

最適制御並びに適応制御に関する統計的研究

阪大 工 今村秀樹 魚崎勝司
田坂誠男 杉山博

I. 序

適応制御とは、一般的に定義すると、時間とともに変動する環境条件にたいして、制御系の最適動作点を追跡し、系をつねに最適状態に保持する制御系であるといえる。さらに、このような適応制御系の発展として、過去の経験を最大限に活用し、随時、跳躍によって最適状態に到達しようとする学習制御系 [1] も、最近の研究の対象となっている。

今回は、適応最適制御の問題を離散的な時間領域で取扱ひ、しかも従来の feed-back 的な制御形式とは異なつて、適応制御 = 予測という feed-forward 的な制御形式についての、Box, むらび Jenkins [2] のアルゴリズムを簡単に紹介し、このアルゴリズムに基づいた、試行法による適応最適化制御についての考察と、入力変数が 2 つの場合の、二次元における学習制御について、それぞれ電子計算機によるシミュレーションの結果

を挙げて説明を加える。

II. 適応最適化制御

(1). アルゴリズム

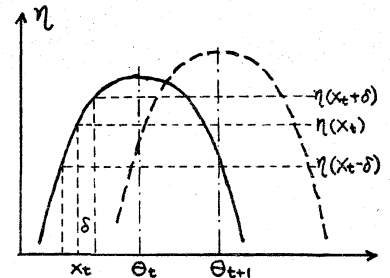
いま、時間軸を等間隔に区切った離散時刻を考える。この区切られた一つの時間中を phase とよび、この phase 内ではプロセスの状態は不変であるとする。この時、プロセスの目的関数を η とすると、 η は時刻 t における操作入力変数 X_t を独立変数とする次のような二次関数で表されるとする。これは、最適点の周辺では、二次関数で十分近似できるとの、Box 等 [3] の仮定によるものである。

$$\eta_t = \eta(X_t | \xi_t) = \eta(\theta_t) - \frac{1}{2} \beta (X_t - \theta_t)^2 \quad \dots \dots (1)$$

$$\theta_t = \{X_{opt} | \xi_t\} : X_t \text{ の最適値}$$

ξ_t : 時刻 t における制御観測共に不能な変数のレベル

β : calibration により既知で ξ_t とは独立不変 Fig. 1



このような離散時刻での取扱いは化学工業における batch-type process などに関して考えられる。さて、このような目的関数を持つプロセスを制御する上での基準としては、 X_t の最適値である θ_t と、 X_t との偏差を最小にする、つまり二次関数の仮定のもとでは、 $X = X_t$ における η_t の規格化された勾配 $\frac{1}{\beta} \frac{d\eta(X_t)}{dX_t} = \theta_t - X_t$ をできるだけ小さくすることが目的となる。そこで

入力変数 X_t を、ある微小な値 δ だけ変動させて X_t における η の勾配を実測すると、その時の規格化された勾配は

$$e_t = \frac{1}{2\delta\beta} \{y(X_t+\delta) - y(X_t-\delta)\} = \epsilon_t + u_t \quad (2)$$

但し $y(X_t \pm \delta)$: $\eta(X_t \pm \delta)$ の平均値

$$\epsilon_t = \theta_t - X_t$$

$$u_t = \text{観測誤差}$$

ϵ_t は理論的偏差である。(2) 式をさらに書き改めて、

$$e_t = \theta_t - X_t + u_t = \bar{x}_t - X_t \quad (3)$$

$$\text{但し } \bar{x}_t = \theta_t + u_t$$

このように \bar{x}_t は、 $\eta(x)$ を最大にする入力の推定値であると考えられる。つぎに時刻 $t+1$ での入力を X_{t+1} とするとき、前時点の入力 X_t に与える補正量を x_{t+1} とすれば、次のような線形関係が成り立つとする。

$$X_{t+1} = X_t + x_{t+1} = X_0 + \sum_{j=0}^t x_{j+1} \quad (4)$$

ところで時刻 $t+1$ での目的関数の値 $\eta(X_{t+1})$ と、その時の最適値 $\eta(\theta_{t+1})$ との差は

$$\frac{1}{2} \beta (X_{t+1} - \theta_{t+1})^2 \quad (5)$$

であるから、 x_{t+1} としては、(5) を最小にするように選ぶのが妥当である。したがって X_{t+1} を、その最適値 θ_{t+1} にとれば良い訳であるが、現時点 t においては直接 θ_{t+1} を知る事ができないので、過去の情報に基づいて θ_{t+1} を推定しなくてはならない。つまり、(3) より

$$X_{t+1} = \hat{\theta}_{t+1} = f(z_t, z_{t-1}, \dots) \quad (6)$$

但し $\hat{\theta}_{t+1}$: θ_{t+1} の推定値

一般に $\hat{\theta}_{t+1}$ は process $\{z_i; i=0, 1, 2, \dots\}$ の非線型関数であるが、簡単のために、いまこれを $\{z_i\}$ の線型関数で表すと

$$\hat{\theta}_{t+1} = \sum_{j=0}^t \mu_j z_{t-j} \quad (7)$$

μ_j は prediction weight と呼ばれるもので、したがって補正量 x_{t+1} は次のように表される。

$$x_{t+1} = X_{t+1} - X_t = \hat{\theta}_{t+1} - \hat{\theta}_t = \sum_{j=0}^t w_j e_{t-j} \quad (8)$$

ここに $\{w_i\}$ は controller weight と呼ばれるものであり、(8) で (7) の $\{z_i\}$ の代りに $\{e_i\}$ を用いたのは、実際には $\{z_i\}$ の測定は不可能であつて、唯一の情報 $\{e_i\}$ のみであるためである。このように、結局、系を最適にする補正量 x_{t+1} は

$$E(e_{t+1}^2) = E(\theta_{t+1} - X_{t+1})^2 = E(\theta_{t+1} - \hat{\theta}_{t+1})^2 \quad (9)$$

を最小にする、つまり予測誤差を最小にするように、重み $\{w_i\}$ を選ぶことにより得られる。

ここで、前述の (3) 式

$$e_t = z_t - X_t \quad (3) ; \quad z_t = \theta_t + u_t \quad (3')$$

において、 θ_t をあるプロセスの品質特性、 z_t をその測定値、 u_t を測定誤差と考えると、この場合の制御の目的は、 θ_t をできるだけ目標値（一般性を失うことなくこれを 0 とする）に近づけることである。したがって時刻 t で z_t に与える全補正

量を $-X_t$ とすると、目標値からの偏差は (3') で表される。つまり、このような問題に対しても同様の議論が展開される。結局、この種の制御問題は、(9) 式で表される、予測誤差を最小にする重みを推定する問題となる。

(2) Controller Weight " w_j " の三項モデル

重み $\{w_j\}$ の形態として最もよく知られているのは、次に示すような“三項モデル”と呼ばれるものである。

$$x_{t+1} = \gamma_1 \Delta e_t + \gamma_0 e_t + \gamma_1 \sum_{j=0}^t e_{t-j} \quad (10)$$

$$\text{但し } \Delta e_t = e_t - e_{t-1}$$

これと (8) 式とを対応させると

$$\left. \begin{aligned} w_0 &= \gamma_1 + \gamma_0 + \gamma_1 \\ w_1 &= \gamma_1 - \gamma_1 \\ w_i &= \gamma_1 \quad (i \geq 2) \end{aligned} \right\} \quad (11)$$

一般に (10) 式のモデルは大部分のプロセス特性変動によく合っているといわれている。この三項モデルを用いる場合に、プロセスを最適状態に導くパラメータの組 $(\gamma_1^*, \gamma_0^*, \gamma_1^*)$ の附近でのパラメータの値 $(\gamma_1, \gamma_0, \gamma_1)$ の変化に対する最適状態の安定性が問題であるが、Box, Jenkins [1] の例にもあるように、 $(\gamma_1, \gamma_0, \gamma_1)$ のかなり広い領域で、最適状態は安定であることが確かめられた。

Fig. 2. は、その例で、プロセス I, II, III の三つのタイプについて、それぞれの場合の偏差の二乗和を、パラメータ $(\gamma_1, \gamma_0, \gamma_1)$ の各格子点について計算したものである。

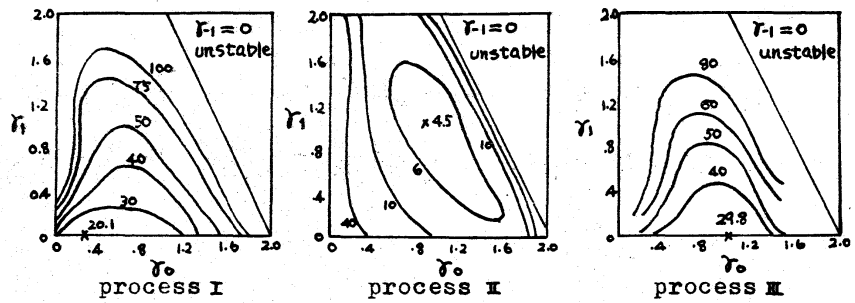


Fig. 2. Response surfaces of process parameter ($\gamma_1, \gamma_0, \gamma_1$)

ここで $(\gamma_1, \gamma_0, \gamma_1)$ の安定性について、我々の得た結果を示す。

結果 1. 補正が施されたいときのプロセス $\{z_i\}$ および、三項モデルにより補正が施されたときのプロセス $\{e_i\}$ の母函数をそれぞれ $z^*(s)$, $e^*(s)$ とする。

$$z^*(s) = \sum_{j=0}^{\infty} z_j s^j, \quad e^*(s) = \sum_{j=0}^{\infty} e_j s^j \quad (12)$$

このとき、 $z^*(s)$ と $e^*(s)$ との間には次の関係が成立する。

$$\{1 - (2-a)s + (1+b-a)s^2 + (c-b)s^3\} e^* = (1-s)^2 z^* - (1-s)X_0 \quad (13)$$

但し X_0 : (4) 参照

$$\left. \begin{aligned} a &= \gamma_1 + \gamma_0 + \gamma_1 \\ b &= \gamma_1 - \gamma_1 \\ c &= \gamma_1 \end{aligned} \right\} \quad (14)$$

この結果と、後に示す結果に加えて、安定条件

$$\lim_{k \rightarrow \infty} \sigma^2(e_k) < \infty \quad (15)$$

より $(\gamma_1, \gamma_0, \gamma_1)$ に対する系の安定領域が求められる。

例: $\gamma_1 > 0$ のとき $\gamma_1 + 2\gamma_0 > 0, \quad 4\gamma_1 - \gamma_1 + 4 > 0$
 $\gamma_1 + 4\gamma_1 + 2\gamma_0 - 4 < 0, \quad \gamma_0 + \gamma_1\gamma_0 + \gamma_1\gamma_1 > 0$

(3). 三項モデルによる偏差 $\{e_t\}$ の統計的性質

一般に、偏差 e_t は補正を施さぬときには z_t の分布と一致し、補正を施すと、 $z_t + X_t$ の分布に等しくなる。そこでこのよう
に e_t について、統計的性質を調べてみる。

$$\text{いま, } E(\theta_t) = m_t, \quad E(u_t) = 0, \quad \text{Var}(u_t) = \sigma^2$$

とし、さらに(4)式より、時刻 $t+1$ での補正量 X_{t+1} を

$$X_{t+1} = X_0 + \sum_{j=0}^t x_{j+1} = \sum_{j=0}^t \sum_{k=0}^j w_k e_{j-k} + X_0 = \sum_{j=0}^t \alpha_{t,j,0} e_j \quad (16)$$

$$\text{但し } \alpha_{i,j,0} = \begin{cases} \sum_{k=0}^{i-j} w_k & (i \geq j) \\ 0 & (i < j) \end{cases} \quad (17)$$

と表すと、偏差 e_t の特性関数は

$$\phi_t(p) = \exp(ipX_0) \prod_{j=0}^t \{ \phi(-\alpha_{j,j,t-j-1} p) \cdot \exp(-i\alpha_{j,j,t-j-1} m_j \cdot p) \} \quad (18)$$

で与えられる。 $\alpha_{i,j,k}$ に関する漸化式は

$$\alpha_{i,j,k} = \alpha_{i+1,j,k-1} - \alpha_{i+1,j+1,k-1} \cdot \alpha_{i,j,0} \quad (19)$$

$$\text{但し } \alpha_{i,j,-1} = -1$$

三項モデルによる補正を施したときの(18)式のパラメータ $\alpha_{i,j,k}$ について、次の結果が得られる。

結果2. 三項モデルによる補正を施すと

$$e_t = - \sum_{j=0}^t \alpha_{j,j,t-j-1} z_j \quad (t=0, 1, 2, \dots) \quad (20)$$

$$\alpha_{i,j,0} = \sum_{k=0}^{i-j} w_k = \begin{cases} r_1 + r_0 + r_1 & (i-j=n=0) \\ r_0 + (n+1)r_1 & (i-j=n \geq 1) \end{cases} \quad (21)$$

で与えられる。ここであらためて、

$$\alpha_{i,j,0} = d_{i-j+1} = d_{n+1} = \begin{cases} \delta_1 + \delta_0 + \delta_i & (n=0) \\ \delta_0 + (n+1)\delta_i & (n \geq 1) \end{cases} \quad (22)$$

とすると, $\alpha_{i,j,k}$ は

$$\alpha_{i,j,k} = \sum_{m=1}^{k+1} d_{i-j+m} \sum_{(k+1-m)} \frac{(l_1 + \dots + l_k + \dots)!}{l_1! \dots l_k! \dots} d_i^{l_1} \dots d_k^{l_k} \dots (-1)^{l_1 + \dots + l_k + \dots} \quad (23)$$

($k \geq 0$)

ただし $\sum_{(S)}$ は $\sum_{j=1}^{\infty} j l_j = S$ とする $(l_1, l_2, \dots, l_k, \dots)$ のすべての組合せについての和である。上記の結果は、数学的帰納法により証明される。

結果3. $E(z_t) = m_t$, $E(z_p z_q) = \sigma^2 \delta_{pq}$ ($p, q = 0, 1, 2, \dots$) とする。ただし, $\delta_{pq} = 0$ ($p \neq q$), $\delta_{pq} = 1$ ($p = q$) である。このとき, z_t に (1b) 式で表されるよう $T_0 - X_t$ は補正を施すと, 偏差 e_t に関し, 次式が成立する。

$$\left. \begin{aligned} E(e_t) &= - \sum_{j=0}^t \alpha_{j,j,t-j-1} \cdot m_j \\ \text{Var}(e_t) &= \sum_{j=0}^t \alpha_{j,j,t-j-1}^2 \cdot \sigma^2 \\ E(e_t^2) &= \sigma^2 \sum_{j=0}^t \alpha_{j,j,t-j-1} + \sum_{k=0}^t \sum_{l=0}^t \alpha_{k,k,t-k-1} \alpha_{l,l,t-l-1} \cdot m_k m_l \end{aligned} \right\} \quad (24)$$

(4). 適応最適化制御シミュレーション装置の構想

以上のアルゴリズムに基づいて, 適応最適化制御のシミュレーション装置についての構想を得た。まず, 偏差 e_t の測定法であるが, (2)式より明らかのように, 目的関数の勾配が偏差に対応するので, ここでは, 比較的簡単で, 連続的に測定可能。

探索信号法を用いる。この方法は、主に正弦波を変数 x に重畳させるもので、この方法を用いると、最適点 θ の左右で、周波数成分の位相が -180° 可成るので、勾配の符号の判定が可能となる。次に、三項モデルのパラメータ $(\gamma_1, \gamma_0, \gamma_i)$ は、逐次探索により求める方法を採用する。ここで用いる逐次探索の方法としては、一変数変動法(単独試行)と多変数同時変動法(複合試行)とも交互に用いる試行法[4],[5],[6]を使用する。このシミュレーション装置はFig.3.で示す、サンプウを含むクロズド=ループで表される。

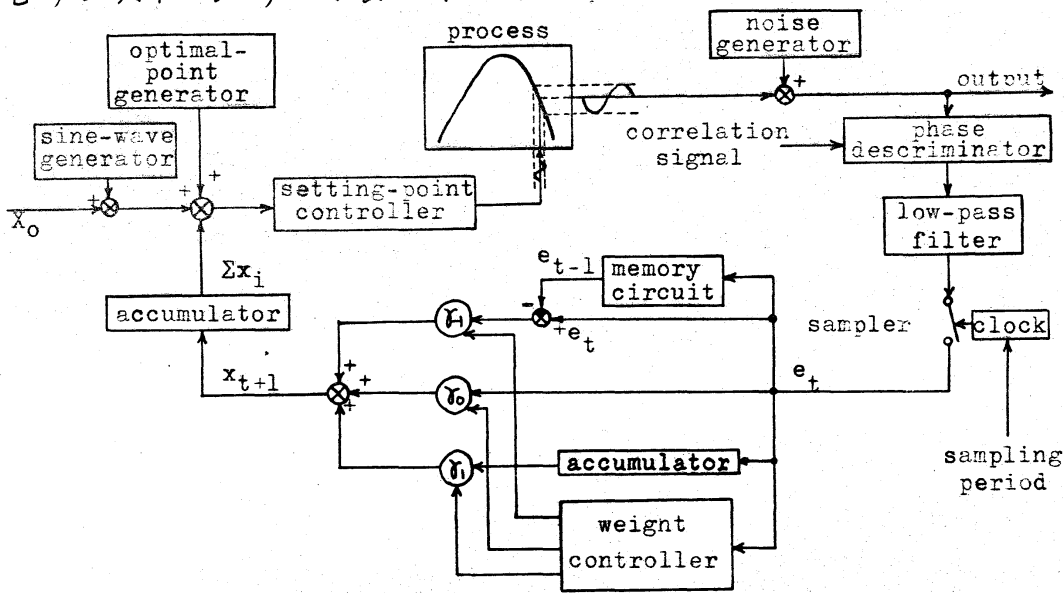


Fig.3. Block diagram of adaptive optimizer

(5). 実験と結果

以上のように、試行法を用いて、最適パラメータ $(\gamma_1, \gamma_0, \gamma_i)$ を探索しながら制御を行って、いく制御装置のシミュレーション実

験を、電子計算機を利用して行った。実験条件としては、

プロセス特性(目的関数) : $y = -x^2$

最適点 θ_t の変動 : 正弦波, 三角波, 変化なし

パラメータ ($\gamma_{-1}, \gamma_0, \gamma_1$) の初期値 : (0.0, 0.0, 0.0)

逐次探索の試行中 $\Delta \gamma_i$: 0.1, 0.01, 0.001 ($i=-1, 0, 1$)

の場合を考えた。その結果を Fig. 4~8 に示す。

(6). 検討

(I). 試行ステップ中 $\Delta \gamma_i$ と最適点変動 (Fig. 4, Fig. 5)

試行ステップ中 $\Delta \gamma_i$ は、最適点の変動状態に固有のもので、その固有の $\Delta \gamma_i$ よりも大きいと、パラメータの最適状態の領域をまたいでしまい、振動状態、若しくは不安定領域に落ち込んで、プロセス変動に追従できなくなる。一方逆に、 $\Delta \gamma_i$ が小さすぎると追従状態に達するのが遅く、また達したとしても追従遅れが生じることが想像されるが、実験によりこれが確かめられた。一般に制御を始める段階では、適当な $\Delta \gamma_i$ は不明であるので、任意の $\Delta \gamma_i$ から出発し、偏差の2乗の時間平均を観測して、その値が異常に大きければ $\Delta \gamma_i$ を小さくし、一方減少のきざしがなく、或いは周期的に変化する場合には $\Delta \gamma_i$ を大きくするよう調整すれば、最適点変動への追従がより可能となる。

(II). サンプリング間隔

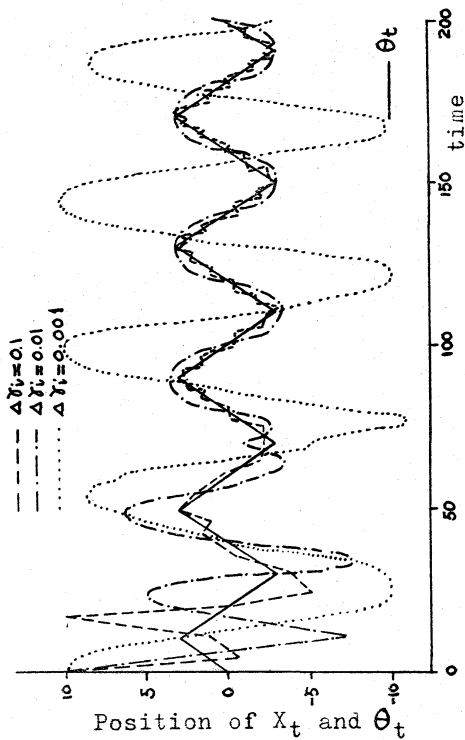


Fig. 5. Pursuit behaviours in the case of triangular-waved optimal point.

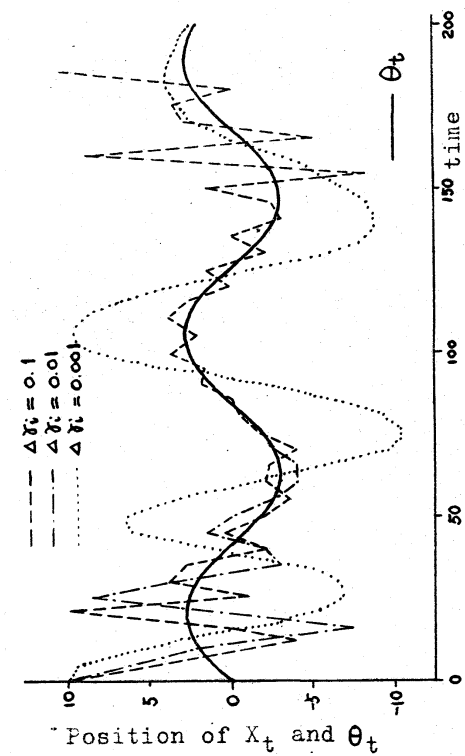


Fig. 4. Pursuit behaviours in the case of sine-waved optimal point.

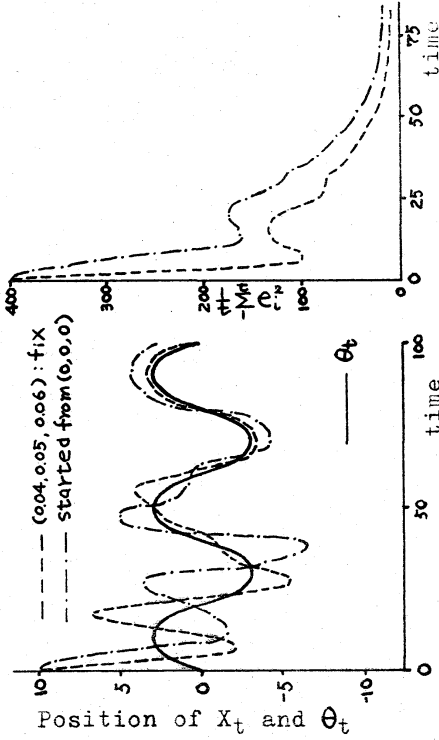


Fig. 7. Behaviours in the case of different starting values of parameter

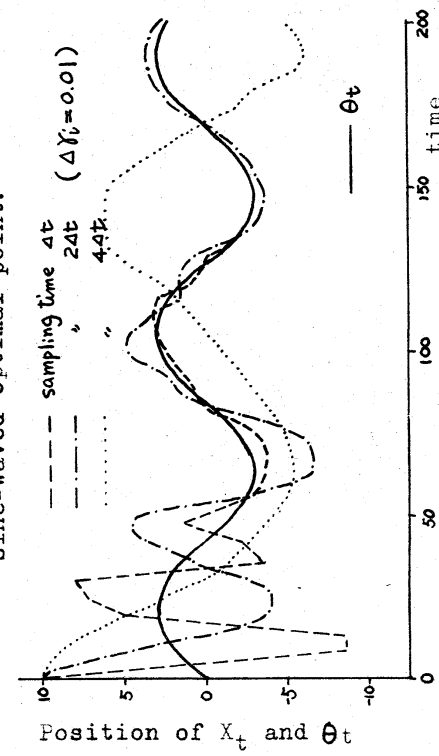


Fig. 6. Pursuit behaviours in the case of different sampling periods.

サンプリング間隔は小さい程、より有効な追跡が可能となるのは、Fig. 6. から明らかなのであるが、実際の場面では、サンプリング頻度を大きくするに要する費用と、それによつて得られる利益との関係から決定しなければならぬ。

(四) 試行法によるパラメータの推定

Fig. 7. は、パラメータの初期値(0.0, 0.0, 0.0)から出発し、追従状態に到達する過程と、到達状態に達した時のパラメータの値(0.04, 0.05, 0.06)に固定して制御を行った場合の追跡の様子を示している。Fig. 8. は、その時の、それぞれの偏差の2乗の時間平均値を示したものである。これより、両者の間には初期の状態を除いて、きわだつた相異はみられない。このように、試行法でパラメータを推定しながら制御を行つていくという方式の可能性が、適応制御の場合にも示される。

II. 二次元における学習制御

学習系および学習制御の発達過程、一般的概念、ならびに具体的な説明は、参考文献[1]に詳しく記しており、さらに一次元での学習制御(Learntrol I~V)についても設計原理と実験結果が載せてあるので、ここではその解説を省略し、一次元のLearntrolのアルゴリズムに基づいて、これを二次元に拡張し

に時の学習制御の設計原理と、系の学習行動について述べ、併せてシミュレーション実験の結果を示す。

(1). 二次元の学習制御の設計原理と学習行動

ここで述べる学習制御とは次のようなものである。つまり制御の初期段階では、制御系の最適動作点を逐次的に探索する適応制御系として行動し、過去の経験が豊富になってくるとにたがひ、それらを活用して跳躍を行い、より早く最適点を探索しようとするものを学習制御系という。一般的にプロ

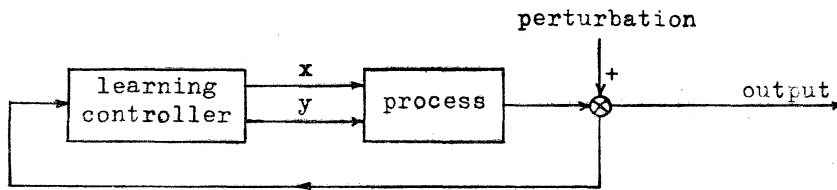


Fig.9. Block diagram of learning control system.

7線図を Fig.9. に示す。操作変数(入力)は二変数 x , y で、出力は性能指数 P であり、制御対象は入力に対して単峰性の山を有する静特性を持つものとする。しかもその頂点の位置だけが負荷(外乱)の突然変化によって、変化する(つまり離散的に変化する)が、負荷に突然変化の無い状態では一定不変であると仮定する。またこの制御系は、過去の経験を活用する上での経験の記憶方法として、以下に説明するような地図と生起度数表、および忘却機能を有する。

地図 系の逐次探索の結果が要約されて、それが後続経験の

ために記憶されているものである。その記憶動作は次の通りである。ある点 (x_k, y_k) より逐次探索を開始したとき、その点での出力 P_k と、曲面の傾き $\frac{\partial P_k}{\partial x_k}, \frac{\partial P_k}{\partial y_k}$ とを初期状態 $(P_k, \frac{\partial P_k}{\partial x_k}, \frac{\partial P_k}{\partial y_k})$ として、この逐次探索により、最終の頂点に到達するまでに操作量が要した変化量 $(\Delta x_m, \Delta y_m)$ とを結合した、 $(P_k, \frac{\partial P_k}{\partial x_k}, \frac{\partial P_k}{\partial y_k}, \Delta x_m, \Delta y_m)$ を、逐次探索の結果として要約する。地図への書き込み方法としては、あらかじめ $(P_k, \frac{\partial P_k}{\partial x_k}, \frac{\partial P_k}{\partial y_k})$ の各成分の領域をグループに分け、各グループに整数番号を与え、それによって決まる整数の組 (L, M, N) に有限個の記憶番地を与える。そして現在点 (x, y) の初期状態 $(P, \frac{\partial P}{\partial x}, \frac{\partial P}{\partial y})$ が、どの整数の組に属するかを判定して、その属する組の記憶番地に逐次探索で要した変化量 $(\Delta x, \Delta y)$ を書き込む。

生起度数表 経験の生起度数をかぞえ、頻度が最大の経験を最も活用し、他方、頻度の小さい経験を忘却するためのもので、同一経験が繰り返し記憶される毎に1を加算していく。

忘却機能と地図、生起度数表の並べ代え 一般には地図に記憶される変化量 $(\Delta x_m, \Delta y_m)$ の番地の数には制限があるため、新しい経験が得られた場合、度数表に照らし合せて、最小の度数を持つ経験を、地図から忘却させると共に、その新しい経験をそこに書き込み、同時に、その生起度数を1にする。

さて、以上の地図、生起度数表、忘却機能を持った学習系

の学習行動は、次のフローチャートに示すようなものである。

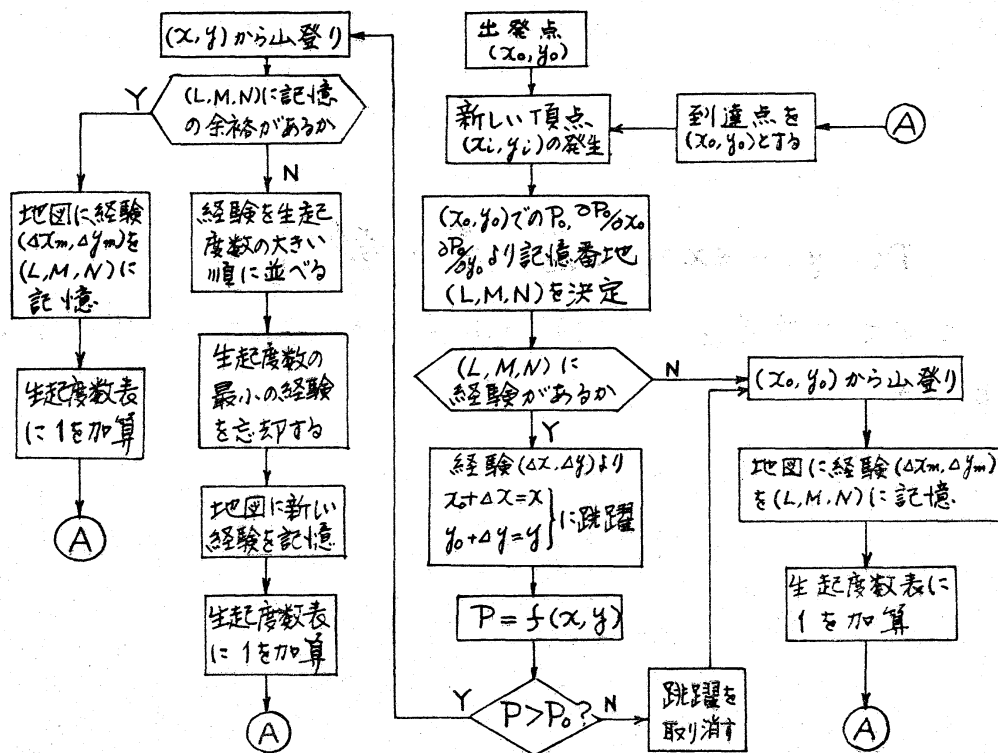


Fig.10. Flow chart of learning behaviour.

つまり現時点の位置から $(P_n, \frac{\partial P_n}{\partial x_n}, \frac{\partial P_n}{\partial y_n})$ を計算してその記憶番地を捜し、そこに経験が記憶されている場合には、その操作変数の変化量 $(\Delta x_m, \Delta y_m)$ だけの跳躍を行って点 $(x_n + \Delta x_m, y_n + \Delta y_m)$ での P が P_n より大きければ、この跳躍を採用し、逆に小さい場合には、この跳躍を破棄して、点 (x_n, y_n) を起点とする逐次探索に切り換え、試行を続けていく。一方、経験の多いときには、逐次探索を行使する。

(2). 実験と結果

以上のアルゴリズムに基づいて、二次元での学習制御の電子計算機によるシミュレーション実験を行なった。実験条件は下記の通りである。なお実験は4つの場合について行なわれた。

①系の静特性の二次曲面

$$P(x, y) = \exp\left[-\left(\frac{x^2}{25} + \frac{y^2}{50} + \frac{xy}{25}\right)\right]$$

この曲面は単峰性の楕円形の山で、その断面は正規分布状を示す。

②逐次探索の試行法[4], [6]

逐次探索に用いる試行法としては、比較的よく知られている次の4方法を用いる。

実験(I). 最急上昇法

実験(II). 直角4試行法

実験(III). 直角4試行最大傾斜直進法

実験(IV). 繰り返し4試行法

③地図への書き込み方法

実験(I), (III). 出発点での山の傾き(方向) : $(P_0, \frac{\partial P_0}{\partial x_0}, \frac{\partial P_0}{\partial y_0})$

実験(II), (IV). 出発点の位置 : (P_0, x_0, y_0)

④山の位置の変動

外乱の突然変動は負荷変動のみによるものとし、これによって山の位置が動くが、その変動発生の間隔は、系の適応時間よりも大であるとする。しかし必ずしもそうでない場合については、実験(V)で行なった。

以上の実験の結果を Fig. 11.~14. に示す。

(3). 検討

Fig. 11.~13. の学習曲線より、あきらかに、経験が増大するにつれ、つまり学習が進行するにつれて学習制御の効果が現れてきている。

(I). 学習方法(地図への書き込み方法)

Fig. 13. の跳躍発生率から推察すると、実験(I), (II)の結果より、山の“傾き”, すなわち“方向”を知るということが、現時点での位置と、且つ頂点の位置を知るということに相当し、たんに現時点での“位置”を知るということよりも多くの情報を含んでいることがわかる。一次元の学習制御では“位置”を問題にしているが、二次元の場合には単なる“位置”よりも“傾き(方向)”が重要であるといえる。

(II). 逐次探索に用いる試行法

逐次探索の試行法としては、学習の進行が著しく早い、短期間での学習制御には、直角4試行直進法、繰返し4試行法、直角4試行最大傾斜直進法が、実験結果から推察すると、より有効であると思われる。一方、長期間に渡る学習制御では最急上昇法がより有効であることが、出力Pを高める点から考えられる。

(III). 負荷変動の間隔: 山の位置の変動

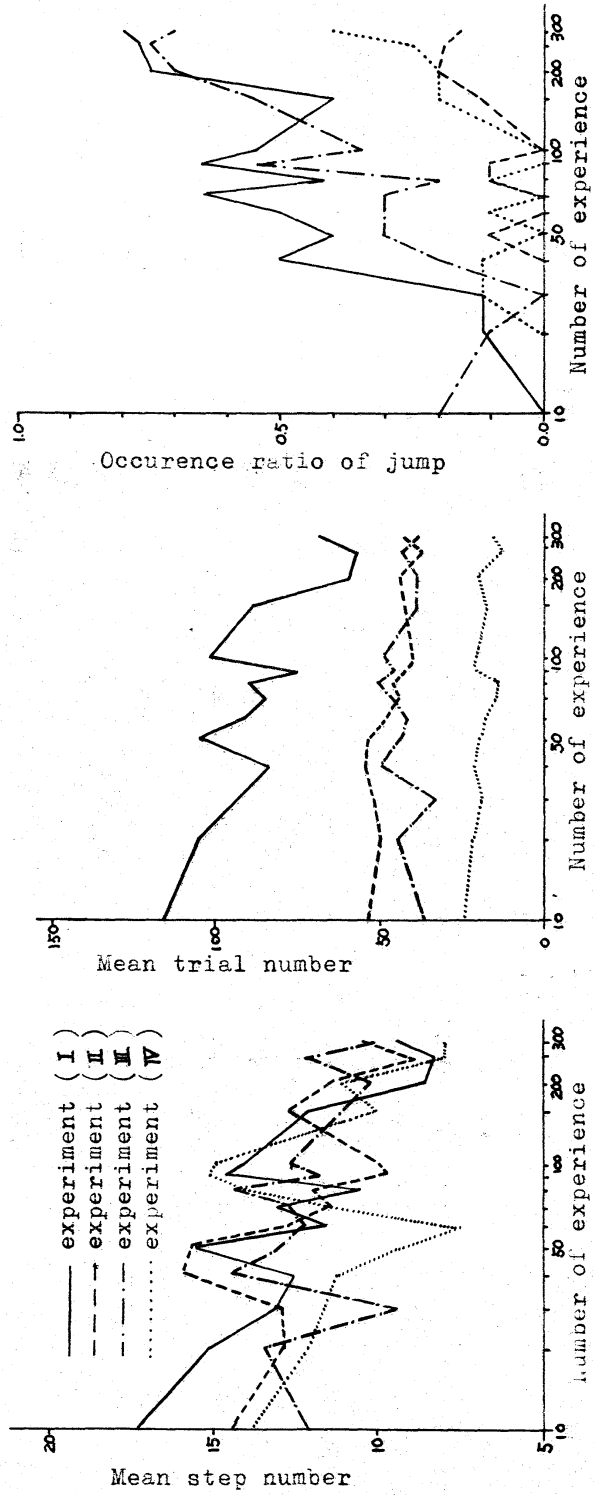


Fig.11. Learning curves

Fig.12. Learning curves

Fig.13. Learning curves

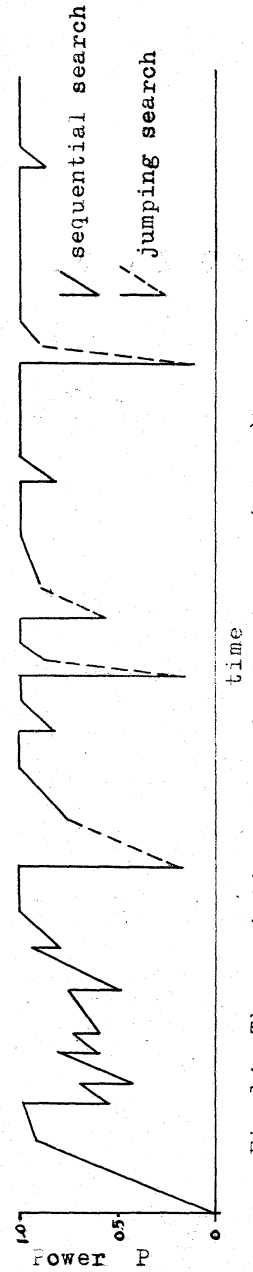


Fig.14. The variation of performance index (power) P by the influence of the peak moving discretely at random time intervals.

実験(V)において、我々は負荷変動の間隔、つまり、山の頂点の位置の変動の時間々隔が、必ずしも系の適応時間よりも大でない場合(実験(I)~(IV)では、これを大とした)の学習制御のシミュレーションを行なった。Fig. 14.に、この場合の出力変化の有様を示す。この図によると、初期のうちには、経験を終えないうちに、山が動くので、経験量が少なくて、逐次探索でしか頂点へ向えないが、中期以降では、序々に経験が増え、その経験を活用して跳躍的に頂点へいき、出力が安定してくる。したがって、この結果より、必ずしも負荷変動の時間々隔を系の適応時間より大にする必要はなく、こうすることにより、より一般的にシミュレーションが可能となる。

IV. あとがき

ここで述べてきた事柄は、まだ基礎的研究の段階であって、まだまだ議論の余地があり、疑問点も残出するものと思う。今後の課題としては、II. については、多変数入力を有する場合のアルゴリズムと、そのシミュレーション装置の設計、さらには、簡便で多方面に応用可能な制御装置の開発にあり、III. の場合については、試行法のステップ中等の各種のパラメータを、学習系自身に決定させること、さらに *stochastic* 要素の存在する場合、多峰性の場合、山の位置と共に、その形態が変

動する場合等についてのアルゴリズムの研究にあつた。

Reference

1. 比川敏男編, "学習制御および学習制御機械," 情報科学講座 E. 17. 4., 共立出版 (1966)
2. Box, G. E. P., Jenkins, G. M., "Some Statistical Aspects of Adaptive Optimization and Control", J. R. S. S. Ser. B, 24, p297 (1962)
3. Box, G. E. P., Wilson, K. B., "On the Experimental Attainment of Optimum Condition", J. R. S. S. Ser. B, 13, (1951)
4. Douglass J. Wilde, "Optimum Seeking Methods", Prentice-Hall, INC, (1964)
5. 福永, "試行法による多変数最適化制御", 制御工学, Vol. 4, (1960)
6. 浅井, 比嶋, 平井, "二変数入力の場合の最適化の試行法および試行装置," 電気学会誌, Vol. 81 (1961)