



Title	PPAシステムを用いた原子力発電プラントの最適階層制御
Author(s)	山岸, 靖明; Yamagishi, Yasuaki; 辻, 雅司 他
Citation	北海道大學工学部研究報告, 133, 73-83
Issue Date	1986-10-31
Doc URL	https://hdl.handle.net/2115/42012
Type	departmental bulletin paper
File Information	133_73-84.pdf



PPAシステムを用いた原子力発電 プラントの最適階層制御

山 岸 靖 明 ** 辻 雅 司 * 小 川 雄 一 *

(昭和61年6月30日受理)

Optimal Hierarchical Control of the Nuclear Power Plant Using the PPA

Yasuaki YAMAGISHI, Masashi TSUJI and Yuichi OGAWA

(Received June 30, 1986)

Abstract

Optimization of a large-scale system is a very difficult task due to the requirement of an extremely large amount of calculation. One effective way to avoid the difficulty is to divide the system into a number of subsystems and carry out the optimization calculation with each subsystem allowing mutual cooperation among the subsystems to meet the need to optimize the whole system. When an algorithm based on the dividing method is realized in the parallel multi-processor system instead of the conventional computer of series type operation, the efficiency of calculation will be largely improved.

In this paper, using the PPA (Parallel Processor Array) located in the "General Purpose Simulator Facility" of this school, and utilizing the hierarchical structure which is furnished within the PPA system, we prepare a software for the multilevel optimization system, and confirm the effectiveness of the hierarchical algorithm. We then, using the prepared algorithm, design the optimal control of the primary system of two loop PWR power plant, and confirmed the effectiveness of the designed optimal control.

1. は じ め に

原子力プラントのような大規模なシステムの最適化問題を考えるとき、一般の数値計画法を適用しても、変数の多さや制約式の複雑さのために解を求められない場合がある。また解が得られるとしても、計算時間や記憶容量などの点で実用上の問題が大きい。したがって与えられた計算制約のもとでいかに効率良く解くかが重要となる。一般に、このような大規模な問題を定式化すると、特殊な構造を持った数値計画問題になることが多く、現在考察されている手法は、この構造的な特徴をうまく利用して分割と協調という操作による最適化をはかるものであり、大規模シ

* 原子工学科原子炉工学講座

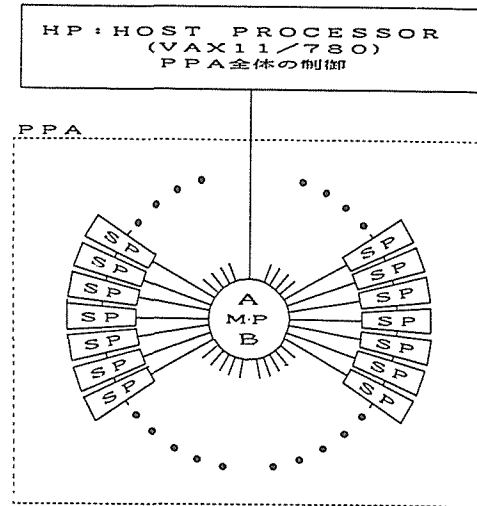
** 現在、ソニー株式会社

システムに対する分割手法と呼ばれている。この分割手法を用いたアルゴリズムが従来の直列方式の計算機ではなく並列方式のマルチプロセッサシステム上で実現された場合には大幅な計算効率の改善が得られると期待される。そこで本研究は、本学汎用シミュレータ室のPPA (Parallel Processor Array) を利用して、その階層構造を生かしたマルチレベル最適化システムのソフトウェアの作成を試み、階層化アルゴリズムの有効性を確認した。また、作成した最適化システムを用いて、状態変数43次、制御変数7次の2ループPWR型原子力発電プラントの一次系のモデルに対する最適制御設計を行なった。

2. PPAおよび分割手法について

2.1 PPA¹⁾

PPAはMIMD (Multiple Instruction Stream - Multiple Data Stream) 方式の隣接密結合型マルチプロセッサシステムであり、複数台のプロセッサの同時 (並列) 動作により処理能力の向上をはかっている。MIMD方式とは、各プロセッサが独自にプログラムとデータを持ち相互に通信し合いながら非同期的に動作し、各プロセッサが並行して各自のプログラムを実行して行く制御方式である。このシステムは、図1のように2台のマスタープロセッサと32台のスレーブプロセッサから構成され、PPAを総括するホストプロセッサを含めると三段階の階層構造を持つシステムとなる。各スレーブプロセッサには、標準的な機械語命令の他に、32ビット浮動小数点データに対する高速浮動小数点演算機構とそれを用いるための特殊な機械



MP: (PDP11/34相当)
MASTER PROCESSOR

SP群の制御
HP-SP間の通信

SP: (PDP11/34相当)
SLAVE PROCESSOR

独立した演算プロセッサ
隣接SP間は自由に通信可能

図1 PPAの構成

語命令や、ベクトル演算ならびに複素演算関係の命令も付加されている。

2.2 分割手法と階層システム

分割とは全体の問題をサイズの小さな部分最適化問題のグループに分けることであり、各部分問題は各々独立に取り扱うことができる。当然、これらの部分最適化問題の解の集合は元の問題の解に一致するように調整(協調)されなければならない。協調は統合変数とよばれるパラメータを修正することによってなされ、各部分問題はこの統合変数によってパラメータ化された問題となっている。現在開発されている手法は分割の方法により2種類に分類される。すなわち、原問題を直接解くことによって最適化をはかる原分割と、原問題に対する双対問題を利用することによって最適化をはかる双対分割である。これらの基本的な分割手法を用いて問題の構造に応じたアルゴリズムがいくつか発表されている。いずれのアルゴリズムも、部分問題群と協調問題との間に階層構造を持たせて、図2に示すような階層分権型最適化システムを構成している。下位レベルのSubsystem群は部分最適解を求め、上位レベルのCoordinatorはそれらを調整するための統合変数の値を決定する。このように原問題をいくつかの独立した部分問題に分割してマルチレベル最適化を行なうことにより、各問題をそれぞれに適したアルゴリズムで解くことができ、また、次元の低下により必要な計算時間や記憶容量を大幅に減らしうるなどの利点もたらされる。本研究で採用しているアルゴリズムは、図3のように、CoordinatorとSubsystem群の間に部分問題間の相互干渉に関する情報を逐一全Subsystemに報告するSubcoordinatorを設けて統合変数の更新操作を分権化し、解の収束性を高めている。

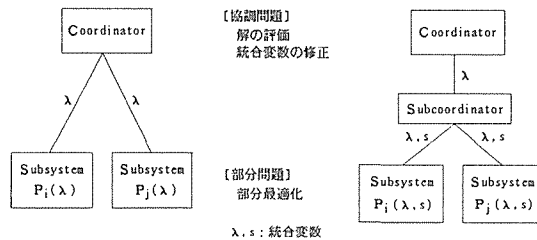


図2 階層最適化システム

図3 3レベル最適化システム

3. マルチレベル最適化システム^{2),3),4),5)}

線形離散時間システム

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad k = 0, 1, \dots, k_{f-1} \tag{1}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (\text{初期条件}) \tag{2}$$

において、つぎの評価関数

$$\begin{aligned}
 J &= \sum_{k=0}^{k_f-1} \frac{1}{2} \{ \mathbf{x}(k)^T \mathbf{Q}\mathbf{x}(k) + \mathbf{u}(k)^T \mathbf{R}\mathbf{u}(k) \} + \frac{1}{2} \mathbf{x}(k_f)^T \mathbf{Q}\mathbf{x}(k_f) \\
 &= \sum_{k=0}^{k_f-1} \frac{1}{2} \{ \|\mathbf{x}(k)\|_Q^2 + \|\mathbf{u}(k)\|_R^2 \} + \frac{1}{2} \{ \|\mathbf{x}(k_f)\|_Q^2 \}
 \end{aligned} \tag{3}$$

を最少にする最適制御列 $\{\mathbf{u}(k)\}$ を求める最適レギュレータ問題を考える。ただし、 $\mathbf{x}(k)$ および $\mathbf{u}(k)$ はそれぞれ時刻 k における状態ベクトルおよび制御ベクトル、 \mathbf{A} および \mathbf{B} はそれぞれ $n \times n$ および $n \times r$ の定数行列、 \mathbf{Q} は $n \times n$ の半正定値行列、 \mathbf{R} は $r \times r$ の正定値行列である。

さらに、 \mathbf{A} 、 \mathbf{B} 、 \mathbf{Q} 、 \mathbf{R} にはつぎのような条件が加えられる。すなわち、 $\mathbf{x}(k)$ 、 $\mathbf{u}(k)$ がそれぞれ

$$\mathbf{x}(k)^T = \{\mathbf{x}_1(k)^T, \mathbf{x}_2(k)^T, \dots, \mathbf{x}_N(k)^T\}, \mathbf{x}_i(k) \in \mathbb{R}^{n_i}, n = \sum_{i=1}^N n_i$$

$$\mathbf{u}(k)^T = \{\mathbf{u}_1(k)^T, \mathbf{u}_2(k)^T, \dots, \mathbf{u}_N(k)^T\}, \mathbf{u}_i(k) \in \mathbb{R}^{r_i}, r = \sum_{i=1}^N r_i$$

のように分離可能であるとき、 \mathbf{A} 、 \mathbf{B} 、 \mathbf{Q} 、 \mathbf{R} は以下のような構造を持つと仮定する。

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \cdot & \cdot & \cdot & \mathbf{A}_{1N} \\ \cdot & & & & \cdot \\ \cdot & & \mathbf{A}_{ij} & & \cdot \\ \cdot & & & & \cdot \\ \mathbf{A}_{N1} & \cdot & \cdot & \cdot & \mathbf{A}_{NN} \end{pmatrix} \quad \mathbf{A}_{ij}; n_i \times n_j \text{ 行列}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 & & & & 0 \\ \cdot & & & & \\ & \cdot & & & \\ & & \mathbf{B}_i & & \\ & & & \cdot & \\ 0 & & & & \cdot \\ & & & & \mathbf{B}_N \end{pmatrix} \quad \mathbf{B}_i; n_i \times r_i \text{ 行列}$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & & & & 0 \\ \cdot & & & & \\ & \cdot & & & \\ & & \mathbf{Q}_i & & \\ & & & \cdot & \\ 0 & & & & \cdot \\ & & & & \mathbf{Q}_N \end{pmatrix} \quad \mathbf{Q}_i; n_i \times n_i \text{ 行列}$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_1 & & & & 0 \\ \cdot & & & & \\ & \cdot & & & \\ & & \mathbf{R}_i & & \\ & & & \cdot & \\ 0 & & & & \cdot \\ & & & & \mathbf{R}_N \end{pmatrix} \quad \mathbf{R}_i; r_i \times r_i \text{ 行列}$$

このとき、(1)~(3)式はそれぞれ N 個に分割された形で表わすことができる。

$$\begin{aligned} \mathbf{x}_i(k+1) &= \sum_{j=1}^N \mathbf{A}_{ij} \mathbf{x}_j(k) + \mathbf{B}_i \mathbf{u}_i(k) \\ &= \mathbf{A}_{i1} \mathbf{x}_1(k) + \sum_{\substack{j=1 \\ j \neq i}}^N \mathbf{A}_{ij} \mathbf{x}_j(k) + \mathbf{B}_i \mathbf{u}_i(k), \quad k=0, 1, \dots, k_{r-1} \end{aligned} \quad (4)$$

$$\mathbf{x}_i(0) = \mathbf{x}_{i0} \quad (5)$$

$$J = \sum_{i=1}^N \left[\sum_{k=0}^{k_{r-1}} \frac{1}{2} \{ \|\mathbf{x}_i(k)\|^2_{Q_i} + \|\mathbf{u}_i(k)\|^2_{R_i} \} + \frac{1}{2} \|\mathbf{x}_i(k_r)\|^2_{Q_i} \right] \quad (6)$$

ここで、相互干渉に関する $\mathbf{x}_j(k)$ を

$$\mathbf{x}_j(k) = \mathbf{s}_j(k)$$

とパラメータ $\mathbf{s}_j(k)$ で置き代えることによって、各サブシステムは他のサブシステムと関係せずに独立に演算できるようになる。このような \mathbf{s}_j を導入し、つぎの Lagrange 関数を定義する。

$$\begin{aligned} L(\mathbf{x}, \mathbf{u}, \mathbf{s}, \mathbf{p}, \lambda) &= \sum_{i=1}^N L_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{s}_i, \mathbf{p}_i, \lambda_i) \\ &= \sum_{i=1}^N \left[\sum_{k=0}^{k_{r-1}} \left\{ \frac{1}{2} (\|\mathbf{x}_i(k)\|^2_{Q_i} + \|\mathbf{u}_i(k)\|^2_{R_i}) \right. \right. \\ &\quad \left. \left. + \mathbf{p}_i(k+1)^T \left(\sum_{j=1}^N \mathbf{A}_{ij} \mathbf{s}_j(k) + \mathbf{B}_i \mathbf{u}_i(k) - \mathbf{x}_i(k+1) \right) \right. \right. \\ &\quad \left. \left. + \lambda_i(k)^T (\mathbf{s}_i(k) - \mathbf{x}_i(k)) \right\} + \frac{1}{2} \|\mathbf{x}_i(k_r)\|^2_{Q_i} \right] \end{aligned} \quad (7)$$

最適解が満たすべき条件は

$$\frac{\partial L}{\partial \mathbf{x}_i(k)} = \mathbf{Q}_i \mathbf{x}_i(k) - \mathbf{p}_i(k) - \lambda_i(k) = 0, \quad \mathbf{p}_i(k_r) = 0 \quad (9)$$

$$\frac{\partial L}{\partial \mathbf{u}_i(k)} = \mathbf{R}_i \mathbf{u}_i(k) + \mathbf{B}_i^T \mathbf{p}_i(k+1) = 0 \quad (10)$$

$$\frac{\partial L}{\partial \mathbf{s}_i(k)} = \sum_{j=1}^N \mathbf{A}_{ji}^T \mathbf{p}_j(k+1) + \lambda_i(k) = 0 \quad (11)$$

$$\frac{\partial L}{\partial \mathbf{p}_i(k+1)} = \sum_{j=1}^N \mathbf{A}_{ij} \mathbf{s}_j(k) + \mathbf{B}_i \mathbf{u}_i(k) - \mathbf{x}_i(k+1) = 0, \quad \mathbf{x}_i(0) = \mathbf{x}_{i0} \quad (12)$$

$$\frac{\partial L}{\partial \lambda_i(k)} = \mathbf{s}_i(k) - \mathbf{x}_i(k) \quad (13)$$

となる。

Lagrange 関数(8)において、 λ, \mathbf{s} を $\lambda = \lambda^*, \mathbf{s} = \mathbf{s}^*$ に固定すると、つぎの独立した部分問題ができる。ただし、 λ は解法の便宜上固定される。

[部分問題] ($i=1, \dots, N$)

$$\begin{aligned} &\text{minimize}_{\mathbf{x}_i, \mathbf{u}_i} f_i(\mathbf{x}_i, \mathbf{u}_i) \\ &= \sum_{k=0}^{k_{r-1}} \left[\frac{1}{2} \{ \|\mathbf{x}_i(k)\|^2_{Q_i} + \|\mathbf{u}_i(k)\|^2_{R_i} \} + \lambda_i^*(k)^T \{ \mathbf{s}_i^*(k) - \mathbf{x}_i(k) \} \right] \\ &\quad + \frac{1}{2} \|\mathbf{x}_i(k_r)\|^2_{Q_i} \end{aligned} \quad (14)$$

$$\text{subject to } \mathbf{x}_i(0) = \mathbf{x}_{i0} \quad (15)$$

$$\mathbf{x}_i(k+1) = \sum_{j=1}^N \mathbf{x}_{ij} \mathbf{s}_j^*(k) + \mathbf{B}_i \mathbf{u}_i(k) \quad (16)$$

これらの部分問題を解いて得られる部分最適解は固定されるパラメータ（統合変数） λ^* 、 \mathbf{s}^* に依存するものであり、それらの総和が元の問題の解に一致するとは限らない。そこで、解を調整するために協調問題を考える。すなわち、固定された λ^* 、 \mathbf{s}^* のもとで解かれた部分問題の解を用いて、 L の λ 、 \mathbf{s} に関する停留条件を満たすような λ 、 \mathbf{s} を求める。この問題を解いて得られた λ 、 \mathbf{s} を新しい統合変数として固定し、前述の部分問題を解く。以上の反復操作を最適解の必要条件のすべてを満たす解が得られるまで続ける手法はマルチレベル最適化アルゴリズムといわれる。

本研究で作成した最適化システムは、上記アルゴリズムを図4に示すようなPPAの3段階の階層構造において実行するものである。すなわちスレーブプロセッサ（SP）群はサブシステム群に対応し、統合変数 λ 、 \mathbf{s} を固定することによって分割された部分問題(14)~(16)を並列的に解き、マスタープロセッサ（MP）はSub-coordinatorに対応し、部分問題間の相互干渉の段階的予告を行ない、ホストプロセッサはCoordinatorに対応し、 λ の更新を行なう。

各SPは、MPから送られる \mathbf{s} と λ をもとにして各ステージ変数(k)毎に部分問題の最適解を求め、結果を逐一MPに報告する。MPは、HPから送られる λ と全SPから集めた相互干渉の情報をステージ変数毎に全SPに放送（ブロードキャスト転送）すると同時に、SPの計算結果をHPに報告する。HPでは得られた結果の評価と、新しい λ の値を決定する。この状況が図4に示される。本研究の手法は、Sub-Coordinatorがステージ変数毎に相互干渉を固定しているため、 λ と \mathbf{s} を一括して各反復の始めに固定する手法（Interaction Prediction Method⁹⁾）に比べて収束特性が改善される利点を持つ。また、 λ の更新に緩和法を用いているために、勾配を用いる降下法で必要な降下方向やステップ幅などの計算が不要になるなどの特徴も有している。以上のアルゴリズムは直列的に述べられているが、実際には図に示されるように、垂直および水平の両方向的に実行される。このように、最適化問題を部分問題群に分割して水平方向に並列して解き、かつ、その協調操作を垂直方向に分権化することによって、計算効率を高めている。

作成した最適化システムで扱えるモデルの規模を以下に示す。

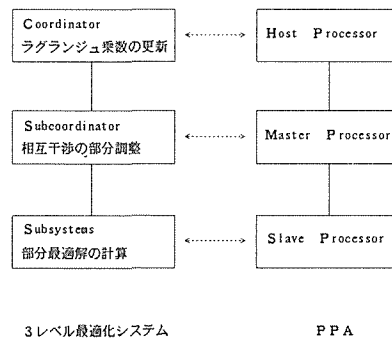


図4 3レベル最適化システムとPPAとの対応

時間分割数	;	170
全状態変数	;	50
全制御変数	;	50
部分問題数	;	10

つぎに、上述の階層分権型アルゴリズムと、サブシステムに分割しない一括的解法について、問題の規模に対する所要計算の比較を行なった。この比較のために用いたモデルは、状態変数がそれぞれ 8, 16, 24, 32, 40 次で制御変数がいずれも 8 次の、特別の意味を持たないモデルである。階層手法において、全系は 8 個のサブシステムに分割されており、協調反復回数 (Coordinator Level) は 2 回としている。使用計算機には、階層手法では VAX11/780 (ホストプロセッサ) + PPA, すなわち作成した最適化システムを用いているのに対して、一括的解法ではメモリー容量の制約から VAX11/780 を用いている。結果は、図 5 のように、一括的解法では所要計算時間がモデルの次数の 4 ~ 5 乗に比例しているのに対して、階層手法では、試験の次数の範囲において、モデルの次数にほぼ直線的に増加している。低次元のモデルに対しては、一括的解法の方が階層手法より幾分計算時間が短くなっているが、これは、VAX11/780 と MAP-16 (PDP11/34 相当) の演算能力の基本的な差によるものと思われる。したがって、3 レベル間の情報転送がネックになっていなければ、同等の計算機で実行された場合には、低次元モデルにおいても、階層手法が一括的解法より計算が速くなると考えられる。

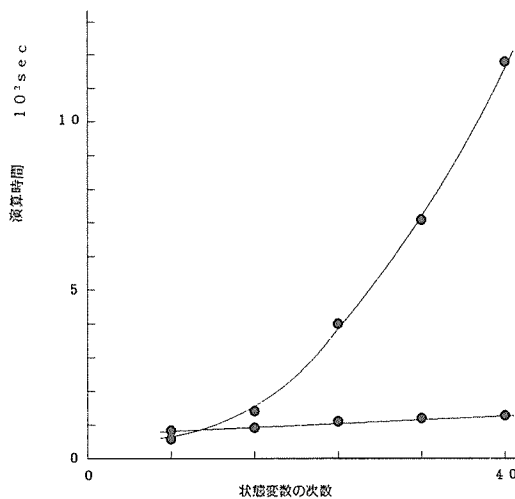


図 5 階層分権型解法と一括的解法の計算時間の比較

4. 原子力発電プラントの最適制御設計

前章で作成した最適化システムを用いて、PWR型原子力発電プラントの最適レギュレータ設計を行なう。PWR型原子力発電プラントは、図 6 に示すように、原子炉心冷却系 (一次系) とタービン-発電機系 (二次系) から構成され、この二つの系を結ぶ蒸気発生器が設けられる。一次系は、原子炉心、反応度を制御するための制御棒とその駆動装置、一次冷却材圧力を高圧に保ち沸騰を抑制するための加圧器、一次冷却材循環ポンプ、一次冷却材のほう酸濃度と体積を制御

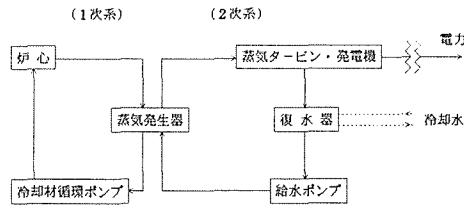


図6 PWR型原子力発電プラント

する化学・体積制御系などから成る。二次系は、タービン-発電機、復水器、給水ポンプなどから成る。炉心で発生した熱は一次冷却材により蒸気発生器に導かれ、二次系に伝達される。一次冷却材は、蒸気発生器を出た後、循環ポンプで炉心に戻される。この冷却ループは、プラントの規模により、2～4ループある。蒸気発生器の2次側に伝えられた熱は二次系の給水を加熱し、蒸気を発生させる。この蒸気はタービンを駆動した後、復水器で冷却されて水に戻り、ポンプで蒸気発生器へ給水される。

本研究で扱うモデルは、図7に示されるような、炉心、2基の蒸気発生器、加圧器、およびそれらの制御装置と、サブクール体積制御装置ならびに蒸気弁開度調節装置とから構成される43個の状態変数と7個の制御変数を持つモデルであり、非線形の前モデルを定常値の近傍で線形近似

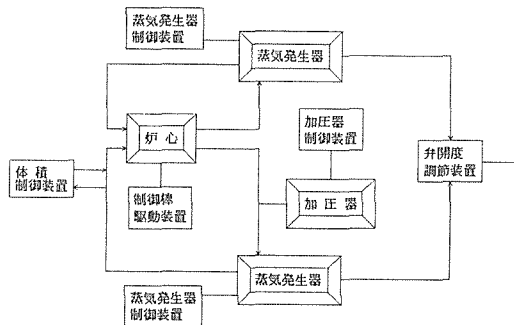


図7 プラントモデルの概略図

しているものである。

本モデルをつぎのような離散系のシステム方程式で表現することができる。

$$\mathbf{x}(k+1) = \mathbf{D}\mathbf{x}(k) + \mathbf{E}\mathbf{u}(k), \quad \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \quad (17)$$

上式の駆動行列 \mathbf{E} は一般には密な行列になるが、前章の最適化アルゴリズムは系をサブシステムに分割し、サブシステム間の相互干渉のみを考慮するものであるため、 \mathbf{E} をブロック対角にする必要がある。そのために、まず(17)式における \mathbf{E} を特異値分解する。

$$\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^T \quad (18)$$

ただし、 \mathbf{U} ; $n \times n$, $\mathbf{E}\mathbf{E}^T$ の固有ベクトルからなる直交行列

\mathbf{V} ; $r \times r$, $\mathbf{E}^T\mathbf{E}$ の固有ベクトルからなる直交行列

$$\Sigma = \begin{pmatrix} \sigma_1 & & \mathbf{0} \\ & \sigma_i & \\ \mathbf{0} & & \sigma_r \\ \hline & & \mathbf{0} \end{pmatrix}, \quad \begin{array}{l} \sigma_i : \mathbf{E} \text{ の特異値} \\ (\mathbf{E}^T \mathbf{E} \text{ の固有値の正の平方根}) \end{array}$$

(18)を(17)に代入すると

$$\mathbf{x}(k+1) = \mathbf{D}\mathbf{x}(k) + \mathbf{U}\Sigma\mathbf{V}^T\mathbf{u}(k) \quad (19)$$

となるが、ここで Σ の行を入れ替えブロック対角 Σ^* にする。このとき $\mathbf{x}(k+1)$, $\mathbf{x}(k)$, \mathbf{D} , \mathbf{U} についても Σ^* の行の並びに対応させて並びかえ $\mathbf{x}^*(k+1)$, $\mathbf{x}^*(k)$, \mathbf{D}^* , \mathbf{U}^* とする。

$$\begin{array}{c} \begin{array}{c} r \\ \left(\begin{array}{ccc} \bullet & & \mathbf{0} \\ & \bullet & \\ \mathbf{0} & & \bullet \end{array} \right) \\ n \\ \hline \mathbf{0} \end{array} \end{array} \Rightarrow \begin{array}{c} \begin{array}{c} r \\ \left(\begin{array}{c|c|c} \bullet & & \mathbf{0} \\ \hline \mathbf{0} & \bullet & \mathbf{0} \\ \hline \mathbf{0} & & \bullet \end{array} \right) \\ \left. \begin{array}{l} \frac{n}{r} \\ \frac{n}{r} \end{array} \right\} \\ \hline \left(\begin{array}{c} \mathbf{0} \\ \bullet \end{array} \right) \\ \left. \begin{array}{l} \frac{n}{r} + \text{mod} \left(\frac{n}{r} \right) \end{array} \right\} \end{array} \end{array}$$

この操作で(19)式は次式になる。

$$\mathbf{x}^*(k+1) = \mathbf{D}^*\mathbf{x}^*(k) + \mathbf{U}^*\Sigma^*\mathbf{V}^T\mathbf{u}(k) \quad (20)$$

つぎに、(20)式の両辺に左から \mathbf{U}^{*T} をかけて以下の式変換を行なう。

$$\begin{aligned} \mathbf{U}^{*T}\mathbf{x}^*(k+1) &= \mathbf{U}^{*T}\mathbf{D}^*\mathbf{x}^*(k) + \Sigma^*\mathbf{V}^T\mathbf{u}(k) \\ &= \mathbf{U}^{*T}\mathbf{D}\mathbf{U}^*\mathbf{U}^{*T}\mathbf{x}^*(k) + \Sigma^*\mathbf{V}^T\mathbf{u}(k) \end{aligned} \quad (21)$$

ここで、

$$\mathbf{z}(k) = \mathbf{U}^{*T}\mathbf{x}^*(k), \quad \mathbf{v}(k) = \mathbf{V}^T\mathbf{u}(k)$$

$$\mathbf{F} = \mathbf{U}^{*T}\mathbf{D}\mathbf{U}^*, \quad \mathbf{G} = \Sigma^*, \quad \mathbf{H} = \mathbf{C}\mathbf{U}^*$$

とおくと、(17)式は次式のブロック対角系に変換される。

$$\mathbf{z}(k+1) = \mathbf{F}\mathbf{z}(k) + \mathbf{G}\mathbf{v}(k), \quad \mathbf{y}(k) = \mathbf{H}\mathbf{z}(k) \quad (22)$$

このとき、評価関数

$$J = \sum_{i=1}^N \left[\sum_{k=0}^{k_i-1} \frac{1}{2} \{ \|\mathbf{z}_i(k)\|_{\mathbf{Q}_i}^2 + \|\mathbf{v}_i(k)\|_{\mathbf{R}_i}^2 \} \right] \quad (23)$$

を最少にするような制御を求めるが、このときの重み行列 \mathbf{Q}_i^* , \mathbf{R}_i^* は(6)の式の最少化を良く反映するように選ばなければならない。

以上によってシステム行列 \mathbf{F} , \mathbf{G} , \mathbf{H} および重み行列 \mathbf{Q}_i , \mathbf{R}_i が得られ、これらとともに初期値、終端時間および緩和係数を前述の最適化システムに入力して計算を行なうと、最適な状態軌跡と制御が得られる。一般に \mathbf{F} , \mathbf{G} が定数であれば、有限時間最適レギュレータ問題の行列 Riccati 方程式の解 $\mathbf{P}(k)$ は、終端時間を大きくとると定数 \mathbf{P}_c となる。したがって、上述の最適化システムにより得られる解を利用して、定常フィードバックゲインを求めることができる。

本研究では、前記の43次の状態変数と7次の制御変数を持つ2ループのPWR型原子力発電プラントモデルにおいて、核分裂発出力 (Neutron Power) が突然に定常値より変位するとし、この状態を最適に定常状態に復帰させるための制御を求めた。全系を、各サブシステムにおける

計算負荷がほぼ同程度になるように、7つのサブシステムに分割して、並列的に計算を行なわせている。最上位レベルの協調操作は3回、すなわち全反復回数は4回となり、計算時間は390秒であった。計算で求めた最適状態軌道のうち、Neutron Power および2次系の蒸気流量についての結果を図8および図9に示す。縦軸の値は定常値を1として規格化した状態の定常値からの変位を示す。この図より、両状態量とも、短時間のトランジェントを経て、十分な速さで定常値に復帰することが見られる。

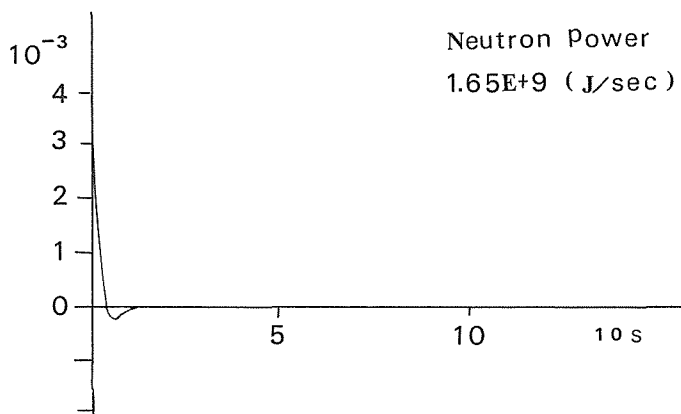


図8 Neutron Power の最適軌道

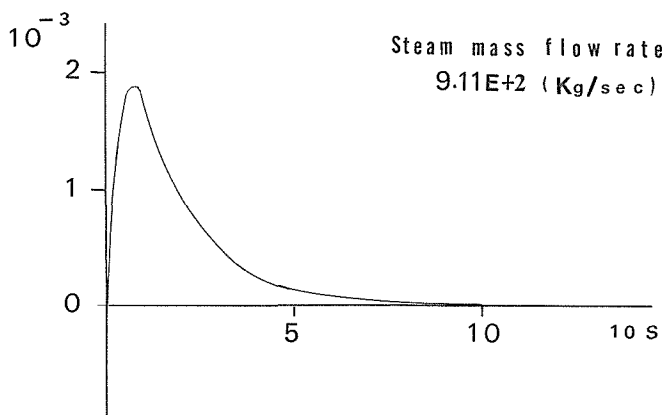


図9 蒸気流量の最適軌道

5. お わ り に

本研究は本学汎用シミュレータ室のPPAが階層構造という非常に興味のある構造を持つ計算システムであることに着目し、その構造を生かした並列演算方式の階層分権型最適化システムのソフトウェアを作成し、その有効性を計算実験で確認した。すなわち、状態変数の次数を種々に変えた計算実験用モデルを用いて、本研究で作成した階層分権型アルゴリズムと、サブシステムに分割しないで全系を一括的に計算する解法との間で、所要計算量の比較を行ない、本アルゴリズムが計算速度の上で極めて有利であり、その優位性が次数の増加とともに益々増大する結果を得ている。つぎに、作成した最適化システムを用いて、PWR型原子力発電プラントの最適出力レギュレータの設計を行なった。用いたプラントモデルは状態変数が43次、制御変数が7次の大規模システムであり、全系を一括して最適計算を行なうことは容易ではない。本研究は、PPAを利用した並列演算方式の階層分権型最適化システムを用いて、原子力発電プラント全系を7つのサブシステムに分割し、各サブシステムを独立的に計算することによって、最適レギュレータの設計を非常に容易なものにしている。求めた最適制御の適用によって、定常値から変位したNeutron Powerや2次系の蒸気流量などの状態変数が十分な速さで定常値に復帰することが見られた。

本学汎用シミュレータ室のPPAは、設計者の意のままに変更することが可能な融通性のある制御構造を持っており、最適なアルゴリズムを容易に構成することができる極めて魅力のあるシステムであることが、本研究をとうして再認識された。

謝 辞

本研究は北大工学部の汎用シミュレータ施設のPPAを利用して行なった。施設の利用において、種々の御指導と御助言をいただきました牧野助手と丸山技官に深く感謝致します。

引用文献

- 1) 牧野圭二, 他: 電子通信学会技術研究報告, EC82 (1982), p.57
- 2) 茨木俊秀: 組合せ最適化一分岐限定法を中心として, (昭和58), 産業図書
- 3) 西川 一, 外: 最適化, (昭和60), 岩波
- 4) 高間宣博, 外: システムと制御, 25 (昭和56) 10, p.642
- 5) M. G. Singh: Decentralized Control, (1981), North Holland
- 6) 渡辺祐二: 修士論文 (昭和59), 北海道大学工学部原子工学専攻