

# 橋梁生産工程における数値制御システム

—主として図形処理システムについて—

## THE APPLICATION OF NUMERICAL CONTROL SYSTEMS IN BRIDGE DESIGN AND PRODUCTION

田中征登\*・高久達将\*\*・渡部 孝\*\*\*・合力俊郎\*\*

By Yukitaka Tanaka, Tatsumasa Takaku, Takashi Watabe and Toshiro Goriki

### 1. 緒 言

ここ数年間における、数値制御（以下 NC-Numerical Control）による工作機器の普及発展は著しいものがあるが、労働集約的組立産業といえる鋼橋生産部門においては、まだ一部の普及に留まっている。

日本の造船界では、ノルウェーのオートコン・システムに刺激されて、原図を中心とした NC 化の開発に重点がおかれ、大手各社独自に図形処理言語開発を核として、生産部門の合理化を押し進めている。

数値制御とは具体的には機械の運動を数値的情報により制御することなのだが、それを生産体系に取り入れていくためには、ハードウェア、ソフトウェア両面の強力なサポートがもちろん必要である。

現在の数値制御のハードウェアの入力機構をみると、入力の数値情報の形式を、その機械の機能に合わせて独自に指定している。入力形式はハードウェアが 10 台あるとすると 10 台すべてが異なっている。これは図形情報の入力形式の標準化、規格化の作業がなされていないという指摘も妥当であるが、それ以上に、図形処理の複雑性を端的に物語っているといえよう。

ハードウェアのメーカー側からいえば、専用の数値制御機構を開発し、売り出せばよいのだが、ユーザー側からいうと、いろいろな機種の入力を別個に作る作業は、やっかいな仕事となる。したがって、個々のハードウェアを総括的に運用していくソフトウェアの開発はユーザー側の負担となってくる。そこで、NC を前提とした生産体制を企画するとき、個々のハードウェアを端末と考え、汎用的な図形処理を行なうソフトウェア開発のための、システム設計の必要が生じる。NC 即図形処理

ということを見ると、造船各界が NC のシステム設計の中心を図形処理言語開発においたのも当然の成りゆきといえよう。

当社では、橋梁を始めとする重工鋼構造部門の新鋭工場建設に当り、原図、内業加工の全面的な NC システムの採用を決定し、それ以降、図形処理言語 BRISTLAN (BRIDGE AND STEEL STRUCTURE LOFTING LANGUAGE) の開発を中心に、NC 一貫処理体系のシステム作りに専心してきた。BRISTLAN の開発により、電子計算機、自動作図機、およびその他の NC 機器を使用して、橋梁をはじめとする鋼構造物の原図作業、切断作業などの省力化が可能となる。更に自動設計的な立場から、設計からの情報、橋梁の基本線形プログラムの出力を結びつけることを考慮し、設計一原図一内業に至る生産工程の一貫した自動化を企画した。

テストケースとして数橋の実施を経験し、実用に十分供し得ることに自信を得たので、図形処理言語 BRISTLAN を中心に橋梁の生産工程の NC システム構成をここに紹介したい。

### 2. システム概要

橋梁上部工の工場製作工事は受注生産をたてまえとした多種少量生産の形式をとっている。現在の NC システムを採用しない鋼橋の設計製作にいたる生産工程みると大体 図-1 のようになる。

この中で生産合理化を意図するとき、その方向は二つに大別できよう。

#### 設計製作自動化の対象

自動設計, NC 原図……ソフトウェアの開発

内業加工の自動化……NC 装置機器の開発

#### 生産管理の対象

加工, 組立の工程計画, 工程管理, 能率管理

……………ソフトウェアの開発

\* 正会員 日本鋼管鋼構造部設計室係長

\*\* 正会員 日本鋼管鋼構造部設計室

\*\*\* 日本鋼管津造船所重工工作部重工計画室

このうち生産管理は別のプロジェクトと考え、設計製作自動化を対象と考え、システム設計が行なわれた。システムの主眼は設計、NC 原図、内業加工の各工程が、

スムーズに有機的に結合されることにおかれた。システムの概要を図-2 に示す。その特徴を列挙すると、次のようになる。

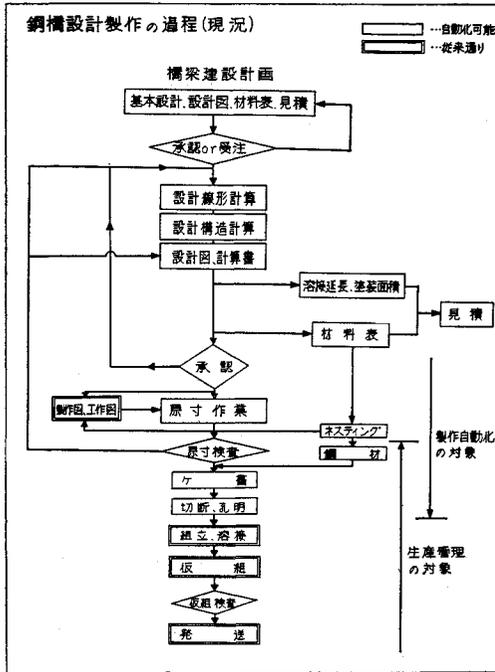


図-1

### (1) 図形処理言語 BRISTLAN の開発

システムの中心は BRISTLAN の開発で、その入力データ形式に合わせて前後の処理プロセッサが組み立てられた。BRISTLAN は問題向け言語 (Problem Oriented Language 略して POL) の形体をとり、図形定義の記述を FORTRAN, COBOL, PL/1 といった手続き向け言語 (Procedure Oriented Language) のレベルから引き上げ、原図作業者がパートプログラムすることにより、容易に図形情報のインプットデータを作成できるようにした。

### (2) 線形計算

橋梁設計における線形計算の比重は大きい。特に都市交通路の線形は複雑多岐化し、橋梁は、その一部を構成する要素と考えられるので、その線形計算は非常にむずかしくなる。

自動設計として線形計算の部分強度計算から独立させ、図形処理一貫システムの中に組み入れ、設計用の出力として、また NC 原図の入力として使用できるように線形計算プログラム (ROAD) が企画され開発された。

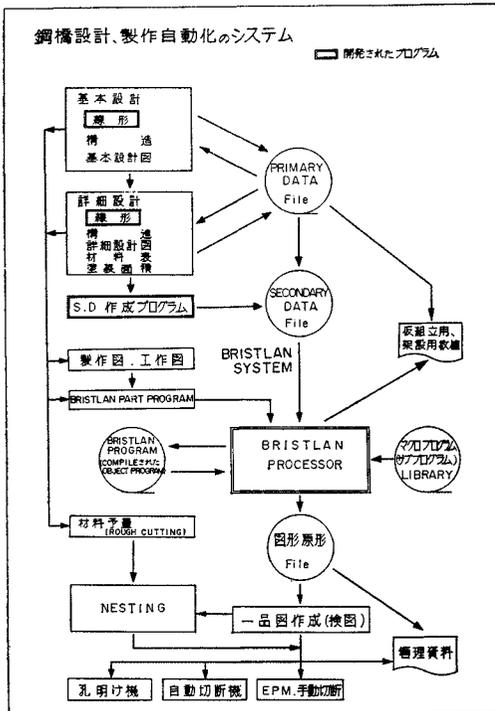


図-2

設計に必要なデータは 1 次データ (Primary Data) としてファイルされ、強度計算に必要なときに参照される。さらに NC 原図用データとして、1 次データを修飾して、2 次データ (Secondary Data) が作成される。橋梁は一般に最終でき上り構造線形に製作上のそり (キャンパー) を考慮して原図作業を行わなければならない。キャンパーをつけることにより構造物の格点は微小に回転変位する。この変位量が加味され、すべての原図作業に必要な格点が、3 次元座標値として、2 次データとしてファイルされる。2 次データは個々のデータが配列 (Array) としてラベル付けされ、BRISTLAN のパートプログラマーが任意に参照できるような、データ構造となっている。

### (3) 図形原形の作成

図形原形は、すべて BRISTLAN 言語を用いたパートプログラミングによって作成されるが、コーディングされた BRISTLAN ソースカードを BRISTLAN PROCESSOR の主入力と考えると、そのほかの補助入力として 2 つ考えられる。1 つは前処理された線形の格点 3 次元座標値であり、もう 1 つは、図形パターンを集めたライブラリーの参照である。図形はパターンの集合

である。しばしば現われる図形および図形要素のパターンは、ライブラリーとしてあらかじめプログラムを作っておき、適時に呼び出せば便利である。したがって BRISLAN PROCESSOR は、図形編集機能につけ加えて、2次データの参照と図形ライブラリーの参照、すなわちサブプログラムの呼び出しが可能である。

(4) 図形の NC 情報としての出力

BRISLAN PROCESSOR により作成された図形原形は、各種ハードウェアに応じた数値情報に変換される (POST PROCESSOR 処理)。

図形は、正しく編集されているかどうか自動製図機を用いて検図される。

正しく編集された図形情報は、次のような各種工程にふり分けられて、内業加工工程にわたされる。

(イ) 自動製図機による 1/10 縮尺原面の作成 (一品図)

大板としての集合作業

(現在手作業による NESTING)

拡大投影機 (EPM 装置……ELECTRONIC PHOTO MARKING) による 1/1 実寸への鋼板上への焼付け (ケ書作業の自動化)

(ロ) 自動ガス切断機による直接鋼板切断

(ハ) 自動孔明け機による直接孔明け

(ニ) 自動製図機による 1/1 のテンプレートの作成

(ホ) 仕上げ寸法定規, その他組立用数表の作成

3. 線形計算

(1) 概要

設計、製作にいたる生産工程全体の一貫した自動化という観点から線形計算プログラムが企画され開発された。プログラムは次の3つの部分に分かれる。

(1) 道路線形プログラム

(2) 桁形状決定プログラム

(3) 製作用データ作成プログラム

(1),(2) によって作成されるものは PRIMARY DATA と呼ばれるもので、設計計算書、設計図作成の

ための資料となる。それに対し (3) によって作成されるものは工場製作用データとして、SECONDARY DATA と呼ばれるもので、ファイルされ BRISLAN パートプログラムで参照される。以上の概略を図-3 に示す。

(2) 道路線形プログラム

A. 計算手順

計算手順を 図-4 に従い説明する。

(1) 道路情報の入力

道路の基本線形の情報を入力する。

(2) 道路中心線

最初に道路中心線の平面線形を求め、その他の線形計算の基準とする。

(3) 地覆線

平行、拡幅の条件より道路中心線を基準にして、地覆線を定める。

(4) 縦断、横断勾配

道路中心線、地覆線を関連させ縦断、横断勾配を決める。

(5) 橋台、橋脚線

橋台、橋脚の線をいれて各スパンに分割する。

(6) 構造情報の入力

上述のステップにより道路の基準線が決まったので、次に構造情報を与え主桁、横桁の配置決定の入力データとする。以下スパンの数だけ繰り返しが行なわれる。

(7) 支承線

橋台、橋脚の中心線を基準にして支承線を求める。

(8) 主桁、横桁の交点

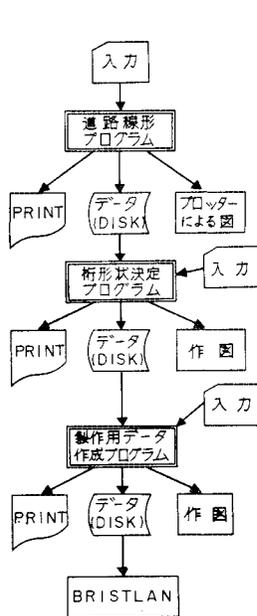


図-3

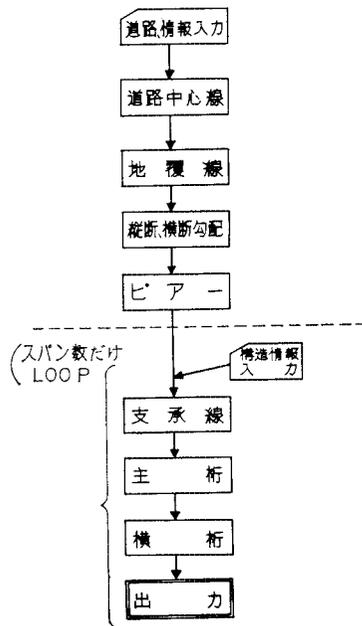


図-4

縦断、横断勾配を考慮し主桁、横桁の交わる道路面を3次元座標値の点列として求める。

#### (9) 出力

出力はスパン単位に行なわれる。主桁位置および横桁取付位置が3次元座標値として求まり、ラインプリンターにより数表が出力される。同時に、これらの3次元座標値は以降の計算に用いられるようPRIMARY DATAとして2次記憶装置(DISK)上にファイルされる。

#### B. 道路線形の取扱い

道路線形を構成する線分の要素として、ここでは次のものを取扱った。

- (1) 直線, 円, クロソイド
- (2) 直線, 円, クロソイドの平行線

各線分は関数で表現され、パラメーター表示がなされる。道路線形は上記の要素の集合として把握される。1つの線形は関数のパラメーターとして表示される線分の組み合わせとして、1つのテーブルを作成している。

#### (3) 桁形状の決定

3.(2) で求められた道路線形座標値をもとに桁形状を決める。桁形状は製作上のそり(キャンパー)を加味して作られる。キャンパーが主桁ごとに異なるものは、キャンパーのちがいをハンチで調整し、製作工数の低減を図れるようにした。形の標準化、統一化のためには、具体的には主桁の縦断勾配、キャンパーを等しくして複数個の主桁の形状を同一とすること、および主桁間の相対的な高低差  $\delta$  (図-5 参照) を等しくして横桁の形状を同一とすることが考えられる。もちろん形状の同一化にも制限があるが、小量のちがいはハンチで逃げるのが有効である。その調整の手法は大体次のようになる。

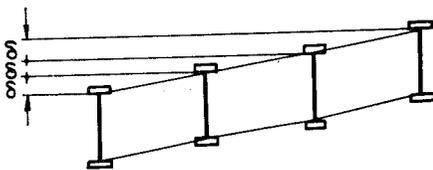


図-5

(1) 主桁  $m$  本、横桁  $n$  本によって構成される構造を考える。第  $i$  番目の主桁線と第  $J$  番目の横桁線の交点を  $Z(i, j)$  とする。 $Z$  は  $(m, n)$  行列で表示される。

(2) このとき  $Z$  は主桁線、横桁線で構成される一つの曲面と考えられる。

(3) この曲面の横方向(横桁方向)の高低差が一定になるような面を見つける。このとき各主桁上のハンチ量は面のとり方により変化する1つの関数で表現される。

(4) これは許容できる最小および最大ハンチ量を与えて、全主桁のハンチ量のうちの最大値を最小にする問

題を解くことに帰着する。

#### (4) 製作用データ作成プログラム

桁形状決定プログラムにより主桁に沿っての座標値が求められた。さらに工場製作用データとして、次の個所についての3次元座標値を求めSECONDARY DATAとしてファイルする。

- (1) 主桁ウェブの添接位置
- (2) 主桁ウェブの板つき位置
- (3) スチフナー取付位置
- (4) 横構取付位置

(1),(2),(3) については、主桁ウェブの上端、中立軸、下端について求める。

キャンパー値(死荷重によるたわみ量)以外に、工場での製作過程上必要な製作上のそりがさらに加味される場合がある。これはスタッドジベル溶接による上フランジ側の一方加熱、または合成桁にみられるように、上下フランジの板厚の差が大きく、溶接時の入熱量のアンバランス等により桁にそりが生じる場合、変形量をみこして、あらかじめ逆そりを与える場合である。上記の座標値には、この逆そりの要素を考慮することができる。

#### 4. 自動作図機

NCシステムでは、自動作画機は必須のものである。点、線の集合として、編集されている図形は、一度人間の目で確かめられねばならない。検図としては、高速プロッター、自動作画機が使用されるが、近い将来はCRT(Cathod Lay Tube)上への投影も有効な手段となろう。

現在当社が使用中の自動作画機の種類、およびその機能を表-1に示す。

#### 5. BRISLAN 開発の重点

- (1) 橋梁、鉄構造物の図形特性を考慮した図形処理言語の開発

図形処理言語としては、APT(Automatically Programmed Tools)をはじめとする種々の言語が存在する。当社内にもすでに造船原図NCに多大の役割をはたしている図形処理言語 LOFTRAN(Lofting Language)があるが、われわれが既成品を使用しなかった最大の理由は、対象とする構造線形の相違によるものである。

対象構造物、あるいは工作法によって、その機能が変わってくるという意味では、図形処理言語は、問題向け



(Problem Orient) であり、狭視的である。

橋梁上部工は、桁のウェブ材を除いて、ガセット、スプライス、スチフナーといった小物部材が多く、しかも図形的にみて同型部材が多い。したがって NC システムで能率よく処理するためにはひとつの基本図形の記述を行ない、さらに同形図形は微小に変化する量を変数として与え、同時に多品図形を得るという機能、すなわち繰り返しの命令が必要になる。

また橋梁は現場架設という宿命を背おって添接部の接合にリベット、ボルトを用いているため、孔の数が非常に多い。

これは2次元的な図形定義とは異質なもので、孔の位置決め情報として別個に把握しておく必要がある。

(2) 床書き原図作業を完全にカバーできるもの

従来の床書き原図作業の内容は大体次のように要約できる。

(イ) 設計図上の不確定寸法の決定、または部材相互位置の確認

(ロ) 全体線形図のチェック

(ハ) 部材の展開、実寸の部材形状の決定

(ニ) 定規、型板等のケ書情報の作成

(ホ) 組立用定規または仮組用数表の作成

(ヘ) 部材表等の管理資料の作成

完全 NC 化を意図するとき上記の事項がすべて満足できるようなシステムでなければならない。このため、BRISTLAN の言語体系では次のような機能が特別に考慮された。

a) 3次元座標値の取扱い

構造物はすべて立体的である。われわれは製図するとき、それを2次元的に表現するが、原図作業は3次元的なものを2次元に変換し、さらに実長を求めるといふ一種の展開作業といえる。任意の構造線形より2次元的な部材を抽出しようとするときは、あらかじめ部材の空間平面を規定する格点座標値が3次元的に求まっていなければならない。われわれの2次データは、実はそのために作成されているのだが、その座標値をもとに空間絶対座標系内に、2次元小座標系を任意に設定し(PLANE

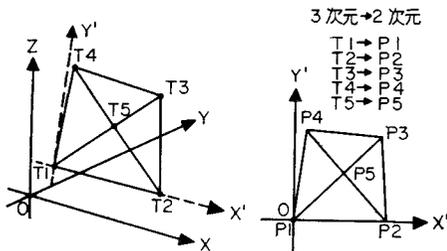


図-6

の定義) 平面部材を展開、抽出できるようにした。展開の原理を図-6 をもとに説明する。

(イ) 大座標系 (X-Y-Z) 内に 5 点 T1, T2, T3, T4, T5 が 3 次元座標値として与えられている。5 点は必ずしも同一平面上になってもよい。

(ロ) 3 点 T1, T2, T3 を用いて、展開面としての 2 次元座標系を定義する。このとき、T1 が原点、 $\overrightarrow{T1 T2}$  が X 軸方向となる。

(ハ) 4 点 T1, T2, T3, T4 を用いて展開面を定義することも可能で、このときは 4 点 T1, T2, T3, T4 からの距離が最小となるような平面が自動的に定義される。

(ニ) 定義面内で、3 次元座標値を用いると自動的に面上への正射影という形で 2 次元変換がなされる。

(ホ) 正射影の代わりに透視射影を定義することにより、変換された 2 次元座標値を用いて、線形の透視図を得ることができる。

以上の機能を言語体系に組み入れることにより 3 次元的な線形座標のチェック、および部材の展開作業を可能にした。

b) 図形上の論理演算の導入

設計図上の不確定寸法の決定、または部材相互の当りのチェックのためには図形相互の論理判定が必要である。

たとえば、ガセットの形状は、その取付く部材形状、または型鋼の方向により決定されるし、さらにそのボルト孔は、孔耳を十分意識して位置が決定される。

こういった問題の解決には次のような論理演算機能が導入された。

(イ) 領域判定 (図-7 参照)

線に方向性を持たせ(ベクトル表示)、与えられた点はその線の進行方向に対して右側にあるかどうかの判定。

(ロ) ON, OFF の判定 (図-7 参照)

与えられた線上(ある精度の領域内)にあるかどうかの判定。

(ハ) 交点数の判定 (図-7 参照)

2 線の交点数の判定、根なし、1 根、2 根、あるいは 3 根以上等の判定。

(ニ) スカラー量の大小判定

スカラー量の大小または等しいかの判定。



図-7

### c) 管理資料, 数表の出力

BRISLAN は図形編集をその主目的とするが, 補助出力として管理資料, 数表の出力を考慮した(表-2 参照)。管理資料は, 一部材ごとに作成された図形情報をもとに自動的に編集される。その内容は, 部材の面積, 重量, 板取用最大寸法 (X 方向, Y 方向寸法), 孔個数, 切断長などである。

そのほかに, 組立用仕上寸法, 仮組立用座標値の出力が必要であるが PRINT の COMMAND を用いることにより演算結果のスカラー量の出力が可能である。

### (3) ソフトウェアとハードウェアの有機的な結合

NC システムでは, 作成された図形原形がいかにか上手に種々のハードウェアの NC 情報に変換されるかが大きな問題になる。

1 つの図形情報に対し, 10 種の NC ハードウェアがあれば, 10 種の変換用の後処理プログラム (POST PROCESSOR) が必要であるが BRISLAN では, 編集される線にあらかじめカテゴリーを与え, 各種 NC 情報に分解抽出できるようなデータ構造にした。

線のカテゴリーは次の 5 つに大別できる。

- |                          |        |
|--------------------------|--------|
| (イ) 外周線 (必ず反時計まわり)       | } 切断線, |
| (ロ) 内周線 (必ず時計まわり)        |        |
| (ハ) ケ書線                  | } 作画線  |
| (ニ) 注記 (ピース番号, 作業指示等の文字) |        |
| (ホ) 孔位置情報……孔明け 作画線       |        |

### (4) 標準図形のライブラリーとしての登録

一工事に必要とする, パートプログラミング, インポートデータの量はかなり多い。各工事ごとに最初からこれらのすべてのインプットソースを作成することは, 作業の能率, あるいは経済性を考えると, 採算ベースに現状では合い難い。そこで過去のデータを保存し, 現工事に使用できるものは適用していくという方法, 換言すれば, コンピューターに学習させるということが絶対必要になる。

コンピューター学習に対しては具体的には, 過去のプログラムをメインプログラムとして持つこと, およびサブプログラムとして持つことが考えられる, 両者ともライブラリーとして, 2 次記憶装置上に登録される。

2 次記憶装置上に登録されたプログラムを使用するためには, 言語開発上次のような必要性が生じた。

(1) サブプログラムをライブラリーから呼び出し, メインプログラムに結合するための連けい編集プログラム (LINKAGE EDITOR) の作成。

(2) ライブラリープログラムをソースプログラムで保存するのではなく, コンパイルされ, コード化された

オブジェクトプログラム (OBJECT PROGRAM) として保存することの必要性。

BRISLAN では, コンパイラー, 連けい編集プログラムが開発された。

コンパイラーによりパートプログラムの文法チェックがなされ, 次に BRISLAN 特有のオブジェクトプログラムへの変換がなされる。さらに連けい編集プログラムによりメインプログラムとサブプログラムの結合がなされ, 実行可能なモジュール (MODULE) が作成される。

[ここでいうオブジェクトプログラムは通常の機械言語 (MACHINE LANGUAGE) レベルに変換されたモジュールでなく, BRISLAN 特有の形を持ち, コード化されているもので, 80 バイトのカードイメージが 16 バイトに縮小されている。]

## 6. BRISLAN 言語

### (1) 言語構成

BRISLAN 言語は BRISLAN 語いを用いた文 (STATEMENT) の連続で記述される。BRISLAN 語いは次の 5 つの要素よりなりたつ。

- (イ) ステートメントナンバー  
(STATEMENT NUMBER)
- (ロ) コントロールワード (CONTROL WORD)
- (ハ) シンボル (SYMBOL)
- (ニ) 定義語 (DEFINITION WORD)
- (ホ) オペランド (OPERAND)

BRISLAN 語いを用いて表-3 のような形式で文が書かれる。

表-3 は, 大別すると次の種類にわかれる。

- (イ) 総括的記述 (CONTROL STATEMENT)
- (ロ) 幾何学的記述  
(GEOMETRIC STATEMENT)
- (ハ) 特殊記述 (SPECIAL STATEMENT)
- (ニ) 論理的記述 (LOGICAL STATEMENT)

BRISLAN 言語は原則として座標系を定義し GEOMETRIC STATEMENT のシークエンスで, 図形および作図動作を表現する。また図形要素の定義と動作とを一つの文で表現することができる。すでに定義されたシンボルは, その後シンボルのみを記入してステートメントとすることができる。

### (2) 図形の基本要素

定義語によって定義される点, 図形, 変数等にはあらかじめ決められた文字を与え, シンボルに特性を与える

表-3

1	4 5	8 9	12 13	16 17	71 72
STATEMENT No.	CONTROL WORD	SYMBOL	DEF. WORD	第1~第5 OPERAND	
Column Restriction				Column Free	
					Continuation Mark

(Reserved Character)。

*P*: 2次元点

*T*: 3次元点

*S*: 直線

*C*: 円弧

*F*: 合成図形

*I~N*: 整数

*W*: キャラクター

*R*: Data Read 命令

### (3) ステートメントの種類

#### a) CONTROL STATEMENT

図形記述のステートメントのグループを規定する文をコントロールステートメントと呼ぶ。一般に、このステートメントはコントロールのみか、それにいくつかの補助情報を加えたもので、次のような種類がある。

- (イ) プログラムの定義, 終了 BSL, END
- (ロ) サブプログラムの定義, 終了 MCST, MCED
- (ハ) 図形の識別名称 JOB, PNUM, MATR
- (ニ) 線分の規定 CONT, LAND, MARK, MEMO, HOLE
- (ホ) データの宣言 DATA, ARRY, EXTS
- (ヘ) 座標系の設定 PLN 1, PLN 2, PLN 3
- (ト) 説明用コメント CMMT
- (チ) 定義のみの宣言 NON
- (リ) 出力の規定 TURN, SCAL, DRFT, REAL, DASH, CROS, DEBG, WRIT
- (ヌ) 図形シンボルの定義 STPT, ENPT, GDPT
- (ル) プログラムコントロール LPST, LPED, GOTO, IF

(オ) サブプログラムの呼出し CALL, LINK

#### b) GEOMETRIC STATEMENT

- (イ) 点の定義 PTCO, PTIT, PTGH, PTST, PTTN, PTNR, PTCC, PTRT
- (ロ) 直線の定義 LNVU, LNVD, LNHL, LNHS, LNST, LNTN, LNNR, LNPT
- (ハ) 円弧の定義 CR2P, CRPT, CR2C, CR2T, CR3P, CR3T

(ニ) 点列曲線の定義 FIGS, FIG 1, FIG 2

(ホ) 線の修飾 PARA, BACK, CUTF, JOIN, COMB, MIRR, MOVE, SHFT

#### e) SPECIAL STATEMENT

(イ) 距離を求める DISP, DIST, SPAC

(ロ) 角度を求める ANGL

(ハ) 半径を求める RADI

(ニ) 点の座標値 XCOR, YCOR, ZCOR

(ホ) 四則演算 ADD, SUB, MUL, DIV

(ハ) 標準関数 SIN, COS, ATAN, ASIN, SQRT, ABS, SIGN

(ト) 等号 ASSN

(チ) 座標変換 TRAS

#### d) LOGICAL STATEMENT

IF (CONTROL STATEMENT) とともに使用され論理演算を行なう。

(イ) 大小比較 EQ, GT

(ロ) 領域判定 ON, PLUS

(ハ) 交点の数 RTO, RT 1, RT 2, RT 3

### (3) サブプログラムとマクロ (MACRO)

あらかじめ用意された補助プログラムを一般にサブプログラムというが、BRISTLAN 言語では図形定義のためにあらかじめ作られた補助プログラムをマクロと呼ぶ。

BRISTLAN マクロは次の性格を持つ。

(イ) BRISTLAN の複数個のステートメントにより1つの図形情報が定義される。

(ロ) マクロは BRISTLAN で書かれたメインプログラムにより呼び込まれる。入力オペランドとして点、スカラーがあり、出力としてはシンボルとしての図形が出力される。オペランドの点、スカラーも補助情報として、マクロ内部で定義してメインプログラムに返すことができる。

(ハ) 座標系がメインプログラムの座標系とは独立に設定される。メインプログラムで呼び込み使用するとき、座標系相互間の移動がなされる。

(ニ) マクロプログラムは BRISTLAN 言語を用いて書かれ、メインプログラムの BSL, END に対し、MCST, MCED で区別される。マクロプログラム単独の出力はできない。プログラム作成上の文法的規約は、

メインプログラムと同じである。

(ホ) マクロプログラムはマクロプログラムを呼び込むことはできない。

BRISLAN マクロは、手続き向け言語 (Procedure Oriented Language) での機能と同様、サブプログラムとして任意に作成され、ライブラリーに登録され、適時に呼び出され、利用される。

マクロライブラリーは作成来歴から

A. すべての工事に共通な標準図形のパターン

B. 一工事のみに特有な使用頻度の多い図形パターンに分けることができる。B. は各工事ごとにそのつど作られる。A. としては、現在次のようなものが作成され、BRISLAN ライブラリーとして使用されている。

(イ) 開図形関係……18 種

スロット、スカラップ、自動ガス切断機に対する切欠き等

(ロ) 閉図形関係……22 種

マンホール、ドレンホール、台形、だ円、ガセット、多角形一般

(ハ) 孔作置決め……16 種

与えられた条件のもとで、2次元的な孔の配列を決める。

## 7. BRISLAN の構成

### (1) BRISLAN プログラムの構成

BRISLAN プログラムは、次の4つの部分から成立っている。

A. COMPILER (コンパイラー)

B. LINKAGE EDITOR (連けい編集プログラム)

C. BRISLAN PROCESSOR (図形処理プログラム)

D. POST PROCESSOR (後処理プログラム)

POL の一般的な機能としては、前処理としてのデータチェック機能があること、処理手順としては、COMMAND 単位に行なうこと、および COMMAND 単位で処理されたデータを編集整理する機能を持つこと等が最低限必要である。

BRISLAN COMPILER は、BRISLAN 言語で書かれた部分プログラムの文法チェックを行ない、コンパイルし、BRISLAN 特有の形式を持つオブジェクトプログラムを作成する。BRISLAN PROCESSOR に対する入力として、オブジェクトプログラムにした理由は、5. でも述べたがさらに次のように要約できる。

(イ) カラムフリー (Column Free) で書かれるソ

ース プログラムを整理し、最小限の情報を縮小し、プログラム全体を CORE にロードできるようにしたい。

(ロ) ライブラリーとしてはコンパイルの必要のないオブジェクト プログラムの形で保ち、LINKAGE EDITOR に対する入力の基本的な形式としたい。

BRISLAN PROCESSOR は、連けい編集された BRISLAN オブジェクト プログラムを入力データとし、必要があれば、線形プログラム (ROAD) で作成された 2 次データのファイルを参照して、図形を作成する。管理資料も同時に計算され、ファイルされる。

POST PROCESSOR は、編集された図形を入力データとし、自動作図機およびその他の NC 機器に対する情報を作成する。以上のステップを要約すると、図-8 のようになる。



図-8

### (2) BRISLAN による図形作成の手順

BRISLAN パート プログラムをインプットとして、図形が作成されるまでの作業手順を図示すると、図-9 のようになる。

## 8. BRISLAN 使用例

### (1) BRISLAN によるコーディング例

EXAMPLE 1 として図-10 の部分プログラムのコーディング例を図-12 に示す。サブプログラム SLO 1 を呼び込むことにより図-11 のような図形原形が得られる。これを親図形の一部として利用するためには、MOVE 命令により、平行移動、回転移動を行なう。CONT, LAND は外周線、内周線を宣言するコントロールワードで閉図形の始点、終点を定義する STPT, ENPT により区切られる。PNUM は管理資料としての部材に名前 (PIECE NUMBER) を与えると同時に、図形出力の一部としての文字を定義する。

演算時間は、IBM 360/65 を用いて、コンパイル、連けい編集、図形編集の各ステップの合計で約 4 秒程度である。

次に EXAMPLE 2 として図-13 で示される立面図、平面図をもとに、実寸の原図展開図形を求めるためのコ

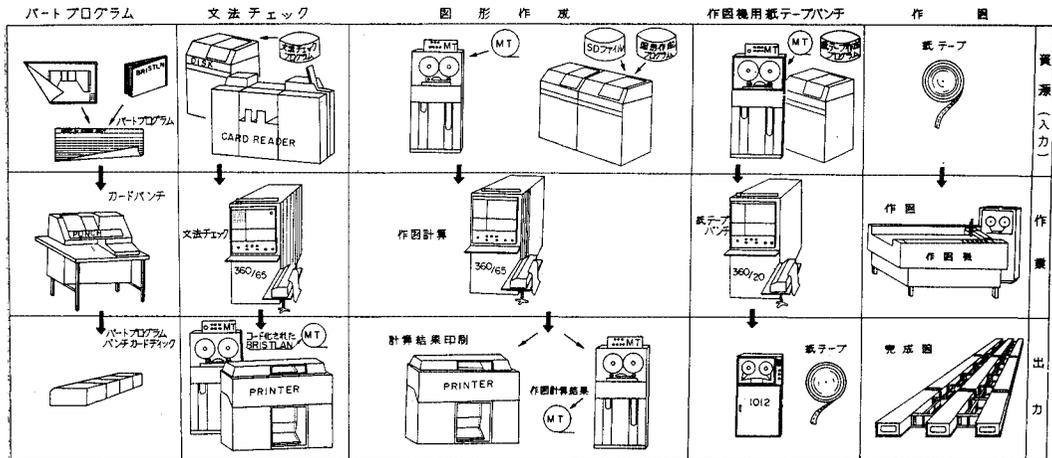


図-9 BRISTLAN による図形作成の手順

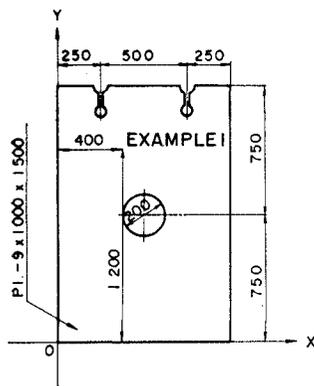


図-10 EXAMPLE 1

```

BSL
JOB          *BRISTLAN*
DEBG        TRAC
SCAL        1/10
PNUM        *EXAMPLE 1/400.0,1200.0
MATR        SS41/9.0
CALL        F
CNT         SL0/9.0/110.0/25.0/35.0/1.0
STPT        P          PTC0 -1.0/-1.0
             LNHL P
             LNVU 1001.0,0.0
             LNHS 0.0,1501.0
             MOVE F/750.0,1500.0/S
             S
             MOVE F/250.0,1500.0/S
             S
             LNVD P
ENPT        P
LAND
STPT
ENPT        CRCR 500.0,750.0/-99.0
END
    
```

図-12 BRISTLAN パート プログラム例 (1)

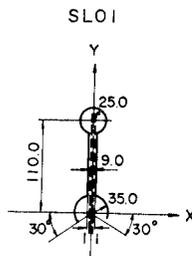


図-11

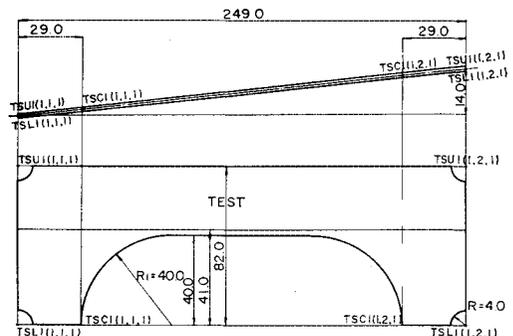


図-13

```

BSL
JOB          *BRISTLAN*
DATA
PLN1        TSL1/TSL1/TSC1
PNUM        TSL1(1,1,1)/TSL1(1,2,1)/TSU1(1,2,1)
CNT         *TEST*/120.0,60.0/12.0
STPT        C1 CRCR TSL1(1,1,1)/-4.0/100.0/-10.0
             S1 LNPT TSL1(1,1,1)/TSL1(1,2,1)
             P PTIT C1/S1
             S1
             S2 LNVU TSC1(1,1,1)
             S3 PARA S1/40.0
             CR2T -40.0/S2/S3
             S3
             S4 LNVD TSC1(1,2,1)
             CR2T -40.0/S3/S4
             S1
             CRCR TSL1(1,2,1)/-4.0/-170.0/80.0
             S5 LNPT TSL1(1,2,1)/TSU1(1,2,1)
             CRCR TSU1(1,2,1)/-4.0/-80.0/170.0
             LNPT TSU1(1,2,1)/TSU1(1,1,1)
             CRCR TSU1(1,1,1)/-4.0/10.0/-100.0
             S6 LNPT TSU1(1,1,1)/TSL1(1,1,1)
             C1
             P
ENPT
MARK
NON        S7 PARA S1/41.0
             CR2F S7/S6/S5
END
    
```

図-14 BRISTLAN パート プログラム例 (2)

ーディング例を 図-14 に示す。この例ではあらかじめ計算された 2 次データファイルが必要で、実行に先立ち、FORTRAN でいうスペシファイケーション ステートメント (Specification Statement) の一種としてのデータ ステートメント (Data Statement) による TSU

1, TSL1, TSC1 のデータ宣言が行なわれる。

(2) BRISTLAN による作図例

BRISTLAN システムは開発終了と同時に、テストケースとして、実際の橋の原図作業に適用された。その

うちプレート ガーダー橋の作図例を、図-15、図-16、図-17 に示す。図-17 は一品図が大板情報として、手作業によりネスティングされたものである。

9. 数値制御システムの方向と問題点

床書き原図作業から NC 原図作業への移行という全く新しい作業方法の導入により、工場における生産体系は全く一新されることになる。この技術革新は、戦後における溶接技術が、鋸構造を駆逐したと同様の意味を持つものといえよう。新しい技術の波が旧来の手法を浸蝕し、生産体系全体を変革していくという発展過程において、種々の問題が生じるのは、一般的な現象であり、数値制御システムの採用についても例外ではない。

社内的には、原図作業者の教育の問題、設備の変換等種々の問題が生じたが、新工場建設というスローガンの

もとに、よくそれらが克服された。さらに鋼橋製作という立場から、数値制御システムの方向と問題点を考えてみたい。

(1) 生産設計から内業加工までの自動化を意図するとき橋梁の構造要素としての部材の「形」を整理し、検討する必要がある。これは主としてスロット、スカラップなどの部材の切片き部、またはガセットの形状など2次部材が標準化の対象として取り上げられる。「形」の標準化を主部材にまで拡張し、整理することももちろん必要であるが、標準化を高度に進め過ぎると、橋の外観的な特質をも失うことになるので、製作上の「形」の標準化にも限界があるろう。そういう意味で「形」の標準化は製作工場内部の工作法の問題として、局所的に整理していくのが正当のようである。これは図形処理一般にいえることで、「形」の標準化をどう考えるかが、これからの図形処理システムの大きな課題となるだろう。

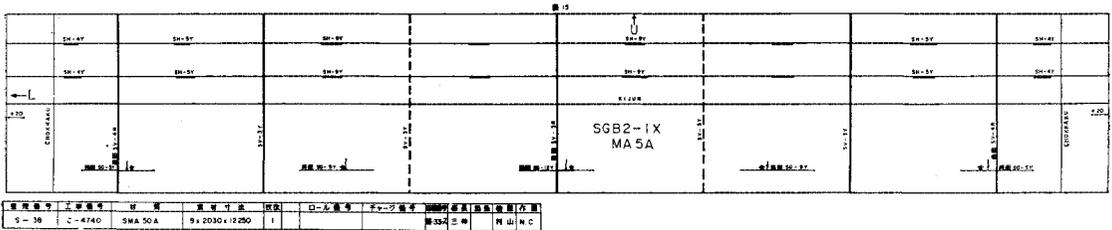


図-15

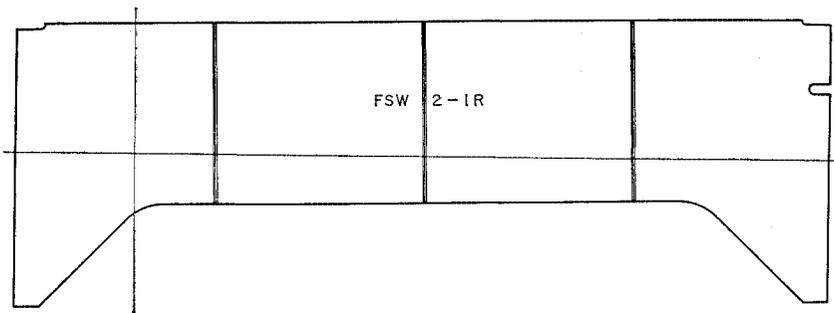


図-16

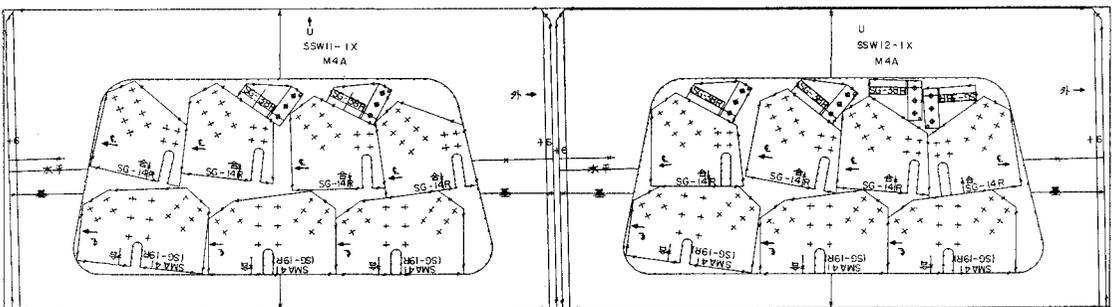


図-17

(2) 電子計算機, CRT, 高速プロッター, 切断機, 自動作画機等のハードウェアは今後急速に進歩し, 数値制御システムの発展に大きな役割を果たすだろう。しかも, ミニコンと呼ばれる小型計算機が普及し, NC 機器のコントローラーとして専用的に使用され, 旧来のアナログ的なコントロールからデジタル的なコントロールに移行するものと思われる。このとき NC 情報のデータ構造の標準化がなされ, ハードウェアに対するデータの互換性を有していなければ, 非常に不便なものとなる。

(3) 鋼道路橋設計製作示方書では, 原寸作業および検査に関する事項が規定されている。

この規定が, 床書き原図作業を前提としているのは論を待たない。NC 原図の登場によりこの規定は大幅に改正の要が生じる。具体的には, 次のような事項について検討が必要であろう。

- (イ) 原図作業の内容とその意義
- (ロ) 原図作業手法の拡大解釈
- (ハ) 原図作業環境の規定
- (ニ) 原図作業とケ書作業の関係
- (ホ) 精度基準の再検討
- (ヘ) 立合検査の内容とその必要性

## 10. 結 言

本システムは昭和 44 年の初めより企画, 開発されたものであり, 昭和 45 年初頭に完成をみた。数橋の実施応用を経て, 同年 10 月から津工場において本格的な本

システムの運用に入った。運営上の種々の問題については, また別の機会に報告したいと思っている。

将来についていえば, CRT による図形の編集および検図方式, 形鋼の処理方式, プレーットのネスティング, 曲面の展開等, 今後われわれは種々の問題をかかえている。

さらに自動設計的な観点からみると MIT で現在開発中の ICES(Integrated Civil Engineering System) にみられるように, BASIC SYSTEM と SUBSYSTEM を独立させ, コンピューターの問題とエンジニアリングの問題を分離して考え, 「問題向け」の SUBSYSTEM を BASIC SYSTEM のコントロールのもとに発展させ, さらに SUBSYSTEM 相互間のデータのつながりも持たせるという手法が, これから POL を発展させる上に必要な概念となろう。

BRISTLAN を用いた本システムが真にその効果を發揮できるのは, これらのテーマが解決されたときであるが, 本報告がこれからの図形処理の問題に何らかの参考になれば幸せである。

なお, 本システムは当設計室, 製作工場, 情報システム部による協同作業で開発された。関係各位に心から感謝の意を表したい。

### 参 考 文 献

- 1) Daniel Roos: ICES System Design, Massachusetts Institute of Technology Cambridge, Massachusetts, The M.I.T. Press, Second Edition, 1967
- 2) 高久達将・渡部 孝: 作図機と作図プログラム, 日本鋼管技報 No. 45

(1970. 8. 8・受付)