ブロック線図データ生成用インタプリタ

7T－3

田沼　正也
バブコック日立（株）　呉研究所

1　まえがき

大規模な非線形ブロック線図の安定な解析方法「準状態変数法」を提案し[1]，ボイラの動特性シミュレーションへの適用を試みている。開発中のシミュレータXsimでは，1入力データで1演算要素を記述する入力データ形式を採っていたが，ボイラのような3000要素以上のモデルでは，入力データの作成やチェックが容易でない。大規模モデルを数式やユーザが定義した関数を用いて効率的に記述でき，シミュレータ内部でブロック線図つまり計算グラフに展開できるインタープリタを開発した。

2　ブロック線図の記述方法

2.1 ブロック線図データ

ボイラの水・蒸気系モデルのブロック線図表現をFig.1に示す。ブロック線図は計算グラフであり，偏導関数の自動計算法を用いると陰解法を規則的な手順で適用でき，安定な計算が可能となる。各要素は基本的な演算式に対応し，次のような形式で記述できる。

変数　：＝　演算子（入力変数＄係数）；　(1)

実規模のボイラモデルは，Fig.1のブロック線図を70以上接続したものになり，データ作成は容易でない。

2.2 インタプリタによるブロック線図の効率的な記述方法

大規模モデルであっても，部分的には相似なものが多いのでユーザが関数を自由に定義でき，また数式を用いることができれば，データ量を大幅に削減できる。しかし，そのような高級言語的な記述は，計算にはあまり適していない。このため，モデルを高級言語的に記述すれば，それを基に内部で(1)の形式のデータを生成するインタープリタを開発した。

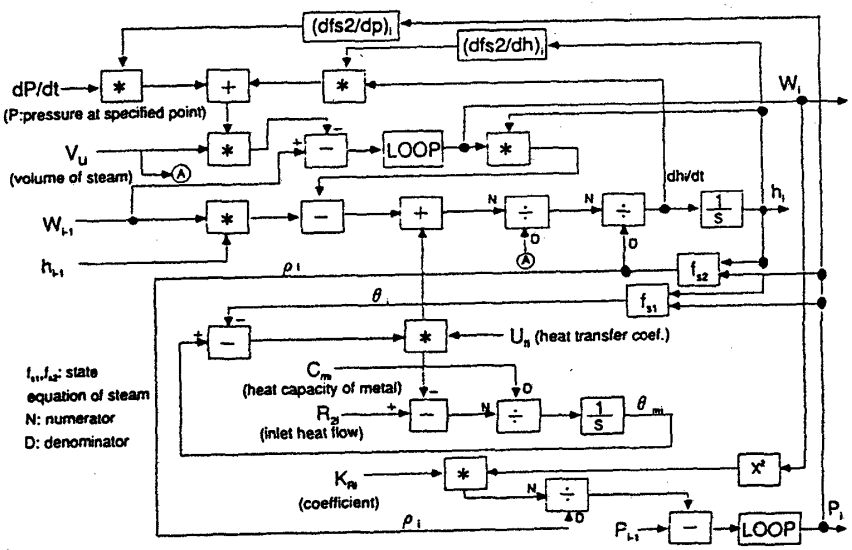

Fig.1 Blockdiagram of steam in boiler

3　ブロック線図記述用インタープリタ

3.1 ブロック線図記述言語の文法

インタプリタで扱う言語の文法は参考文献[2]に倣って，構文解析が左再帰降下法により規則的な手順で行える正規右辺文法を用いた。四則演算式は次のように表わせる。

式：：＝　項［＋｜－項］　(2)

項：：＝因子［＊｜／因子］　(3)

因子：：＝（式）｜変数｜定数｜標準関数　(4)

Interpreter generating Block Diagram Data for Complicated Simulation Model

Masaya Tanuma
Babcock-Hitachi K.K.　Kure Research Lab
1-2-5 Isogo Isogo-ku Yokohama　Kanagawa 235 Japan

ただし，：：＝は定義，［］は０回以上の繰り返し，｜は選択を表わす。処理の優先度は因子→項→式の順となる。文は式の集合となる。ブロック線図特有の積分や１次遅れ要素などは標準関数として用意した。ユーザ定義関数等を含めた実際の言語構成をFig.2に示す。

3.2 インタプリタの構成

Fig.3に示すように，字句解析器，構文解析器，出力器より構成する。

(1)構文解析器　：字句解析器から出力されるトークンの種類に応じて，Fig.3の各要素に対応した関数により，逆ポーランド記法に変換し出力する。式の中の因子に（式）が含まれるので，処理は再帰的になるがC言語を用いて実現できる。

(2)出力器：逆ポーランド記法の式から，基本演算を抽出し，(1)のような形式に変換する。中間変数が発生するが，それは内部で自動的に生成，演算式に割り付ける。結果は内部ファイルに出力するとともに，外部ファイルへASCIIデータで出力し，チェックや再利用を可能とした。本言語で記述した５分割の過熱器モデルをFig.4に示す。１つの関数定義と５回の関数呼び出しでモデルの定義が可能となっており，ブロック線図データの作成時間と作成誤りを大幅に削減できた。

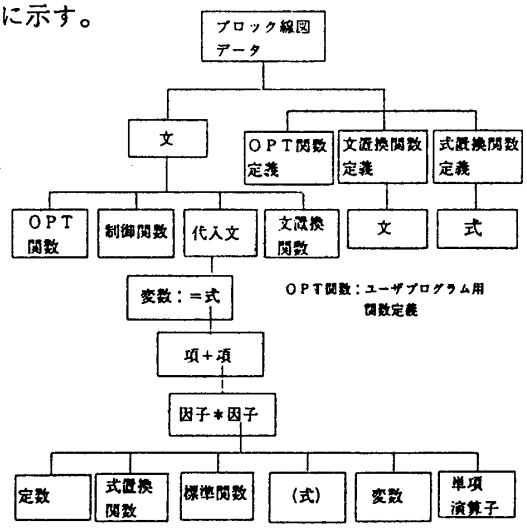

Fig.2 Structure of simulation model language description

4 結び

大規模なボイラモデルの定義を効率良く行うため，ブロック線図入力用インタプリタを開発し，準状態変数法を用いたボイラ動特性解析に利用している。

参考文献

(1)田沼：偏導関数の自動計算法を用いたブロック線図の安定な計算法，ＳＩＣＥ論文集，30-10(1994)

(2)高田：インタプリタ進化論，ＣＱ出版社(1992)

```
DEFINE  SUPERHT(DPDT, HI, WI, PI,
                VLT, VL, UFTT, UFT, TCMT, TCM, QGMT, QGM, KRT, KR,
                H, W, P, T, TM, ROU, DF2DP, DF2DH, DHDT, NS)
 STEAM2(INPUT( P, HS),
        OUTPUT(DF2DP,DF2DHS)S);
 VL := VLT / N;
 UFT := UFTT / N;
 TCM := TCMT / N;
 QGM := QGMT / N;
 KR := KRT / N;
 W := ALOOP( WI - (DPDT * DF2DP + DHDT * DF2DH) * VLS);
 DHDT := (WI * HI - W * H + UFT * (TM - T)) / (VL * ROU);
 STEAM1(INPUT(P, HS),
        OUTPUT(T, ROUS)S);
 H := INTGRL(DHDTS 1.0);
 TM := INTGRL((QGM - UFT * (TM - T)) / TCMS 1.0);
 P := ALOOP(PI - SQRT(W S) * KR / ROUS)!;
```

関数定義

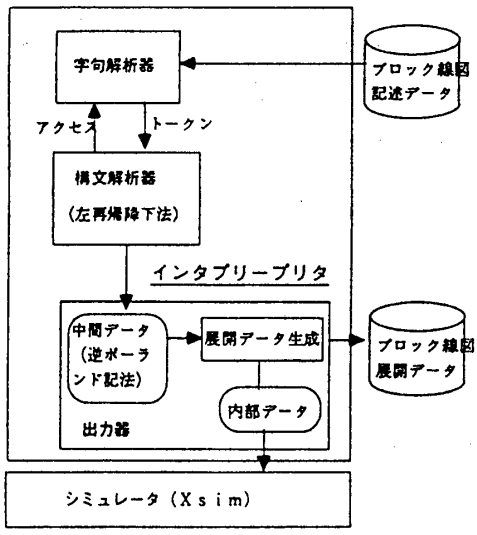

Fig.3 Structure of interpreter for blockdiagram data

```
{ 1ST SECTION }
  SUPERHT(DPDT, H0, W0, P0, VLT, VL1, UFTT, UFT1, TCMT, TCM1, QGMT, QGM1, KRT, KR1,
                H1, W1, P1, T1, TM1, ROU1, DF2DP1, DF2DH1, DHDT1, NSS);

{ 2ST SECTION }
  SUPERHT(DPDT, H1, W1, P1, VLT, VL2, UFTT, UFT2, TCMT, TCM2, QGMT, QGM2, KRT, KR2,
                H2, W2, P2, T2, TM2, ROU2, DF2DP2, DF2DH2, DHDT2, NSS);

{ 3ST SECTION }
  SUPERHT(DPDT, H2, W2, P2, VLT, VL3, UFTT, UFT3, TCMT, TCM3, QGMT, QGM3, KRT, KR3,
                H3, W3, P3, T3, TM3, ROU3, DF2DP3, DF2DH3, DHDT3, NSS);

{ 4ST SECTION }
  SUPERHT(DPDT, H3, W3, P3, VLT, VL4, UFTT, UFT4, TCMT, TCM4, QGMT, QGM4, KRT, KR4,
                H4, W4, P4, T4, TM4, ROU4, DF2DP4, DF2DH4, DHDT4, NSS);

{ 5ST SECTION }
  SUPERHT(DPDT, H4, W4, P4, VLT, VL5, UFTT, UFT5, TCMT, TCM5, QGMT, QGM5, KRT, KR5,
                H5, W5, P5, T5, TM5, ROU5, DF2DP5, DF2DH5, DHDT5, NSS);
```

モデル定義

Fig.4 Description of model by developed methed