学術論文

多面体間の接触による拘束条件を幾何モデルから導出する一般的なアルゴリズム

比留川博久*　松井俊浩*　高瀬國克*

A General Algorithm for Deriving Constraint of Contact between Polyhedra from Geometric Model

Hirohisa HIRUKAWA　Toshihiro MATSUI　Kunikatsu TAKASE

When a body is under bilateral constraint, the constraint condition for the body is represented as homogeneous linear equations of velocity and force and their solution set for velocity is an orthogonal complement of that for force. In case of unilateral constraint, the condition is represented as homogeneous linear inequalities generally and their solution set for velocity is a dual polyhedral convex cone of that for force.

However, no general method has been proposed yet for deriving such constraint condition automatically, when a body contacts another body in an arbitrary state.

In this paper, we propose a general algorithm for deriving constraint of contact between bodies from geometric model of a body and working environment constraining it. The algorithm has been implemented in EusLisp. The algorithm is able to be applied in the case that a polyhedron with an arbitrary shape contacts another one in an arbitrary state and therefore very general. We believe it becomes a key technology for the analysis, planning and control of constrained motion and the assembly sequence planning.

Key Words: Manipulation, constraint, Contact, Convex Cone, Geometric Model

1. はじめに

物体が自由空間にあるとすると，その物体には任意の微小変位が可能であるが，他の物体に対して力を発生することは出来ない．逆に，物体が十分重い物体に固定されている場合は，その物体は変位することは出来ないが，固定している物体に対して任意の力を発生することが可能である．この両極端の間の状態として，両側拘束された物体の場合を考えると，拘束条件は微小変位と力に関する同次方程式になり，その解である許される微小変位の集合と発生可能な力の集合は直交補空間になる[1]．また，片側拘束された物体の運動についても，瞬間中心に着目した解析[2]や，許される微小変位と発生可能な力の集合に関する研究が行なわれている[3,4]．片側拘束の場合，物体に対する拘束条件は同次不等式になり，許される微小変位と発生可能な力の集合は双対凸多面錐になる．

しかしながら，3次元の物体と物体が任意の接触状態にある場合に，これらの物体に対する拘束条件を求めることの出来る一般的なアルゴリズムは従来提案されていない．この方法の確立は，拘束状態にある物体に許される微小変位の集合と発生可能な力の集合を求め，これに基づいて動作を解析，計画，制御したり，組立手順を計画したりするために不可欠である．

本稿では，作業対象の物体とそれを拘束している作業環境の幾何モデルのみが与えられていると仮定し，これから作業対象に対する拘束条件を求めるアルゴリズムを提案する[6]．作業対象と作業環境の形状は多面体に限定し，作業環境は任意の力を加えても動かず，これらの間に摩擦はないものとする．

原稿受付　1990年9月10日
* 電子技術総合研究所

微小変位に対する拘束条件を求めることが出来れば，発生可能な力に対する条件は静力学的な考察により求まるので，本稿では微小変位に対する拘束条件についてのみ考察する．

2. 多面体間の接触による拘束

2.1 1点の接触による拘束

多面体と多面体の接触点の集合は，一般には点・線分・多角形およびこれらの組合せになる．本節では，全ての接触点から受ける拘束条件を求めるための準備として，このうちの1点に注目し，この点での接触による拘束条件を求める方法を示す．

2つの多面体は点 X_c で接触しているとし，それぞれの多面体の内点の集合を A, B とする．以下の議論では，点 X_c の近傍に含まれる集合 A, B の部分集合，

$$U_A(X_c)=\{X|X\in A, \rho(X, X_c)<\delta\} \qquad (1)$$

$$U_B(X_c)=\{X|X\in B, \rho(X, X_c)<\delta\} \qquad (2)$$

についてのみ考え，全てを点 X_c を原点とする座標系で記述することにする．ここに，$\rho(X, Y)$ は点 X と Y の距離，δ は微小量である．

多面体の面の内点で点 X_c に接触している場合，点 X_c の近傍におけるその多面体の内点の集合 $U(X_c)$ は，

$$U(X_c)=\{X|F_i^TX>0,\ \rho(X, X_c)<\delta\} \qquad (3)$$

と表せる．ここに，F_i は 3×1 である．稜とその両側の2つの面およびこれらに囲まれる内点の集合は，2面角を構成する．多面体の稜で点 X_c 接触している場合，この2面角の平面角が π より小さいときは，

$$U(X_c)=\bigcap_{j=1}^{2}\{X|F_j^TX>0,\ \rho(X, X_c)<\delta\} \qquad (4)$$

と，$U(X_c)$ は2面角の2つの面を境界とする半空間の積 (and) になる．平面角が π より大きいときは，

$$U(X_c)=\bigcup_{i=1}^{2}\{X|F_i^TX>0,\ \rho(X, X_c)<\delta\} \qquad (5)$$

と，和 (or) になる．頂点は，これらの2面角の辺の交点で，この場合，$U(X_c)$ は一般に次の様に表せる．

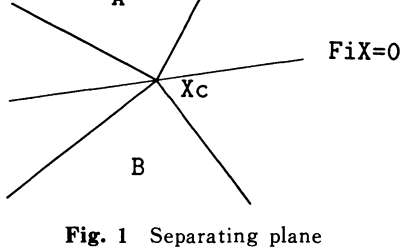

Fig. 1 Separating plane

$$U(X_c)=\bigcup_{i=1}^{n}\bigcap_{j=1}^{m(i)}\{X|F_{ij}^TX>0,\ \rho(X, X_c)<\delta\} \qquad (6)$$

まず，前述した2面角の平面角が全て π より小さい場合について考える．この場合，点 X_c の近傍における各多面体の内点の集合は，次の様にかける．

$$U(X_c)=\bigcap_{j=1}^{m}\{X|F_j^TX>0,\ \rho(X, X_c)<\delta\} \qquad (7)$$

ここに，面の内点で接触している場合 $m=1$，稜の場合 $m=2$，頂点の場合 $m\geq3$ である．ここで，次の定義をする．

定義1 ある平面 $F_{si}^TX=0$ が次の条件に満たすとき，2つの多面体は点 X_c の近傍において線形的に分離可能 (linearly separable) であるといい，この平面を分離面という．(**Fig. 1** 参照)

$$\forall X\in U_A(X_c)\ ;\ F_{si}^TX>0 \text{ and}$$
$$\forall X\in U_B(X_c)\ ;\ F_{si}^TX<0 \qquad (8)$$

ここで，不等号の向きがどちらも逆になっても同じ様に定義する．定義から，分離面は接触点 X_c を含む必要がある．

以下，一般性を失うことなしに，A の多面体を作業対象の物体，B の多面体を固定された作業環境と考えて，A の物体に対する拘束条件について考察する．また，分離面 $F_{si}^TX=0$ の係数ベクトルの符号は，(8) 式が成り立つ様に決めるものとする．さらに，点 X_c の近傍で2つの多面体の干渉を引き起こさない A の多面体の微小変位の集合を $\mathrm{SAFE}(A, X_c)$，干渉を引き起こす微小変位の集合を $\mathrm{COL}(A, X_c)$ と表記することにする．本節では，1点の接触による拘束を考え，座標系の原点を接触点にとっているので，拘束される微小変位としては並進要素のみを考えれば良い．従って，以下本節では微小変位として微小並進のみを考察の対象とする．

以上の準備に基づいて，次の定理が成立する．

定理 1 ある面 $F_{si}^TX=0$ が2つの多面体を分離するとき，その平面の法線ベクトルとの内積が正である任意の微小変位 ΔX は，点 X_c の近傍において2つの多面体の干渉を引き起こさない．すなわち，A の多面体が動くとして，

$$\forall\Delta X\ ;\ F_{si}^T\Delta X\geq0 \Longrightarrow \Delta X\in\mathrm{SAFE}(A, X_c) \qquad (9)$$

である．逆に，(9) 式を満たす面は，点 X_c の近傍において2つの多面体を分離する面だけである．

(証明) **付録参照**．

分離面が複数あるときは，集合 $\mathrm{SAFE}(A, X_c)$ はこれらの面によって決まる集合の和 (or) になる．$\mathrm{SAFE}(A, X_c)$ を分離面 $F_{sj}^TX=0$ により，

$$\mathrm{SAFE}(A, X_c)=\bigcup_{j=1}^{M}\{\Delta X|F_{sj}^T\Delta X\geq0\} \qquad (10)$$

と表現するのに必要で，かつ，最小の数の面の集合の要素を SAFE(A, X_c) の境界を決める面と呼ぶことにすると，次の定理が成り立つ．

定理 2　2つの多面体のいずれかの境界上にある任意の2点を X_1, X_2 とする．このとき，同一直線上にない3点 X_c, X_1, X_2 を含み，かつ，2つの多面体を分離する面は，集合 SAFE(A, X_c) の境界を決める面である．逆に，ある分離面が SAFE(A, X_c) の境界を決める面であるときは，このような3点を含む．

（証明）　**付録**参照．

次に問題なのは，点 X_c 以外にどのような2点の組合せを調べれば，SAFE(A, X_c) の境界を決める面をもれなく求めることが出来るか，ということである．これに関しては，次の定理が成り立つ．

定理 3　集合 SAFE(A, X_c) の境界を決める面の集合は，次の様な境界の点を含む分離面の集合である．すなわち，多面体のどちらかが点 X_c に面の内点で接触している場合は，その面が唯一の分離面になる．この場合，点 X_c 以外の境界の点を2点選ぶ必要はない．点 X_c に稜の内点で接触している場合は，分離面はこの稜を含む必要があり，境界上のもう1点の候補としては，2つの面の適当な内点を各々1点ずつ選ぶ．点 X_c に接触しているのが頂点の場合は，境界の2点の候補としては点 X_c を端点とする m 本の稜上に，各々点 X_c 以外の点を1点ずつ選ぶ．

（証明）　**付録**参照．

さらに，前述した2面角の平面角に π より大きいものがある場合，すなわち，少なくともどちらかの近傍が(6)式で表される様な一般の場合について考える．各近傍が

$$U_A(X_c)=\bigcup_{i_A=1}^{n_A}\bigcap_{j_A=1}^{m_A(i_A)}\{X|F_{Aij}{}^T X>0,\ \rho(X, X_c)<\delta\} \tag{11}$$

$$U_B(X_c)=\bigcup_{i_B=1}^{n_B}\bigcap_{j_B=1}^{m_B(i_B)}\{X|F_{Bij}{}^T X>0,\ \rho(X, X_c)>\delta\} \tag{12}$$

と凸集合の和で表されるとし，A の i_A 番目の部分凸集合と B の i_B 番目の部分凸集合の組合せに対して前述した処理により得られる集合を SAFE$(A_{i_A}, X_c)_{i_B}$ とすると，これは，

$$\mathrm{SAFE}(A_{i_A}X_c)_{i_B}=\bigcup_{j=1}^{M(i_A,i_B)}\{\varDelta X|F'^T_{si_Ai_Bj}\varDelta X\geq 0\} \tag{13}$$

と表せる．ここに，$F'^T_{si_Ai_Bj}X=0$ は，集合 SAFE$(A_{i_A}, X_c)_{i_B}$ の境界を決める面である．これらの積集合が SAFE(A, X_c) になる．すなわち，

$$\mathrm{SAFE}(A, X_c)=\bigcap_{i_A=1}^{n_A}\bigcap_{i_B=1}^{n_B}\mathrm{SAFE}(A_{i_A}, X_c)_{i_B}$$

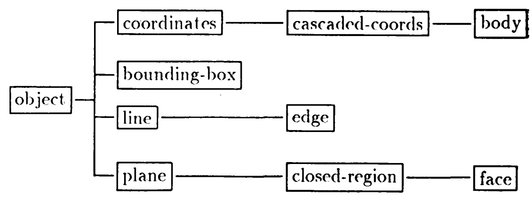

Fig. 2　Hierarchy of geometrical object

$$=\bigcap_{i_A=1}^{n_A}\bigcap_{i_B=1}^{n_B}\bigcup_{j=1}^{M(i_A,i_B)}\{\varDelta X|F'^T_{si_Ai_Bj}\varDelta X\geq 0\} \tag{14}$$

が求める集合である．表現を簡単にするため，

$$F_{(si_A+n_A\times(i_B-1))j}\overset{\mathrm{def}}{=}F'_{si_Ai_Bj}$$

とおき直すと，1点が接触しているときに，点 X_c の近傍で干渉を引き起こさない A の微小変位の集合は，最終的には

$$\mathrm{SAFE}(A, X_c)=\bigcap_{i=1}^{N}\bigcup_{j=1}^{M(i)}\{\varDelta X|F_{sij}{}^T\varDelta X\geq 0\} \tag{15}$$

で与えられる．ここに，$N=n_An_B$ である．

2.2　多点の接触による拘束

前節の結果に基づいて，本節では接触点の集合が点・線分・多角形およびこれらの組合せになる場合の拘束を求める方法を示す．各接触点 X_k における拘束条件を，(15)式を簡単にかくことにより，X_k に原点を持つ座標系において

$$\bigcap_{i=1}^{N(k)}\bigcup_{j=1}^{M(k,i)}F_{skij}{}^T\varDelta X_k\geq 0 \tag{16}$$

の様に表現することにする．この座標系の位置と姿勢が，被拘束物体に固定した物体座標系から見て，

$$T=\begin{pmatrix}E_{kx} & E_{ky} & E_{kz} & P_k\\ 0 & 0 & 0 & 1\end{pmatrix} \tag{17}$$

なる同次変換で与えられるとすると，この拘束条件は物体座標系では，

$$\bigcap_{i=1}^{N(k)}\bigcap_{j=1}^{M(k,i)}F_{skij}{}^T J_k\begin{pmatrix}\varDelta X\\ \varOmega\end{pmatrix}\geq 0 \tag{18}$$

と表される．ここに，J_k は，

$$\varDelta X_k=J_k\begin{pmatrix}\varDelta X\\ \varOmega\end{pmatrix}=\begin{pmatrix}E_{kx}{}^T & (P_k\times E_{kx})^T\\ E_{ky}{}^T & (P_k\times E_{ky})^T\\ E_{kz}{}^T & (P_k\times E_{kz})^T\end{pmatrix}\begin{pmatrix}\varDelta X\\ \varOmega\end{pmatrix} \tag{19}$$

と，X_k の座標系で表された微小変位 $\varDelta X_k$ と物体座標系で表された微小変位・回転 $\varDelta X, \varOmega$ との関係を与えるヤコビ行列である．(18)式により，複数の接触点における拘束を物体座標系で表現することが出来る．接触点の集合が線分や多角形になる場合については，次の定理が成り立つ．

定理 4　1点の接触による拘束が条件式に和演算を含ま

ない場合，接触点の集合が線分になるときに受ける拘束は，この線分の両端点から受ける拘束と等価である．従って，接触点の集合が多角形になるときに受ける拘束は，この多角形の凸包の全ての頂点に受ける拘束と等価になる．

(証明) **付録**参照．

この定理により扱えないのは，近傍が（4）式で表される様な稜が線分で接触している場合のみである．そのような場合，拘束条件は線形の不等式にはならない．この場合を除くと，接触点の集合が点・線分・多角形およびこれらの組合せになる一般的な場合の拘束条件は，次式で与えられる．

$$\bigcap_{k=1}^{L}\bigcap_{l=1}^{N(k)}\bigcup_{j=1}^{M(k,l)} F_{sklj}{}^{T} J_k \begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} \geq 0 \qquad (20)$$

ここに，Lは等価拘束点の数である．

3. 拘束条件導出アルゴリズム

本節では，前節で述べた諸定理に基づいて，被拘束物体と拘束物体の幾何モデルと位置から，拘束条件を導出するアルゴリズムを示す．本アルゴリズムは，電総研で開発された，単一継承型のオブジェクト指向に基づくCommon Lispのサブセット，EusLisp[5]を用いてインプリメントされている．オブジェクトとして定義されている幾何モデルの要素の階層構造を **Fig. 2** に示す．ここに，coordinatesは座標系あるいは座標変換を表すクラス，cascaded-coordsは座標系の親子関係を管理するクラス，bodyは多面体，bounding-boxは稜がx, y, z軸に平行である直方体のクラスである．lineは直線，edgeは稜，planeは境界のない平面，closed-regionは稜に囲まれた平面の領域，faceは穴を含む多面体の面のクラスである．主なオブジェクトのスロット構成を，**Table 1** に示す．ここで，constrained-pointは被拘束点のクラスで，被拘束点の座標，被拘束点における拘束物体の面の一つ，被拘束点における自分自身の近傍，被拘束点に対する拘束条件の4つのスロットを持ち，拘束条件の解析の際に用いる．以下に示すメソッド中では，各クラスのインスタンスの作成は

(instance クラス名)

と表記する．また，selfは自分自身のオブジェクトを指す変数名として使う．さらに，selfのスロットは単にスロット名で表し，他のオブジェクトのスロットは，

オブジェクト名．スロット名

という記法で表す．メソッドの起動は，

(send オブジェクト名 :メソッド名 引数)

という形式で行なう．

アルゴリズム全体の構造を **Fig. 3** に示す．アルゴリズムのトップレベルは，bodyのメソッド:constraintである．このメソッドから呼ばれる，クラスbodyのメソッド:contact-verticesと:contact-edgesにより始

Table 1 Slots of objects

body slot name	*body* value	*face* slot name	*face* value	*edge* slot name	*edge* value
worldcoords	coordinates	normal,distance	face equation	pvert	starting vertex
box	minimal box	edges	edge list	nvert	ending vertex
faces	face list	vertices	vertex list	pface	left face
edges	edge list	convexp	convex flag	nface	right face
vertices	vertices list				
convexp	convex flag				
constraint	constraints				

constrained-point

slot name	value
position	position of constrained point
hisface	one of faces constraining body at constrained point
myneighborhood	self's neighbors at constrained point
condition	constraint conditions

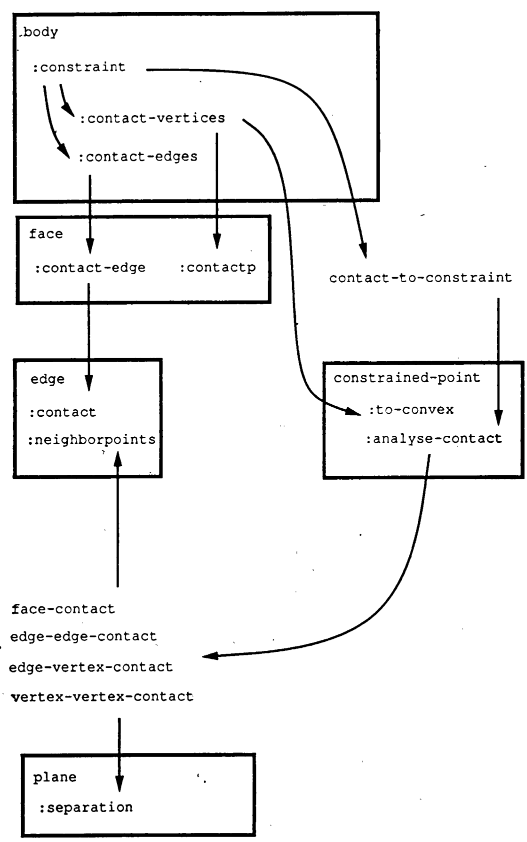

Fig. 3 Structure of the Algorithm

:constraint (*b*)

1. **if** (send *box* **:contact** *b*'s *box*) is empty **then**
   self is not in contact with *b*. Terminate the algorithm
2. *mycontact* ← (send *self* **:contact-vertices** *b*) + (send *self* **:contact-edges** *b*)
3. *hiscontact* ← (send *b* **:contact-vertices** *self*) + (send *b* **:contact-edges** *self*)
4. *constraint* ← **contact-to-constraint** (*mycontact hiscontact*)

Fig. 4 Body's method : constraint

まっている処理は，等価拘束点の位置を求めるための処理である．これに対して，関数 contact-to-constraint により始まっている処理は，前節の諸定理に基づいて，各等価拘束点における拘束条件を求める処理で，クラス constrained-point のメソッド：analyse-contact により拘束点が分類され，最終的にはクラス plane のメソッド：separation による，ある平面が分離面になるかどうかの判定に帰着する．

まず，これらのメソッドと関数の機能を箇条書にする．

body のメソッド：constraint　多面体 self が多面体 b から受ける拘束の，等価拘束点の位置と拘束条件を求める．(17) 式と (16) 式を求めることに対応している．これから，(19)式を用いてヤコビ行列を求めれば，最終的な拘束条件式である (20) 式が求まる．

body のメソッド：contact-vertices　多面体 b に接触している self の頂点，対応する b の接触面，凸集合の和集合で表現した self の近傍を求める．

body のメソッド：contact-edges　多面体 b の稜に接触している self の稜の内点，対応する b の接触面，半空間の積または和集合で表した self 近傍を求める．ここで，多面体 b に接触している self の稜の内点ではなく，b の稜に接触しているものを求めていることに注意されたい．

bounding-box のメソッド：contact　self と box が接触しているかどうかをチェックする．接触している場合は，両者の共通箱を，していない場合は nil を返す．

face のメソッド：contactp　点 p が self に接触しているかどうかをチェックする．接触しているときは inside，境界上のときは border，接触していないときは outside を返す．

face のメソッド：contact-edge　稜 e1 と self が平行でないとき，e1 と self のどれかの稜との接触点を返す．

edge のメソッド：contact　self の内点と稜 e との接触点を求める．接触していない場合は，nil を返す．

constrained-point のメソッド：to-convex　等価拘束点の凸でない近傍を，凸集合の和集合に変換する．

関数 contact-to-constraint　各等価拘束点における拘束条件を求める．

constrained-point のメソッド：analyse-contact　自分の接触点とその近傍を表す self と，相手の接触点とその近傍を表す hispoint から，自分に対する接触条件を求める．

関数 face-contact　近傍が半平面の和で表される場合の拘束条件を求める．

関数 edge-edge-contact　稜の内点どうしで接触している場合の拘束条件を求める．

関数 edge-vertex-contact　稜の内点と頂点で接触している場合の拘束条件を求める．

関数 vertex-vertex-contact　頂点と頂点で接触している場合の拘束条件を求める．

edge のメソッド：neighborpoints　self の近傍を表す次の 4 点のリストを返す．self 上の点で point とは異なる点，その点と point を挟んで反対側の点，self の pface 上の点，self の nface 上の点．

plane のメソッド：separation　面 self が mypoints と hispoints の分離面になるかどうかをチェックする．分離面になる場合は，mypoints 側に向く様に符合を合わせた，self の法線ベクトルを返す，分離面にならない場合は，nil を返す．

以下，主なメソッドと関数の詳細を説明する．

body のメソッド：constraint を **Fig. 4** に示す．ステップ 1 は，各々の多面体が含まれる最小の bounding-box どうしが接触しているかどうかのチェックで，接触していないときはアルゴリズムを終了する．ステップ 2 では，メソッド：contact-vertices により，多面体 b に接触している self の頂点，対応する b の接触面，凸集合の和集合で表現した self の近傍を求め，メソッド：contact-edges により，多面体 b の稜に接触している self の稜の内点，接触面，近傍を求めている．ステップ 3 は，同じ処理を self と b を逆にして行なっている．このような，self と b の頂点と稜の内点の接触状態を調べれば，(20) 式の等価拘束点は全て求まる．ただし，この方法だと接触領域が凹多角形になる場合は，冗長な条件が出てくるが，拘束条件全体では等価である．ステップ 4 では，関数 contact-to-constraint により，各々の等価拘束点における拘束条件を求める．ここで，ステップ 2 と 3 の処理の例を示す．**Fig. 5** は，直方体 A が穴 B により拘束されている例である．図中で，ステップ 2 では，5 番から 8 番までの点が：contact-vertices により求まり，1 番から 4 番までの点が：contact-edges により求まる．ステップ 3 では，1 番から 8 番までの点が全て：contact-vertices により求まる．**Fig. 6** は，1 番と 5 番の等価拘束点の近傍を凸集合の和集合に分解した例で

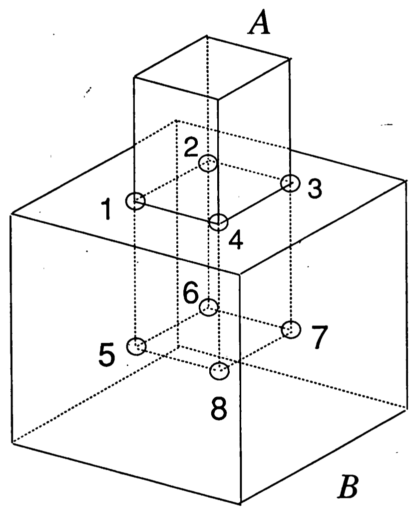

Fig. 5　Equivalent Contact Points

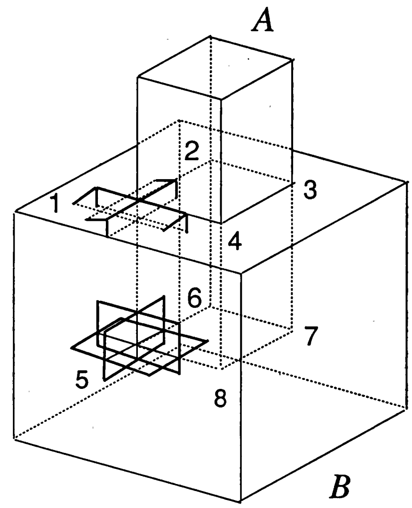

Fig. 6　Decomposition of neighborhood to Convex Set

ある．1番の頂点の近傍は2つの稜の近傍の和に，5番の頂点の近傍は3つの平面の近傍の和に分解される．

body のメソッド：contact-vertices を **Fig. 7** に示す．ステップ5では，b に接触している self の頂点を求め，これを position スロットの値とし，b の接触面を hisface スロットの値とする，クラス constrained-point のインスタンスを作成している．ステップ6では，self の接触頂点から，それが載る self の稜へのバックポインタを myneighborhood スロットにセットしている．ステップ7では，self の近傍が凸でない場合に，それを constrained-point のメソッド：to-convex により，凸集合の和集合に変換している．このメソッドの詳細は省くが，基本的考え方は，稜から面へのバックポインタを順に辿って行って，凸でない稜を見つけたらその両側の面で近傍を切断して，(7)式の様な近傍を新たに生成するというものである．

body のメソッド：contact-edges を **Fig. 8** に示す．ステップ5では，self の稜の内点が b の面に接触している場合に，face のメソッド：contact-edge を用いて，その接触点の位置と，その接触面を求めている．

関数 contact-to-constraint を **Fig. 9** に示す．この関数の引数は，クラス constrained-point のインスタンスのリストである．ステップ1では，b に接触している

```
:contact-vertices (b)
  1. cbox ← (send box :contact b'sbox)
  2. if cbox is empty then
       self is not in contact with b. Terminate the algorithm
  3. myvertices ← self's edges in cbox
  4. hisfaces ← b's faces in cbox
  5. for each vertex v of myvertices
       for each face f of hisfaces
          if (send f :contactp v) is not outside then
             p ← (instance constrained-point)
             p.position ← self's contacting vertex
             p.hisface ← one of b's contacting face
             conp ← conp + p
  6. for each element p of conp
       p.myneiborhood ← self's edges in contact with the vertex
  7. for each element p of conp
       (send p :to-convex)
  8. return conp
```

Fig. 7　Body's method : contact-vertices

```
:contact-edges (b)
  1. cbox ← (send box :contact b'box)
  2. if cbox is empty then
       self is not in contact with b. Terminate the algorithm
  3. myedges ← self's edges in cbox
  4. hisfaces ← b's faces in cbox
  5. for each edge e of myedges
       for each face f of hisfaces
          if (send f :contact-edge e) is not empty and not an element of conp then
          p ← (instance constrained-point)
          p.position ← contacting point on e
          p.hisface ← one of b's contacting face
             if plane angle of e's dihedral angle < π then
                p.myneighborhood ← ((contacting self's edge))
                conp ← conp + p
             else
                p.myneighborhood ← ((left face of contacting self's edge) (right face
                 of contacting self's edge))
                conp ← conp + p
  6. return conp
```

Fig. 8　Body's method : contact-edges

contact-to-constraint(*mycontact hiscontact*)

1. **while** *mycontact* is not empty **do**
   - *point* ← one element of *mycontact*
   - anotherpoint ← empty
   - **for each** element *hispoint* of *hiscontact*
     - **if** *point.position* = *hispoint.position* **then** *anotherpoint* ← *hispoint*
   - **if** *anotherpoint* is not empty **then**
     - *constraints* ← *constraints* + (send *point* **:analyse-contact** *anotherpoint*)
     - *hiscontact* ← *hiscontact* − *another* − *point*
   - **else**
     - *p* ← (instance constrained-point)
     - *p.position* ← *point.position*
     - *p.condition* ← normal vector of *point.hisface*
     - *constraints* ← *constraints* + *p*
   - *mycontact* ← *mycontact* − *point*
2. **while** *hiscontact* is not empty **do**
   - *point* ← one element of *hiscontact*
   - *p* ← (instance constrained-point)
   - *p.position* ← *point.position*
   - *p.condition* ← −1.0× normal vector of *point.hisface*
   - *constraints* ← *constraints* + *p*
   - *hiscontact* ← *hiscontact* − *point*
3. **return** *constraints*

Fig. 9 Function contact-to-constraint

self の頂点と稜，self に接触している b の頂点と稜に，同じ点があるかどうかを調べている．これらの等価拘束点に共通点がないのは，その接触点が頂点と面の内点の組合せの場合のみである．この場合，この面の法線ベクトルを self の方に向けたものが，この点における拘束条件になる．同じ点がある場合は，オブジェクト constrained-point のメソッド：analyse-contact で解析する．

クラス constraind-point のメソッド：analyse-contact を **Fig. 10** に示す．まず，このメソッドでは，それぞれの凸近傍の組合せが，面・稜・頂点のどのような組合せであるかを分類する．Fig. 6 の例だと，1 番から 4 番の点は稜と稜，5 番から 8 番の点は頂点と面の組合せになる．そして，定理 3 に基づいてそれぞれの組合せによる拘束条件を，関数 face-contact, edge-edge-contact, edge-vertex-contact, vertex-vertex-contact のいずれかを用いて求めている．

face-contact (*face sign*)

1. **return** (normal vector of *face* × *sign*)

Fig. 11 Function face-contact

edge-edge-contact (*position myedge hisedge*)

1. *mypoints* ← (send *myedge* **:neighborpoints** *position*)
2. *hispoints* ← (send *hisedge* **:neighborpoints** *position*)
3. *splane* ← plane including myedge and hisedge
4. **return** ((send *splane* **:separation** *mypoints hispoints*))

Fig. 12 Function edge-edge-contact

:analyse-contact(*hispoint*)

1. **for each** (*mine,his*) ← combinations of all convex subset of *self*'s neighborhood and that of *hispoint*'s neighborhood
   - **if** *mine* is face **then** *andcond* ← *andcond* + **face-contact**(*mine* −1.0)
   - **if** *his* is face **then** *andcond* ← *andcond* + **face-contact**(*his* 1.0)
   - **if** contact between edge and edge **then** *andcond* ← *andcond* + **edge-edge-contact**(*position mine his*)
   - **if** contact between edge and vertex **then** *andcond* ← *andcond* + **edge-vertex-contact**(*position mine his* 1.0)
   - **if** contact between vertex and edge**then** *andcond* ← *andcond* + **edge-vertex-contact**(*position his mine* −1.0)
   - **if** contact between vertex and vertex **then** *andcond* ← *andcond* + **vertex-vertex-contact**(*position mine his*)
2. *cpoint* ← (instance constrained-point)
3. *cpoint.position* ← *position*
4. *cpoint.condition* ← *andcond*
5. **return** *cpoint*

Fig. 10 Constrained-point's method : analyse-contact

関数 face-contact を **Fig. 11** に示す．近傍が半平面の和で表される場合は，それぞれの面が唯一の分離面なので，face が自身の面のときは sign を −1.0，相手の面のときは 1.0 にして，この面の法線ベクトルにかけたものを返す．

関数 edge-edge-contact を **Fig. 12** に示す．稜の内点どうしで接触している場合は，定理 3 から，求める分離面はこれら 2 つの稜を含む必要があり，この様な面は 1 つしかないから，この面の法線ベクトルを被拘束物体側に向けたものが，求める拘束条件になる．

関数 edge-vertex-contact を **Fig. 13** に示す．稜の内点と頂点で接触している場合は，定理 3 から，求める分離面はこの稜と，この他にこの稜の載る 2 つの面上の各々適当な 1 点またはステップ 2 の hispoints 中の 1 点を

edge-vertex-contact (*position myedge hisvertex sign*)

1. *mypoints* ← (send *myedge* **:neighborpoints** *position*)
2. *hispoints* ← another verties of all edges which hisvertex lies
3. **for each** *f* ← two face which myedge lies
   - *cvector* ← (send *f* **:separation** *mypoints hispoints*)
   - **if** *cvector* is not *nil* **then** *orcond* ← *orcond* + *cvector*
4. *p*1 ← a point on myedge which is not identical with *position*
5. **for each** *p*2 ← all elements of *hispoints*
   - *splane* ← plane including *position*, *p*1, *p*2
   - *cvector* ← (send *splane* **:separation** *mypoints hispoints*)
   - **if** *cvector* is not *nil* **then** *orcond* ← *orcond* + *cvector*
6. **if** *sign* = −1.0 **then** reverse all element vector of *orcond*
7. **return** *orcond*

Fig. 13 Function edge-vertex-contact

含む必要がある．稜と面上の点を含む場合というのは，稜の載る2つの面そのものになるので，ステップ3では，この2つの面が各々分離面になるかどうかをチェックし，分離面になる場合は対応する拘束条件を orcond に加えている．hispoints 中の1点を含む場合は，稜とその点を含む面を求め，同様にそれが分離面になる場合は対応する拘束条件を追加している．これがステップ4である．

関数 vertex-vertex-contact を **Fig. 14** に示す．頂点と頂点で接触している場合は，定理3から，求める分離面はステップ3の bothpoint から選んだ2点と接触点を合わせた3点の載る分離面である．それを求めているのが，ステップ4である．

以上，EusLisp によるインプリメントを示した．本アルゴリズムは，オブジェクト指向プログラミングの特徴によって，簡潔かつ拡張性高く実現されているが，アルゴリズム自体は一般的なもので，通常のプログラミング言語でインプリメントすることも勿論可能である．

4. 実 行 例

前節で述べたアルゴリズムを実行した例を以下に示す．**Fig. 15** は両側拘束の例，**Fig. 16** は片側拘束の例である．図中で矢印で示してあるのが，求められた等価拘束点と，その点に対する拘束条件である．Fig. 15 では直方体に対する拘束，Fig. 16 では上に載っている物体に対する拘束を表示してある．この様に，本方法では両側

vertex-vertex-contact (*position myvertex hisvertex*)

1. *mypoints* ← another verties of all edges which *myvertex* lies
2. *hispoints* ← another verties of all edges which *hisvertex* lies
3. *bothpoints* ← union set of *myvertex* and *hisvertex*
4. **for each** $p1, p2$ ← all combination of 2 points from *bothpoint*
   splane ← plane including *position*, $p1, p2$
   cvector ← (send *splane* **:separation** *mypoints hispoints*)
   if *cvector* is not *nil* **then** *orcond* ← *orcond* + *cvector*
5. **return** *orcond*

Fig. 14 Function vertex-vertex-contact

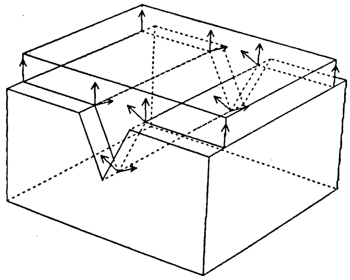

Fig. 16 Unilateral constraint

拘束も区別することなく，拘束条件を求めることが出来る．

Fig. 17 は，頂点と頂点の接触，および，頂点と稜の接触を含む例である．頂点と頂点の接触点に示されている4つの矢印と，頂点と稜の接触点の2つの矢印は，いずれも論理和の条件で，各々の点において少なくとも一つの条件を満たせば良い．これらは，(20) 式の一番内側の論理和に対応している．組み立て動作等の過程で，この様な接触状態になることは確率的には高くない．しかしながら，この様な状態を特異状態として対象外とするのではなく，同じアルゴリズムで扱えるところが本方法の特徴であり，アルゴリズムの一般性を示している．この例について，(20) 式を示す．ワールド座標系は，Fig. 17 中に小さい矢印で示した底面の中心を原点とし，矢印方向を z 軸にとる．このとき，ワールド座標系で表現した拘束条件は，

$$\begin{pmatrix} 0.0 & 1.0 & 0.0 & -100.0 \\ .707107 & .707107 & 0.0 & -70.7107 \\ 0.0 & .816499 & .577346 & -81.6499 \\ .500002 & .500002 & .707103 & -50.0002 \end{pmatrix}$$

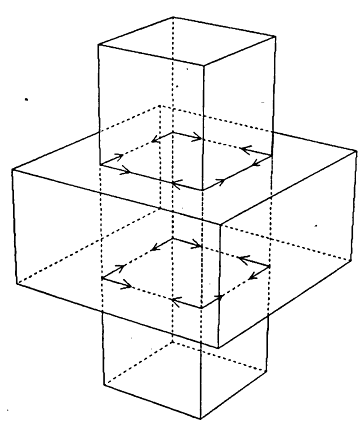

Fig. 15 Bilateral constraint

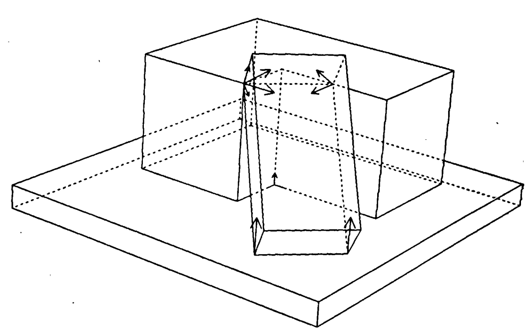

Fig. 17 Vertex-to-vertex and vertex-to-edge contact

$$\begin{pmatrix} \cdots & 0.0 & 50.0 \\ \cdots & 70.7107 & 35.3553 \\ \cdots & -28.8673 & 40.8250 \\ \cdots & 14.6451 & 25.0001 \end{pmatrix} \begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} \geq 0 \tag{21}$$

$$\begin{pmatrix} .816499 & 0.0 & .577346 & 28.8673 & 81.6499 & -40.8250 \\ 1.0 & 0.0 & 0.0 & 0.0 & 100.0 & -50.0 \end{pmatrix} \begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} \geq 0 \tag{22}$$

$$(0.0 \quad 0.0 \quad 1.0 \quad 70.7100 \quad -120.710 \quad 0.0) \times \begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} \geq 0 \tag{23}$$

$$(0.0 \quad 0.0 \quad 1.0 \quad 120.710 \quad -70.7100 \quad 0.0) \times \begin{pmatrix} \Delta X \\ \Omega \end{pmatrix} \geq 0 \tag{24}$$

なる4つの線形不等式になる．ただし，(21)式，(22)式は少なくとも1つの不等式が成立すれば良い．

Fig. 18 は，ペグを穴に挿入する作業中における拘束条件の例である．1番目の状態では等価な拘束条件の数は3，2番目では10，3番目では16になり，最終的に穴の底まで到達すると，20になる．この様に，本方法は拘束された動作の解析，計画，制御の基礎になる．

Fig. 19 は，組立パズルのある状態における拘束状態の例で，真中の部品が他の2つの部品から受ける拘束を表示してある．この例には，鞍点の接触も含まれている．この様に，本手法は，組立作業手順の計画等を実現するための，基本的な要素技術にもなり得る．

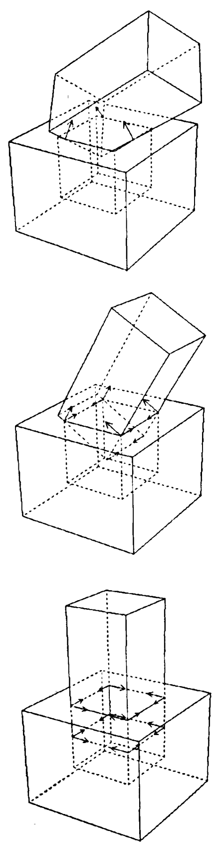

Fig. 18 Peg in a Hole

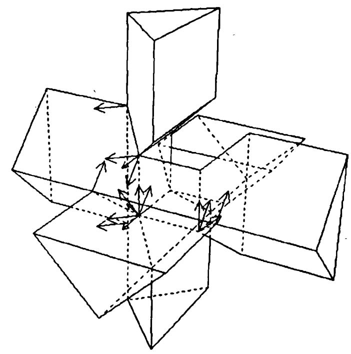

Fig. 19 Example of Assembly

5. おわりに

本稿では，作業対象とそれを拘束している作業環境の幾何モデルのみが与えられていると仮定し，これから作業対象に対する力学的な拘束条件を求める一般性のあるアルゴリズムを提案し，Eus Lisp によるインプリメント方法を示した．本方法で得られた拘束条件式は，凸解析のアルゴリズムによって解を求めることが出来[3,4,7]，被拘束物体に許される微小変位，被拘束物体が拘束物体に作用可能な力を求めることが可能である．

本アルゴリズムは，両側拘束や片側拘束を区別することなく，任意の形状の多面体と多面体が接触している場合に適用可能な非常に一般的なアルゴリズムであり，拘束された動作の解析，計画，制御や，組立作業手順の計画等を実現するための基本的な要素技術になると考えられる．

今後の課題としては，円筒面や球面等の2次曲面を含む物体間の接触の場合への拡張が挙げられる．

6. 謝辞

本研究の機会を与えて下さいました電子技術総合研究所弓場敏嗣前知能システム部長に感謝します．また，日頃貴重なご意見を戴いている築根秀男行動知能研究室長，佐藤知正自律システム研究室長，前自律システム研究室長柿倉正義氏（現東京電機大学教授）を始めとする電総研ロボットグループ諸氏に感謝します．

参考文献

1) Mason, M. T., Compliance and Force Control for Computer Controlled Manipulators, IEEE Trans., SMC-11, 6, pp. 418-432, 1981.

2) 末広尚士，高瀬國克，接触運動の表現と制御およびその組立作業への応用，日本ロボット学会誌，6巻6号，pp. 499-506，1988.
3) 平井愼一，浅田春比古，得丸英勝，凸多面錐理論を用いたマニピュレーションの運動学とその把握および組立作業への応用，計測自動制御学会論文集，24巻12号，pp. 1284-1291，1988.
4) 比留川博久，高瀬國克，拘束された物体の離脱動作の計画，計測自動制御学会学術講演会前刷，pp. 633-634，1988.
5) Matsui, T. and Inaba, M., Euslisp : An Object-Based Implementation of Lisp, Journal of Information Processing, Vol.13, No.2, 1990.
6) 比留川博久，高瀬國克，多面体間の接触による拘束条件を幾何モデルから導出する一般的なアルゴリズム，計測自動制御学会学術講演会前刷，pp. 659-660，1989.
7) 比留川博久，松井俊浩，高瀬國克，多面体間の接触による拘束条件を満たす微小変位を幾何モデルから求める方法，日本ロボット学会学術講演会前刷，pp. 525-528，1989.

付録

［定理１］の証明

まず，(9)式が成り立つことは，

$$\forall \Delta X, \forall X \in U_A(X_c)\,;$$
$$F_{st}{}^T \Delta X \geq 0,\ F_{st}{}^T X > 0 \Longrightarrow$$
$$F_{st}{}^T(X+\Delta X) = F_{st}{}^T X + F_{st}{}^T \Delta X > 0 \quad (25)$$
$$\forall X \in U_B(X_c)\,;\ F_{st}{}^T X < 0 \quad (26)$$

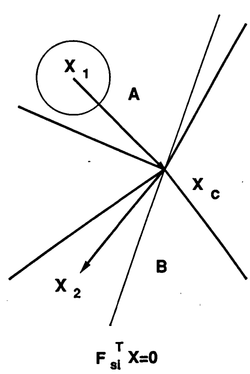

Fig. 20 Plane not separating two bodies

より明らかである．以下，定理の後半を証明する．ある面 $F_{st}{}^T X=0$ について，

$$\exists X_1 \in U_A(X_c),\quad \exists X_2 \in U_B(X_c)\,;$$
$$F_{st}{}^T X_1 > 0,\quad F_{st}{}^T X_2 > 0 \quad (27)$$

と仮定する（**Fig. 20** 参照）．このとき，Aの多面体の方が $\Delta X = X_2 - X_c$ なる変位をしたとすると，$\rho(X_2, X_c) < \delta$ よりこれは微小変位で，$F_{st}{}^T X_c = 0$ だから，

$$F_{st}{}^T \Delta X = F_{st}{}^T X_2 - F_{st}{}^T X_c = F_{st}{}^T X_2 > 0 \quad (28)$$

が成り立ち，A の多面体の境界上の点 X_c は $X_c + \Delta X = X_c + (X_2 - X_c) = X_2$ より $X_2 \in U_B(X_c)$ に移動するので，$\Delta X \in \mathrm{COL}(A, X_c)$ である．一方，Aの多面体の方が $\Delta X = X_c - X_1$ なる変位をしたとすると，これも同様に微小変位で，

$$F_{st}{}^T \Delta X = F_{st}{}^T X_c - F_{st}{}^T X_1 = -F_{st}{}^T X_1 < 0 \quad (29)$$

が成り立つ．また，X_1 は点 X_c の近傍に含まれる A の多面体の内点だから，X_1 の δ_1 近傍を $U(X_1;\delta_1) = \{X | \rho(X, X_1) < \delta_1\}$ とおくと，

$$\exists \delta_1\,;\ U(X_1;\delta_1) \subset U_A(X_c) \quad (30)$$

である．この $U(X_1;\delta_1)$ は $X_1 + \Delta X = X_1 + (X_c - X_1) = X_c$ より $U(X_c;\delta_1)$ に移動する．このとき，Bの多面体は点 X_c に接触していることから，

$$\exists X\,;\ X \in U(X_c;\delta_1) \subset U_A(X_c),\ X \in U_B(X_c) \quad (31)$$

となるので，$\Delta X \in \mathrm{COL}(A, X_c)$ である．以上により，$F_{st}{}^T X=0$ が分離面でない場合は，

$$\exists \Delta X\,;\ F_{st}{}^T \Delta X > 0,\ \Delta X \in \mathrm{COL}(A, X_c) \quad (32)$$
$$\exists \Delta X\,;\ F_{st}{}^T \Delta X < 0,\ \Delta X \in \mathrm{COL}(A, X_c) \quad (33)$$

であるので，(9)式は成立しない．従って，命題は証明された．

［定理２］の証明

$F_s{}^T X=0$ が $\mathrm{SAFE}(A, X_c)$ の境界にならないと仮定する．このとき，一般性を失うことなしに，

$$\forall \alpha, \beta, \forall X_a \in U_A(X_c),$$
$$\forall X_b \in U_B(X_c),\ \exists F_{st}{}^T X = 0\,;$$
$$F_{st}{}^T(\alpha(X_1 - X_c) + \beta(X_2 - X_c)) \geq 0 \quad (34)$$
$$F_{st}{}^T X_a > 0 \text{ and } F_{st}{}^T X_b < 0 \quad (35)$$
$$F_{st}{}^T X_1 \neq 0 \text{ or } F_{st}{}^T X_2 \neq 0 \quad (36)$$

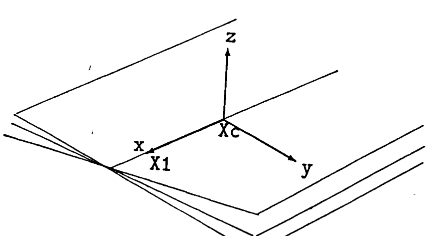

Fig. 21 Diffential rotation of separating plane

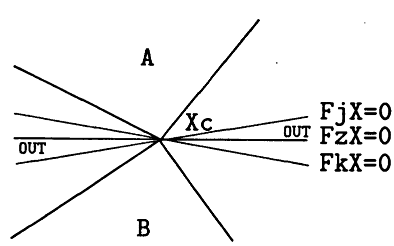

Fig. 22 New separating plane

がこの仮定が成立するための必要条件である．これは，$F_s{}^TX=0$ の任意の点に対して，その点から面までの距離が非負になり，$F_s{}^TX=0$ とは一致しない分離面 $F_{si}{}^TX=0$ が存在する，という条件である．このような面 $F_{si}{}^TX=0$ がないと，$F_s{}^TX=0$ は SAFE(A,X_c) の境界を決める面として必須の面になるので，仮定に反する．ここでは，2つの多面体のいずれかの境界に載っている点 X_c 以外の2点を X_1,X_2 として選ぶものとする．まず，例えば X_1 が A の多面体の境界の場合を考える．(34) 式から，$\alpha=-1$，$\beta=0$ に対して，

$$F_{s1}{}^T(X_c-X_1)\geq 0 \tag{37}$$

となる分離面 $F_{s1}{}^TX=0$ が存在する必要がある．このとき，

$$F_{s1}{}^TX_1=F_{s1}{}^T(X_1-X_c)+F_{s1}{}^TX_c\leq 0 \tag{38}$$

となり，X_1 は A の多面体の境界の点であるので，(35) 式に矛盾しないのは，$F_{s1}{}^TX_1=0$ の場合のみである．

一方，点X_1 が B の多面体の境界上の点の場合を考えると，同様に $\alpha=1$，$\beta=0$ に対して，

$$F_{s2}{}^T(X_1-X_c)\geq 0 \tag{39}$$

となる面 $F_{s2}{}^TX=0$ が存在する必要がある．このとき，

$$F_{s2}{}^TX_1=F_{s2}{}^T(X_1-X_c)+F_{s2}{}^TX_c\geq 0 \tag{40}$$

となり，X_1 は B の多面体の境界の点であるので，(35) 式に矛盾しないのは，やはり $F_{s2}{}^TX_1=0$ の場合のみである．X_2 についても同様であるので，これは (36) 式に矛盾するので，定理の前半は証明された．

逆に，ある分離面 $F_s{}^TX=0$ が SAFE(A,X_c) の境界を決める面であるとき，2多面体のいずれかの境界の同一直線上にない3点を含まないと仮定する．ここで，この面が2多面体のいずれかの境界上の点を X_c 以外に1点含む場合はその点を X_1 とし，そうでない場合はこの面上の X_c を除く任意の点を X_1 とおく．点 X_c を原点とし，2点 X_c,X_1 を含む直線が x 軸になり，面 $F_s{}^TX=0$ が $xy-$ 平面になる様な座標系を考え，以下この命題の証明はこの座標系で記述することにする．このとき，(35) 式の条件は，

$$F_z{}^TX_a>0 \text{ and } F_z{}^TX_b<0 \tag{41}$$

になる．ここに，$F_z{}^T=(0\ 0\ 1)$ である．x 軸を軸として面 $F_z{}^TX=0$ を角度 θ だけ回転したとすると (**Fig. 21** 参照)，この直線上にない面上の任意の点 $X_2=(x_2,y_2,z_2)$ は

$$\tilde{X}_2=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\theta & -\sin\theta\\ 0 & \sin\theta & \cos\theta\end{pmatrix}X_2=\begin{pmatrix}x_2\\ y_2\cos\theta\\ y_2\sin\theta\end{pmatrix} \tag{42}$$

に移動する．この2点間の距離は，

$$\rho(X_2,\tilde{X}_2)=y_2\sqrt{2(1-\cos\theta)} \tag{43}$$

で与えられる．また，仮定から点 X_2 は A,B の外点の集合 OUT$_{A,B}(X_c)$ の要素なので，

$$\exists\delta_2;\ U(X_2;\delta_2)\subset \mathrm{OUT}_{A,B}(X_c) \tag{44}$$

である．従って，$\delta_2=y_2\sqrt{2(1-\cos\Theta)}$ となる様に Θ を選べば，

$$\forall X_2,\exists\Theta;\ |\theta|<\Theta \Longrightarrow \tilde{X}_2\in \mathrm{OUT}_{A,B}(X_c) \tag{45}$$

となる．ここで，面 $F_z{}^TX=0$ を $\pm\Theta$ 回転した2面を $F_{sj}{}^TX=0$，$F_{sk}{}^TX=0$ とする．このとき，(45) 式から，

$$\{X|F_{sj}{}^TX>0,\ F_z{}^TX\leq 0\}\subset \mathrm{OUT}_{A,B}(X_c) \tag{46}$$

$$\{X|F_{sj}{}^TX<0,\ F_z{}^TX\geq 0\}\subset \mathrm{OUT}_{A,B}(X_c) \tag{47}$$

となり，回転の軸上の点 $F_{sj}{}^TX=0$，$F_z{}^TX=0$ は動かないので，(41) 式に注意すれば，

$$X\in U_A(X_c),\ F_{sj}{}^TX<0 \text{ or } X\in U_B(X_c),\ F_{sj}{}^TX>0 \tag{48}$$

となる点 X は存在しないので，面 $F_{sj}{}^TX=0$ は2つの多面体の分離面になる (**Fig. 22** 参照)．面 $F_{sk}{}^TX=0$ も同様に分離面になる．ところが，$\Delta X=(\Delta x,\Delta y,\Delta z)$ に対して，

$$F_z{}^T\Delta X=\Delta z \tag{49}$$

$$F_{sj}{}^T\Delta X=-\sin\Theta\Delta y+\cos\Theta\Delta z \tag{50}$$

$$F_{sk}{}^T\Delta X=\sin\Theta\Delta y+\cos\Theta\Delta z \tag{51}$$

であるので，任意の $\Delta z\geq 0$ なる微小変位に対して，Θ を十分小さくとれば $\cos\Theta\geq 0$ だから，

$$F_{sj}{}^T\Delta X=-\sin\Theta\Delta y+\cos\Theta\Delta z\geq -\sin\Theta\Delta y \tag{52}$$

$$F_{sk}{}^T\Delta X=\sin\Theta\Delta y+\cos\Theta\Delta z\geq \sin\Theta\Delta y \tag{53}$$

となる．従って，

$$\forall\Delta X;\ F_z{}^T\Delta X\geq 0 \Longrightarrow F_{sj}{}^T\Delta X\geq 0 \text{ or } F_{sk}{}^T\Delta X\geq 0 \tag{54}$$

が成立する．これは，

$$\{\Delta X|F_z{}^T\Delta X\geq 0\}\subset \{\Delta X|F_{sj}{}^T\Delta X\geq 0 \text{ or } F_{sk}{}^T\Delta X\geq 0\} \tag{55}$$

を意味しており，面 $F_z{}^TX=0$ が SAFE(A,X_c) の境界を決める面であることに矛盾する．以上により，命題は証明された．

［定理3］の証明

点 X_c に接触しているのが面の内点の場合，その多面体の X_c の近傍における任意の境界の点は，X_c が座標系の原点であることに注意すると，

$$X=(X_{21}\quad X_{22}\quad -X_{21}\quad -X_{22})\begin{pmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \varepsilon_4\end{pmatrix} \tag{56}$$

$$\varepsilon_i\geq(i=1,\cdots,4) \tag{57}$$

と表せる．ここに，X_{21},X_{22} は接触面 $F_{1s}{}^TX=0$ 上の点で，X_{21},X_{22} が1次独立，となる様な任意の2点である．このとき，ある面 $F_s{}^TX=0$ が分離面になるために

は，$F_{s1}{}^TX=0$ がAの多面体の面である場合は，

$$\left.\begin{array}{l} F_s{}^TX_{21}\geq 0,\ F_s{}^T(-X_{21})\geq 0, \\ F_s{}^TX_{22}\geq 0,\ F_s{}^T(-X_{22})\geq 0 \end{array}\right\} \tag{58}$$

が必要である．ところが，これらの式から

$$F_s{}^TX_{21}=0,\ F_s{}^TX_{22}=0 \tag{59}$$

となり，$F_s{}^TX_c=0$ が必要なことに注意すると，$F_s=F_{s1}$ であることが分かる．$F_{s1}{}^TX=0$ がBの多面体の面の場合も同様なので，面の内点で接触する場合の命題は示された．

点 X_c に接触しているのが稜の場合，その多面体の X_c の近傍における任意の境界の点は，

$$X=(X_{11}\quad X_{12})\begin{pmatrix}\varepsilon_1\\ \varepsilon_2\end{pmatrix}+(X_{21}\quad -X_{21})\begin{pmatrix}\varepsilon_3\\ \varepsilon_4\end{pmatrix} \tag{60}$$

$$\varepsilon_1\varepsilon_2=0 \tag{61}$$

$$\varepsilon_i\geq 0(i=1,\cdots,4) \tag{62}$$

で与えられる．ここに，X_{ij} は，この稜の近傍を（4）式の様に表記したとき，

$$F_1{}^TX_{11}=0,\ F_2{}^TX_{11}>0 \tag{63}$$

$$F_1{}^TX_{12}>0,\ F_2{}^TX_{12}=0 \tag{64}$$

$$F_1{}^TX_{21}=0,\ F_2{}^TX_{21}=0 \tag{65}$$

$$X_{21}\neq X_c \tag{66}$$

なる様な任意の点である．このとき，ある面 $F_s{}^TX=0$ が分離面になるためには，(59) 式と同様に，

$$F_s{}^TX_{21}=0 \tag{67}$$

が必要なことが分かる．X_{21} は稜上の点だから，これは分離面がこの稜を含む必要があることを意味している．この式に注意すると，この分離面が X_{21} と1次独立な境界の任意の点Xを含むならば，(60) 式から，

$$F_s{}^TX=F_s{}^T(X_{11}\quad X_{12})\begin{pmatrix}\varepsilon_1\\ \varepsilon_2\end{pmatrix}=0 \tag{68}$$

となる．この式において，$F_s{}^TX=0$ が分離面であることから，一般性を失うことなしに $F_s{}^TX_{11}\geq 0$，$F_s{}^TX_{12}\geq 0$ であり，$\varepsilon_1\geq 0$，$\varepsilon_2\geq 0$ に注意すると，

$$F_s{}^TX_{11}=0 \text{ or } \varepsilon_1=0 \tag{69}$$

and

$$F_2{}^TX_{12}=0 \text{ or } \varepsilon_2=0 \tag{70}$$

となる．ところが，$\varepsilon_1=\varepsilon_2=0$ だと，この点Xは X_{21} と1次従属になってしまう．従って，

$$F_s{}^TX_{11}=0 \text{ or } F_s{}^TX_{12}=0 \tag{71}$$

であることが分かる．以上により，分離面 $F_s{}^TX=0$ が X_{21} と1次独立な任意の点を含むならば X_{11} または X_{12} を必ず含むことが示されたので，稜で接触する場合の命題は示された．

点 X_c に接触しているのが頂点の場合，その多面体の X_c の近傍において，j 番目 ($j=1,\cdots,m$) の面上の任意の点は，

$$X=(X_{j1}\quad X_{j2})\begin{pmatrix}\varepsilon_1\\ \varepsilon_2\end{pmatrix} \tag{72}$$

$$\varepsilon_i\geq 0(i=1,2) \tag{73}$$

と表せる．ここに，X_{j1}, X_{j2} は j 番目の面の点 X_c を端点とする2つの稜上の任意の点で，点 X_c とは一致しない点とする．このとき，ある分離面 $F_s{}^TX=0$ が j 番目の面の任意の点Xを含むならば，

$$F_s{}^TX=F_s{}^T(X_{j1}\quad X_{j2})\begin{pmatrix}\varepsilon_1\\ \varepsilon_2\end{pmatrix}=0 \tag{74}$$

となり，分離面であることから一般性を失うことなしに，$F_s{}^TX_{j1}\geq 0$，$F_s{}^TX_{j2}\geq 0$ であり，$\varepsilon_1=\varepsilon_2=0$ だと点 X_c に一致してしまうことから，(71) 式の場合と同様にして，

$$F_s{}^TX_{j1}=0 \text{ or } F_s{}^TX_{j2}=0 \tag{75}$$

が成り立つ．以上により，分離面 $F_s{}^TX=0$ が j 番目の面上の任意の点を含むならば X_{j1} または X_{j2} を必ず含むことが示されたので，頂点で接触する場合の命題は示された．

［定理4］の証明

接触点の集合が物体座標系で P_1, P_2 と表される2点を端点とする線分であるとすると，この線分上の任意の点から受ける拘束は，(18) 式，(19) 式より，

$$\bigcap_{i=1}^{N}\bigcup_{j=1}^{M(i)}F_{sij}{}^T\begin{pmatrix}E_x{}^T & (((1-t)P_1+tP_2)\times E_x)^T\\ E_y{}^T & (((1-t)P_1+tP_2)\times E_y)^T\\ E_z{}^T & (((1-t)P_1+tP_2)\times E_z)^T\end{pmatrix}\begin{pmatrix}\Delta X\\ \Omega\end{pmatrix}\geq 0,\ 0\leq t\leq 1 \tag{76}$$

となる．ここで，一般性を失うことなしに，各点における座標系の姿勢は E_x, E_y, E_z が線分上の全ての点で同じになる様にとったので，F_{sij} は接触によらず一定の値である．また，t はパラメータである．$E_x=(1-t)E_x+tE_x$ 等に注意して，これを整理すると，

$$\bigcap_{i=1}^{N}\bigcup_{j=1}^{M(i)}\left((1-t)F_{sij}{}^TJ_1\begin{pmatrix}\Delta X\\ \Omega\end{pmatrix}+tF_{sij}{}^TJ_2\begin{pmatrix}\Delta X\\ \Omega\end{pmatrix}\right)\geq 0,\ 0\leq t\leq 1 \tag{77}$$

となる．各接触点における拘束条件が和演算を含まない場合，この式は，

$$\bigcap_{i=1}^{N}\left((1-\mathrm{t})F_{si}{}^TJ_1\begin{pmatrix}\Delta X\\ \Omega\end{pmatrix}+tF_{si}{}^TJ_2\begin{pmatrix}\Delta X\\ \Omega\end{pmatrix}\right)\geq 0,\ 0\leq t\leq 1 \tag{78}$$

となる．一方，両端点に対する拘束条件は，

$$\bigcap_{i=1}^{N}F_{si}{}^TJ_1\begin{pmatrix}\Delta X\\ \Omega\end{pmatrix}\geq 0 \tag{79}$$

$$\bigcap_{i=1}^{N}F_{si}{}^TJ_2\begin{pmatrix}\Delta X\\ \Omega\end{pmatrix}\geq 0 \tag{80}$$

で与えられ，これらの2式が等価であることから命題の前半は証明された．命題の後半は，凸包の定義から明らかである．