準最短時間計算機制御のアルゴリズム†

馬 場 順 昭

Algorithms of Time-Suboptimal On-Line Computer Control

Yoriaki BABA*

This paper is concerned with algorithms for the on-line closed-loop computer control of a linear process. The control signals are determined so as to force the disturbed state to achieve the desired state as soon as possible. The control concepts are based on a time-suboptimal control for a linear discrete-time system. In the proposed algorithms neither long computation time nor excessive computer memory are required, but some computation time delay exists.

1. ま え が き

線形定係数系のオンライン閉ループ計算機制御において評価関数が準最短時間のときのアルゴリズムについて述べる。計算機制御では計算時間と記憶容量が重要である¹⁾。計算時間はプロセスの変化の速さより十分速くなければならないし、必要とされる容量は大きすぎないほうがよい。外乱に対しては早目に正しい制御を行わねばならないし、また状態を目標値に完全に整定させるには計算時間遅れを考慮しなければならない。

計算機制御系は離散値系となるので基本的にばアルゴリズムの開発は入力に拘束を有する準最短時間離散値制御問題の解法に帰着する。この種の解法はいくつか提案されているが²⁾⁻⁹⁾、これらは必ずしも上の条件を満足していないし、またそのままの形で利用できない場合が多い。

本論文では準最短時間制御の三つの方法について触れ、これらの欠点を克服するため組合せて用いることを提案する。また実際問題として全状態空間にわたって制御を求める必要はないので⁷⁾、目標値のまわりの適当な集合のみを考え、最小サンプリング数の最大値

を求め、計算時間遅れを定める。さらに、外乱に対し早目に対処するとともに整定するまでの所要時間を短縮するように逐次制御の更新を行う方法を考慮している。

2. 問題の設定

オンライン閉ループ計算機制御系においてプロセスの動特性を次式とする。

$$\dot{X}(t) = AX(t) + BU(t) \quad (1)$$

ここで、 $X(t)$ は n 次元状態ベクトル、 $U(t)$ は r 次元入力ベクトル、 A は $n \times n$ 定数行列、 B は $n \times r$ 定数行列とし、 $\text{rank } B = r$ とする。制御入力には制限があり、この許容制御の集合を Ω とする。

$$\Omega = \{U(t) : \alpha \leq U(t) \leq \beta\} \quad (2)$$

ただし、 α, β は r 次元定数ベクトルとする。状態変数はすべて直接測定でき、プロセスは許容制御可能とする。制御量は $X(t)$ そのものであり目標値を X_d とする。以上の条件のもとに $X(t)$ が外乱により X_d からずれたとき、または X_d が変更されたとき、 $X(t)$ をできるだけ早く X_d へ整定させる準最短時間制御を求めよう。ただし、外乱の統計的性質は不明であるとする。ここで、制御はつぎの不等式で与えられる集合 Γ 内の初期点に対してのみ考えれば十分である。

$$\Gamma = \{X(0) : \gamma \leq X(0) - X_d \leq \delta\} \quad (3)$$

ただし、 γ, δ は n 次元定数ベクトルとする。この制御目的を達成するため計算機は一定周期 T_0 秒ごとにプロセス状態を監視し、常時新しい制御信号を算出する。この系で制御入力はサンプリング周期ごとの階段波となり、(1)式の状態推移方程式を次式で表わし S_d とおく。

$$X(k+1) = \Phi(T)X(k) + H(T)U(k) \quad (S_d) \quad (4)$$

ただし、 T はサンプリング周期、 k はサンプリング数、 $X(k), U(k)$ はそれぞれ $t = kT$ における状態ベクトルおよび制御ベクトル、 $\Phi(T), H(T)$ は $n \times n$ 遷移行列、 $n \times r$ 駆動行列である。また、 S_d は許容制御可能であり、かつ $\text{rank } H(T) = r$ となるように T は選

† 第18回自動制御連合講演会で発表 (昭50・11)

* 防衛大学校 横須賀市走水 1-10-20

* National Defence Academy, Yokosuka

(Received February 18, 1976)

ばれるものとする⁸⁾.

3. 有限整定時間応答

3.1 最小サンプリング数制御

S_d は許容制御可能なので任意の初期点 $X(0)$ と任意の目標値 X_d に対して有限な長さ m の入力列が存在し次式が成立する.

$$P(T; m)U(T; m) = X_d - \Phi^m(T)X(0) \quad (5)$$

ただし,

$$P(T; m) = [\Phi^{m-1}(T)H(T) \mid \Phi^{m-2}(T)H(T) \mid \dots \mid \Phi(T)H(T) \mid H(T)]$$

$$U(T; m) = [U'(0) \mid U'(1) \mid \dots \mid U'(m-1)]' \quad (6)$$

ここで, 制御入力に拘束がないとすると任意の初期点を X_d へ整定させるための最小サンプリング数 m_0 は S_d の可制御性指数に等しく⁸⁾, このときの入力列はつぎの二つの場合に分けられる.

$$(a) \quad m_0 = [n/r]; \quad (b) \quad m_0 > [n/r]$$

ただし, $[\cdot]$ は $\cdot$ の整数部を示す. (a) の場合, $P(T; m_0)$ は $n \times n$ 正則行列となり入力列は次式となる.

$$U(T; m_0) = P^{-1}(T; m_0)[X_d - \Phi^{m_0}(T)X(0)] \quad (7)$$

(b) の場合, $\text{rank } P(T; m_0) = n < m_0 r$ であり, (5) 式は一意的に解けないが, 3.2 に示すような最小ノルム解を利用すると入力列は(7)式と同様に初期値の1次式で与えられる.

3.2 最小ノルム解

(5)式において, $m \geq m_0$ なる任意の m に対して入力列が一意的に定まらない場合(8)式で与えられるノルムを最小にするという条件を加えると最小ノルム解は(9)式のようなになる⁹⁾.

$$V[U(T; m)] = \sum_{k=0}^{m-1} [U(k) - U_0]' Q [U(k) - U_0] \quad (8)$$

ただし, $U_0 = (\alpha + \beta)/2$, Q : 実対称正定 $r \times r$ 行列

$$\begin{aligned} U(T; m) &= \theta_m^{-1} P'(T; m) [P(T; m) \theta_m^{-1} P'(T; m)]^{-1} \\ &\quad \cdot [X_d - \Phi^m(T)X(0) - P(T; m)U_0] + U_0 \end{aligned} \quad (9)$$

ただし,

$$\begin{aligned} U_0 &= [U_0' \mid U_0' \mid \dots \mid U_0']' \quad (r \text{ 次元ベクトル}) \\ \theta_m &= Q + Q + \dots + Q \quad (m \text{ 個の } Q \text{ の直和}) \end{aligned}$$

4. 準最短時間離散値制御

多入出力系の計算機制御のアルゴリズムとして実用的であると思われる三つの準最短時間制御について述べる.

第1の方法はサンプリング数を m_0 に固定し, 周期 T を逐次変えながら(7)式または(9)式より適当な許容制御を得ようとするものであり⁵⁾, **Method 1** とする. この方法では前もっていくつかの周期 T_i に対し必要な計算を行っておけば計算時間は非常に短くなる. しかし, 任意の系について T を増すことにより任意の $X(0)$ を $m_0 T$ 後に X_d へ整定させるような許容制御が常に得られるとは限らないし, また T をあまりにも大きくすると最短時間の意味で良好な制御は得られない. このように本手法は X_d の近傍ではきわめて有効であるが, X_d から遠く離れた点に対してはあまり好ましくない.

第2の方法は周期 T を一定とし, サンプリング数 m を逐次増加させながらそのたびに線形計画問題を繰返し解く方法であり²⁾, **Method 2** とする. 得られた m サンプリング間の入力列は(6)式のように $U(T; m)$ と表わそう. この方法では T が小さいほど最短時間性はよくなるが, X_d から遠く離れた点に対しては m が増加し, 結局計算時間と容量が増大する. 他方, T が大きいとき, 一般に X_d の近傍の点に対しても m_0 サンプリング必要なので, 整定するまでの所要時間 $m_0 T$ が必要以上に長くなる.

第3の方法は最小ノルム解を利用するものであり, **Method 3** とする. これは **Method 2** と同様, T を固定し $X(0)$ に対し m を m_0 より逐次増加させながら(9)式より $U(T; m)$ を算出し, 許容制御かどうか判定する方法である⁶⁾. 得られた入力列は準最短時間制御ではあるが, 前もっていくつかの m に対し必要な計算を行っておけば計算時間は **Method 2** に比べ非常に短い. しかし, 本手法も T が小さいとき X_d から遠く離れた点に対しては m が増大するし, T が大きいと X_d の近傍では最短時間の意味で良好な制御が得られないのは **Method 2** と同様である.

5. 最小サンプリング数の最大値

Method 2 か **3** により Γ 内のすべての初期点を X_d へ整定させるための最小サンプリング数には最大値が存在し, これを m_x とする.

はじめに **Method 2** について述べる. 許容制御により周期 T , m サンプリングで X_d に整定するようなあらゆる初期点の集合を $R(T, m)$ と定義する.

$$R(T, m)$$

$$\begin{aligned} &= \left\{ X(0) : X(0) = \Phi^{-m}(T)X_d \right. \\ &\quad \left. - \sum_{i=1}^m \Phi^{-i}(T)H(T)U(i-1), U(i-1) \in Q \right\} \end{aligned}$$

$R(T, m)$ は X_d を含む閉凸集合であり, 任意の m に対し $R(T, m) \subset R(T, m+1)$ が成立する³⁾. 他方, Γ は 2^n 個の端点の凸閉包でありこれらの端点は容易に得られる. $R(T, m)$ と Γ のこれらの性質より $\Gamma \subset R(T, m_z)$ なる最小の m_z を得るには, 結局, Γ の 2^n 個の端点を X_d へ整定させるための最小サンプリング数を算出し, これらの中で最大のものを m_z とすればよい.

つぎに **Method 3** について述べる. 上の $R(T, m)$ に対し, つぎのような集合 $R'(T, m)$ を定義する.

$$R'(T, m)$$

$$= \{X(0) : (9) \text{ 式の } U(T; m) \text{ が許容制御列となる}\}$$

$R'(T, m)$ 内の任意の 2 点 $\tilde{X}(0), \tilde{X}(0)$ に対する最小ノルム解 $\tilde{U}(T; m), \tilde{U}(T; m)$ は許容制御列となる. ここで, $\tilde{X}(0), \tilde{X}(0)$ の線分 $[\tilde{X}(0) : \tilde{X}(0)]$ を考える.

$$[\tilde{X}(0) : \tilde{X}(0)]$$

$$= \{X(0) : X(0) = \lambda \tilde{X}(0) + (1-\lambda) \tilde{X}(0), 0 \leq \lambda \leq 1\}$$

線分上の任意の点 $X(0)$ に対する最小ノルム解 $U(T; m)$ を算出し, $\tilde{U}(T; m), \tilde{U}(T; m)$ との関係の求めるとつぎを得る.

$$U(T; m) = \lambda \tilde{U}(T; m) + (1-\lambda) \tilde{U}(T; m)$$

ここで, $\lambda \geq 0, (1-\lambda) \geq 0$ であり $\tilde{U}(T; m), \tilde{U}(T; m)$ は許容制御列なので $U(T; m)$ も許容制御列となり,

$$X(0) \in R'(T, m)$$

となる. $X(0)$ は任意なので

$$[\tilde{X}(0) : \tilde{X}(0)] \subset R'(T, m)$$

これより $R'(T, m)$ は凸集合である¹⁰⁾. しかしながら, 一般に $R'(T, m) \subset R'(T, m+1)$ は成立しない. そこで, $R'(T, m)$ と Γ の性質より $\Gamma \subset R'(T, m_z)$ なる最小の m_z を得るには Γ の 2^n 個の端点に対する最小ノルム解がすべての端点について同時に許容制御となるような最小のサンプリング数を算出すればよい.

6. アルゴリズム

6.1 アルゴリズムの開発

4. で述べたように **Method 1, 2, 3** はそれぞれ長所短所を有しており単独で用いると欠点を克服できない. 前の議論より一般に X_d の近傍においては **Method 1** がすぐれており X_d から遠く離れた点に対しては周期 T をあまり小さくとらず **Method 2** か **3** を用いるとよいことがいえる. なお, **Method 2, 3** に関しては計算時間を短くかつ容量を節約したい場合, 最適ではないが **Method 3** がすぐれている. このように三つの方法をそれぞれ独立して用いるより初期点の位置に応じ使い分けるほうが有利である. そこで T の効果を考慮したうえ, **Method 1** と **2** または **Method 1** と **3** を組合せるとそれぞれの欠点が除かれすぐれたアル

ゴリズムを得ることができる. **Method 1** と **2** を結合したものを **Algorithm 1** とし **Method 1** と **3** を結合したものを **Algorithm 2** としよう.

結合した方法において **Method 1** のために s 個の周期

$$T_1 < T_2 < \dots < T_s$$

を選び, **Method 2** または **3** のための周期を T_s とする. このようにすると Γ 内の点に対して **Algorithm 1, 2** により得られる入力列はすべて次式で与えられる.

$$U^1 \triangleq U(T_1; m_0),$$

$$U^2 \triangleq U(T_2; m_0), \dots, U^s \triangleq U(T_s; m_0),$$

$$U^{s+1} \triangleq U(T_s; m_0+1),$$

$$U^{s+2} \triangleq U(T_s; m_0+2), \dots, U^{m_z-m_0+s}$$

$$\triangleq U(T_s; m_z)$$

U^1 から U^s までは **Method 1** で U^{s+1} から $U^{m_z-m_0+s}$ までは **Method 2** か **3** で得られる. もしこれらが許容制御ならば整定するまでの所要時間は $U^1, \dots, U^{m_z-m_0+s}$ に対し順に

$$m_0 T_1 < m_0 T_2 < \dots < m_0 T_s < (m_0+1) T_s < \dots < m_z T_s$$

となるのである点 $X(0) \in \Gamma$ に対する最良な入力列は U^1 から $U^{m_z-m_0+s}$ の中にただ一つ存在する. しかしこれを直接求める方法はないので, 結局, Γ 内のあらゆる点に対し最短の計算時間で最良な U^i を見出す探索法を考えねばならない. いろいろの状況に応じて種々の探索法が考えられるが, 一般的には **Method 1, 2, 3** に対する二つの基本的な探索法の組合せである.

はじめに **Method 1** が有効である場合 (特に $m_0 = [n/r]$) について述べる. 今, 集合 $R(T_1, m_0), \dots, R(T_s, m_0)$ を考えると初期点 $X(0)$ はこれらのいずれかに含まれる. 第 1 の探索法はこれらの集合について次式が成立するととき有効であり **Search 1** とする.

$$R(T_1, m_0) \subset R(T_2, m_0) \subset \dots \subset R(T_s, m_0) \quad (10)$$

この場合, 最良の入力列は $X(0) \in R(T_i, m_0)$ を満たす最小の T_i に対する制御 U^i である. さらに $i < j \leq s$ を満たす任意の j に対する入力列 U^j はすべて許容制御列である. そこで $U^1, \dots, U^s$ の s 個の中から U^i を見出すには逐次中間の入力列の許容性を判定し, U^i の含まれる範囲を狭くしていけばよい. たとえば, 第 1 回目は $s_1 = 1 + [s/2]$ とし U^{s_1} の許容性を判定する. これが (2) 式を満たさないとすれば $s_1 < i \leq s$ である. そこで第 2 回目は $s_2 = s_1 + [(s-s_1+1)/2]$ とし U^{s_2} の許容性を判定する. これが (2) 式を満たすならば $s_1 < i \leq s_2$ である. そこで, 第 3 回目は $s_3 = s_1 + [(s_2-s_1+1)/2]$ とし U^{s_3} の許容性を判定する. これを繰返せば U^i を得ることができる. 最大の繰返し回数 w は次式

となる。

$$w = \left\lceil \frac{\log s}{\log 2} \right\rceil + 1 \quad (11)$$

第2の探索法は(10)式が成立しない場合についてであり **Search 2** とする。この場合最良の入力列は前と同様 $X(0) \in R(T_i, m_0)$ を満たす最小の T_i に対する U^i であるが(10)式が成立しないので $i < j \leq s$ なる任意の j に対する U^j はすべて許容制御とはならない。そこで正確に U^i を得るには **Search 1** は無効であり、結局、 U^1 から U^s にいたるまで逐次検討しなければならない。この探索法での最大の繰返し回数は s である。 $m_0 > [n/r]$ の場合も同様である。

Method 2 の場合には **Search 1** が有効である。実際、任意の m に対し $R(T, m) \subset R(T, m+1)$ となるのでこのことは明白である。

Method 3 の場合には一般に $R'(T, m) \subset R'(T, m+1)$ なので厳密には **Search 2** を利用すべきであるが、**Method 1** と同様 **Search 1** も有効な場合がある。

Algorithm 1, 2 の探索法は **Method 1, 2, 3** に対する方法として **Search 1, 2** のいずれを選んだかによってきまる。 Γ 内のいかなる点に対しても最良の入力列を得たい場合、一般に(10)式の成立の判別は困難なので **Algorithm 1** においては U^1 から U^s に対して **Search 2** を、 U^{s+1} から $U^{m_z - m_0 + s}$ に対して **Search 1** を用いるが、**Algorithm 2** においては U^1 から $U^{m_z - m_0 + s}$ に対して **Search 2** を用いばよい。他方、最適性は犠牲にしても計算時間を最も短くしたいならば **Algorithm 1, 2** の両方において $U^1, \dots, U^{m_z - m_0 + s}$ に対し **Search 1** を用いばよい。**Search 1** に対する繰返し回数は(11)式で与えられるので $m_z - m_0 + s$ が大きいとき、計算時間の点から **Search 1** はきわめて有効である。

6.2 計算時間遅れと各周期

計算時間遅れ ΔT はできるだけ短いほうがよいが使用するアルゴリズムが定まると最良の入力列を得るまでの最大の繰返し回数が決まるので結果的に決定される。ここで、 ΔT と計算機へ情報のはいる周期 T_0 と離散値系 S_d のサンプリング周期 T_i の関係について考える。プロセス情報はなるべく多いほうが好ましいが、計算機が処理できないほど多くの情報を得ても無意味であるので $T_0 = \Delta T$ となる。他方、制御入力 U は T_i ごとの階段波なので T_i を T_0 の整数倍に選べば入力信号は測定時点においてのみ変わりうる。

また、測定値 $X(0)$ へ対する最良な入力 U_p は前段で算出し

た入力を用いるものとすれば最良な入力 U_p が実際に加えられる時点での状態は外乱を無視してつぎのように推定される。

$$\hat{X}(\Delta T) = \Phi(\Delta T)X(0) + H(\Delta T)U_p \quad (12)$$

この $\hat{X}(\Delta T)$ を改めて $\hat{X}(0)$ とおく。 $\Phi(\Delta T), H(\Delta T)$ は前もって計算しておくことができるので $\hat{X}(0)$ は $X(0)$ より容易に得られる。 $\hat{X}(0)$ へ対するこのような入力列を $X(0)$ へ対するものと区別する意味で $\tilde{U}(T_i, m_0), \tilde{U}(T, m)$ などとおくことにする。

6.3 制御の更新

T_0 ごとの新しいプロセス情報をもとに外乱に対して早目に正しい制御を行うとともに制御の途中で外乱が存在しなくとも適当な方法で制御の更新を行い整理するまでの所要時間の短縮を図る。

現時刻を $t=0$ とする。測定値 $X(0)$ と ΔT 前の測定値をもとにした現時点の(12)式による推定値 X_e との差

$$\Delta X = X(0) - X_e \quad (13)$$

が0でなければこの ΔT 間に系は外乱の影響を受けたことになり新しく算出した入力列を用いねばならない。他方、 ΔX が0ならば外乱は存在しない。この場合、無条件に新しい入力列を用いるのではなく、所要時間が短くなるときのみ切換えねばならない。すなわち、 $t=0$ において得られた入力列を U' 、サンプリング数を m' 、周期を T' とし、これより t_f 後において得られた入力列を U'' 、サンプリング数を m'' 、周期を T'' とする。ここで、 U' を用いたとき残りの所要時間は $m'T' - t_f + \Delta T$ であるが、 U'' を用いたときは $m''T'' + \Delta T$ である。そこで、

$$m'T' - t_f > m''T''$$

ならばこの時点で U'' に切換えたほうが有利である。

6.4 Algorithm 1

例として **Method 1** に対し **Search 2, Method 2** に対し **Search 1** を採用した場合のアルゴリズムを記す。

(1°) 最小サンプリング数 m_0 を求める。 m_0 は周期に依存するが、多くの場合 n が r の整数倍なら $m_0 = [n/r]$ であり、整数倍でないなら $m_0 = [n/r] + 1$ である。

(2°) 適当な計算時間遅れ ΔT を与え、 $T_0 = \Delta T$ とする。

(3°) 適当な s 個の周期をつぎのように定める。

$$T_i = n_i T_0 \quad (i=1, 2, \dots, s)$$

ここで、 n_i は正の整数であり、 $n_1 < n_2 < \dots < n_s$ とする。

(4°) 周期 T_i に関して m_z を算出。

(5°) 最大の計算時間 $\Delta T'$ を求め ΔT と比較する。

$\Delta T < \Delta T'$ であったり、または ΔT が $\Delta T'$ より十分大きすぎる場合 ΔT を修正し (2°)~(5°) を繰返す。

(6°) 前もって各周期 T_i ($i=1, 2, \dots, s$), ΔT および各サンプリング数 m ($m_0 \leq m \leq m_s$) に対し必要な計算 [遅れについて $\Phi(\Delta T)$, $H(\Delta T)$, **Method 1** について $P^{-1}(T_i; m_0)$, $\Phi^{m_0}(T_i)$ または $\Theta_{m_0}^{-1}P'(T_i; m_0)$ [$P(T_i; m_0)\Theta_{m_0}^{-1}P'(T_i; m_0)$] $^{-1}$, $\Phi^{m_0}(T_i)$, $X_d - P(T_i; m_0)\hat{U}_0$, $\hat{U}_0$, **Method 2** について $\Phi^{m-i}(T_s)H(T_s)$ ($j=1, 2, \dots, m$), $\Phi^m(T_s)$, $\pm X_d \mp \sum_{i=1}^m \Phi^{m-i}(T_s)H(T_s)\alpha$, $\beta - \alpha$] を行っておく。

以上の準備のもとに実際に制御入力を求める手順はつぎのごとく与えられる。

(7°) 現在の観測値を $X(0)$ とし, $\hat{X}(0)$ を (12) 式より算出し $j=1$ とおく。

(8°) $\hat{X}(0)$ に対し (7) または (9) 式より $\tilde{U}(T_j, m_0)$ を算出。

(9°) $\tilde{U}(T_j, m_0)$ が許容制御ならば $T_j \rightarrow T$, $m_0 \rightarrow l$ において (15°) へ, そうでないならつぎへ。

(10°) $j < s$ ならば $j+1 \rightarrow j$ とおき (8°) へ, $j \geq s$ ならば (11°) へ。

(11°) $m_0 \rightarrow s_1$, $m_s \rightarrow s_2$ とおく。

(12°) $s_1 + [(s_2 - s_1 + 1)/2] \rightarrow m$ においてつぎの線形計画問題を解く。

(制約式)

$$qI_n \pm \sum_{i=1}^m \Phi^{m-i}(T_s)H(T_s)\hat{Y}(i-1)$$

$$\geq \pm X_d \mp \Phi^m(T_s)\hat{X}(0)$$

$$\mp \sum_{i=1}^m \Phi^{m-i}(T_s)H(T_s)\alpha \quad (\text{異符号同順})$$

$$\hat{Y}(i) \leq \beta - \alpha \quad (i=0, 1, \dots, m-1)$$

(非負条件)

$$q \geq 0; \quad \hat{Y}(i) \geq 0, \quad (i=0, 1, \dots, m-1)$$

(目的関数)

$$f = -q$$

ただし, q は新しく導入された変数, I_n はすべての要素を1とする n 次元ベクトル, 0_r は r 次元零ベクトル, $\hat{Y}(i)$ は制御入力 $\tilde{U}(i)$ とつぎの関係にある r 次元ベクトルである。

$$\hat{Y}(i) = \tilde{U}(i) - \alpha$$

(13°) $|f| \leq e_f$ ならば $m \rightarrow s_2$ とし, そうでないなら $m \rightarrow s_1$ とする。ただし, e_f は収束判定定数。

(14°) $s_2 = s_1 + 1$ ならば $T_s \rightarrow T$, $s_2 \rightarrow l$ において (15°) へ, そうでないときは (12°) へ。

(15°) (13) 式より ΔX を算出し $|\Delta X| \leq e_f I_n$ ならばつぎへ, そうでないなら (19°) へ。

Fig. 1 Results of digital computer simulations

(16°) $\Sigma_0 - kT_0 \rightarrow \Sigma$ とおき Σ が 0 ならば (19°) へ, そうでないならつぎへ。

(17°) $\Sigma > lT$ なら (19°) へ, そうでないならつぎへ。

(18°) $k+1 \rightarrow k$, $\hat{X}(0) \rightarrow X_e$ とおく。制御は更新しない。(7°) へ。

(19°) $1 \rightarrow k$, $lT \rightarrow \Sigma_0$, $\hat{X}(0) \rightarrow X_e$ とおく。制御を $\tilde{U}(T, l)$ へ更新する。(7°) へ。

(7°)~(19°) を T_0 ごとく新しい観測値に対し繰返す。制御を始める時点では $X_e = X_d$, $\Sigma = 0$ と設定する。

6.5 Algorithm 2

例として **Method 1, 3** に対し **Search 1** を採用した場合の手順を示す。(1°)~(6°) は **Algorithm 1** と同じ。ただし **Method 3** について必要な計算は $\Theta_m^{-1}P'(T_s; m)[P(T_s; m)\Theta_m^{-1}P'(T_s; m)]^{-1}$, $\Phi^m(T_s)$, $X_d - P(T_s; m)\hat{U}_0$, $\hat{U}_0$ である。

(7°) 現在の観測値を $X(0)$ とし, $\hat{X}(0)$ を (12) 式より算出。

(8°) $1 \rightarrow m$ とし (13°) へ。

(9°) $1 \rightarrow s_1$, $m_s - m_0 + s \rightarrow s_2$ とおく。

(10°) $s_1 + [(s_2 - s_1 + 1)/2] \rightarrow m$ とおく。

(11°) $m > s$ ならば (12°) へ, $m \leq s$ ならば (13°) へ。

(12°) (9) 式より $\tilde{U}(T_s, m-s+m_0)$ を算出し, これが許容制御なら $T_s \rightarrow T$, $m-s+m_0 \rightarrow l$ において (14°) へ, そうでないなら (15°) へ。

(13°) (7) または (9) 式より $\tilde{U}(T_m, m_0)$ を算出し, これが許容制御なら $T_m \rightarrow T$, $m_0 \rightarrow l$ において (14°) へ, そうでないなら (15°) へ。

(14°) $m=1$ なら (17°) へ, そうでないなら $m \rightarrow s_2$ とおき (16°) へ。

(15°) $m=1$ なら (9°) へ, そうでないなら $m \rightarrow s_1$ とおき, (16°) へ。

(16°) $s_2 = s_1 + 1$ ならば (17°) へ、そうでないなら (10°) へ。

(17°)~(21°) は **Algorithm 1** の (15°)~(19°) と同じ。

7. 例 題

6.4, 6.5 において述べた **Algorithm 1, 2** をつぎの例題に適用する。

(動特性)

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3 + v(t), \quad \dot{x}_3 = -x_2 - 0.6x_3 + u$$

ただし、 $v(t)$ は統計的性質は不明な外乱。

(拘束条件)

$$|u| \leq 1, \quad (\text{目標値}) \quad X_d = (0, 0, 0)'$$

(集合 Γ)

$$-17 \leq x_1 \leq 17, \quad -7 \leq x_2, \quad x_3 \leq 7.$$

ここで両手法において $m_0 = 3$, $\Delta T = 1.5$, $T_1 = 1.5$, $T_2 = 3.0$, $T_3 = 4.5$ とする。このとき **Algorithm 1** および **2** においてそれぞれ $m_z = 8$ および 10 となる。実際、NEAC-2200-575 で所要記憶容量と Γ 内の任意の点に対する最大の計算時間は **Algorithm 1** に関して約 4,100 語, 1.4 秒, **Algorithm 2** に関して 1,600 語, 0.015 秒であった。**Fig. 1(a)** は初期点を (4., 4., 5.) とし途中で外乱のない場合の結果である。**Algorithm 1** の場合 0.0~7.5 秒間は **M.2(Method 2)**, 7.5~16.5 は **M.1** であり, **Algorithm 2** の場合 0.0~12.0 は **M.3**, 12.0~19.5 は **M.1** であった。**Fig. 1(b)** は同じ初期点に対し 7~9 秒間に大きさ 5 の矩形状の外乱が加わった場合である。**Algorithm 1** に対し 0.0~7.5 は **M.2**, 7.5~9.0 は **M.1**, 9.0~18.0 は **M.2**, 18.0~25.5 は **M.1** であり, **Algorithm 2** に対し 0.0~19.5 は **M.3**, 19.5~27.0 は **M.1** であった。周期を 4.5 とするとき, (4., 4., 5.) の点を原点へ整定させるための最小サンプリング数は両手法において 5 であり, そのまま制御の更新をしなければ ΔT も考慮し, 24.0 秒必要である。しかし, 途中で更新を行うことにより **1** に対し 16.5 秒, **2** に対し 19.5 秒に減少している。また, 外乱に対してもじきに正しい処理が行われていることがわかる。**Algorithm 2** については ΔT を 1.5 にとる必要はなく, 0.05 以下で十分であるが比較のため同じ条件を選んだ。応答の速いプロセスに対しては **Algorithm 2** が有効であることがわかる。

8. あとがき

本文は最短時間計算機制御の実用的なアルゴリズム

に関するものである。このアルゴリズムにおいて, T_i を小さくすると理論上の最適性はよくなるが, m_z が増加し結局容量も ΔT も大きくなる。他方, T_i を大きくすると理論上の最適性は失われるが, m_z が小さくなり結局容量も ΔT も小さくなる。このように T_i や ΔT の決定にはプロセスの特性のみならず導入される制御用計算機の大きさと計算速度が考慮されねばならない。また, 状態がすべて直接測定できない場合や観測雑音などを含む場合, 観測器やカルマンフィルタを利用しなければならないが, このとき状態の推定に必要な計算時間は当然 ΔT の中に含まれねばならない。プラントの特性がわずかに変動する場合や観測された状態値に振動性の微小外乱の存在する場合 dual mode system にする必要がある。本文では T_i や ΔT の具体的決定法については触れなかったが, ここで示したアルゴリズムは種々の状況に応じ基本的構成を与えるものである。

最後に本研究に対し終始懇切なご指導を賜りました防衛大学 高尾吉郎教授, また日頃ご指導を賜ります同校岡田剛研究室ならびに中溝高好研究室の方々に衷心より感謝申し上げます。

参 考 文 献

- 1) 志水清孝: システム制御と数計画法, 238/273, コロナ社 (1971)
- 2) 高田 勝: 飽和のある線形プロセスの最短時間サンプル値制御, 制御工学, 9-12, 669/676 (1965)
- 3) J. Wing and C.A. Desoer: The Multiple-Input Minimal Time Regulator Problem (General Theory), IEEE Trans. on Automatic Control, AC-8, 125/136 (1963)
- 4) 長田 朗, 西村卓也: 線形離散システムの最短時間問題, 電子通信学会論文誌, 56-D-9, 507/514 (1973)
- 5) 馬場順昭: 不等式拘束条件を有する線形プロセスの計算機制御, 計測自動制御学会論文集, 10-5, 533/539 (1974)
- 6) J.A. Cadzow and H.R. Martens: Discrete-Time and Computer Control Systems, 297/298, Prentice-Hall, Inc. (1970)
- 7) 市川邦彦: 最適制御法則の直接計算, 制御理論シンポジウム資料, 1/8 (1971)
- 8) 須田信英, 児玉慎三: 制御工学者のためのマトリックス理論 (27), システムと制御, 17-11, 680/687 (1973)
- 9) Roger W. Brockett: Finite Dimensional Linear Systems, 126/127, John Wiley & Sons (1970)
- 10) William A. Porter: Modern Foundation of Systems Engineering, 273/275, The Macmillan Company (1966)