

ユーザーオリエンテッドのDDC向け プログラミング言語と保守ツール

User-Oriented Programming Language and Maintenance Tools for Direct Digital Control

火力発電プラント最新形監視制御システムHIACS-3000のボイラ、タービン、バーナなど異なる制御対象のDDCプログラムを、共通手法で記述作成できるようにしたDDC向けシンボリック言語と、上記の制御対象を共通の方法で、しかも統括的に保守できるようにしたエンジニアコンソールを完成した。

前者のシンボリック言語は、DDCプログラムをプラントエンジニアに親しみやすい制御ロジック図と1対1に対応したマクロ命令で記述でき、しかも高速演算が実現できるようにしたシステム統一言語である。後者のエンジニアコンソールは、上記言語を用いたDDCプログラムの作成・変更や伝送ネットワークを介したシステムのオンライン調整・制御信号やプロセス量のモニタなどの広範囲の機能によってシステムを構成するコントローラHISEC 04-M/L, HISEC 04-M/Fのいずれにも共通して使用できるようにした統一保守ツールである。この保守ツールは、誤操作やプラントに影響を与える変更を阻止する機能ももつので、プラントエンジニアでも容易に、かつ安全に変更作業が行なえる。本論文では、上記言語とツールの機能、実現手段及び開発思想について述べる。

宮垣久典* Hisanori Miyagaki

皆川武司* Takeshi Minakawa

下山和彦* Kazuhiko Shimoyama

1 緒 言

日立製作所では、火力発電所を取り巻く新しいニーズに対して、デジタル式ボイラ自動制御装置HIACS-2000¹⁾、タービン自動制御装置HITASS^{2)~5)}、デジタル式電子油圧式ガバナD-EHG^{5)~6)}、デジタル式自動バーナ制御装置や補機シーケンス制御装置など、最新のマイクロエレクトロニクス技術とシステム制御技術を駆使した装置を開発することによりこたえてきた。しかし、これらの装置は、ボイラ、タービン、バーナなど異なる制御対象向け、もしくは異なる時期に開発され発展してきたこともあり、コントローラのハードウェアやソフトウェア又は保守ツールの面で必ずしも統一されているとは言えない点があった。

今回開発した新監視制御システムHIACS-3000では、本システムを構成する高機能形コントローラHISEC 04-M/L, HISEC 04-M/Fのアプリケーションソフトウェア〔以下、DDC (Direct Digital Control) ソフトウェアと呼ぶ。〕を、両方に共通のDDC向けプログラミング言語「DDC向けシンボリック言語」とその言語構造を基にしたDDC保守用コンソール「エンジニアコンソール」とを用いて、統一的に作成・調整・保守ができるようにした。

本論文では、上記DDC向けシンボリック言語、及びDDC用保守ツールの技術内容と機能を中心に報告する。

2 統一言語及び保守ツールの開発課題と対応策

図1に高機能形コントローラHISEC 04-M/L, HISEC 04-M/Fの統一言語、及び保守ツールの開発課題と対応策を示す。

まず、ソフトウェアの作成と保守の面で、DDCソフトウェアは、その性格上計算機ソフトウェアの知識がなくても容易に取り扱え、かつプログラムが効率よく生産、保守できるも

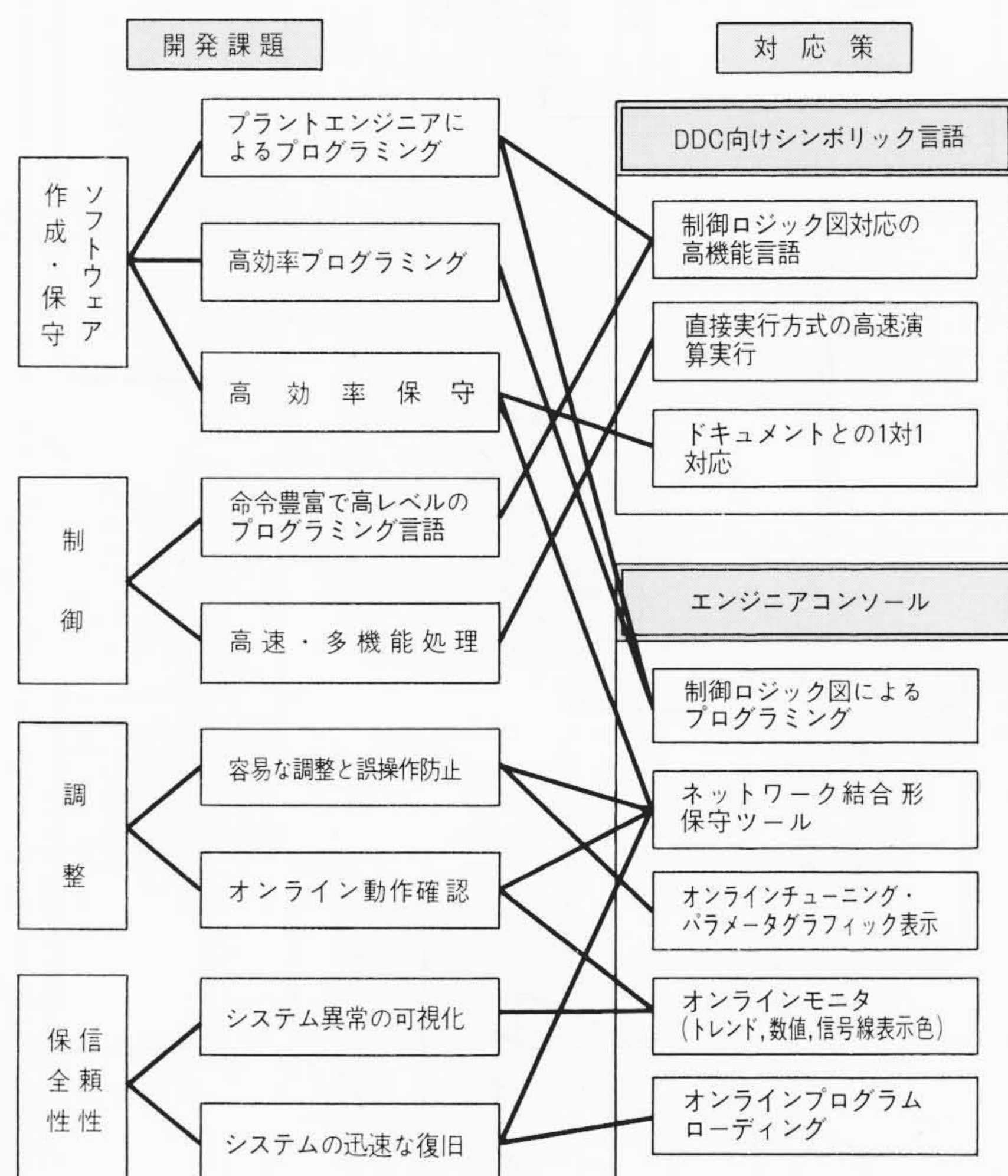


図1 統一言語及び保守ツールの開発課題と対応策 プラントエンジニアに扱え、高速かつ多機能のDDC向けシンボリック言語と、本言語による作成・保守及びオンライン調整保守をネットワーク結合により行なうエンジニアコンソールにより、システムの統一言語による構築・保守が可能となる。

* 日立製作所大みか工場

のでなければならない。

次に制御の面で、高度又は多種多様なロジックが簡単にプログラミングできる豊富な命令語をもった高レベルプログラム記述言語とともに、DDCプログラムをコントローラで高速処理するための言語構造と実行処理構造が必要である。

一方、制御特性の調整の面で、制御パラメータがコントローラのオンライン動作中に、簡単な操作で誤りなく設定でき、調整の結果が迅速かつ経時的に確認できなければならない。

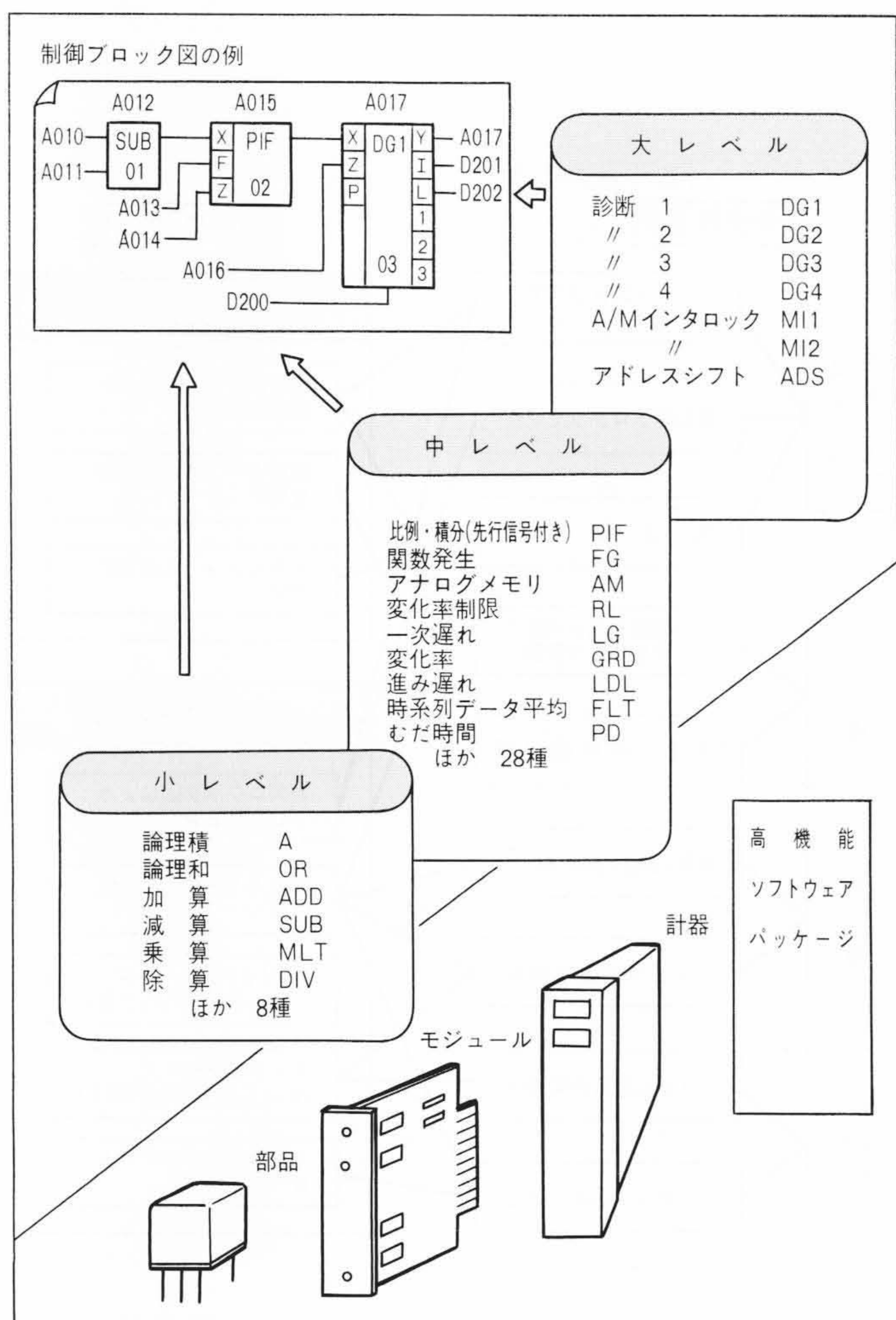
最後に、システムの信頼性、保全性の面では、コントローラの演算渋滞などのシステム異常の可視化やDDCプログラムの変更、復旧が迅速かつ安全に行なえることが重要な課題である。

以上述べた課題をHIACS-3000システムではDDC向けシンボリック言語とエンジニアコンソールによって解決した。

3 DDCプログラムの記述言語とソフトウェア構造

3.1 DDCプログラム記述言語

DDCプログラムを記述する方式としては、(1)機能ブロックシンボルによるデータフロー方式、(2)ラダーシンボルによるラダー方式、(3)ビットパターンによるテーブル方式が代表として挙げられるが、調整制御とシーケンス制御を混在させて



注：制御ブロック図で、

1. AXXXX…アナログ点番号(アナログ信号の番号)
2. DXXXX…デジタル点番号(論理又はデジタル信号の番号)
3. マクロ命令シンボルの中の番号は、演算実行の順序を示す。

図2 DDC向けシンボリック言語 調整制御及びシーケンス制御及び簡単なロジックから複雑高度なロジックまで、機能対応の小レベルから大レベルのマクロ命令を用いて、フレキシブルに構築できる。

記述させても制御の流れが分かりやすい、制御ロジック図と1対1に対応したシンボルでプログラミングができるなどの点に着目し、(1)の方式を採用した。すなわち、図2に示すようなDDC向けシンボリック言語を開発し、制御ロジックを制御ブロック図で記述できるようにした。本言語は同図に示すように、論理演算、四則演算などハードウェアに例えればリレーのような機能を実行する小レベルの命令から、演算モジュールや計器の機能に対応する中レベルの命令、更に入出力診断などハードウェアでは実現困難な大レベルの命令まで幅広いレパートリーをもち、簡単なロジックから複雑、高度なロジックまでをフレキシブルに実現することができる。以下では、上記小レベルから大レベルまでの命令を総称してマクロ命令と呼ぶ。今回はまずこの中レベル命令のレパートリーを拡充しボイラ、タービン、補機と広範囲の制御が実現できるようにする一方、プロセス入力信号や操作用出力信号の診断、手動・自動切換インタロック用の高機能標準マクロ命令を用意した。また、このほかに上述のマクロ命令を組み合わせたプログラムをサブルーチン化するマクロ命令(ADS)を支援し、共通のロジックによって異なる操作端を駆動するバーナ制御などのDDCプログラムをコンパクトに構成できるようにした。

3.2 DDCプログラムのプログラミング

DDCプログラムを図3(a)に示すようにDDCシンボリック言語を用いてエンジニアコンソール(以下、コンソールと略す。)からのキー操作によってCRT(Cathode Ray Tube)画面上に制御ブロック図を表示させながら対話形式で作成、修正できるようにした。また、マクロ命令のブロックシンボルの配置や信号線の結線ルートが同図(b)に示すように特に指定しなくても同図(a)に示すようなキー操作だけで自動的に決定されるようにした。

上述のようなプログラミング方式を実現するため、マクロ命令を1個ずつコントローラが実行できる目的プログラムに変換するコンパイラ(逐次形コンパイラと呼ぶ。)と、この目的プログラムを制御ブロック図に変換し、CRTに表示する制御ブロック図表示処理プログラムをコンソールに内蔵させた。

上述のDDCプログラムは、コントローラごとのDDCプログラムファイルとして目的プログラムの形でハードディスクに記憶させ集中管理できるようにした。そして、プログラムの追加・修正時にはまず該当コントローラのDDCプログラムをファイルからコンソールのメモリ上に読み出して、シートという分割単位で制御ブロック図として表示させて追加、修正する。

その結果をDDCプログラムファイルに書き込んだ後、該当コントローラのプログラムをネットワーク経由で書き換える方式とし、常にハードディスクにコントローラの最新のプログラムが保存されるようにした。

3.3 DDCプログラムのソフトウェア構造

DDCプログラムがコントローラで高速処理でき、かつそれがコンソールで迅速に制御ブロック図として表示できるようにするため、その言語構造と演算実行処理構造を次のようにして決定した。

(1) マクロ命令実行処理の高速化

DDCプログラムをコントローラに実行させる方式としては、直接実行方式とインタプリタ方式がある。前者は実行処理対象である目的プログラムが機械語のもので、そのため高速演算が可能であるが、原始言語(ここではマクロ命令)を機械語に翻訳するプログラム(コンパイラ)が複雑になるという欠点がある。これに対して後者は、原始言語と機械語の中間に位

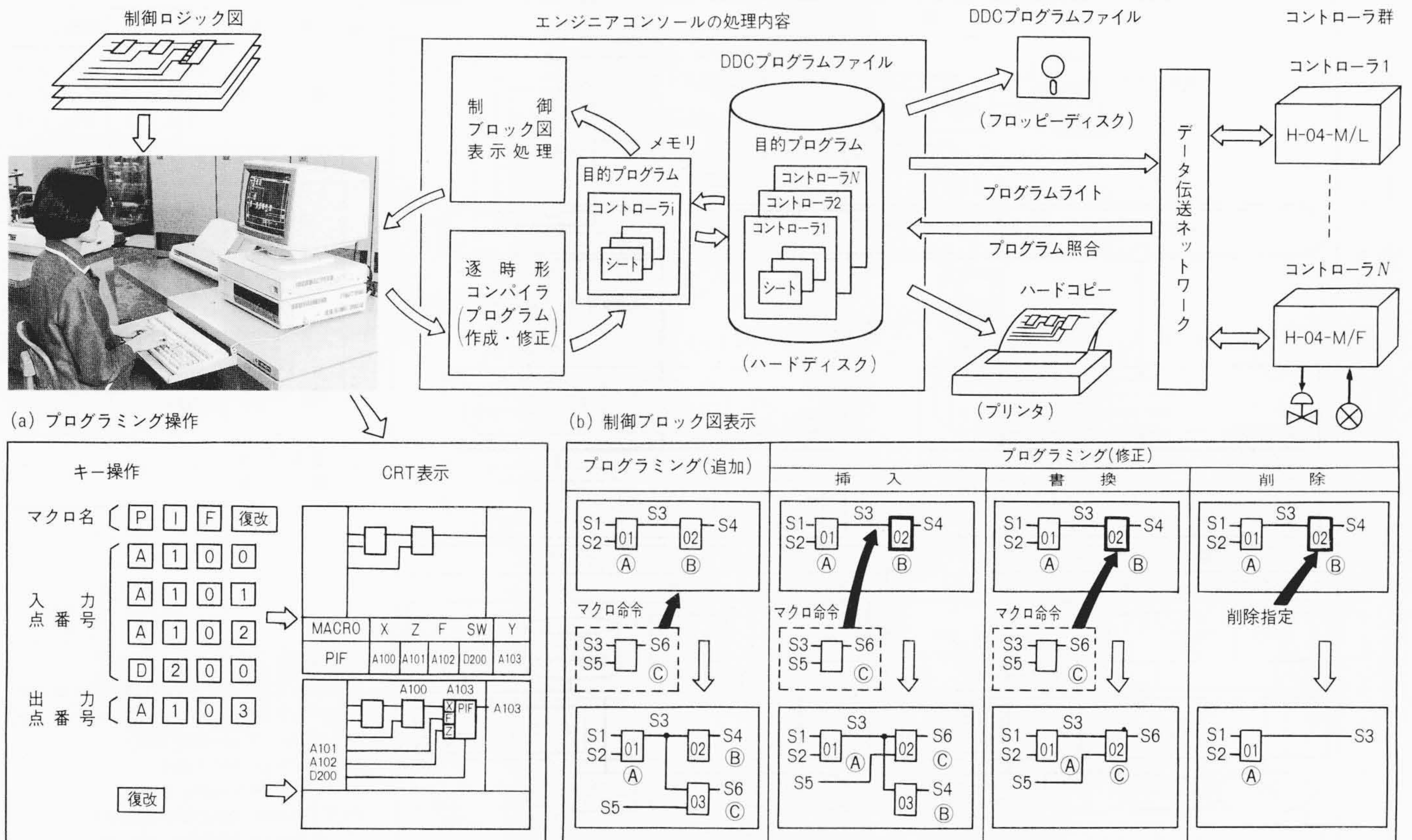


図3 プログラミング操作及び処理内容 エンジニアコンソールを用いて、DDCプログラムを制御ブロック図イメージで確認しながらプログラミングできる。

置する中間語を実行する方式であり、中間語を制御ブロック図表示に都合のよい形態にしやすく、かつコンパイラが簡単に構成できるなどの長所をもっているが、コントローラにインタプリタという中間語を解釈実行するプログラムが必要となり、実行処理速度が遅くなるという欠点がある。そこで、高速から低速まで広範囲の制御演算を同一のソフトウェア構造で実現できることから直接実行処理方式を採用した。これに伴い、コンパイラはマクロ命令を機械語に直接変換する方式とし、容量が大きくなることからコンソールに内蔵させた。

(2) 制御ブロック図表示の高速化

マクロ命令すべてについて機械語に展開した形を標準化し、コンソールにマクロ命令の機械語パターン認識用のテーブルをもたせて制御ブロック図表示の高速化を図った。

(3) DDCプログラム実行処理の高速化

DDCプログラムは、一般に論理演算のような高速処理が必要な部分と比例積分演算のように必ずしもその必要のない部分とから構成されているので、DDCプログラムをループという実行単位に分割し、各々を指定周期又はトリガ信号で起動できる実行処理構造とし、計算機負荷率の平均化が図れるようにした。なお、前述したシートという区分単位に対しては、各ループ内で個別に番号をとり、ループNo.とこの番号を組み合わせたシートNo.コードによりコンソールでは管理することとした。

図4に以上の検討結果に基づいて実現させたコントローラのソフトウェア構造を示す。同図は特にループ02に属するシートNo.0203のDDCプログラムの実行処理フローを拡大したもので、各ループを各々の起動条件に応じてシートNo.の順、シート内ではマクロ命令の順に実行管理タスクにより高速処理させる。

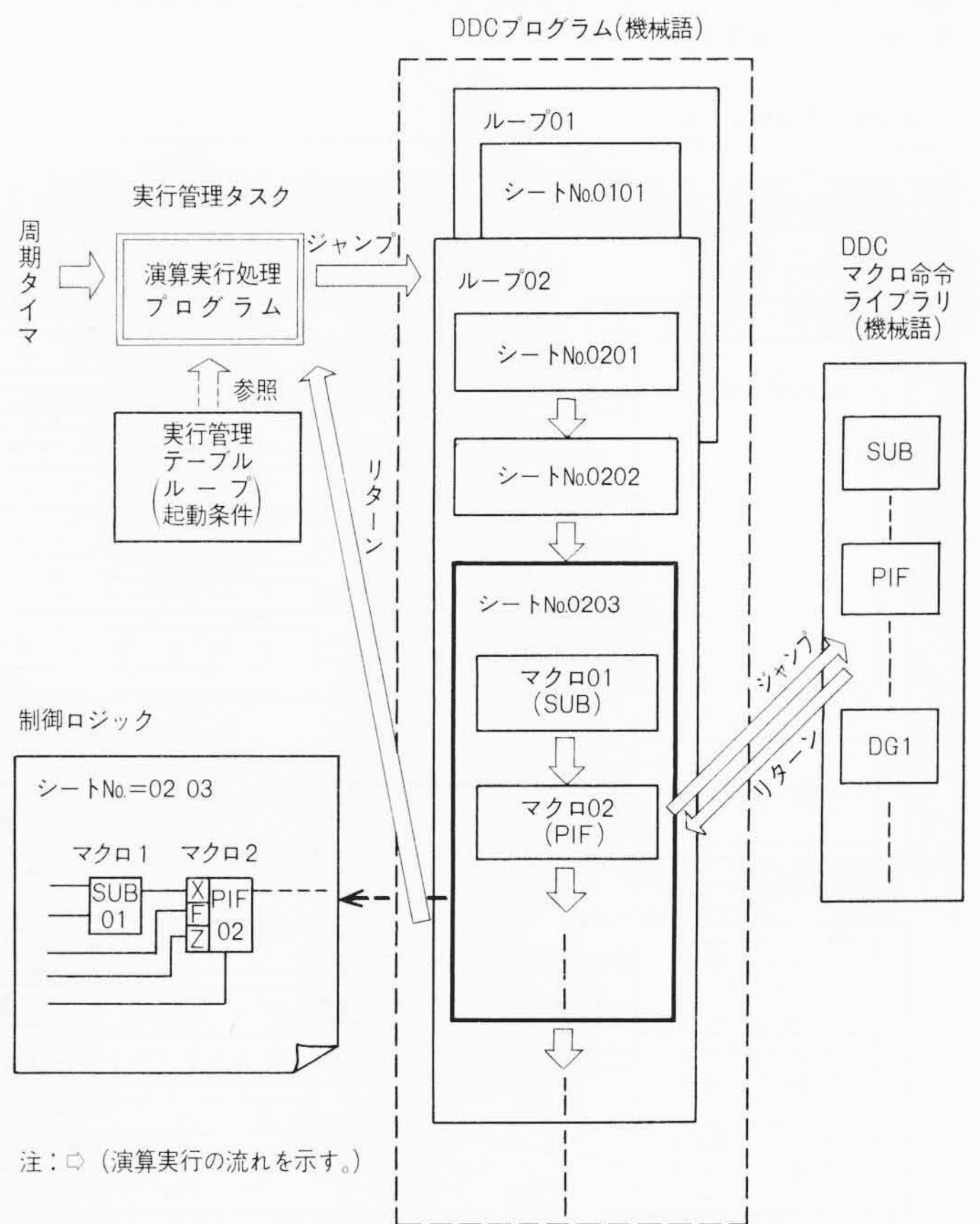
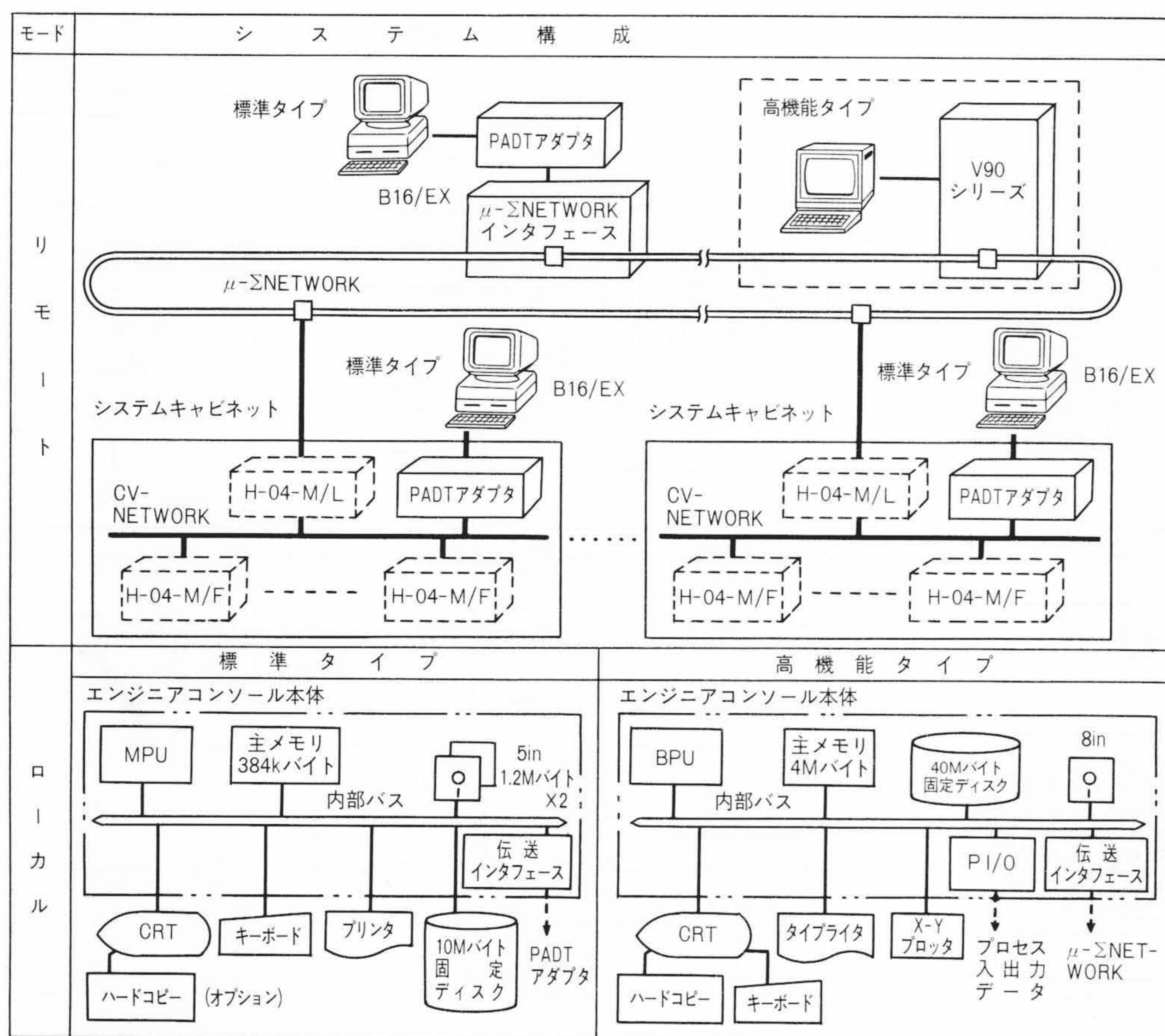


図4 コントローラのソフトウェア構造 実行管理タスクの演算実行処理プログラムは、周期タイマによって起動され、実行管理テーブルを参照して、DDCプログラムを起動条件下にあるループごとに実行する。DDCプログラムを機械語に展開することにより高速化を図り、シートNo.コードに区分することでコンソールによるファイル管理を容易化している。



No.	項目	標準タイプ	高機能タイプ
1	機種	マイクロコンピュータ (B16/EX)	スーパーマイクロコンピュータ (V90シリーズ)
2	伝送速度	4.8kビット/秒	1Mビット/秒
3	CRT		
	サイズ	14in	20in
	表示色	7色	15色
4	ネットワーク構成	μ-SUM NETWORK用 CV-NETWORK用のみ。	μ-SUM NETWORK用のみ。
	表示マクロ数	6~15個	20~30個
5	運転監視機能	なし	あり

注：略語説明

H-04-M/L (HISEC 04-M/L形コントローラ)

H-04-M/F (HISEC 04-M/F形コントローラ)

MPU (Micro Processing Unit)

CRT (Cathode Ray Tube)

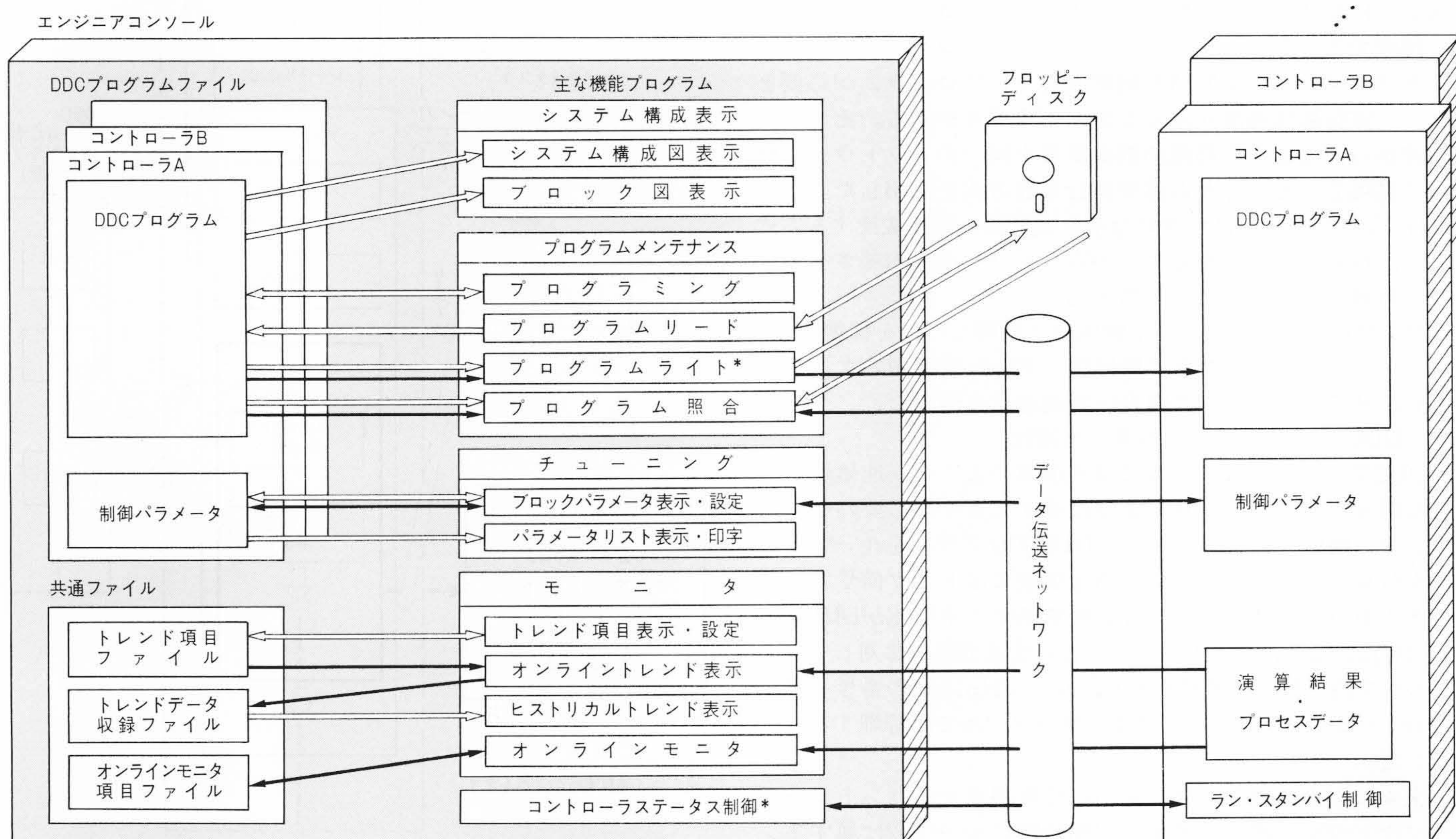
BPU (Basic Processing Unit)

PI/O (Process Input and Output)

CV-NETWORK (システムネットワーク)

μ-SUM NETWORK (ユニットネットワーク)

図5 エンジニアコンソールのシステム構成と主仕様 エンジニアコンソールには、標準タイプと高機能タイプの2種類があり、リモートモードでは、接続したネットワーク及びその下流のネットワークのコントローラをオンライン保守できる。ローカルモードでは、ハードディスクベースでのプログラミング機能などオフライン保守が行なえる。



注：⇨ (ローカルモードのデータの流れ), ⇨ (リモートモードのデータの流れ), * (μ-SUM NETWORK接続時は未支援)

図6 エンジニアコンソールの主な機能 DDCプログラムの検索表示を行なうシステム構成表示, DDCプログラムの作成・修正や書き込み, 読出しなどを行なうプログラムメンテナンス, 制御パラメータの設定を行なうチューニング, コントローラの演算結果などのオンライントレンドや収録データのヒストリカルトレンドを行なうモニタなど, 多種多様な機能をもっている。

4 DDC用保守ツール

4.1 システム構成

図5に、HIACS-3000システムのDDC用保守ツールであるコンソールのシステム構成と主な仕様を示す。コンソールとして保守専用の標準タイプとプラントの運転監視機能ももつ高機能タイプの2種類を用意した。いずれもネットワークに接続して、コントローラのオンライン調整・保守機能を実行するためのリモートモードと、コンソール単体でプログラミングなどの機能を実行するローカルモードの二通りの使用方法がとれるようにした。

リモートモードとしては両タイプ共に μ - Σ NETWORKに接続し、 μ - Σ NETWORK及びCV-NETWORKを介してすべてのコントローラをコンソールとの1対Nデータ伝送により集中オンライン保守ができるようにした。また、標準タイプの場合は、CV-NETWORKに接続し、該当したCV-NETWORK上のコントローラとの1対Nデータ伝送によりオンライン保守ができるようにした。本構成により、異なるCV-NETWORKごとにコンソールを接続してシステムの並列保守が行なえ、特にプラントの定期点検や試運転期間を短縮できる効果は大きい。このほか、標準タイプではコントローラと直結した1対1の保守も可能とした。

4.2 機能及び特長

図6にコンソールの主な機能を、コントローラとのデータの流れが分かる形で示した。以下、本図と具体例を基に主な機能と特長について説明する。

(1) システム構成表示

図7に示すように手元に図面がない場合でも、コントローラ間の階層構成の表示内容からコントローラNo.を知り、そのDDCプログラムの内容を制御ブロック図のレベルまで容易に検索することができるシステム構成図表示、コントローラNo.とシートNo.をもとに制御ブロック図を直接検索することができるブロック図表示の二つの機能を用意した。

(2) プログラムメンテナンス

3.2節で説明したプログラミング機能、DDCプログラムファイルの内容をコントローラのメモリに書き込むプログラムライト、書き込み後に双方のプログラムの照合を行なうプログラム照合の外に、フロッピーディスクをDDCプログラムファイルとする上述と同様の機能ももたせた。なお、プログラミング機能では、マクロ命令の入力時に、その入出力点番号の種別、出力点番号の重複、プログラム用メモリ容量のチェックなどを行ない、文法上の誤りが作り込まれるのを未然に防止するようにした。また、安全のためDDCプログラムが動作中はプログラムライトができないようにした。

(3) チューニング

コントローラの制御パラメータを、マクロ命令単位にオンラインで設定変更するブロックパラメータ表示・設定、DDCプログラムファイルの制御パラメータを、マクロ命令の種別ごと又はループ別にリスト表示あるいはプリントアウトするパラメータリスト表示・印字の機能を用意した。パラメータ変更操作に対しては、入力データの合理性チェックや現在値との比較表示を行なうなどの誤設定防止策を講ずる一方、変更前後の値を自動的にプリントアウトし、変更来歴として残せるようにした。図8に、ブロックパラメータ表示・設定による関数発生マクロ命令のパラメータ変更時の表示例を示す。

(4) モニタ

(a) コントローラの演算結果やプロセスデータの中から任意

オペレーション

C R T 表 示

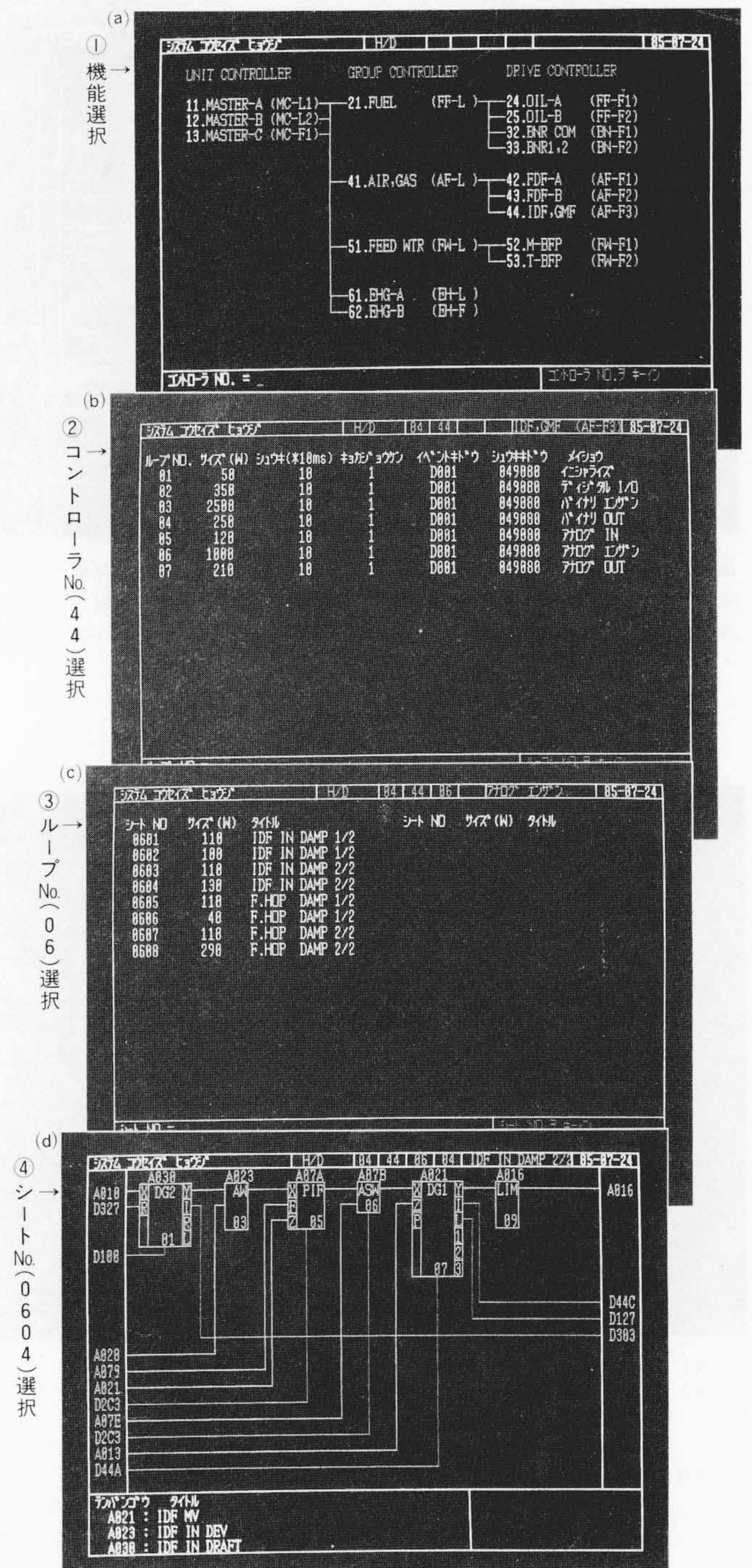


図7 システム構成図表示例 エンジニアコンソールのDDCプログラムファイルを読み出して、コントローラ相互間のシステム構成、指定したコントローラのループ情報、シート情報、制御ブロック図を順次階層的に表示させ、図面がなくても迅速な情報検索ができる。

のものをトレンド項目として選択するためのトレンド項目表示・設定、(b)選択データを図9に示すようにカラートレンドグラフで実時間表示するオンライントレンド表示、(c)この表示データをトレンドデータファイルに収録し、後で必要な部分をトレンドグラフ表示するヒストリカルトレンド表示、(d)図10に示すように制御ブロック図と実時間データ表示を同時に行なうオンラインモニタの四つの機能を設けた。オンライントレンド表示では図9に示すように複数台のコントローラの信号(点番号)を一画面に6点までアナログ信号とデジタル信号を混在させて表示できるようにするとともに、制御パ

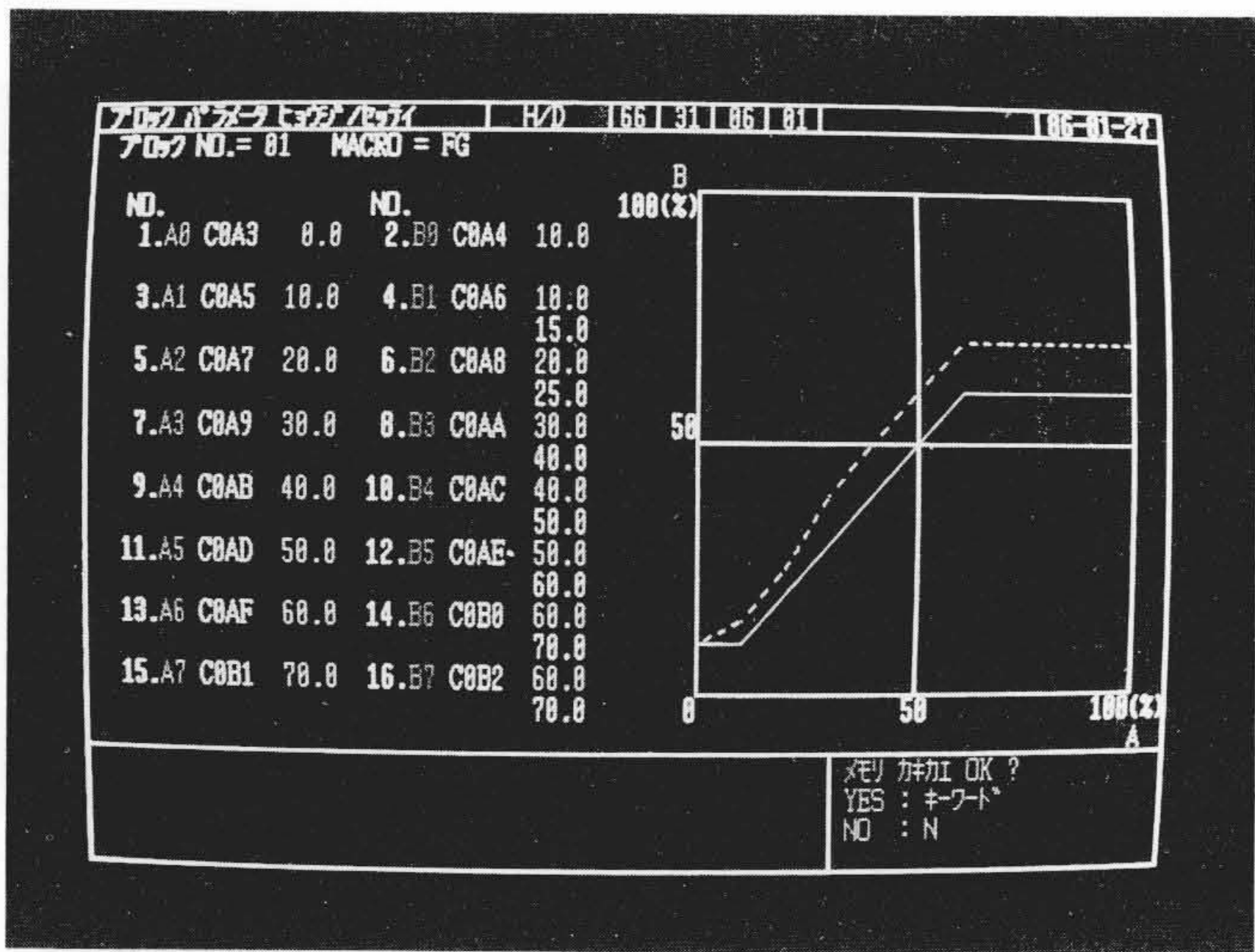


図8 関数発生マクロ命令のパラメータ表示例 関数発生マクロ命令のパラメータ設定時のCRT表示画面の例を示す。実線が現状の設定カーブ、破線が入力データの確認表示カーブを示す。設定指令を入力することにより、パラメータが実際に書き換えられる。

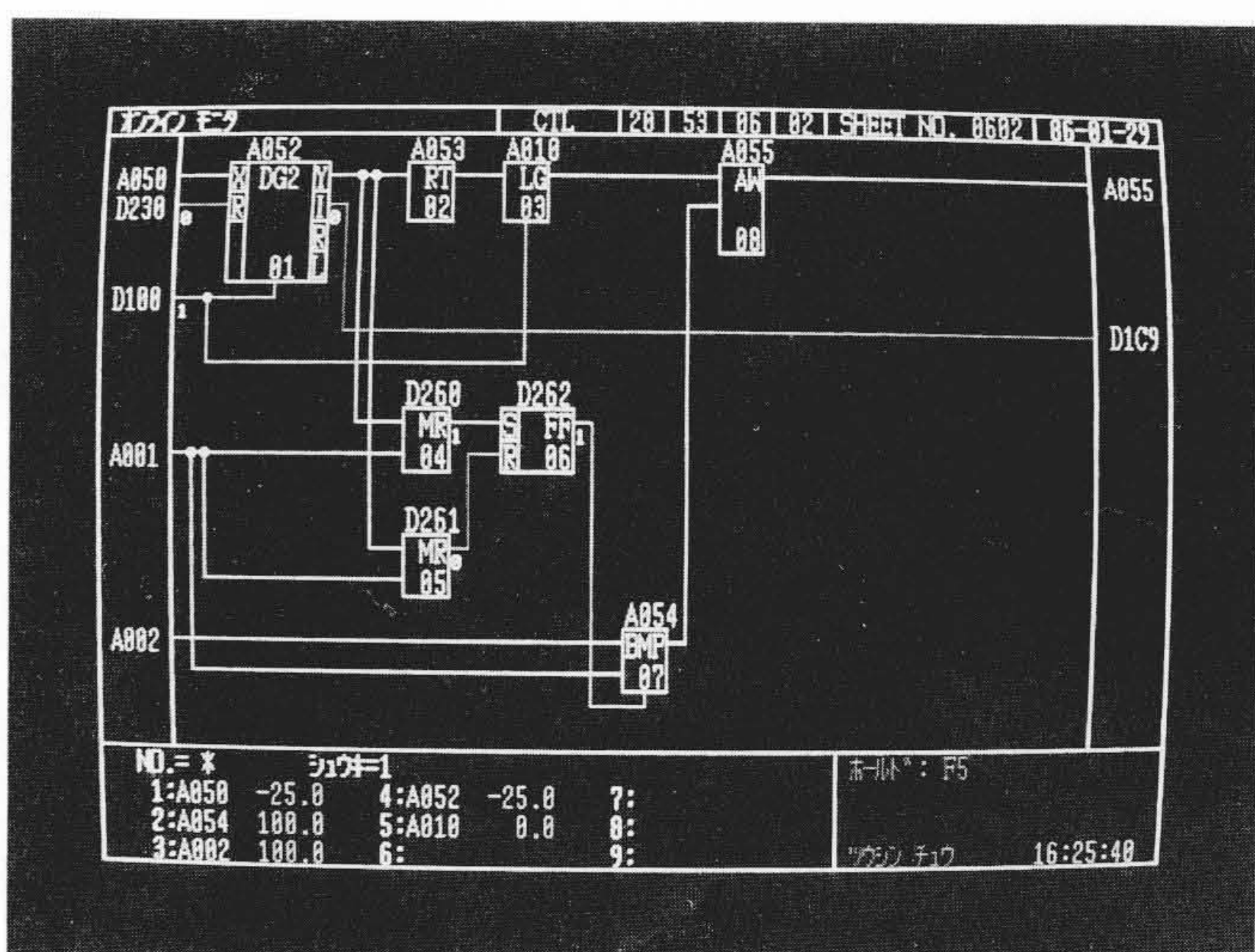


図10 オンラインモニタ表示例 選択表示した制御ブロック図のデジタル信号については、論理値1, 0に対応して数値を信号線近傍に表示するとともに、信号線の色をそれぞれ黄、緑で色別表示する。また、最大9点の信号を、この表示と平行して画面下方に数値で更新表示することができる。

(5) コントローラステータス制御

DDCプログラムを、プログラムライト機能でコントローラのメモリに書き込む場合などに、コントローラのスイッチを直接操作しなくてもリモートモードでスタンバイ(停止)にしたり、書き込み完了後にラン(動作)させたりすることができるように本機能を設けた。

5 結 言

新監視制御システムHIACS-3000を構成する高機能形コントローラによる制御を統一言語と保守ツールを用いて構築・保守できるようにすることを目的として開発したDDC向けシンボリック言語とエンジニアコンソールの機能、実現手段及び開発思想について述べた。本開発の結果、火力発電制御システムの(1)低速制御から高速制御、(2)調整制御とシーケンス制御、(3)異なる制御対象の制御のすべてを、統一言語を用いてプラントエンジニアでも容易に構築可能となるとともに、これら制御を行なうコントローラ群を統一保守ツールを用いることにより、システムのMTTR(Mean Time to Repair)を従来の $\frac{1}{7}$ にすることができるようになった。今後とも、新規ニーズを反映してより使いやすいものとするため、機能及び性能の充実を図っていく考えである。

参考文献

- 1) 中野, 外: デジタル式ボイラ自動制御装置, 火力原子力発電, 31, No. 6, 609~621(昭55-6)
- 2) 天日, 外: ロータ熱応力制御によるタービン自動制御装置, 火力原子力発電, 29, No. 5, 473~482(昭53-3)
- 3) 松本, 外: ロータ熱応力予測タービン制御システム, 計測自動制御学会論文集, 第16巻, 第6号, 905~912(昭55-12)
- 4) 本田, 外: 蒸気タービン自動制御装置, 日立評論, 61, 3, 199~202(昭54-3)
- 5) 本田, 外: 大容量タービンデジタル制御システム, 日立評論, 65, 9, 625~628(昭58-9)
- 6) 東, 外: 蒸気タービン用デジタル式電子油圧式ガバナ, 日立評論, 59, 5, 403~408(昭52-5)
- 7) 中田: コンパイラ, 産業図書(昭56-9)

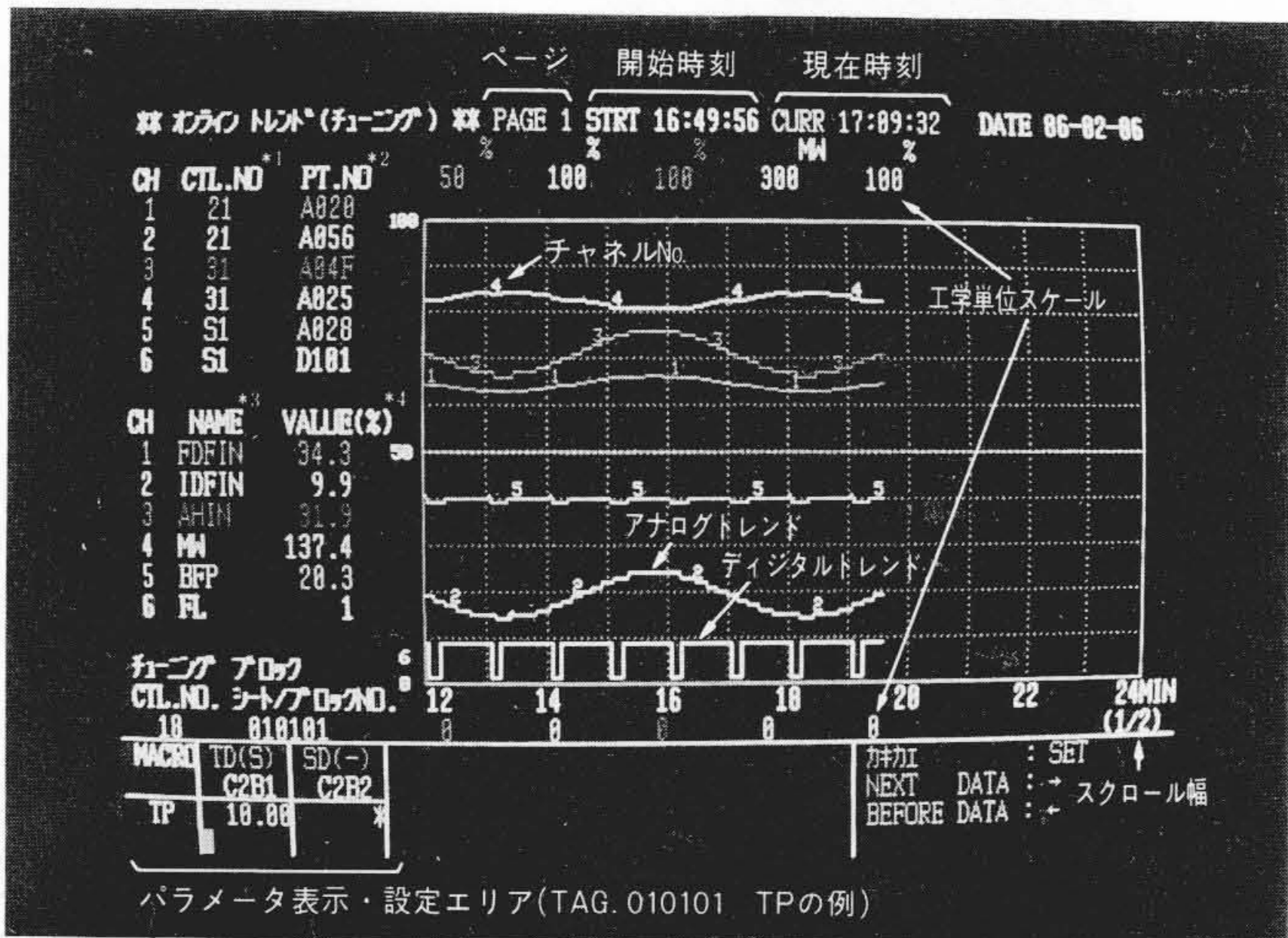


図9 オンライントレンド表示例 ページNo.を指定することにより、あらかじめトレンド項目設定機能で指定された点番号のデータを、1ページ当たり最大6点(6色)までトレンドグラフで表示する。各点の現在値も数値更新表示する。また、この機能の中で、マクロ命令の制御パラメータのチューニングが行なえる。

ラメータの調整をトレンドグラフと同一画面上で制御応答を見ながら行なえるようにした。ヒストリカルトレンド表示では、試運転記録の整理や異常現象の解析などの用途を想定し、トレンドデータファイルはデータサンプリング周期を2秒として最大30点のデータを8時間分収録できるようにした。次に、オンラインモニタでは、図10に示すように制御ロジックの動作状態やシーケンスの渋滞状況をチェックしやすいように、最大9点の信号の数値表示と制御ブロック図のデジタル信号線の色変更を行なわせるようにした。また、オンラインモニタの中で制御ブロック図中のマクロ命令の制御パラメータをカーソル指定により容易に変更できるようにした。