自動設計システムの一方式

沖　野　教　郎*

(昭和43年11月26日受理)

A Study on Automated Design Systems

Norio OKINO

(Received November 26, 1968)

Abstract

In recent years, a new use of computers in the field of design and drawing has been developed. This is known as "CAD", "DAC" or "ADE" etc.

This paper is a proposal on a new form of a system for automated design in which the following two main points are included.

(1) Man-Machine communication system managed by the computer side.

(2) Expression of the design information by the "Pattern Matrix Method"

The first is a method of communication from designer to computer and in this paper, this new system is compared with the CAD system which has CRT and a light pen. A description of the process to realize the design was made.

The second is a concept concerning the mathematical system of the automated design, in which all patterns for designs, e.g. graphic patterns, constraints for the engineering design or characteristics of the Materials etc. are expressible in the simple mathematical formulae by the Pattern Matrix Method.

Some merits of this system are as follows :

(1) A small computer may be sufficiently used to organize the system.

(2) This system makes expert technical knowledge of designers unnecessary.

(3) The results of the automated design are directly linked with numerically controlled machine tools and drafters.

1. 緒　　言

設計にコンピュータを利用することは，コンピュータ開発の当初より盛んに行なわれて来たことであるが，これらは手計算で不可能な計算量を短時間に処理することがその主目的であった。これをさらに一歩進めて，設計過程の中に占める人間の役割を出来る限りコンピュータ

* 精密工学科　精密機器学第一講座

側にうつして自動化し，設計時間の飛躍的短縮，設計精度の著しい向上など，設計の能率化，合理化を目標にした自動設計に関する研究が，世界各地で意欲的に始められている。

このような意味での自動設計は設計を中心に自動製図や自動加工などを含むシステムとして構成してこそ，その効果を十分に発揮できるものと考えられる。すなわち，この分野の研究課題の一つとして，どのようなシステムが適当かという，システムの構造（方式）を決定する問題がある。

この論文では北海道大学工学部精密工学科創立十周年を記念して出版される研究報告のうちの一つとして，精密機械器学第一講座を中心に進められている自動設計システムの方式の特徴を説明する。

2. システムの基本構成

対比によってシステムの特徴を明確にするため，まず従来の方式（数年前 MIT において提唱されたもので[1,2]，最近研究の進められている中で最も一般的な方式である）について考える。これは CAD (Computer-Aided Design), DAC (Design Augmented by Computer), ADE (Automated Design Engineering) などの名称である程度実用化されており，そのシステム（以下 CAD とよぶ）の基本構成は図1のごとくである。

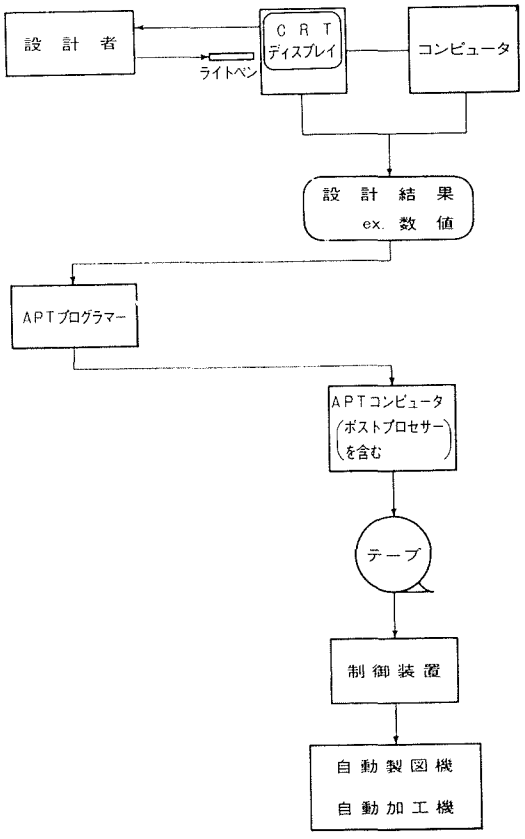


図1 “CAD”システムの基本構成

自動設計とはいっても，すべてをコンピュータに行わせることは不可能であるから，人間すなわち設計者がシステムの一要素として適当に設計過程に介在しなければならない。この場合，人間とコンピュータとの関係は丁度両者が対話をするような形で連接され

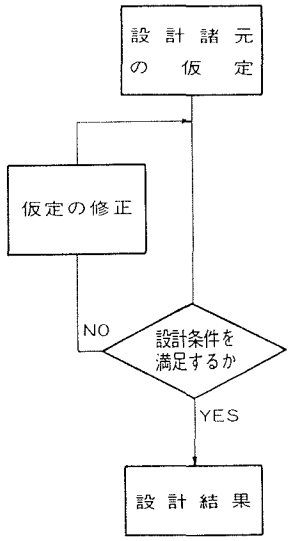


図2 Cut and Try 法

るならば最も能率的である。CAD システムはこの対話のために，ハードウエアとして CRT デイスプレイ装置とライトペンを，ソフトウエアとして図形処理を含む各種の問題向き言語を用意して，対話を円滑に行なわせようとしているところに特徴がある。図1のようにこのような対話用機器を間にはさんで，設計者はコンピュータに種々な仮定に立つ問題を解かせ，その結果を判断して，仮定を修正する過程を，条件に合致した設計諸元を得るまで繰り返す。いわゆる Cut and Try 法をコンピュータシステムに移したと考えてよく，そのブロック線図は図2のように示される。

この場合コンピュータは一種のシミュレータとして使用されているものと考えてもよく，対話の形式は図3 (a) に示すように人間が問を発し，これに対してコンピュータが答えるので，判断の後つぎの問を発するという過程を繰り返す。すなわち主導権は人間側にある。したがってこれを人間主体の対話方式とよぶことにする。

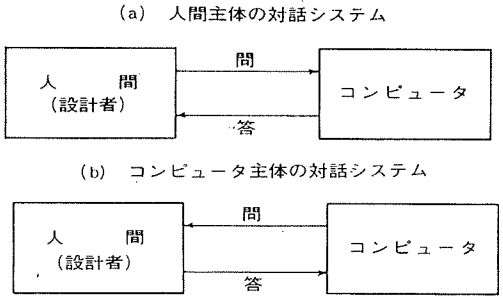


図3 対話の2方式

再び図1にかえって，設計結果は数値，言語，あるいは CRT 上の図形として得られるが，これを自動製図機，自動加工機と結ぶには，図のように APT プログラマー (使用する言語は APT に限らない) によって図形の記号化処理が行なわれると，APT コンピュータ (ポストプロセッサーを含む) によって制御用テープが作られる。すなわちこのシステムではここにも人間が介在して重要な仕事を行なっている。

つぎにこのような従来のシステムに対比して，本論文で提案する新しいシステムの基本構成を示すと図4のようになる。ここで対話用機器の主体はオンラインタイプライターで，プロッターや CRT デイスプレイなどを出力専用機として補助的に使用する。対話の方式の大きな相違は，先述の CAD システムが図3の (a) のように人間が問を発し，コンピュータがこれに答えることの繰返し，すなわち人間主体の対話方式であったのに対し図3 (b) のようにコンピュータ側からの問に対して人間が答える方式をとっていることである。このような方式は医療用の自動診断機，テイーチングマシン

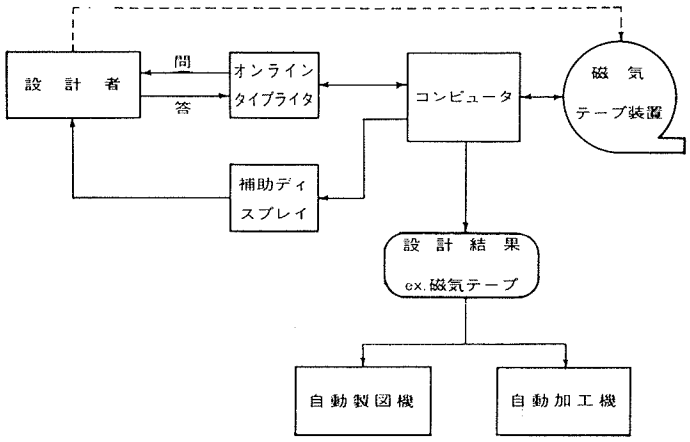


図4 提案システムの基本構成

あるいは適当な相談業務などにおいてコンピュータと人間との結合の際採られている方法であり，設計にこの方式を導入したことに当る。このような対話の方式はコンピュータ側に主導権があるので，コンピュータ主体の対話方式とよぶことにする。

つぎに第2の相違点は，この方式によれば設計結果として自動製図機や自動加工機を駆動するためのテープがコンピュータから直接得られることである。図1における APT などによる図形処理過程の省略をはかったもので，この部分には人間の介在が不必要となる。

設計の過程はコンピュータ側であらかじめきめられた手順に従って進めるので，図2のような試行的方法をそのまま自動化することは困難であり，図5のような最適設計のための過程をそのまま採り入れることとした。

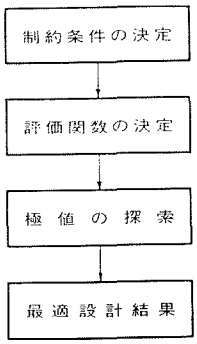


図5 最適設計過程

システムの詳細な構成については次項以下において述べることとし，ここではこのシステムの特徴について触れておく。この方式の主なるものを列挙すると，

(1) 設計過程が単純であるから，小型のコンピュータでもシステムを構成できること。

(2) コンピュータ側でほとんどの Job を処理するので設計者に特別な専門的知識がなくとも最適な設計結果をうることができる。

(3) 自動製図や自動加工との連接がスムーズである。

ただし，このシステムは標準化された設計対象以外には適用不可能で，創造的な設計は図1の CAD システムによるのが適当である。すなわち，ここに提案するシステムは CAD システムと競合するものではなく，それぞれの適合分野で生かされるべきものである。もちろん，両者を組み合わせた大規模なシステムを構成することも可能である。

3. 対話方式の構成

3-1. コンピュータ主体の対話方式

前項で対話の原理的方式については説明した。ここではコンピュータ主体の対話を実行するための具体的な方法をのべる。

図6は筆者の研究室において，この方式の実験のために設置した小型コンピュータシステム (OKITAC 5090 C 型システム) である。CPU (中央処理装置)，TW (タイプライタ)，MT (磁気テープ装置) が，この対話方式実行のための基本構成であり，LP (ラインプリンタ)，plotter (作図機)，PTR (光電式テープリーダ) は補助機器として使用し，あれば便利であるが無くても用は足せ

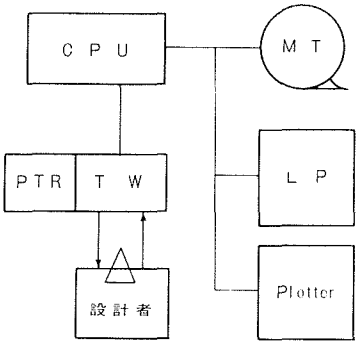


図6 コンピュータ主体の対話実行のために設置したシステム

る。CPU は1語48ビットのメモリを4K 語保有している。

このようなシステムは普通に見られる科学技術計算用の小型あるいは中型の汎用システムであり，これをそのまま自動設計用として使用できることはこの方式の一つの特徴でもある。

対話はオンラインのタイプライタによって行い，コンピュータからの問に対して YES, NO, あるいは適当な数値または指定の番号や記号によって返答すれば再び次の問がタイプされる。この問答の繰り返しが対話であるが，その目的はコンピュータ側が自動設計に必要な公式や数値の決定など図5における極値探索のための準備を整えることにある。すなわち制約条件および評価関数の決定が最終目的である。

制約条件の決定はあらかじめメモリ中に用意された多数の条件式中から適当なものを選択すること，および選択された条件式中の未確定定数に適当な数値を与えることである。

いま m 個の制約条件 C_i $(i=1\sim m)$ が

$$G_i(x_1,\ x_2,\ \cdots\cdots,\ x_n)\geqq 0 \quad (i=1\sim m) \tag{1}$$

なる一般形で表わされるものとする。ここで x_j $(j=1\sim n)$ は求める設計諸元である。m 個の中に異種の対象に対するそれぞれの条件と同種の対象に対する複数の条件とが含まれる。後者はたとえば厳密な条件式と近似条件式，あるいはいくつかの実験式などである。

条件が近似式によって与えられるとき，C_a，C_b，C_c，……の順に厳密度が高くなっているとすれば

$$C_a\subset C_b\subset C_c\subset\cdots\cdots \tag{2}$$

のようになるが，厳密な式と近似式では計算機使用時間に大きな違いがあるのが普通だから，設計に必要な精度に応じて適当な近似式を選択しなければならない。選択の基準はコンピュータ側にあらかじめ用意しておく。

条件が多数の実験式によってきめられる場合も多い。油の粘度と温度の関係など十数種の式があり，いずれも場合に応じて使われる。この選択も対話を通じて行なわれる。

G_i には係数あるいは定数項としていくつかの未確定定数を含んでいる。これは設計仕様やその他の設計条件（たとえば材料の強度，規格，工場の特殊事情）から決まるので，決定に必要な諸数値をコンピュータからの問に応じて入力してやる。

評価関数についても，複数の評価関数中から適当な一つの式の選択，およびウエイトなどの定数の決定が行われねばならない。複数の評価関数が存在することについて補足すると，評価関数は何が最適であるかを決める基準であるから設計者の主観に左右される要素の強いものであり，またそうでなければ意味がないので，式の形についても定まったものはなく，複数の式が考えられる。もし，人間主体の対話方式ならば，評価関数は設計者が与えることができるが，コンピュータ主体の方式でも，いくつかの評価関数をあらかじめコンピュータ側に用意しておき，その選択および式中のウエイトを設計者にまかせることによって設計者の主観を導入

できる。

対話によって決定するのは上記までであるが，極値の探索以下のプログラムとあわせてコンピュータ側に用意しなければならないメモリ量は莫大なものになる。この問題点を合理的に解決するために記憶装置としては図6に見るように磁気テープを使用し，テープ中のブロックを順次 CPU 中に移し，そのプログラムによって対話を進行させる，対話のプログラムをとりだしやすくするために図7のように樹枝状に質問事項を配置する。

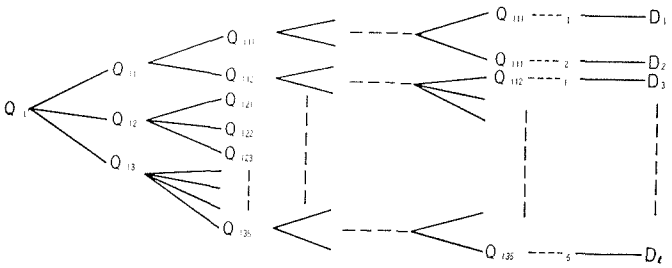


図7 質問の樹枝状配置

すなわち，図において質問 Q_1 の内容は Q_{11}, Q_{12}, Q_{13} のいずれかを設計者に選択させるための問である。仮りに Q_{11} が選択されたとすれば，つぎには Q_{111}, Q_{112} のいずれかを選択する問題がタイプされる。設計者からの答は分岐をどれにとるかの選択に他ならない。数値を入力させるような質問は分岐が1つしかない問であると考えることができるので，樹枝状配置の中に含めることができる。図7の D_1, D_2, ……, D_t は対話につづいてコンピュータ側でなすべき Job のプログラムを意味し，最適値の探索や自動製図，自動加工との連結のプログラムなどが含まれる。すなわちこのシステムではあらかじめ用意した何種類かの標準的な設計プログラム中から一種類をとりだす方式であるといえる。もちろんこの一種類のプログラムにもある範囲をカバーできる内容を持たせ，未定の定数に数値を与えることによって設計諸元を確定する。

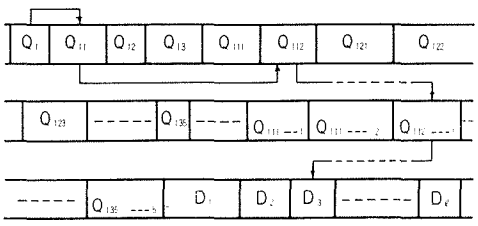


図8 磁気テープ中のブロック配置

図7の樹枝状の質問を磁気テープ中に納めるには図8のように各々を1列に並べ，必要なブロックだけを CPU にうつす。たとえば図7で質問と答えの選択が $Q_1 \rightarrow Q_{11} \rightarrow Q_{112} \rightarrow \cdots\cdots \rightarrow Q_{112\cdots1} \rightarrow D_3$ のような順序で進んだとすれば，磁気テープのブロックは図8の矢印で示したような順序で CPU に移され他は読みとばされる。すなわち磁気テープの一方向のみの巻取りの間に対話を進行させる。

つぎに対話のプログラムの作成について述べる。この対話方式による自動設計が効力を発揮するには大規模で完全なプログラムを用意しておく必要があるが，このようなプログラムを作るにはあらかじめあらゆる設計仕様を想定して，それぞれに対処できるように作らなければならない。これはしかし非常な労力と期間を要し，実際的ではない。最も簡単に実行できる方法は，設計の必要が生ずるたびにそのプログラムを作成して，すでに出来ているものに追加す

る方法である。図7においてある設計のためには対話が $Q_1 \rightarrow Q_{11} \rightarrow Q_{112} \rightarrow \cdots\cdots$ の順に進むとする。そのとき Q_{12}, Q_{13} につづく質問はその時点ではプログラムを作成せず，磁気テープ中に1番地分のメモリを確保するだけにとどめる。将来，これらの質問につづく対話のプログラムが必要になったときにはその番地に行先ブロックの番号を入れて，そのブロックまで飛び，そこに必要なプログラムを格納する。この方法で追加，修正は自由にできるので，プログラムの生長が可能であり，これは本方式の特徴の一つである。

3-2. 簡単な具体例

つぎに上記の対話方式の補足説明として，軸の自動設計を例題としてあげる。

軸の設計における最も基本的な制約条件は，強度，変形，振動の3点である。強度については曲げ，ねじり，座屈の3応力に対して式 (3) のように条件を設定できる。

$$\left\{\frac{\pi(1-n^4)\tau_y}{16S}\right\}^2 d_2^6 \geqq \left\{M+\frac{d_2}{8}(1+n^2)P\right\}^2+T^2 \tag{3}$$

d_2: 軸外径

d_1: 中空軸の場合の内径

n: d_1/d_2

τ_y: 降伏せん断応力

S: 安全率

M: 曲げモーメント

T: ねじりモーメント

P: 軸力

負荷が変動する場合には

$$\begin{aligned} M &= M_m + KM_a \\ T &= T_m + KT_a \\ P &= P_m + KP_a \end{aligned} \tag{4}$$

添字 m: 平均値を意味する

添字 a: 変動量の振幅

K: 等価静荷重係数

式 (3), (4) において d_2 は曲げモーメント最大の位置について考え，段つき軸には各段に対して上式を適用する。

変形については曲げとねじりの両面から制約の要があり，

曲げ変形

$$d_2^4 \geqq \frac{64k_\delta Wl^3}{\pi E(1-n^4)\delta_m} \tag{5}$$

$$d_2^4 \geqq \frac{64k_\theta Wl^2}{\pi E(1-n^4)\theta_m} \tag{6}$$

ねじり変形

$$d_2^4 \geqq \frac{32Tl}{\pi G(1-n^4)\alpha} \tag{7}$$

δ_m: 限界たわみ

θ_m: 限界たわみ角

α: 限界ねじれ角

E: 縦弾性係数

G: 横弾性係数

l: 軸受（支点）間の長さ

W: 荷重（作用点が二つ以上のときは別の式を使用の要あり）

k_δ, k_θ: 負荷状態で変る定数，別表により入力（たとえば機械工学便覧改訂第5版1—45頁を使用）

なお段つき軸の場合には一連の計算式を含む別のプログラムで取扱う必要がある。

振動あるいは危険速度に関しては最も単純に共振点の前後に危険振動域として $\pm\beta_m$, $\pm\beta_t$（添字 m は曲げ振動，v はねじり振動に対応する）だけの振動数範囲を設定して，この領域内に入らないことを条件とすれば次式を得る。

$$\left| f_{mo} - \frac{\lambda^2 d_2^2}{2\pi l}\sqrt{\frac{\pi E g(1-n^4)}{64\gamma A}} \right| > \beta_m \tag{8}$$

$$\left| f_{to} - \frac{\lambda}{2\pi l}\sqrt{\frac{Gg}{\gamma}} \right| > \beta_t \tag{9}$$

f_{mo}, f_{to}: 軸の回転あるいはその他の強制振動源の曲げおよびねじりの振動数

γ: 軸材料の密度

A: 軸断面積

この他に Rayleigh, Dunkerley, Holzer などの式が用意されているので，段つき軸の場合や，強制振動力の作用点が多いときなどはこれらの式によらねばならない。

以上の制約条件式について，その選択は，たとえば変動荷重の有無による式 (4) の選択，曲げ，ねじり，いずれも考慮の要があるかどうかによって式 (5), (6), (7) あるいは式 (8), (9) の選択，段つきの有無によりここに示した以外の式の選択を行うなどである。

評価関数については前述のように数例を用意して選択に供しなければならないが，これの作製は“最適”の定義とも関連して最適設計における最も困難な問題の一つである。ただこの点は本論文の本質とは直接関係がないので，この点についての議論に紙面をさくことを避け，ここには線形の評価関数の1例を示しておく。

$$Y = w_1\left(\frac{d_2}{d_0}\right) + w_2\left(\frac{\tau_y}{\tau_0}\right) - w_3\left(\frac{\tau}{\tau_y}\right) + w_4\left(\frac{\delta}{\delta_m}\right) + w_5\left(\frac{\theta}{\theta_m}\right) - w_5\left(\frac{f_m}{f_0}\right) - w_7\left(\frac{f_t}{f_{to}}\right) \tag{10}$$

Y は小さいほど評価点を高くするものとし，最適値は Y の最小点として発見される。各項の

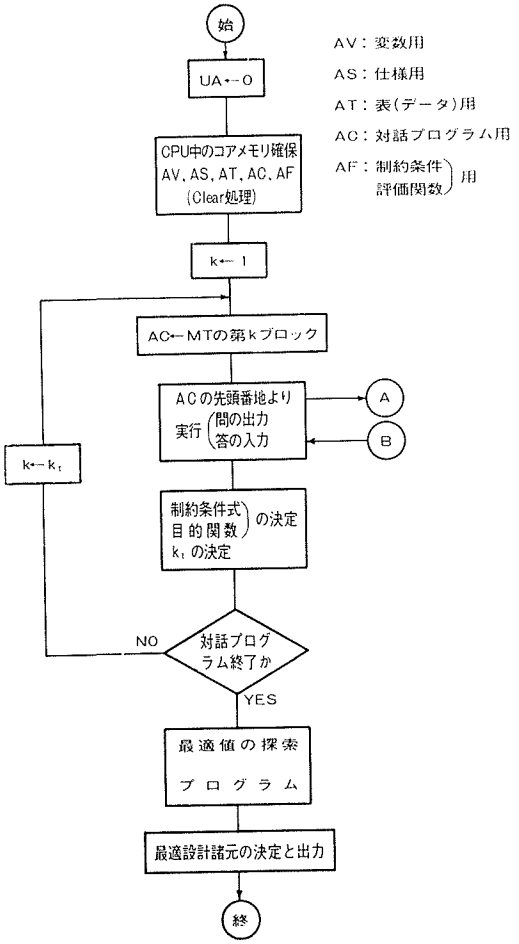


図9　メインプログラム

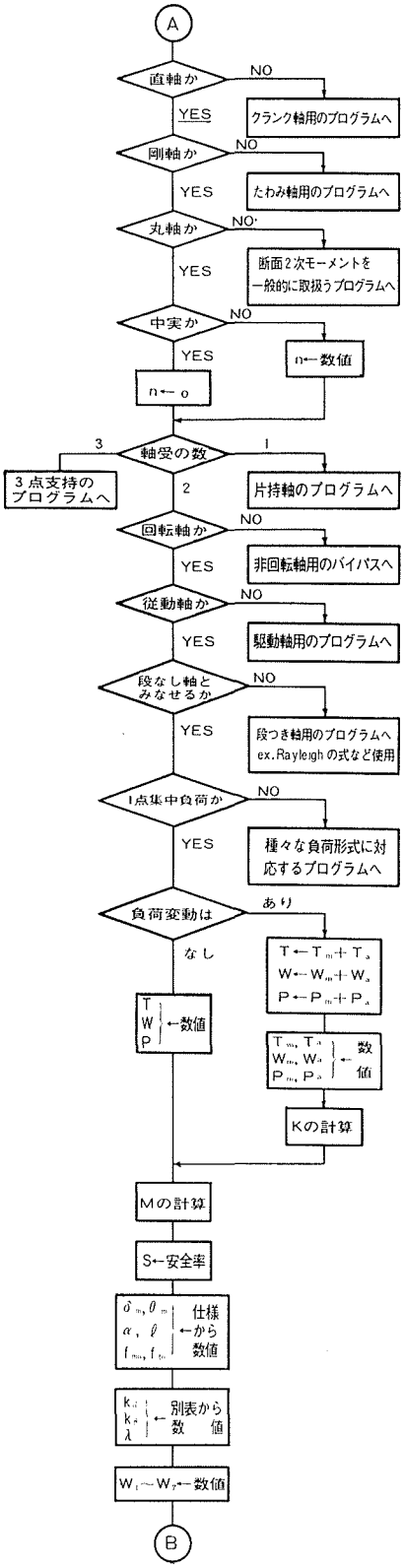


図10　対話のプログラムの一部

もつ意味についてそれぞれ説明すると，第1項：材料節約，軽量化の点から直径を小さくする目的をもつ。なお，この場合軸の長さは設計対象外としている。第2項：安価な材料を使用する経済性の項，第3項：強度に関する安全率の項，第4，5項：剛性の項，第6，7項：危険振動からの安全性の項，w_1，w_2，……，w_7 は各項に付加されたウエイトであり，設計者の判断により重要度に応じて適当な値を入力する。各項の分母は無次元化するための基準であり，d_0 には設計結果に近いと考えられる値を想定して入力し，τ_0 には，軟鋼の値を標準値として入力しておく。τ は式 (3) において等号のみとし，τ_y の代りに τ とおいて求めた値で実際に軸表面に発生する剪断応力である。f_m，f_t は式 (8)，(9) の左辺第2項の値である。

つぎに対話のためのプログラムをフローチャートによって示す。基本的な流れは図9に示すごとくである。

前段処理の後，磁気テープより順次必要な質問事項を読みだし，制約条件と評価関数の選択，各式への数値の代入などを対話の過程を通して行ない，つづいて最適設計諸元の探索のための計算にうつり，結果を印刷することによって終了する。このうちの対話の部分のプログラムを図10に示す。樹枝状の分岐が特徴であるが，分岐の枝の方は紙面の都合で省略した。先述のように分岐は必要があるまでは単に1個の番地として残しておき，プログラムを作成したときはじめてそこに補充すればよい。

図10では最初分類をきめるための対話を YES, NO の答を主要な対話として進め，その間制約条件式の選択を自動的に行ない，つづいて数値の決定に移っている。数値を設計者が与える際，設計者の判断によって自由にその値がきめられる定数も多い。(たとえば安全率など) これらは一方では比較的経験の少ない設計者を困惑させることにもなるので，この対話方式では，それらの数値は従来使用された標準的な数値の一覧表を参考数値として，その都度ラインプリンターに打出させる。これは丁度設計者が便覧片手に設計をする場合に似ている。

4. 図形処理の方式

前項の対話方式によって自動設計を進行させる場合，最も問題になるのは図形を如何に処理するかである。第2項で述べた CAD システムにおけるライトペンや CRT ディスプレイは図形処理のための装置であるといっても過言ではない。逆にいえば図形処理は自動設計の程度をきめる大きな要因の一つである。なぜなら図形を考えない設計はほとんどないからである。

しかし，現在のデイジタルコンピュータは記号を扱いうるのみだから，図形をそのままの形で処理することはできない。現在，自動設計において図形はつぎのいずれかによって取扱われている。

i) 図形を言語によって記述する方法

ii) 図形を点の集合におきかえて取扱う方法

iii) 図形を数式化して取扱う方法

(i) の言語による表現法は APT に代表される図形処理用問題向き言語を用いる方法で，現在最も実用されている方式である。しかし，図形はかなり高次な情報であるから，これを問題向き言語（僅かの語彙しか持っていない）によって表現することには当然限界があり，複雑な図形の表現は困難である。現に APT 以外にも多数の図形処理用言語が作られ，あるいは新しく研究されていることは表現能力の高い言語の作製がいかに困難事であるかを物語るものである。(ii) の点集合におきかえる方法は必要な記憶容量が大きすぎて問題にならない。記憶する点を少なくし，使用の際補間する方法なども試みられているが，自動設計にはごく限られた場合以外には一般性がない。(iii) の数式化して取扱う方法は，もし図形が数式化されたならばコンピュータ側での取扱いが容易である点，および数学に乗せて高度な処理を行ない得る点から，理想的といえるが，図形そのものを数式化することが困難である。ここでは図形の数式化を容易にしかも自動的に行う方法を考える。

4-1. 数式化パターン

ここで論ずる図形の数式化の基本的考え方は，図形を単純な数式で表現できる要素にまで分解し，その集合として図形を取扱うことである。したがって図形は各要素の数式の集合として表現されることになる。一般の図形を，単純な数式によって表現可能な要素の組み合わせとして表現できることに関しては証明するまでもないが，問題はその要素の数である。この方法が実用性を持つためには要素の数をコンピュータで扱いうる範囲内にとどめること，およびコンピュータで扱いやすい様に整然とした形にならべることである。

要素の数については，設計で扱う図形そのものが単純な要素に分解しやすいものが多いので実際上は問題にならないと考えられる。特殊な場合でも後述の変換関係による図形の発生方式を利用することによって解決できる。

ここでは図形(以後は場合によってパターンとよび次節とも関連して一般性を持たせる)の数式集合による表現の整理した記述法をのべる。

パターン P が P_1, P_2, ……, P_m なる m 個の要素の“結び”によって構成されるとき，これを

$$P=[P_1 \cup P_2 \cup P_3 \cup \cdots\cdots \cup P_m] \tag{11}$$

と表わすことにする。左辺はパターンの名称，右辺はその内容を意味し，特に [ ] をつけることとする。式 (11) をまとめると

$$P=\bigcup_{j=1}^{m} P_j \tag{12}$$

P_j が 1 個の数式，あるいは不等式で完全に表現できる場合はそれでよいが，一般には P_j は数式で表わされる集合の一部として表現した方が式の形が単純になる。したがって P_j はいくつ

か（一般には n 個とする）の集合の“交わり”としてその部分を決定する。

$$P_j=P_{1j}\cap P_{2j}\cap\cdots\cdots\cap P_{nj}$$

すなわち

$$P_j=\bigcap_{i=1}^{n}P_{ij} \tag{13}$$

式 (12) に代入

$$P=\bigcup_{j=1}^{m}\left(\bigcap_{i=1}^{n}P_{ij}\right) \tag{14}$$

このような形に表現されたパターンを数式化パターンと呼ぶことにする。

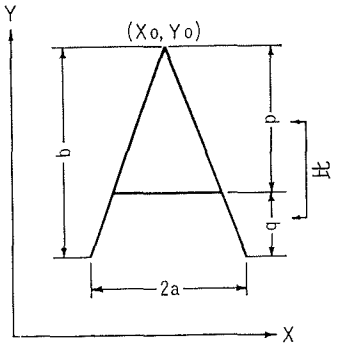


図 11 文字 A の寸法と位置

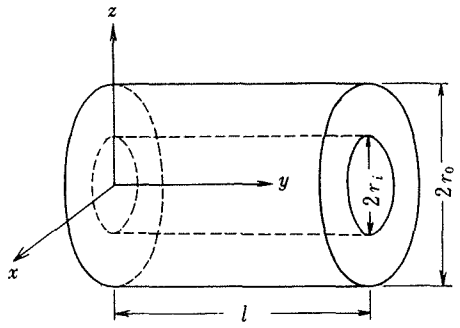


図 12 中空円筒の寸法と座標系

なお，二三の実例を示しておく。図 11 の文字 A を数式化すると式 (15) のようになる。

$$\begin{aligned}P=&\left[\left\{a(y-y_0)=\pm b(x-x_0)\right\}\cap(y_0-b\leqq y\leqq y_0)\right]\\&\cup\left[\left(y=y_0-\frac{pb}{p+q}\right)\cap\left\{(x-x_0)^2\leqq\left(\frac{pa}{p+q}\right)^2\right\}\right]\end{aligned} \tag{15}$$

これは 3 個の線分の結びであり，各線分は，1 個の直線と，線分の存在区間を示す 2 個の半平面の“交わり”として示される。

図 12 に示すようなシリンダーを数式で表示すると式 (16) である。

$$P=\left[(r_i^2\leqq x^2+z^2\leqq r_0^2)\cap(0\leqq y\leqq l)\right] \tag{16}$$

すなわち二つの円柱面と端面の二つの平面とで囲まれた部分として与えられる。

4-2. パターンマトリクス

上述の数式化パターンは $\cup$, $\cap$ で各数式を結合した集合演算であるから，これをコンピュータで取扱う際 $\cup$, $\cap$ をまとめて整然とした形にしておくと便利である。そこで数式化パターンを式 (14) の形に整理し，これをつぎのようにマトリクスに並べる。

$$P=\left\{\begin{pmatrix} P_{11}\ P_{12}\cdots\cdots P_{1m} \\ P_{21}\ P_{22}\cdots\cdots P_{2m} \\ \cdots\cdots\cdots\cdots\cdots \\ P_{n1}\ P_{n2}\cdots\cdots P_{nm} \end{pmatrix}\right\} \tag{17}$$

これは列が $\cap$ で結合され，行が各列をまとめて $\cup$ で結合されたもので，普通の行列と区別するために〔〔　　〕〕を使用する。

コンピュータでこれを扱うには P_{ij} 中には $\cap$, $\cup$ による結合を含まないようにあらかじめ分割しておき，1個の数式のみで構成させる。もし P_{ij} のどれかに不等号を含むならばすべてを陰関数におき，

$$P_{ij}=[f_{ij}(x, y, z)\geqq 0] \tag{18}$$

の形に標準化する。

P_{ij} のすべてが陽関数，あるいは媒介変数によって表示できるときはそのまま扱った方が能率的である。

ただし，これらの標準化はあくまで原則論で実際には臨機応変に処理することも必要である。

式 (18) はプログラム上で

$$\mathrm{PIJ}=f_{ij}(x, y, z) \tag{19}$$

とかかれたとする。(FORTRAN では f_{ij} が簡単な関数でそのルーチンが用意されているとき，そのまま式 (19) がプログラムになる) 式 (18) の等号，不等号については，式 (19) の記憶場所と対になる別の記憶場所に適当な記号によって等号，不等号を格納する。すなわち +1, 0, −1 の3種の値をとる記号を ε とすれば

$$\left.\begin{aligned} &f_{ij}>0 \text{ のとき } \varepsilon_{ij}=+1 \\ &f_{ij}\geqq 0 \text{ のとき } \varepsilon_{ij}=0 \\ &f_{ij}=0 \text{ のとき } \varepsilon_{ij}=-1 \end{aligned}\right\} \tag{20}$$

とおき，PIJ と対の記憶場所を EIJ ときめて

$$\mathrm{EIJ}=\varepsilon_{ij}$$

とする。

式(17)のパターンマトリクスはサブルーチンとして格納しておき必要に応じてメインプログラムに呼び出す形にする。そのとき図13のように P_{ij} の選択プログラムを含めて構成すると便利である。ここで [SUBIJK] には PIJ と EIJ が [SUBSK] の選択に応じて，いつでもとりだせ

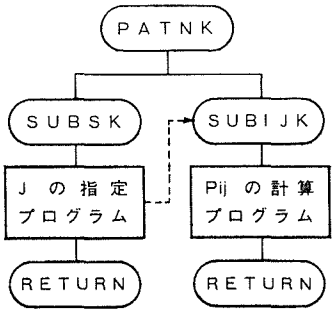


図13 パターンマトリクスのサブルチンプログラム

るように，マトリクスの列を一つのサブプログラムとして整然とプログラムされねばならない。たとえば FORTRAN ではつぎのようにするとよい。

```
        SUBROUTIEE SUBI 1 K
        GO TO (ν1, ν2, ……, νl), i
        PIJ=f11 (x, x2, ……, xn),
        EIJ=ε11
ν1      RETURN
        PIJ=f21 (x1, x2, ……, xn)
ν2      EIJ=ε2
 ⋮      RETURN
 ⋮      ……
 ⋮      ……
νl      PIJ=fl1 (x1, x2, ……, xn)
        EIJ=εl1
        RETURN
        END
```

[SUBI 2 K], [SUBI 3 K], ……と j については 1 から m までが同様にしてプログラムされる。

このようにしてプログラムされた [SUBIJK] はパターンマトリクスの第 j 列に相当し，しかも i によって，この列の第 i 番目の要素 P_{ij} が選び出されて，その PIJ と EIJ が x_1, x_2, ……, x_n のある値に対して計算されるしくみになっている。すなわちステートメント [GO TO (ν_1, ν_2, ……, ν_l), i] はステートメント番号 ν_i の命令から実行せよとの意味であるから，ν_i に PIJ と EIJ のプログラムが書かれてあればそれが実行される。

上記のプログラムで，たとえば $i=2$ ならば，左側に記されたステートメント番号 ν_2 から実行されるので，まず f_{21} が

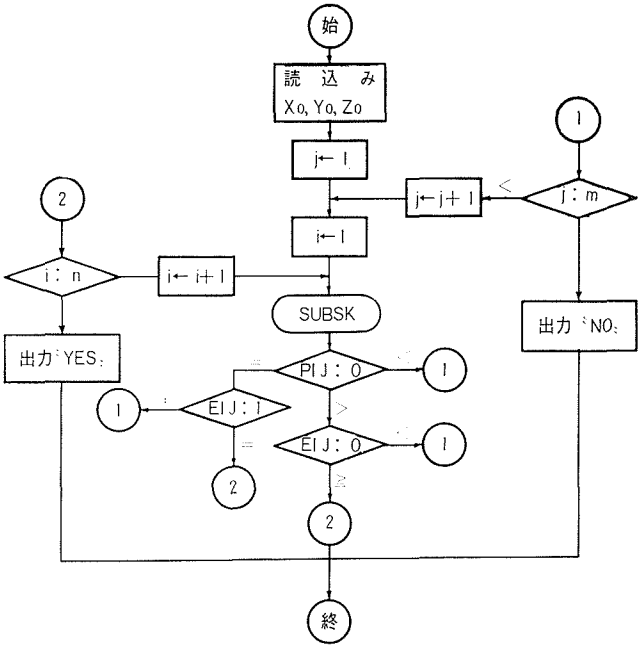


図14 点 (x_0, y_0, z_0) が図形 P 内にあるかどうかの判定プログラム

計算され，つづいて EIJ が ε_{21} になる。そのつぎのステートメント [RETURN] は，このサブプログラムを呼出したもとのプログラムに帰れとの意であるから，結局このサブプログラムでは P_{21} だけが計算されることになる。

つぎに，パターンマトリクスの列，すなわち j を選択するためのプログラムは同様の考え方をして

	SUBROUTINE SUBSK
	GO TO (ω_1, ω_2, ⋯, ω_m), j
ω_1	CALL SUBI 1 K
	RETURN
ω_2	CALL SUBI 2 K
	RETURN
⋮	.....
	.....
ω_m	CALL SUBI m K
	RETURN
	END

メインプログラムによって j を指定すると，ステートメント番号 ω_j のステートメントから命令が実行され，[CALL] ステートメントによって [SUBIJK] がよびだされる。

したがってメインプログラムにおいて，サブルーチン [SUBSK] の名前と，i, j を指定すれば P_{ij} をとりだしてその計算結果をメインプログラムで利用することができる。

たとえばある一点 (x_0, y_0, z) が図形 P 内にあるかどうかを判定し，あれば YES，なければ NO と出力させるプログラムを考える。そのフローチャートは図 14 のようになる。〔P_{ij}〕は第 1 列から順次各要素がとりだされて計算され，もし一つでも不満足になる要素があれば残りの要素を計算することなく，つぎの列にうつる。もし，ある列の全部の要素を満足すれば，残りの列を計算するることなく，YES と出力されて計算は終了する。どの列も満足しないときは最後の列を計算し終えてから NO と出力されることになる。

4-3. 変換関数による図形の発生

以上述べた数式化パターンおよびパターンマトリクスは図形処理のための基礎であり，以下にこれを使って，図形の発生および図形認識（次項）に対する試みを述べる。

ここにいう図形の発生とはコンピュータによって図形を造りだすことと限定する。さらにここではコンピュータのメモリ中の内容をプロッター，CRT ディスプレイなどを用いて図形として表わすこと自体には問題を考えず，人間の意志を計算機に伝達し，図形として表現させる方法を考える。従来，この目的のためにとられた方法は APT などの図形処理用言語を用いて意志を伝達するか，または CRT ディスプレイとライトペンで直接入力する方法が有力であ

った。これに対して筆者はさきのコンピュータ主体の対話方式とも関連して，コンピュータ側の数式の変換による修正によって意図する図形を発生させる方法を採っている。

これはつぎの2段階を経る。

(1) 原パターンとして意図する図形に類似し，単純な数式で表わされる図形をコンピューターに与える。

(2) 適当な変換関数によって修正を繰り返し，意図する図形に近ずける。

この過程をフローチャートで示すと図15のように判定と修正の繰り

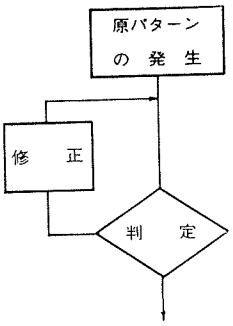


図15 修正によるパターンの発生過程

(a)

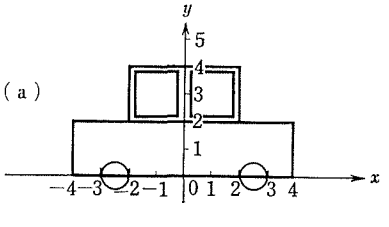
$$P_a=\{P_{a1}\ P_{a2}\ P_{a3}\ P_{a4}\}$$

$$P_{a1}=\left\{\left|\frac{x}{8}+\frac{y-1}{2}\right|+\left|\frac{x}{8}-\frac{y-1}{2}\right|=1\right\}$$

$$P_{a2}=\left\{\left|\frac{x}{4}+\frac{y-3}{2}\right|+\left|\frac{x}{4}-\frac{y-3}{2}\right|=1\right\}$$

$$P_{a3}=\left\{|(x\pm1)+(y-3)|+|(x\pm1)-(y-3)|=\frac{8}{5}\right\}$$

$$P_{a4}=\left\{\left(x\pm\frac{5}{2}\right)^2+y^2=\left(\frac{1}{2}\right)^2\right\}$$



(b)

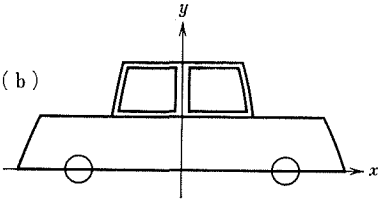
$$P_b=\varphi_b P_a,\quad P_b=\{P_{b1}\ P_{b2}\ P_{b3}\ P_{b4}\}$$

$$P_{b1}=\left\{\left|\frac{5x}{4(15-y)}+\frac{y-1}{2}\right|+\left|\frac{5x}{4(15-y)}-\frac{y-1}{2}\right|=1\right\}$$

$$P_{b2}=\left\{\left|\frac{5x}{2(15-y)}+\frac{y-3}{2}\right|+\left|\frac{5x}{2(15-y)}-\frac{y-3}{2}\right|=1\right\}$$

$$P_{b3}=\left\{\left|\left(\frac{10x}{15-y}\pm1\right)+(y-3)\right|+\left|\left(\frac{10x}{15-y}\pm1\right)-(y-3)\right|=\frac{8}{5}\right\}$$

$$P_{b4}=\left\{\left(x\pm\frac{15}{4}\right)^2+y^2=\left(\frac{1}{2}\right)^2\right\}$$



(c)

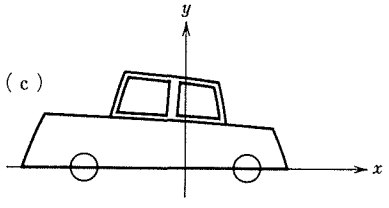
$$P_c=\varphi_c P_b,\quad P_c=\{P_{c1}\ P_{c2}\ P_{c3}\ P_{c4}\}$$

$$P_{c1}=\left\{\left|A+\frac{B}{2}-\frac{1}{2}\right|+\left|A-\frac{B}{2}+\frac{1}{2}\right|=1\right\}$$

$$P_{c2}=\left\{\left|2A+\frac{B}{2}-\frac{3}{2}\right|+\left|2A-\frac{B}{2}+\frac{3}{2}\right|=1\right\}$$

$$P_{c3}=\{|8A\pm1+B-3|+|8A\pm1-B+3|=1.6\}$$

$$P_{c4}=\left\{\left(x\pm\frac{15}{4}\mp\frac{45}{64}-\frac{801}{1024}\right)^2+y^2=\left(\frac{1}{2}\right)^2\right\}$$



ただし $A=\dfrac{5(13-4\sqrt{10-x})(2+\sqrt{10-x})}{6(10+5\sqrt{10-x}-2y)}$,

$B=6y/(2+\sqrt{10-x})$

図16　変換関数による図形の発生

返しであり，図形発生に限らず，デザインにおける基本的なプロセスである。

いま原パターン P を変換関数 φ によって修正し，新しいパターン P' を得たとし，これを，

$$P'=\varphi P \tag{21}$$

として表示する。φ の内容としては一般には x を $f(x, y)$, y を $g(x, y)$ と置きかえることである。1例をあげると，図16のように最初単純な式 P_a で表現されるパターンとして，図の(a)を仮定し，これに φ_b として

$$\begin{cases} x\rightarrow 10x/(15-y) \\ y\rightarrow y \end{cases}$$

なる変換を行うと，

$$P_b=\varphi_b P_a$$

の結果，図の (b) を得る。さらに φ_c として，

$$\begin{cases} x\rightarrow 26-8\sqrt{10-x} \\ y\rightarrow 6y/(2+\sqrt{10-x}) \end{cases}$$

を用い，$P_c=\varphi_c P_b$ の結果，図 (c) のようなややスマートな図形を得る。

ただし，車輪に相当する円については変換の対象から除外した。

変換関数をあらかじめ多数用意しておき，順次変換に利用することによって，つぎつぎと新しい図形を発生させることができる。しかもこの場合の大きな特長は，最初のパターンが数式化されておれば，何回変換を繰り返しても発生した図形は常に数式化されていることで，コンピュータ内で取扱いを著しく容易にするものである。

4-4. 図形認識の試み

自動設計のシステムの中で図形認識もまた重要な問題である。本論文で提案している方式では図形を数式化してとり扱うところに特徴があるので，図形認識はとりもなおさず図形を数式として認識することにほかならない。筆者が試みているのはどんな複雑な図形でも認識することを目的として走査により，黒白の境界点を全部認識し，これを処理して数式を求めようとする方法である。

図17のように図形と走査線の交点1，2，3，……を走査線の始点からの距離として，すなわち1次元の座標として認識し，磁気テープ中に，あるいは CPU のメモリ中に直接格納する。コンピュータ内ではこれを2次元座標に直し，さらに各点を結ぶ曲線を適当な数式化できる曲線の集合におきなおす。すでに直線集合，円弧集合によって近似

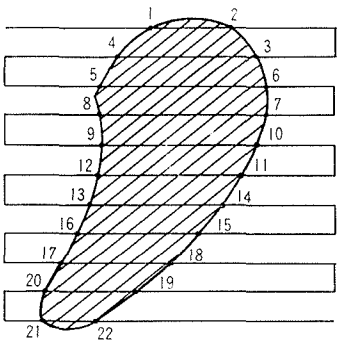


図 17 走査による図形認識

する方法を実験し報告している[3]。

以上は完全な図形認識に対するアプローチであるが，先述のコンピュータ主体の対話方式による自動設計中で取扱う図形認識ではより簡単な方法を用いることが可能である。すなわち設計の標準化が進むと，設計結果として得られる製品の形状に関してもある程度標準化されること，および工業製品そのものが，比較的単純な形状の要素を組み合わせて構成されているものであることに着目して，すでに数式化された標準形状要素の組み合せとして認識させる方法である。

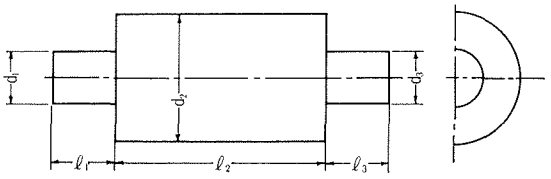


図18 3個の円柱の組み合せ

たとえば図18のような形状を考えると，これは3個の円柱をその中心線を一致させて並べたものである。直径 d，長さ l の円柱は数式化パターンとして次式のように表現される。

$$P_c=\left[\left(x^2+y^2\leqq\frac{d^2}{4}\right)\cap\left(L\leqq Z\leqq l\right)\right] \tag{21}$$

ただし，円柱端面から L の距離に原点をもち，円柱の中心線が Z 軸に一致するような直交座標系 $(x\text{-}y\text{-}z)$ を設定した。

図18のように中心線を共有する円柱の列をなすような形状は，式 (21) で示される円柱を標準形状要素として，その組み合わせによって認識することができる。式 (21) は d と l および L を与えるとその寸法もきまるので，記号 $P_c(d,\ l,\ L)$ によって式 (21) の左辺を表わすものとすれば図18の場合

$$P_c=\left[P_c(d_1,\ l_1,\ L_1)\cup P_c(d_2,\ l_2,\ L_2)\cup P_c(d_2,\ l_3,\ L_3)\right] \tag{22}$$

すなわち，式 (22) の形で認識することによって，コンピュータは図17を寸法可変の状態のまま把握したことになる。寸法については自動設計の過程において決定されるものである。

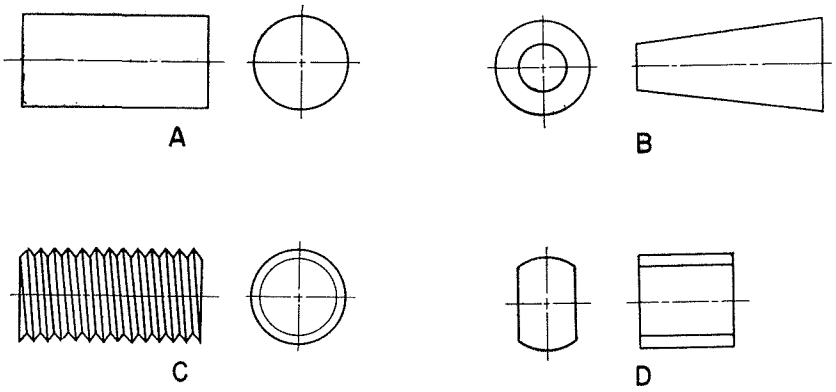


図19 軸の設計のための標準形状要素

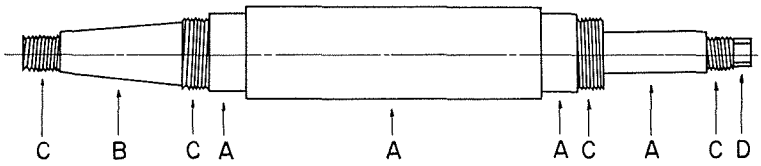


図 20　標準形状要素の組み合せ

さきに例として示した軸の自動設計では図 19 のような標準形状要素を用い，A, B, C, D なる記号で代表させると，たとえば図 20 のような軸であれば，CBCAAACACD の順序をもった記号列としてその形状をコンピュータに把握させることができる。

このような方式は図形認識といえるものではなく，設計者の形状に関する意志をコンピュータに伝達する一つの手段で，APT のような一般的言語を用いない方法であるといった方が現状では適当である。しかしこの方法は認識し得る小数の標準形状要素をコンピュータ内に用意してある点で文字認識などと同種のものであり，自動設計にもこの考え方を導入できることを述べたわけである。

5. データ処理の方式

自動設計におけるデータ処理，特にはじめに述べたコンピュータ主体の対話方式によって自動設計を行う際のデータ処理は，主として多量の設計用データを磁気テープにあらかじめ用意することと，これを必要に応じて中央処理装置にとりだし，設計計算に組込む仕事である。普通は表の形でデータを記憶するが，大量になると記憶容量が問題であり，さらに必要なデータの検索など取扱いが複雑になる。

筆者はデータ処理に対しても図形処理の場合同様，近似的に数式化して記憶装置内に格納し，これより必要なデータをとりだす場合には，式を計算して数値を得る方法を試みている。

図形の存在する空間は一応 3 次元であるが，データ空間は n 次元になる。すなわち，前項での数式化の方法は n 次元でも全く同様であり，単に変数が増えたものとして取扱えばよいので，数式化パターンおよびパターンマトリクスとして扱うことができる。

6. 設計解の発見と最適値探索の方式

対話方式，図形処理，データ処理につづいて自動設計において最後に問題になるのは設計解の発見と最適値探索の方式である。なおここにいう設計解とは設計のための制約条件を満足する設計諸元を意味する。設計においては最適値を得ることが最善ではあるが，設計解を発見することも次善の策である。現に普通に行なわれている設計は最適値ではなく，制約条件を満足するだけで十分としている。自動設計においてもまず設計解を求め，つづいて必要あれば最適値を求めるという 2 段階に分割する方法が効率的である。

ここではコンピュータ主体で設計を進めるに都合のよいように上記の 2 段階を機械的に進

める方法について述べる。

まず設計解の発見のためには田川氏（北大・工，この研究は精密工学科在職中に行なわれた。）が極値探索に用いた方法[4]に類似の方法を使用する。いま制約条件を

$$\left.\begin{aligned} &P_1=[G_1(x_1,\ x_2,\ \cdots\cdots,\ x_n)\geqq 0]\\ &P_2=[G_2(x_1,\ x_2,\ \cdots\cdots,\ x_n)\geqq 0]\\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\ &P_R=[G_R(x_1,\ x_2,\ \cdots\cdots,\ x_n)\geqq 0] \end{aligned}\right\} \tag{23}$$

の形に整理する。これはパターンマトリクスとして，

$$P=\left(\left(\begin{array}{c} P_1\\ P_2\\ \vdots\\ P_n \end{array}\right)\right) \tag{24}$$

のようにまとめることができる。制約条件の中には A or B のような形のものもあるので，一般には式(17)に示すパターンマトリクスの一般形にまとめられるが，1列の場合の繰り返しと考えることができるので，式(24)を考えると十分である。

いま，

$$R=\begin{cases} 0 & (G_i\geqq 0 \text{ のとき})\\ G_i & (G_i<0 \text{ のとき}) \end{cases} \tag{25}$$

で定義される R_i を用いて

$$Y=\sum_{i=1}^{n} C_i R_i^2 \tag{26}$$

なる関数 Y を定義する。ここで C_i には十分大きな正の数をとる。Y は図21に見るように制約領域の外周に作った壁である。適当な地点から出発して制約領域を探す場合には壁に沿ってすべり降りるならば，容易に制約領域内に到達する。すべり降りる方法は極値探索のために開発された種々の方法をいずれも使用できるが，われわれは規格の不連続性を利用した単一変数変化法を試みている。これは変数の変化幅を規格の幅に一致させて探す方法で，変化幅を大きくとれるので探索に要する時間は単一変数変化法でもあまり大きくならないですむ。

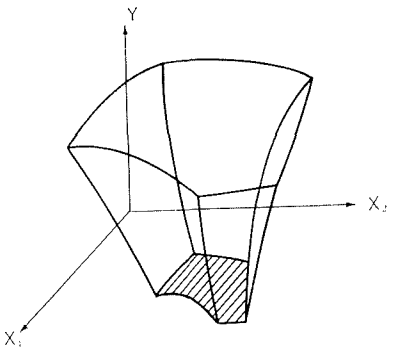


図21 制約領域外周の壁

制約領域内に到達したかどうかは Y が0になることから知ることができる。制約領域が存在しない場合には，同じ試行を繰り返す状態になっても Y が0にならないので，そのことを知る。制約領域に到達後は Y をあらためて評価関数におき直して同じ単一変数変化法を適

用する。

7. 自動製図および自動加工との結合

設計結果は最適な設計諸元として得られるが，この中には形状の寸法に関する諸元を含んでいる。これより自動製図機あるいは数値制御加工機を動作させるための指令テープを作成するには，従来，APT などの図形作成用言語によってプログラムし，これを APT コンピュータに翻訳していた。前項までに述べてきたコンピュータ主体に進める自動設計の方式ではこの場合にも APT のような言語を一切使用せず，得られた設計結果から指令テープを作成する作業は人間を介することなくコンピュータ側に行わせる。

すなわち，図7の D_1，D_2，……，D_l には標準化された形状が，未確定の定数を含んだ数式化パターンとして格納されている。最適設計の過程を経て未確定の定数が決まる，とその形状は寸法が確定し，しかも数式表示されている。この数式より指令テープを作製することの可能性については論を待たない。しかし具体的方法については現在二三の方法を実験中であり，結論を得ていないが，実用性についての見通しを得ている。図17に示されたロールの加工を例にとると，これを旋盤で削る場合，刃物を中心線に平行に走査すると同じ動きをさせ，図13の方法によって決められた図形内に刃物がくい込まないよう指令をきめるのも一法である。また刃物の動きをより能率的にするために，3個の円柱を一つ一つ仕上げるように刃物の動きを制限することも可能である。いずれにしてもこのようなプログラムは場合に応じて最適のものをあらかじめ磁気テープ中に納めておくことが可能である。

8. 結　　言

以上，まだ研究が完成された状態ではないが，自動設計システムに関する一つの新しい方式について，現在進行中の研究の骨子を述べた。御批判をいただければ幸いである。なお，この研究のために，本学精密工学科の諸先生から多大の御支援，御指導をいただいている。本稿を借りて深甚の謝意を表する。

引 用 文 献

1) I, E. Sutherland: Proc. SJCC, 1963, p. 329.
2) R, W. Mann: Mechanical Engineering, May, 1965, p. 41.
3) 沖野・斎藤・村上：情報処理学会第9回大会予稿集.
4) 田川：Proc, IFAC Tokyo Symposium, 1965, p. 31.