 를 탐색하고, 결함이 있다면 $[u_0, u_{mid}]$ 를 계속 탐색한다. 구간 길이가 ϵ 보다 작으면 탐색을 멈추고 이 때의 중간점을 전이점으로 출력한다. 논문의 예제들은 $\epsilon=10^{-5}$ 값을 사용하였다.

다음은 u_c 를 가지고 이동 벡터 Δu 를 구하는 것을 보자. 먼저 u_c 에서 각 변형 축에 수직인 공차 경계면까지의 거리 $d[i]$ 를 구한다:

$$d[i] = \text{sign}(u_c[i]-u_0[i])(u_c[i]-u_0[i]+\delta_i) \quad (3)$$

u_c 를 공차 범위 밖으로 보내기 위한 최소 이동 방향은 $k = \arg \min(|d[i]|)$ 이 되고, 이동 거리는 $d[k]$ 가 된다. 이를 정리하면, 최적의 이동 벡터는 다음과 같이 된다.

$$\Delta u[i] = \begin{cases} d[i] & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

여러 개의 파라미터 벡터들을 동시에 제거하는 이동 벡터 Δu 는 위의 방법을 차례로 각 벡터마다 적용하여 구할 수 있다. 편의상 아래첨자를 제거 리스트에 있는 파라미터 벡터들의 인덱스로 사용하고, 특별히 0은 명목 설계 점으로 사용하겠다. 먼저 Δu 는 0으로 초기화한다. 그리고, 각 파라미터 벡터에 대한 전이점 $u_{c,i}$ 를 구하고 공차 경계까지의 거리 벡터 $d_{c,i}[j]$ 를 계산한다. 다음은 k_i 를 찾아서 다음과 같이 수정한다.

$$\Delta u[k_i] = \max_{|j|} (d_{c,i}[k_i], \Delta u[k_i]) \quad (5)$$

$\max_{|j|}$ 은 두 값의 절대값을 비교하여 큰 값을 돌려주는 연산이다.

그런데, 위 방법으로 구해진 이동 벡터는 파라미터

벡터들을 처리하는 순서에 따라 다른 결과를 낳게 된다. 그 이유는 한 파라미터 벡터를 제거할 때 동시에 제거되는 다른 벡터들이 생기기 때문인데, 예를 들어 두 파라미터 벡터 $u_{c,1}$ 와 $u_{c,2}$ 가 있고 각각 거리 벡터 $d_1=(3,2)$ 와 $d_2=(3,3)$ 를 가진다고 하자. $u_{c,2}$ 을 먼저 처리하면 이동 벡터는 $\Delta u=(3,0)$ 이 되고, 이 때 $u_{c,1}$ 은 자동적으로 제거된다. 이와 반대로 처리하면, $u_{c,1}$ 을 먼저 제거하기 위해 $\Delta u=(0,2)$ 가 되고, 다음으로 $u_{c,2}$ 를 제거하면 최종적으로 $\Delta u=(3,2)$ 가 된다. 따라서, $u_{c,2}$ 를 먼저 처리하는 것이 최적의 결과를 낳게 된다. 이 순서는 $d_i[k_i]$ 값이 감소하는 순서와 일치한다. 위의 경우를 보면,

$$d_1[k_1] = d_1[2] = 2 < d_2[k_2] = d_2[1] = 3$$

이기 되어 $u_{c,2}$, $u_{c,1}$ 의 순서가 된다. 본 연구에서는 이 순서를 이용하여 제거 리스트의 벡터들을 정렬시킨 후 이동 벡터를 구하였다.

Δu 를 업데이트하는 과정에서, $d_i[k_i]$ 의 부호가 $\Delta u[k_i]$ 와 다른 모순된 상황이 발생할 수 있다. 이러한 경우 $\Delta u[k_i]$ 를 $d_i[k_i]$ 로 단순히 치환하면 이전에 제거된 파라미터 벡터가 다시 공차 범위로 들어오게 되는 상황이 발생한다. 따라서, 이런 상황에서는 k_i 대신, 다음으로 작은 이동 축을 선택하여 이동 벡터를 업데이트시킨다. 만일 모든 이동 축들에서 모순 상황이 발생하면 공차 구간을 이동시키는 것만으로는 $u_{c,i}$ 를 제거할 수 없기 때문에 마지막으로 공차 한계를 축소시킨다.

5. 실험 결과

우리는 공차 설계 알고리즘을 세 가지의 고차대우를 가지는 기구들에 적용해 봤다. 알고리즘은 실제로 여러 개의 기계 요소를 가지는 기구에도 적용할 수 있다. 이 공차 설계 알고리즘은 각 예제에 대해 결함 변형이 발견되지 않을 때까지 반복해서 적용되었다. 공차 설계를 통한 얻어진 결과의 개선 여부는 접촉역의 중복이 해소되는 것으로 확인하였다.

5.1 제네바 페어

첫번째 예제는 제네바 페어로 접촉면의 모양과 회전축의 위치를 26개의 설계 파라미터로 사용하여 정의하였고, 각 파라미터의 공차 한계는 ± 0.018 cm로 설정하였다(Fig. 1a). 제네바 페어의 접촉역을 구하면 Fig. 3(a)에서 볼 수 있듯이 자유 공간의 위 아래 접

촉역이 서로 중복되는 것을 발견할 수 있다. 따라서, 공차 범위 안에 결함 변형들이 있음을 예상할 수 있다. 공차 설계 알고리즘을 처음 실행시키면 컨피규레이션 공간의 접촉역으로 부터 44개의 파라미터 벡터들이 얻어지고, 이 벡터들을 가지고 파라미터 조합을 하면 137개의 파라미터 벡터들이 얻어져 검사 리스트에 저장되었다. 이 중에서 21개의 파라미터 벡터들이 결함 변형인 경로 판단되어 제거 리스트에 저장되고, 이 점들을 제거하기 위해 명목 설계의 파라미터 값들을 업데이트하였다. 제거 리스트의 크기는 두 번째 실행에서는 5개로 줄게 되고, 그 다음에는 2개, 1개로 점차적으로 줄어들어 9번째 실행 후에 0이 되었다. 초기 설계와 최종 결과의 차이의 측정치는 최대 파라미터의 차이값을 이용하였다:

$$e = \max_i |u_0'[i] - u_0[i]| \tag{6}$$

제네바 페어에 대한 최종 결과의 차이 e 는 0.0032224가 되었다. 이 값은 공차 한계 0.018과 비교하면 상당히 작은 변화라고 할 수 있다. 원래 중복되었던 접촉역은 최종 결과에서는 Fig. 3(b)에서 볼 수 있듯이 중복이 거의 해소된 것을 확인할 수 있다. 실제로는 약간의 중복이 여전히 남아 있는데 이것은 접촉역 계산의 오차(overestimation)에 의한 것으로 무시할 수 있다.

5.2 광학 필터 캠 페어

두 번째 예제(Fig. 6)는 이스라엘 항공 산업에서 개발한 광학 필터 메카니즘의 일부로 세 쌍의 캠 페어로 구성되어 있다. 세 개의 캠들은 하나의 회전축에 90도씩 어긋나게 배열되어 있고, 세 개의 종질(follower)들은 각각 독립적으로 회전할 수 있도록 프레임에 핀 이름으로 연결되어 있다. 캠이 반시계 방향

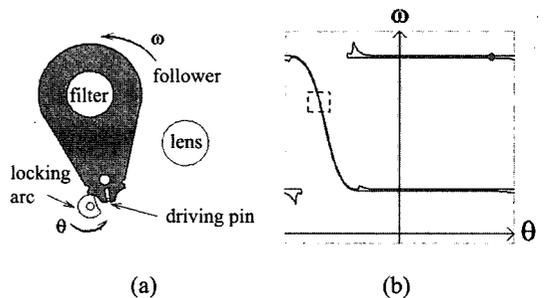


Fig. 6. (a) Cam/follower pair for an optical filter developed by Israel Aircraft Industries, (b) configuration space.

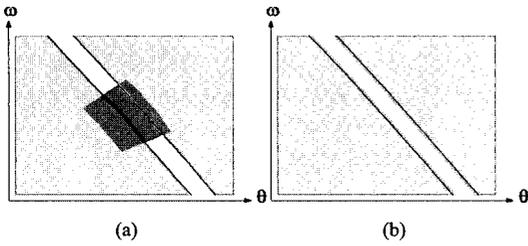


Fig. 7. (a) Detail of contact zone marked as a dashed box in Fig. 6(b), (b) contact zone of modified design.

으로 회전하면 캠에 붙어있는 핀이 종결의 슬롯으로 들어가면서 종결을 시계 방향으로 일정한 각도만큼 회전시키고, 이때 종결에 올려져 있는 광학 필터가 렌즈를 덮게 된다. 캠이 계속 회전하면서 핀이 슬롯을 떠나게 되면 캠의 마찰 면이 종결의 오른쪽 오목면과 접하면서 종결의 위치를 고정시켜 필터가 렌즈를 계속 덮고 있게 해준다. 캠을 반대 방향으로 회전시키면 종결을 반시계 방향으로 회전시키면서 필터를 원래 위치로 돌려 준다. 캠들이 90도씩 어긋나게 배열되어 있기 때문에, 필터들의 움직임은 약간의 시간차를 두고 일어나게 된다.

캠 페어의 접촉면의 모양과 회전축은 모두 17개의 파라미터들을 가지고 정의되었고, 각 파라미터들의 공차 한계는 $\pm 0.002 \text{ cm}$ 로 설정되었다. 공차 설계 알고리즘을 실행시키면, 접촉역으로부터 77개의 파라미터 벡터들이 얻어지고, 이 점들을 파라미터 조합하면 700개의 벡터들로 확장되었다. 이 중에서 결함 변형으로 판단되는 것들은 첫번째 실행에서는 146개가 되고, 알고리즘을 반복함에 따라 점차적으로 줄어들어 19번째 실행에서 결함 변형이 모두 사라졌다. 최종 결과의 접촉역을 초기 설계의 접촉역(Fig. 7a)과 비교해 보면 F중복되어 있던 부분이 사라진 것을 확인할 수 있다(Fig. 7b). 최종 결과의 최대 파라미터 차이 e 는 0.002709 cm 으로 공차 한계를 넘어간다. 따라서, 이 결과를 그대로 받아들일지는 사용자가 설계 목표하고의 차이를 보고 판단하여야 한다.

5.3 왕복 운동 캠 페어

마지막 예제는 왕복 운동 캠 페어이다. 캠은 Fig. 8(a)에서와 같이 세 가락의 긴 돌기가 120도 간격으로 회전 축을 중심으로 배열되어 있는 모양을 가지고 있다. 각각의 핑거는 종결의 기울어진 내부 면을 번갈아 밀어 주어 종결의 좌우 운동이 반복해서 이루어지도록 해준다. Fig. 8(b)는 왕복 운동 캠 페어의 컨피규레이션 공간을 나타내는데, 가로 x 방향의 θ 축은 캠의

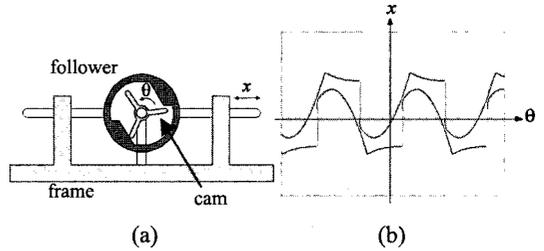


Fig. 9. (a) A reciprocating cam pair, (b) configuration space.

회전 각을 나타내고 세로 방향의 x 축은 프레임 중심으로부터 종결의 수평 변위를 나타낸다. 그림을 보면 자유 공간의 모양이 위와 아래를 향하는 컵이 반복되는 모양을 하고 있는 걸 알 수 있다. 각각의 컵은 캠의 한 핑거가 종결을 한쪽 방향으로 밀어주는 구간에 해당된다.

왕복 운동 캠 페어의 접촉면과 운동축은 11개의 파라미터를 가지고 정의되었고, 각 파라미터들의 공차 한계는 $+0.02 \text{ cm}$ 로 주어졌다. 초기 설계치를 가지고 공차 분석을 위해 접촉역을 계산해 보면 자유 공간의 두 컵을 연결해 주는 좁은 연결통로에서 위아래의 접촉역이 서로 약간씩 중복되는 것을 볼 수 있다(Fig. 9a). 이것은 좁은 연결통로가 막히는 결함 변형이 발생할 수 있다는 것이다. 여기에 공차 설계 알고리즘을 실행시키면 파라미터 조합 후의 검사 리스트에는 45~57개의 파라미터 점들이 구해지고, 여기서 결함 변형들은 초기에는 15개가 나오기 점차적으로 줄어서 9번의 실행 후에는 모두 제거되었다. 최종 결과의 접촉역은 Fig. 9(b)에서 보여주는데 접촉역의 중복이 사라진 걸 발견할 수 있다. 최종 결과의 최대 파라미터 차이 e 는 0.016054 로 공차 한계와 비교했을 때 비교적 작은 차이라고 할 수 있다.

위의 세가지 예제를 통해 본 알고리즘이 고차대수들로 구성된 기구의 결함 변형들을 공차 한계를 바꾸지 않고도 효과적으로 제거할 수 있음을 보였다.

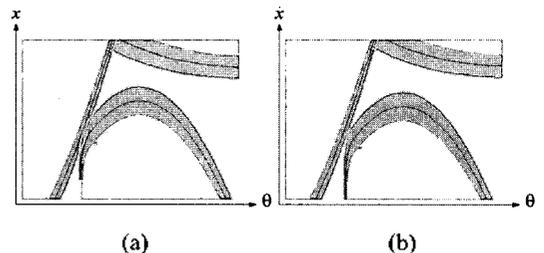


Fig. 9. (a) Detail of contact zone, (b) contact zone of modified design

6. 결 론

본 연구에서는 고차대우로 구성된 기구의 공차 설계를 자동으로 해주는 알고리즘을 처음 제안하였다. 이 알고리즘은 세 단계로 구성되어 있다. 먼저, 기구의 공차 구간에서 기구학적 변형이 최대가 되는 파라미터 벡터들을 접촉역 계산과 파라미터 조합을 이용해 찾는다. 다음은 각 파라미터 벡터들의 기구학적 운동을 명목 설계와 비교하여 결함 여부를 판단한다. 마지막으로 결함 변형에 해당하는 파라미터 벡터들을 설계 파라미터들의 명목값을 수정하거나 또는 공차 구간을 축소하여 제거한다. 이 알고리즘을 세 종류의 고차 대우들로 구성된 기구에 대해 적용하여 성공적으로 결함 변형들이 공차 구간에서 제거되는 것을 확인하였다.

앞으로의 연구 방향은 몇 가지로 생각해 볼 수 있다. 먼저 공차 구간의 축소나 명목 파라미터값을 바꿀 때 발생하는 가공과 생산 비용의 변화를 설계 변경에 반영하는 것이다. 경우에 따라 명목 파라미터값의 변경이 공차 구간 축소보다 비용을 상승시키거나 혹은 설계 목적에 부합되지 않은 결과를 낼 수 있다. 따라서 비용 상승 정도에 따라 공차 변경을 선택적으로 결정하도록 하는 것이 바람직하다. 다른 하나로 현재 접촉역의 계산과 컨퍼레이션 공간 매치는 자유도가 2인 고차대우에 대해서만 구현되어 있다. 따라서, 두 기계 요소의 동작이 자유도 3을 가지는 경우에는 현재 구현된 프로그램을 적용할 수 없다. 이러한 부분들은 3차원 컨퍼레이션 공간 상에서 해결해야 하는 문제로 더 높은 복잡도를 가지고 있기 때문에 새로운 구현 방법이 필요하다.

감사의 글

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성 지원 사업의 연구 결과로 수행되었습니다.

참고문헌

1. Hong-Chao Zhang, *Advanced Tolerancing Techniques*, Wiley Series in Engineering Design and

Automation, 1997.

2. Clement, A., Riviere, A., Serre, P. and Valade, C., "The ttrs: 13 Constraints for Dimensioning and Tolerancing", *Proceedings of the Fifth CIRP International Seminar on Computer-Aided Tolerancing*, Toronto, 1997.
3. Whitney, D., Gilbert, O. and Marek, M., "Representation of Geometric Variations Using Matrix Transforms for Statistical Tolerance Analysis", *Research Engineering Design*, Vol. 6, No. 4, pp. 191-210, 1994.
4. Arther, G. E., *Modern Kinematics: Developments in the Last Forty Years*, New York, Wiley, 1993.
5. Glancy, C. G. and Chase, K. W., "A Second-order Method for Assembly Tolerance Analysis", *Proceedings of the ASME Design Automation Conference*, 1999.
6. Elisha Sacks and Joskowicz, L., "Parametric Kinematic Tolerance Analysis of Planar Mechanisms", *Computer Aided Design*, Vol. 29, No. 5, pp. 333-342, 1997.
7. Elisha Sacks, L. Joskowicz, R. Schultheiss and U. Hinze, "Computer-assisted Kinematic Tolerance Analysis of a Gear Selector Mechanism with the Configuration Space Method", *Proc. of 25th ASME Design Automation Conference*, Las Vegas, 1999.
8. Joskowicz, L. and Elisha Sacks, "Computational Kinematics", *Artificial Intelligence*, Vol. 51, No. 1-3, pp. 381-416, 1991.
9. Min-Ho Kyung and Elisha Sacks, "Nonlinear Kinematic Tolerance Analysis of Planar Mechanical Systems", *Computer-Aided Design*, Vol. 35, pp. 901-911, 2003.
10. Min-Ho Kyung and Elisha Sacks, "Parametric Synthesis of Higher Kinematic Pairs", *Computer-Aided Design*, Vol. 35, pp. 567-575, 2003.



경민호

1993년 포항공과대학교 전자계산학과 학사
 1995년 포항공과대학교 전자계산학과 석사
 2001년 Purdue University, 전자계산학과 박사
 2002년 Purdue University, Post Doc.
 2002년~현재 이주대학교 미디어학부 조교수
 관심분야: Computer Graphics & Animation, Kinematic Analysis, Motion Planning, Tolerance Design