

Title	輸送網および生産工程のシステム制御に関する研究
Author(s)	田村, 坦之
Citation	大阪大学, 1971, 博士論文
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/985">https://hdl.handle.net/11094/985</a>
rights	
Note	

*Osaka University Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

Osaka University

輸送網および生産工程のシステム制御  
に関する研究

田村 坦之

昭和45年11月

## 内容梗概

本論文は、第I部「輸送網のシステム制御に関する研究」と、第II部「生産工程のシステム制御に関する研究」から成っている。

第I部では、陸上輸送網の解析、設計、制御の問題を取り扱う。第1章には、将来の都市内陸上輸送系を対象にして、系の重要な変数、パラメータ、制限条件等を明きらかにし、これらの間の基本的な関係を示し、システムの最適運用の問題を論じる。第2章には、鉄道輸送における列車プログラム自動運転のための制御系の最適設計法を与える。第3章には、鉄道輸送における複数の列車を対象にした列車群のスケジュール制御方式を示す。

第II部では、生産工程におけるプロセス制御、生産スケジューリングの問題を取り扱う。第4章には、特性方程式未知のプロセスの多変数最適化制御方式を示し、簡単なアナログ要素で on-line 制御を行う方式を示す。第5章には、機械加工工程を想定し、シミュレーションによって逐次生産工程のスケジュールを改良して行く方式を示す。第6章では、互に関連した複数のサブプロジェクトから成る大規模な生産工程のスケジューリングの問題を取り扱う。生産工程を大規模な二次計画法の問題としてモデル化し、Fenchel の双対原理を応用して、できるだけ少ない計算量で最適スケジュールを得るアルゴリズムを示す。

本論文の理論的背景になっているものとして、線形系最適制御理論、線形計画法、非線形計画法、分解原理、双対原理等があげられる。

# 目 次

	頁
緒 言	4
第I部 輸送網のシステム制御に関する研究	6
第1章 陸上輸送網の解析と最適運用	7
1.1 まえがき	7
1.2 システムのモデル	7
1.3 システムの最適運用	14
1.4 各変数, パラメータの間の trade off 関係	17
1.5 むすび	19
第2章 鉄道輸送制御系の最適設計	20
2.1 まえがき	20
2.2 問題の設定	20
2.3 最適設計	23
2.4 直接法と分解原理による結果の比較	26
2.5 プログラム自動運転制御系の構成	30
2.6 むすび	30
第3章 鉄道輸送制御系のスケジュール制御	31
3.1 まえがき	31
3.2 地上の制御用計算機の制御機能	32
3.3 中央制御のアルゴリズム	33
3.4 特徴	39
3.5 むすび	39

第Ⅱ部	生産工程のシステム制御に関する研究	40
第4章	線形計画法による数学モデル未知のプロセス制御	41
4.1	まえがき	41
4.2	制御対象および問題の設定	41
4.3	最適化制御系の構成	43
4.4	アナログ計算機によるシミュレーション	45
4.5	むすび	50
第5章	シミュレーションによる生産スケジューリング	51
5.1	まえがき	51
5.2	生産工程とそのシミュレーション	51
5.3	逐次スケジュールを改良する方式	57
5.4	シミュレーションの結果と検討	60
5.5	むすび	62
第6章	双対原理による大規模生産システムのスケジュー リング	63
6.1	まえがき	63
6.2	システムの数式モデル	63
6.3	Fenchelの双対原理	67
6.4	双対問題の定式化	68
6.5	双対問題を解くアルゴリズム	70
6.6	数値例と検討	72
6.7	むすび	74
結 言		77

謝 辭	79
參考文獻	80
圖表目次	83

## 結 言

近年、電子計算機のもつ情報処理能力の大規模化、高速化、多様化にともない、システム工学、制御工学およびその理論は急速な発展途上にある。ジャーナリストの言を借りれば、世の中は「システム時代」、「情報革新時代」に入りつつあると言われている現代において、システムと制御の分野が、今後益々その重要度を高めてくるものと思われる。

しかしながら、過去、約10年同程の間にあらわれてきたシステム理論、現代制御理論と称する新しい学問分野は、著しく数学的な色彩をおび、実務家にとっては、極めて難解なものとなってきた。システム理論、制御理論の研究において、高度の数学が不可欠の道具となってきたのは事実であるが、これらの理論と、実社会に山積みしているシステム制御の実際の問題との差が益々広がりつつあるように思われる。従って、理論と実際問題橋渡しの役割を果たす研究、すなわち、

- 1) 実社会の問題をモデル化し、このモデルに最も適した手法を見出すこと。
  - 2) 純粋理論を、実際問題に適用し得る形にかみ砕くこと。
  - 3) 実際問題を解く過程において、新しい理論、アルゴリズムを生み出すこと。
- 等の研究が必要である。

本論文は、このような立場で行なわれた研究をまとめたものである。いろいろな種類のシステムの内、輸送システムと生産システムを取りあげ、これらのシステムのモデル化、解析、設計、制御の問題を取り扱っている。本論文の理論的背景になっているものとして、線形系最適制御理論、線形計画法、非線形計画法、分解原理、双対原理等をあげることができ、これらの理論と、輸送システムおよび生産システムの間の橋渡しを試みている。

本論文は、第I部「輸送網のシステム制御に関する研究」(第1, 2, 3章)と、第II部「生産工程のシステム制御に関する研究」(第4, 5, 6章)で構成されている。第1章は関連発表論文(1)に、第2章および第3章は関連発表論文(2)に、第4章, 第5章, 第6章は各々関連発表論文(3), (4), (5)に基づいて書かれている。

## 関連発表論文

- (1) L. P. Hajdu, K. W. Gardiner, H. Tamura and G. L. Pressman, "Design and Control Considerations for Automated Ground Transportation Systems," Proceedings of the IEEE, Vol. 56, No. 4, pp.493-513, 1968.
- (2) H. Tamura, "Optimal Design of a Programmed Train Control System and Computer Control of Traffic Schedule Keeping," Preprints of IFAC Symposium on Systems Engineering Approach to Computer Control, pp.515-520, Kyoto, Aug.11-14, 1970.
- (3) 福永, 田村, "線形系における多変数最適化制御—数学モデル未知のリニア・プログラミング," 制御工学, 10巻, 1号, pp. 32-37, 昭41-1.
- (4) 福永, 田村, 羽根田, "シミュレーションによる生産スケジューリング制御," 制御工学, 10巻, 7号, pp. 352-358, 昭41-7.
- (5) 田村, "双対原理による大規模生産システムのスケジューリング," システムと制御(制御工学改題), 投稿中



# 第 I 部

## 輸送網のシステム制御に関する研究

# 第1章 陸上輸送網の解析と最適運用

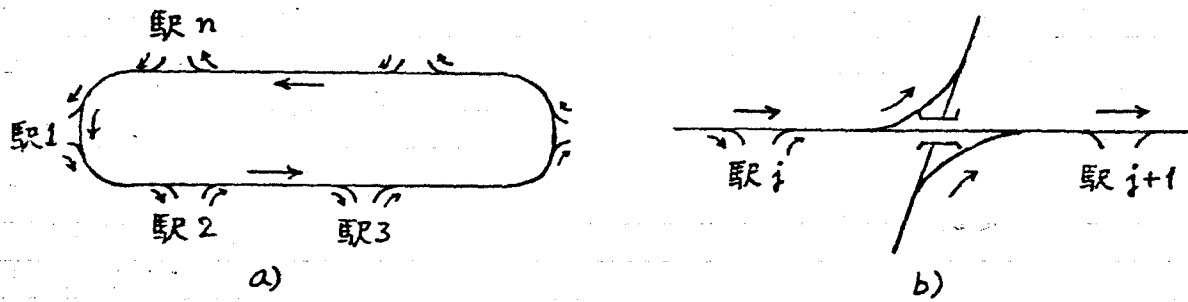
## 1.1 まえがき

本章では、都市交通の一つとして軌条式の自動化された輸送網におけるシステムの解析および最適運用の問題を取り扱う。ここで言う軌条式の輸送網とは、いわゆる鉄道形式のもののみならず、鉄道の長所（経済性、安全性、運転の必要のないこと等）と、自動車の長所（便利さ、融通性、乗り心地良さ等）を組合わせた自動式小型車方式<sup>5,37</sup>のものを意味する。ここで単一モード系とデュアルモード系を区別しておく。前者では、従来の鉄道と同じ形態を持ち、各車両は常に輸送網の軌条の上に拘束されるが、後者では、各車両とも輸送網の軌条から脱出することができ、一旦軌条を脱出すると自動車と同じ機能をもつものである。システム的には、単一モード系はデュアルモード系の特別な場合と見ることができると考える。

解析的な手法の適用を可能にするために、簡単な輸送網、すなわちループ状の案内路線をもつ輸送網を取り扱っているが、この解析は、さらに複雑な輸送網の解析を行うための出発点になるものと考えられる。

## 1.2 システムのモデル

第1.1図a)に示すような、本線の外に駅の側線をもつ閉じた案内路線の系を考える。この系に対する解析の方法を明らかにすれば、第1.1図b)に示すような交叉点を持つ系に対する解析が容易に行える。何故ならば、交叉点における車両の流入、流出は、駅における車両の流入、流出と同じ扱いをすることができると考えられる。



第1.1図 ループ状の案内路線と交差点を持つ案内路線

### 1.2.1 輸送網における乗客需要, 車両の流入・流出量, 平均移動距離

輸送網の解析において最も基本になる独立変数は乗客輸送需要  $d_{jk}$ , すなわち単位時間に駅  $j$  から駅  $k$  へ移動する乗客数である。ピーク時に、一つの車両で運ばれる乗客の平均人数を  $p$  とすると、単位時間に駅  $j$  から駅  $k$  へ移動する車両の数  $u_{jk}$  は

$$u_{jk} = d_{jk}/p. \quad (1.1)$$

駅  $k$  において、単位時間に本線へ流入する車両の数  $i_k$ , 本線から流出する車両の数  $e_k$  は

$$i_k = \sum_{\substack{l=1 \\ (l \neq k)}}^n u_{kl} \leq i_k^M \quad (1.2)$$

$$e_k = \sum_{\substack{j=1 \\ (j \neq k)}}^n u_{jk} \leq e_k^M$$

であらわせる。ただし、 $i_k^M, e_k^M$  は駅  $k$  における最大可能な車両の流入・流出量, すなわち駅の能力を示す。

発駅  $j$  と着駅  $k$  の向の距離を  $s_{jk}$  とすると、駅  $j$  を出発する車両の平均移動距離  $r_j$  は

$$r_j = \frac{\sum_{\substack{k=1 \\ (k \neq j)}}^n s_{jk} \cdot u_{jk}}{\sum_{\substack{k=1 \\ (k \neq j)}}^n u_{jk}} = \frac{\sum_{\substack{k=1 \\ (k \neq j)}}^n s_{jk} \cdot u_{jk}}{i_j}. \quad (1.3)$$

特に、各駅がほぼ等間隔に配置している時には、 $r_j$  は駅数(無次元量)であらわすことができ、

$$r_j = \frac{u_{j,j+1} + 2 u_{j,j+2} + \dots + (n-1) u_{j,j-1}}{i_j} \quad (1.3)'$$

で与えられる。  $i_j$  が一定の場合、平均移動距離  $r_j$  が長くなればなる程、本線上にいる車両の数は大になる。すなわち本線上の負荷が大になる。

### 1.2.2 輸送網における車両の流量

輸送網におけるシステム運用の状況をあらわす最も重要な量は、本線上を流れる車両の流量、すなわち本線上の或る点を単位時間に通過する車両の数である。今、駅  $k$  と  $k+1$  の間の車両の流量を  $F_k$  とすると

$$\max_k \{ F_k \} \leq C \quad (1.4)$$

を満たさねばならない。たゞ  $C$  は線路容量、すなわち本線上の或る点を、単位時間に通過し得る最大車両数である。また、 $e_k, i_k, F_k, F_{k-1}$  の間には

$$\begin{aligned} F_k &= F_{k-1} - e_k + i_k, \quad k=2, 3, \dots, n \\ F_1 &= F_n - e_1 + i_1 \end{aligned} \quad (1.5)$$

の関係がある。これより

$$\begin{aligned} F_1 &= F_n - e_1 + i_1 \\ F_2 &= F_n - e_1 - e_2 + i_1 + i_2 \\ &\vdots \\ F_{n-1} &= F_n - \sum_{k=1}^{n-1} e_k + \sum_{k=1}^{n-1} i_k \end{aligned} \quad (1.6)$$

を得る。然るに  $n$  個のすべての流量の和は、本線への流入量  $i_k$  と (1.3)' による平均移動距離  $r_k$  を用いて

$$\sum_{k=1}^n F_k = \sum_{k=1}^n r_k i_k \quad (1.7)$$

であらわせる。また、定常状態では

$$\sum_{k=1}^n e_k = \sum_{k=1}^n i_k \quad (1.8)$$

なので、(1.6) と (1.7) より

$$F_n = \frac{1}{n} \left\{ \sum_{k=1}^n r_k i_k + n(e_1 - i_1) + (n-1)(e_2 - i_2) + \dots \right. \\ \left. \dots + (e_n - i_n) \right\} \quad (1.9)$$

を得る。(1.9)と(1.6)式より各線区における車両の流量は、各駅における本線への流入量、本線からの流出量、平均移動距離によってあらわされる。車両の流量  $F_k$  を、各駅における  $i_k, e_k, r_k$  によって陽にあらわすことの物理的意義は次の通りである。 $r_k$  は経験的に評価でき、 $i_k, e_k$  は容易に測定できる。この測定値にもとずいて、各リンクにおける負荷を計算し、流入量を制御することができる。

以上は、ループ状の輸送網以外の場合にも容易に拡張できる。互に交叉するいくつかのループがある場合には、各ループに対して流れの式を立てる。交叉点における流入、流出は、各駅における流入、流出と同等に取り扱える。平均移動距離は、各ループ毎に評価しておく。このようにして、複雑な輸送網の問題も、簡単なループ状の輸送網の問題に分解して解くことができる。

### 1.2.3 主要な拘束条件

$i_k, e_k, F_k$  に関する主要な拘束条件は

- a) 本線への流入に関する制限
- b) 本線からの流出に関する制限
- c) 路線容量に関する制限

である。

a) b) は駅の構造、乗車、降車の方法、側線から本線への流入に関する設計方法等からきまり(1.2)式で示される。c) に関しては、先行車の後端と後続車の前端の距離を  $h$  とし、この最小許容値を  $h_0$  とすると、路線容量  $C$  は

$$C = \frac{v}{h_0 + L} \quad (1.10)$$

であらわされる。ただし  $L$  は車両の長さをあらわす。

速度  $v$  で走行している車両を停止させるのに必要な距離  $d_0$  は、非常時に使用する減速度を  $a_0$  とすると

$$d_0 = \frac{v^2}{2a_0} + \tau v \quad (1.11)$$

である。ただし、 $\tau$  はブレーキをかけるまでの時間遅れであり、 $\tau v$  のことを空走距離という。 $h_0$  は空走距離を無視すると

$$h_0 = k d_0 \approx k \frac{v^2}{2a_0} \quad (1.12)$$

で定められる。

系を  $k > 1$  となる様に設計すると、 $h_0$  は停止距離  $d_0$  よりも大きい。従って、先行の車両が突然停止（例えば脱線等）しても、後続車両は安全に停止できる。 $k \approx 1$  のときには、 $h_0$  は停止距離  $d_0$  にほぼ等しい。 $k \approx 1$  となる様に系を設計するためには、従来の鉄道に採用されているブロック方式の信号系統（固定閉塞方式）はもはや使えなくなり、後続車は、先行車の走行位置を連続的に検知しなければならない。 $k < 1$  のときには、 $h_0$  は停止距離  $d_0$  よりも小さいので、先行車が突然停止したときには衝突が起り得る。一般に、 $k$  の値を小さく設計すればする程、後続車の安全性を確保するのに必要な、先行車に関する情報（位置、速度等）量は多くなる。鉄道では  $k = 2.0 \sim 3.0$ 、高速道路における自動車は  $k = 0.3$  近傍で走行している。

(1.12) 式を (1.10) 式へ代入すると

$$C = \frac{2a_0 v}{k v^2 + 2a_0 L} \quad (1.10)'$$

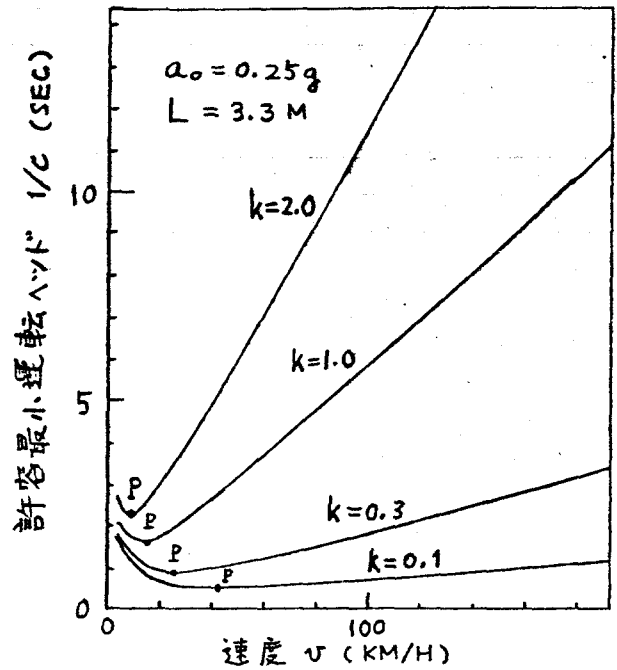
となる。路線容量と車両の速度との関係を第 1.2 図に示す。

### 1.2.4 システムの評価規準

輸送システムの性能を示す規準で最も重要なものの一つは、乗客が発駅  $j$  から輸送網を通して目的駅  $k$  に着くまでの平均所要時分  $T_{jk}$  であり

$$T_{jk} = (T_r)_{jk} + (T_w)_{jk} \quad (1.13)$$

であらわされる。ここで  $(T_r)_{jk}$  は車で移動する時間、 $(T_w)_{jk}$  は車に乗るまでに待たされる時間を示す。



第1.2図 路線容量と速度の関係

デュアルモードのループ状の系で、 $n$ 個の駅が等間隔  $\Delta$  ごとにある場合には、

$$T_{jk} = \begin{cases} [t_a + (k-j)\frac{\Delta}{v} - \frac{b}{v}] + \frac{1}{u_{jk}}, & k > j \\ [t_a + (n+k-j)\frac{\Delta}{v} - \frac{b}{v}] + \frac{1}{u_{jk}}, & k < j \end{cases} \quad (1.14)$$

ただし、 $t_a$  は加速、減速、駅停止に使われる時間、 $b$  は駅のバイパスの長さである。(1.14)式右辺の括弧内は走行時分を、 $1/u_{jk}$  は駅における待ち時間を示す。

すべての発駅/目的駅の組合せに対して輸送所要時分の荷重平均値

$$T = \frac{\sum_{i,k} T_{jk} u_{jk}}{\sum_{i,k} u_{jk}} \quad (1.15)$$

を考える。(1.14)を(1.15)へ代入し、(1.2)および(1.3)'の関係を用いて整理すると

$$T = \left[ t_a + \frac{\Delta}{v} \frac{\sum_{k=1}^n r_k i_k}{\sum_{k=1}^n i_k} - \frac{b}{v} \right] + \frac{v}{\sum_{k=1}^n i_k} \quad (1.16)$$

$$v = \sum_{j,k} \delta_{jk}, \quad \delta_{jk} = \begin{cases} 1, & u_{jk} > 0 \\ 0, & u_{jk} = 0 \end{cases}$$

を得る。ここで、定数  $t_a$  および  $b$  はシステムの設計段階でおよそ

$$\begin{aligned} t_a &\cong 2 \frac{v}{a} + t_d \\ b &\cong \frac{v^2}{a} \end{aligned} \quad (1.17)$$

に選ばれる。すなわち、 $t_a$  は加速、減速に費される時間と駅停止時分  $t_d$  の和であらわされ、 $b$  は加速、減速に使われる距離であらわされる。また、(1.8) 式であらわされる量を

$$S = \sum_{k=1}^n e_k = \sum_{k=1}^n i_k \quad (1.18)$$

と

$$S^* = p \cdot S \quad (1.19)$$

とすると、 $S^*$  は単位時間あたり各駅から運び出される乗客数の和で、定常状態では、各駅へ運びこまれる乗客数の和に等しい。また、

$$F_k^* = p \cdot F_k \quad (1.20)$$

とすると  $F_k^*$  は駅  $k$  と駅  $k+1$  の間の単位時間あたりの乗客の流量をあらわす。(1.17)(1.19)(1.20) 式を (1.16) 式へ代入すると、輸送所要時分の荷重平均値  $T$  は  $v$  と  $p$  の関数として

$$T(v, p) = \frac{v}{a} + \frac{v}{v} \frac{1}{S^*} \sum_{k=1}^n F_k^* + \frac{v}{S^*} p + t_d \quad (1.21)$$

であらわされる。

デュアルモードのループ状の系に対して (1.21) 式を求めたが、これは単一モードの系、ダイナミックな駅構造を持つ系、各駅にバイパスのない系等、種々の系に対しても同様に導くことができ、<sup>16</sup> いずれの場合にも

$$T(v, p) = c_1 v + c_2 \frac{1}{v} + c_3 p + c_4 \quad (1.22)$$

の形になる。ただし  $c_1 \sim c_4$  はシステムパラメータ、乗客需要等からきまる定数である。



### 1.3 システムの最適運用

前節で議論したシステムのモデルを用いて、システム運用の最適化の問題は、

$$F^M = \text{Max}_k \{ F_k \} \leq C = \frac{2a_0 v}{k v^2 + 2a_0 L} \quad (1.4)$$

$$\left. \begin{array}{l} i_k \leq i_k^M \\ e_k \leq e_k^M \end{array} \right\} \quad (k=1, 2, \dots, n) \quad (1.2)$$

さらに車の許容最高速度を  $v^M$  とすると

$$v \leq v^M \quad (1.23)$$

の制限条件のもとで

$$T(v, p) = c_1 v + c_2 \frac{1}{v} + c_3 p + c_4 \quad (1.24)$$

を  $v$  と  $p$  に関して最小にする問題となる。

車の流れと乗客の流れは  $p$  を介して

$$\left. \begin{array}{l} F^{*M} = p F^M \\ i_k^* = p i_k \\ e_k^* = p e_k \end{array} \right\} \quad (k=1, 2, \dots, n)$$

の関係にあるので、(1.4)(1.2)(1.23)式の制限条件は

$$g_1(v, p) = v^2 - c_5 p v + c_6 \leq 0 \quad (1.25)$$

$$g_2(v) = v - v^M \leq 0 \quad (1.26)$$

$$g_3(p) = -p + p^m \leq 0 \quad (1.27)$$

ただし

$$\left. \begin{array}{l} c_5 = \frac{2a_0}{k F^{*M}} \\ c_6 = \frac{2a_0 L}{k} \\ p^m = \text{Max}_k \left\{ \frac{i_k^*}{i_k^M}, \frac{e_k^*}{e_k^M} \right\} \end{array} \right\} \quad (1.28)$$

であらわされる。

(1.25)~(1.27) 式の  $v$  と  $p$  で (1.24) 式を最小にする問題は、二変数の非線形計画法の問題である。Lagrange 乗数  $\lambda_1, \lambda_2, \lambda_3$  を導入すると Lagrange 関数  $\mathcal{L}$  は

$$\mathcal{L} = T + \lambda_1 g_1 + \lambda_2 g_2 + \lambda_3 g_3 \quad (1.29)$$

となる。  $v$  および  $p$  は 0 にはなり得ないので、Kuhn-Tucker の定理<sup>22</sup>より

$$\mathcal{L}_v = (c_1 - c_2 \frac{1}{v^2}) + \lambda_1 (2v - c_5 p) + \lambda_2 = 0$$

$$\mathcal{L}_p = c_3 - \lambda_1 c_5 v - \lambda_3 = 0$$

$$g_1 \leq 0, \quad g_2 \leq 0, \quad g_3 \leq 0$$

$$\lambda_1 \geq 0, \quad \lambda_2 \geq 0, \quad \lambda_3 \geq 0 \quad (1.30)$$

$$\lambda_1 g_1 = \lambda_1 (v^2 - c_5 p v - c_6) = 0$$

$$\lambda_2 g_2 = \lambda_2 (v - v^M) = 0$$

$$\lambda_3 g_3 = \lambda_3 (-p + p^M) = 0$$

$$v \geq 0, \quad p \geq 0$$

ここで、 $c_3 > 0$  であるので、 $\lambda_1$  と  $\lambda_3$  が同時に 0 になることはない。 $\lambda_1, \lambda_2, \lambda_3$  の符号に応じて、システムの最適運用条件は、第 1.1 表に示すごとく 6 つの場合に分類することができ、すなわち、第 1.1 表に示す  $v^0, p^0, \lambda_1, \lambda_2, \lambda_3$  が (1.30) 式の解であり、(1.25)~(1.27) 式の制限条件のもとで、(1.24) 式の評価関数を最小にする解を与える。

第 1.1 表に示す Case 1 ~ Case 6 のうち、システム設計の段階で Case 2 または Case 5 の条件をみたす様に各パラメータを設定しておき、システム運用の段階で  $v$  および  $p$  を、各々  $v^0$  および  $p^0$  に設定することによって、システムの最適運用が行なえる。Case 6 は  $v^M$  および  $p^M$  がたまたま  $g_2 = 0$  をみたす場合を示し、減速に起り得ない case である。Case 1, 3, および 4 は、システム設計の段階で設定したパラメータの値が必ずしもを示す。なぜならば、これらの場合には最高許容速度  $v^M$  で走行するとは

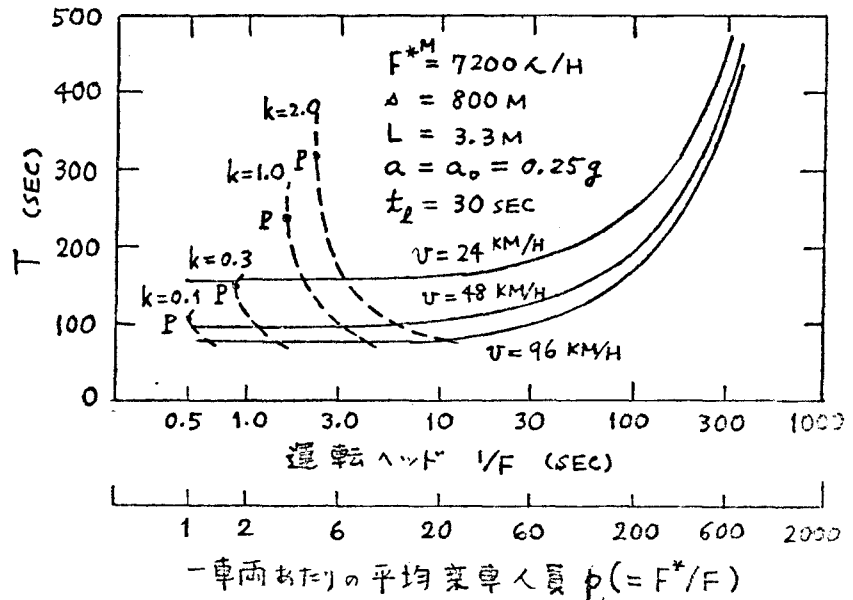
第1.1表 システムの最適運用条件

	$\lambda_1$	$\lambda_2$	$\lambda_3$	$g_1$	$g_2$	$g_3$	$v^0$	$p^0$	備考
Case 1	$\frac{c_3}{c_5 v^0} (>0)$	0	0	0	$\leq 0$	$\leq 0$	$\sqrt{\frac{c_2 + (c_3 c_6 / c_5)}{c_1 + (c_3 / c_5)}}$	$\frac{v^0 + c_6}{c_5 v^0}$	$v^0, p^0$ が $g_2 \leq 0, g_3 \leq 0$ をみたす場合, 路線容量に関する制限のみがきく。
Case 2	$\frac{c_3}{c_5 v^0} (>0)$	$-c_1 + c_2 \frac{1}{v^0}$ $-\lambda_1(2v^0 - c_5 p^0)$	0	0	0	$\leq 0$	$v^M$	$\frac{v^0 + c_6}{c_5 v^0}$	$p^0$ が $g_3 \leq 0$ をみたし, $\lambda_2 > 0$ がみたまはたされている場合, 路線容量と速度制限がきく。
Case 3	0	0	$c_3 (>0)$	$\leq 0$	$\leq 0$	0	$\sqrt{c_2 / c_1}$	$p^m$	$v^0, p^0$ が $g_1 \leq 0, g_2 \leq 0$ をみたす場合, 駅設備に関する制限のみがきく。
Case 4	$-\frac{c_1 - (c_2 / v^0)}{2v^0 - c_5 p^0}$	0	$c_3 - \lambda_1 c_5 v^0$	0	$\leq 0$	0	$\frac{c_5 p^0 + \sqrt{c_5^2 p^0 - 4c_6}}{2}$	$p^m$	$v^0$ が $g_2 \leq 0$ をみたし, $\lambda_1 > 0, \lambda_3 > 0$ がみたまはたされている場合, 路線容量と駅設備に関する制限がきく。
Case 5	0	$-c_1 + c_2 \frac{1}{v^0}$	$c_3 (>0)$	$\leq 0$	0	0	$v^M$	$p^m$	$v^0, p^0$ が $g_1 \leq 0$ をみたし, $\lambda_2 > 0$ がみたまはたされている場合, 速度と駅設備に関する制限がきく。
Case 6	任意の正の定数	$-c_1 + c_2 \frac{1}{v^0}$ $-\lambda_1(2v^0 - c_5 p^0)$	$c_3 - \lambda_1 c_5 v^0$	0	0	0	$v^M$	$p^m$	$v^0, p^0$ が $g_1 = 0$ をみたし, $\lambda_2 > 0, \lambda_3 > 0$ がみたまはたされている場合, 路線容量, 速度, 駅設備の制限がすべてきく。

は、システムの最適運用を意味しなくて、他の case にくらべて明らかに輸送所要時分を長びかせているからである。

#### 1.4 各変数, パラメータの間の trade-off 関係

第1.3図に、 $v$  および  $p$  の、 $T$  へおよぼす影響を示す。この図では、 $v$ 、 $p$ 、 $k$  の間の trade-off 関係を明らかにするために、出発駅と到着駅の2駅だけからなっている輸送システムを取りあつかっている。



第1.3図 輸送所要時分の荷重平均値  $T(v, p)$  (2駅問題)

単位時間あたりの乗客の流れ  $F^*$  および車の流れ  $F$  の間には、一車両あたりの平均乗車人員  $p$  を介して (1.20) に示すごとく

$$F^* = pF$$

の関係にあるので、 $F^*$  が与えられれば、 $p$  を減らすことと、車両の運転ヘッド  $1/F$  を減らすこととは等価である。第1.3図より明らかなように、一定の車両速度のもとでは、 $1/F$  を減らせば減らす程、輸送所要時分の荷重平均値  $T(v, p)$  は減少する。しかし、 $v$  一定の曲線にいたがって  $1/F$  を減らして行くと  $k$  一定の曲線と交叉する。 $k$  の値は、システム設計の段階で、或る一定値に設定されているため、さらに  $1/F$  を減らそうとすることもはや  $v$  一定の曲線にいたがうことはできず、 $k$  一定の曲線にいたがって  $1/F$  を減らすなければならない。第1.3図では、 $v$  一定の曲線と  $k$  一定の

曲線の交点が最小輸送所要時分を与え、前節で分類した Case 2 の場合に  
あっている。k一定の曲線は、路線容量の制限を示している。k一定の  
曲線にいたがってvを減少させながら1/Fを減らして行くと第1.3図のP  
点に到達する。1/Fをこの値以下に減らすことはできない。第1.3図のP  
点は、第1.2図のP点と対応しこの点が運転ヘッド最小、言いかえれば路  
線容量の範囲内で車両の流量すなわち輸送量を最大にする点である。

複雑な輸送網に対する第1.3図と同様の線図を求めることは容易である。  
例として、始発駅と終着駅の間に中間駅が3つある、5駅からなるシステ  
ムを考へる。単位時間あたり輸送  
なければならぬ乗客の数は第1.2  
表のように与えられているものとす

第1.2表 単位時間あたりの乗客  
輸送量 (人/H)

発着駅	1	2	3	4	5
1	0	900	900	900	3100
2		0	400	400	1600
3			0	400	1600
4				0	1600
5					0

る。また、システム設計の段階で

$$v^M = 96 \text{ km/h}$$

$$L = 3.3 \text{ m}$$

$$a = a_0 = 0.25 \text{ g}$$

$$t_d = 20 \text{ sec}$$

$$k = 0.3$$

が与えられているものと  
する。

第1.2表から

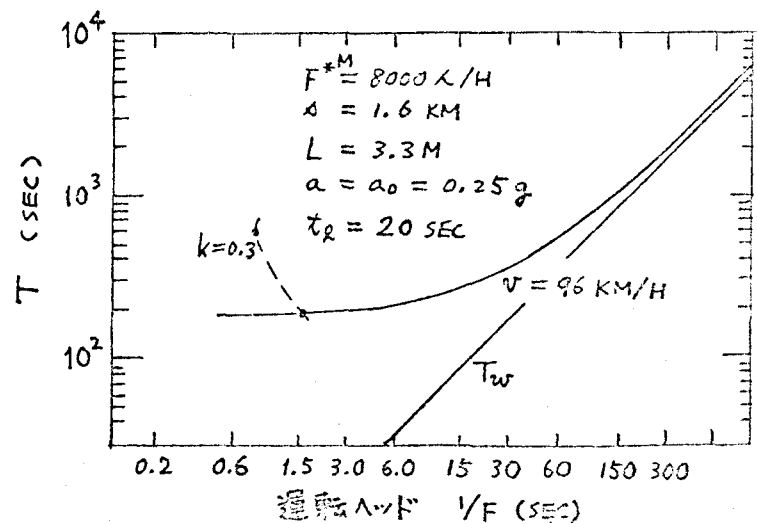
$$F^{*M} = F_3^* = 8000$$

を得る。輸送所要時分の  
荷重平均値を第1.4図に  
示す。これより、Tを最

小にするvおよびpは

$$v^0 = 96 \text{ km/h}$$

$$p^0 = 3.9 \text{ 人}$$



0.55 1.7 3.3 6.6 13.3 33.3 66.6 133 333 663

一車両あたりの平均乗車人員  $p (= F^*/F)$

第1.4図 輸送所要時分の荷重平均値  
T(v, p) (5駅問題)

となる。また、この時の車両の運転ヘッドは

$$\frac{1}{F^0} = p^0 \cdot \frac{1}{F^*M} = 3.9 \times \frac{3600}{8000} = 1.8 \text{ sec}$$

となる。ここでは、 $p \geq p^m$  の制限条件を考慮していないが、もし  $p^0 = 3.9$  人が、 $p \geq p^m$  をみたしていないければ  $p^0$  は  $p^m$  でおきえられる。この場合には、システムを路線容量以下の状態で運用することになる。

### 1.5 むすび

将来の都市内陸上輸送系を対象にして、系の重要な変数、パラメータ、制限条件等を明きらかにし、これらの間の基本的な関係を示した。また、システム運用の評価規準として、輸送所要時分の荷重平均値をとりあげ、その最適化問題を議論し、さらに重要な変数が評価規準に及ぼす trade-off 関係を明きらかにした。

## 第2章 鉄道輸送制御系の最適設計

### 2.1 まえがき

本章では、問題の対象を鉄道輸送に限定し、各列車のプログラム自動運転<sup>32,34</sup>を行うための制御系の最適設計法を検討する。

列車運転の自動化の方式には、大別して

#### a) 車上制御方式

個々の列車の走行制御は、車上の制御機器が担当し、中央からは列車群の総合的な制御を行ない、車上制御機能と地上制御機能を合理的に配分して総合システムの自動化をはかる方式。

#### b) 地上制御方式

個々の列車の走行制御、列車群の制御等をすべて地上の制御用計算機を通じて行う方式。

があり、その利害得失が検討されている。<sup>36</sup>

本章では、車上制御方式における車上の制御器の設計方法を検討し、第3章では、車上制御方式における列車群の総合的な制御方式を検討する。

### 2.2 問題の設定

車上プログラム式列車自動運転では、車上プログラムとして、

a) 速度 ( $v$ ) - 距離 ( $s$ ) プログラム

b) 時間 ( $t$ ) - 距離 ( $s$ ) プログラム

を持ち、両プログラムとも距離ベースのプログラムである。時間偏差 ( $\Delta t$ ) の測定および速度変化指令 ( $\Delta v$ ) を、一定のサンプリング距離 ( $\Delta s$ ) ごとに行う。

地点  $s$  を通過する列車の速度  $v(s)$  と、時間  $t(s)$  の間には、

$$t(s) = t(s_0) + \int_{s_0}^s \frac{3600}{v(\xi)} d\xi \quad (2.1)$$

$$v(s) = \tilde{v}(s) + \Delta v(s) \quad (2.2)$$

$$t(s) = \tilde{t}(s) - \Delta t(s) \quad (2.3)$$

の関係がある。ここで  $\tilde{v}(s)$  および  $\tilde{t}(s)$  は、各々標準の速度-距離プログラムおよび標準の時間-距離プログラムを示し、 $\Delta t(s)$  は、実時刻の  $\tilde{t}(s)$  からの偏差、 $\Delta v(s)$  は  $\tilde{v}(s)$  を基準にした速度指令変化値を示す。(2.2) および (2.3) 式を (2.1) 式に代入して整理すると、

$$\Delta t(s) = \Delta t(s_0) + \int_{s_0}^s \frac{3600}{\tilde{v}^2(\xi)} \Delta v(\xi) d\xi \quad (2.4)$$

を得る。ここで  $\Delta v(s)$  は、 $\tilde{v}(s)$  に対して十分小さい、1 に対して  $(\Delta v(s)/\tilde{v}(s))^2$  以上の項を無視した。

時間偏差  $\Delta t(s)$  を  $s$  に関して連続的に測定し、速度指令変化  $\Delta v(s)$  を  $s$  に関して連続的に与える場合には、両者の間には (2.4) 式またはこれと等価な

$$\frac{d}{ds} \Delta t(s) = \frac{3600}{\tilde{v}^2(s)} \Delta v(s) \quad (2.5)$$

の関係があるが、制御は一定のサンプリング距離  $\Delta s$  ごとに行うものとし



ているので、(2.4) または (2.5) 式を、各サンプリング地点で満足する関係に変形する必要がある。すなわち、 $\Delta v(s)$  は  $k\Delta s \leq s < (k+1)\Delta s$ ,  $k=0, 1, 2, \dots$  において一定にたもたれる。従って (2.4) 式より、

$$\Delta t(k\Delta s + \Delta s) = \Delta t(k\Delta s) + \int_{k\Delta s}^{(k+1)\Delta s} \frac{3600}{v^2(\xi)} d\xi \cdot \Delta v(k\Delta s)$$

を得る。 $\Delta t(k) = \Delta t(k\Delta s)$ ,  $\Delta v(k) = \Delta v(k\Delta s)$  と書くと、線形の差分方程式

$$\Delta t(k+1) = \Delta t(k) + \gamma(k) \cdot \Delta v(k) \quad (2.6)$$

$$\gamma(k) = \int_{k\Delta s}^{(k+1)\Delta s} \frac{3600}{v^2(\xi)} d\xi$$

$$k=0, 1, 2, \dots, K$$

を得る。各サンプリング地点  $k$  で与える速度指令変化値  $\Delta v(k)$  に対する制限として

$$\Delta v_{\min}(k) \leq \Delta v(k) \leq \Delta v_{\max}(k) \quad (2.7)$$

を考慮しなければならぬ。ただし、 $\Delta v_{\max}(k)$  および  $\Delta v_{\min}(k)$  は、各サンプリング地点にて許容し得る  $\Delta v(k)$  の最大値および最小値を示す。

制御系の最適設計を行うための評価関数を、

$$J = \sum_{k=1}^K \{ q_1(k) \Delta t^2(k) + q_2(k-1) \Delta v^2(k-1) \} \quad (2.8)$$

とする。これは、列車走行の始点から終点にいたる間の時間偏差と、速度指令変化の荷重自乗和を示す。時間偏差および速度指令変化にかゝる重み  $q_1(k)$  および  $q_2(k)$  の値は、この制御系を使用する使用者の意志に従って決定される。例えば、速度指令変化（この量はエネルギー消費と密接な関係を持つ）よりも時間偏差を小さくすることをより重要な場合には、 $q_2(k)$  よりも  $q_1(k)$  を相対的に大きく選ぶ。また駅中間よりも駅停止附近の時間偏差を小さくしたい場合には、 $q_1(k)$  の駅中間の値よりも、駅停止附近の値の方を相対的に大きく選ぶ。

以上は (2.6) および (2.7) 式で与えられる変係数離散値型線形システムにおいて (2.8) 式を最小にする列車自動運転の大域的な制御系を設計するための数式モデルである。

## 2.3 最適設計

### 2.3.1 直接法による設計

(2.7) 式を無視すると、(2.6) 式のもとで (2.8) 式を最小にする最適なフィードバック制御  $\Delta \hat{u}(k)$  は、

$$\Delta \hat{u}(k) = -w(k) \Delta z(k) \quad (2.9)$$

で与えられる。ただしゲイン  $w(k)$  は

$$w(k) = \gamma(k) p(k+1) / \{ q_2(k) + p(k+1) \gamma^2(k) \} \quad (2.10)$$

$p(k)$  は次の Riccati の差分方程式

$$p(k) = q_1(k) + p(k+1) - \frac{\gamma^2(k) p^2(k+1)}{q_2(k) + p(k+1) \gamma^2(k)} \quad (2.11)$$

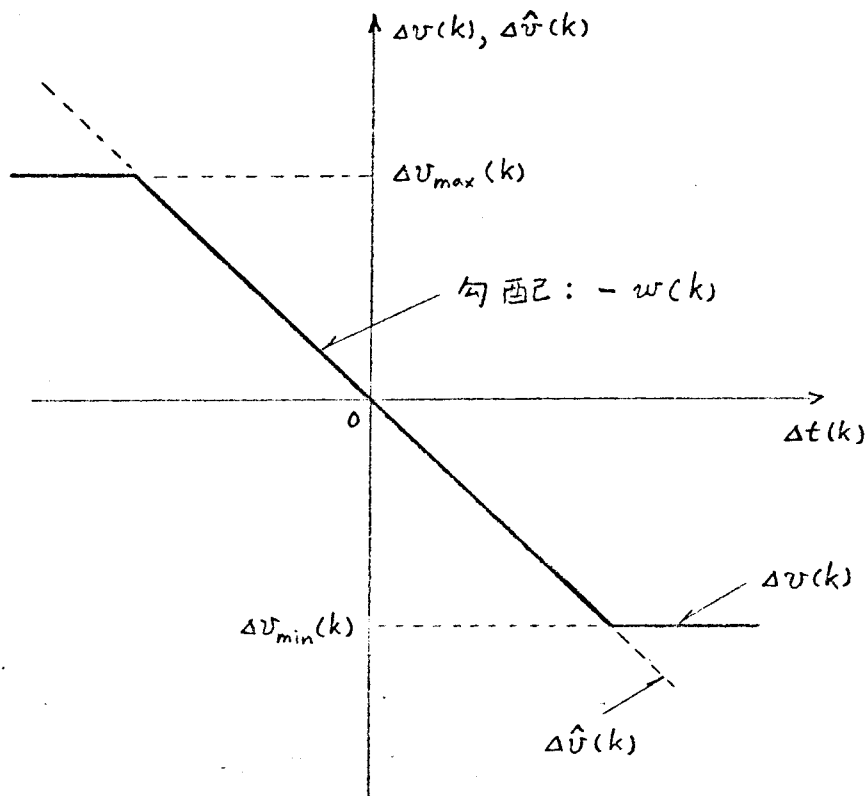
$$p(k) = q_1(k)$$

を後方に解くことにより得られる。<sup>18, 25</sup>

実際に列車に与える速度指令変化値としては、(2.7)式を考慮して、

$$\Delta v(k) = \begin{cases} \Delta v_{\min}(k), & \Delta \hat{v}(k) \leq \Delta v_{\min}(k) \\ \Delta \hat{v}(k), & \Delta v_{\min}(k) \leq \Delta \hat{v}(k) \leq \Delta v_{\max}(k) \\ \Delta v_{\max}(k), & \Delta \hat{v}(k) \geq \Delta v_{\max}(k) \end{cases} \quad (2.12)$$

を採用する。第2.1図に  $\Delta v(k)$ ,  $\Delta \hat{v}(k)$  および  $w(k)$  の関係を示す。



第2.1図 時間偏差と速度指令変化値の関係

## 2.3.2 分解原理による設計

ここでは (2.12) 式に示すフィードバック制御が, (2.7) 式を考慮に入れた上での最適制御にほぼ一致することを数値的に示すために分解原理<sup>20, 41</sup>を適用する。

### A. サブシステムレベルの最適化

$$\text{Minimize } \left\{ \begin{array}{l} q_1(k) \Delta t^2(k) + q_2(k-1) \Delta v^2(k-1) \\ \Delta t(k), \\ \Delta v(k-1) \end{array} \right. + \lambda_1(k) \Delta t(k) + \lambda_2(k-1) \Delta v(k-1) : \quad (2.13)$$

$$\Delta v_{\min}(k) \leq \Delta v(k-1) \leq \Delta v_{\max}(k) \left. \right\}$$

$$k = 1, 2, \dots, K$$

ここで  $\lambda_1(k)$  および  $\lambda_2(k-1)$  は補助パラメータであり, その初期値は任意に選べるので, ともに 0 に選ぶ。

$$\Delta v'(k-1) = -\lambda_2(k-1) / 2 q_2(k-1)$$

とおくと (2.13) 式の解は簡単に求まり,

$$\Delta t(k) = -\lambda_1(k) / 2 q_1(k) \quad (2.14)$$

$$\Delta v(k-1) = \begin{cases} \Delta v_{\min}(k-1), & \Delta v'(k-1) \leq \Delta v_{\min}(k-1) \\ \Delta v'(k-1), & \Delta v_{\min}(k-1) \leq \Delta v'(k-1) \leq \Delta v_{\max}(k-1) \\ \Delta v_{\max}(k-1), & \Delta v'(k-1) \geq \Delta v_{\max}(k-1) \end{cases}$$

で与えられる。

## B. センタレベルの協調

$$e(k) = \Delta t(k) - \Delta t(k-1) - \gamma(k-1)\Delta v(k-1) \quad (2.15)$$

$$k = 1, 2, \dots, K$$

と、

$$\|e(\lambda)\|^2 = e^2(1) + e^2(2) + \dots + e^2(K) \quad (2.16)$$

とおく。十分小さい  $\varepsilon$  に対して

$$\|e(\lambda)\| > \varepsilon \quad (2.17)$$

ならば

$$\lambda_1(k) + \{e(k) - e(k-1)\}\Delta\tau \rightarrow \lambda_1(k)$$

$$\lambda_2(k-1) - \gamma(k-1)e(k)\Delta\tau \rightarrow \lambda_2(k) \quad (2.18)$$

$$k = 1, 2, \dots, K$$

に従って補助パラメータ  $\lambda_1(k)$  および  $\lambda_2(k-1)$  を修正し、A. に戻る。

もし  $\|e(\lambda)\| \leq \varepsilon$  ならば繰り返し計算を停止する。ここで  $\Delta\tau$  は補助パラメータ修正のステップ中を示す。

## 2.4 直接法と分解原理による結果の比較

直接法による設計では、

a) 速度指令変化値に対する制限条件 (2.7) 式を無視して、時間偏差と速度指令変化値の関係を規定する制御系のゲインを求め、

b) a) で得たゲインにもとづいて速度指令変化値を順次求め、これが制

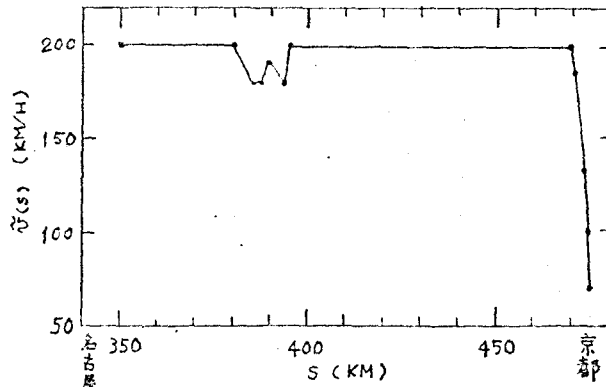
限值をはみ出した場合には、制限値を優先して採用する。

という手順に従うので、この結果得られる制御は理論的にはかならずしも最適な制御ではない。

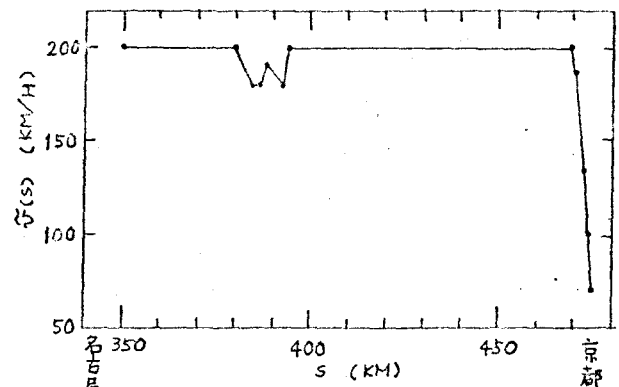
一方、分解原理を適用して得られる結果は、速度指令変化値の制限を考慮して、各サンプリング地点における制御が順次規定され、かつ理論的に最適性が保証される。

上記に示した二つの手法を東海道新幹線の走行曲線に適用し、直接法によって簡略に得られる結果の最適性を裏づける。第2.2図および第2.3図に、得た結果を示す。図中(a)に示す標準速度-距離曲線に従って走行する列車が、各々のサンプリング地点において、(c)に示す時間偏差を検知したときに加える速度指令変化値を(d)に示す。(b)は時間偏差と速度指令変化値の関係を規定するゲインの値を示している。第2.2図にはサンプリング距離  $\Delta s$  を 10 km に選んだ場合、第2.3図には  $\Delta s$  を 5 km に選んだ場合の結果を示している。時間偏差の収束の様子はよく一致しており、実用的に名古屋-京都間 120 km 程度の線区に対しては、サンプリング距離間隔は 10 km 程度で十分であると言える。第2.2図および第2.3図に示し得る範囲において、直接法によって得た結果と分解原理適用の結果はまったく一致した。第2.1表に両手法を適用して得た評価関数の値を比較しているが、ここでも両者の結果が良く一致している。IBM 360/50 による計算所要時間は直接法で1ケースあたり約10秒、分解原理で約60秒であった。

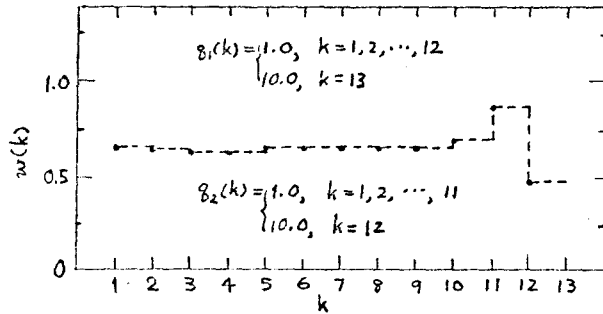
以上を示した検討の結果、速度指令変化に制限のある場合にも、直接法による簡略な設計で十分満足な結果が得られることがわかった。



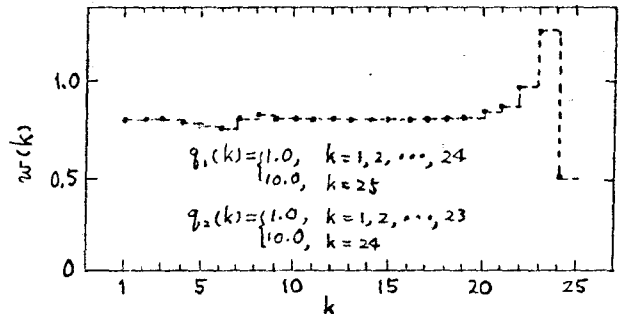
(a) 標準速度



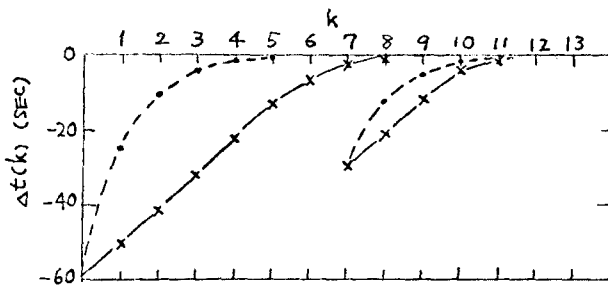
(a) 標準速度



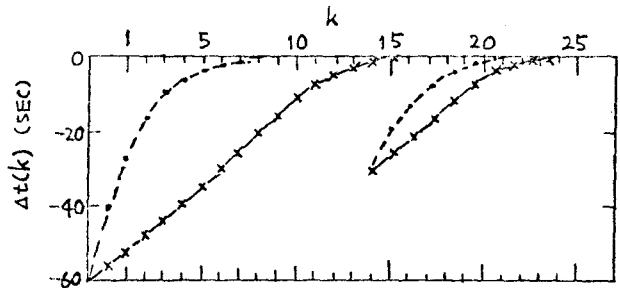
(b) ゲイン



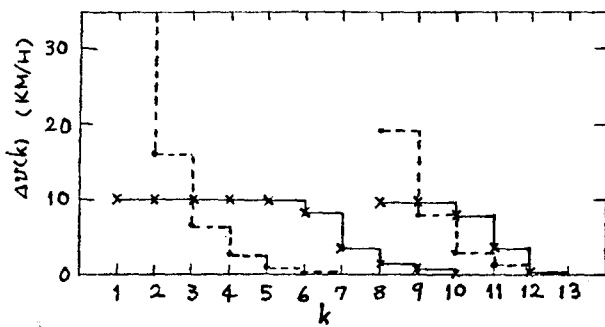
(b) ゲイン



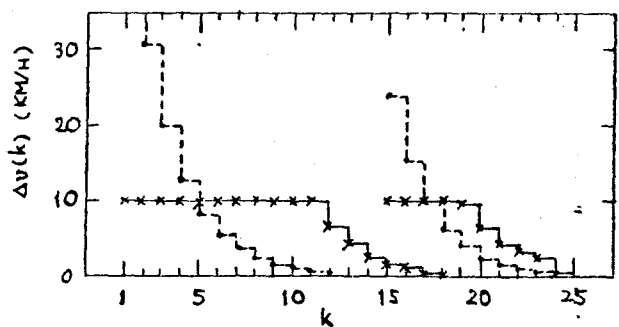
(c) 時間偏差



(c) 時間偏差



(d) 速度指令変化



(d) 速度指令変化

第 2. 2 図 プログラム自動運転:

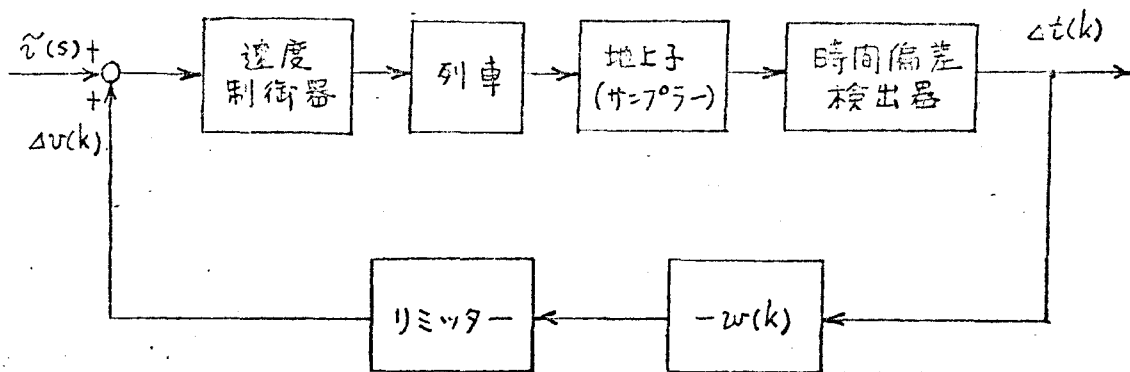
東海道新幹線  $\epsilon \alpha 1 \Delta s = 10 \text{ km}$

第 2. 3 図 プログラム自動運転:

東海道新幹線  $\epsilon \alpha 2 \Delta s = 5 \text{ km}$

第 2. 1 表 直接法および分解原理によって得た評価  
関数値の比較,  $\Delta S = 10 \text{ km}$

		直接法	分解原理		
		J	J	$\ e(\lambda)\ ^2$	繰返し数
例 1	制限条件なし	$2.58 \times 10^3$	$2.58 \times 10^3$	$6.72 \times 10^{-3}$	15
	$\Delta v_{\min}(k) \leq \Delta v(k) \leq \Delta v_{\max}(k)$	$6.74 \times 10^3$	$6.69 \times 10^3$	$9.58 \times 10^{-3}$	147
例 2	制限条件なし	$6.48 \times 10^2$	$6.46 \times 10^2$	$9.40 \times 10^{-3}$	15
	$\Delta v_{\min}(k) \leq \Delta v(k) \leq \Delta v_{\max}(k)$	$8.94 \times 10^2$	$8.87 \times 10^2$	$9.63 \times 10^{-3}$	35



第 2. 4 図 プログラム自動運転制御系のブロック図



## 2.5 プログラム自動運転制御系の構成

第2.4図にプログラム運転の制御系のブロック図を示す。標準ユニカーブのデータにもとづいて、直接法によりゲイン  $w(k)$  をあらかじめ求めておく。論理装置には S-UPプログラム, S-OPプログラムを記憶させておくと同時に  $w(k)$  の値を記憶させておく。この論理装置を車上に積載し、地上の一定距離間隔毎に設置される地上子(サンプラー)を検知するたびに時間偏差  $\Delta t(k)$  を検出し、論理装置から読み出されるゲインの値を用いて速度指令変化値  $\Delta v(k)$  を求め、速度指令値を列車の速度制御器にフィードバックする。

以上を示すごとくプログラム運転の制御系は、可変利得型比例制御系で構成される。S-UPプログラムの代りに信号保安の為の制限速度の数を減らす速度を用いると、論理装置に S-UPプログラムを記憶させる必要がなく、記憶する情報量を減らし得るので経済的に有利であるが、きめのこまかい速度制御はできなくなる。

## 2.6 むすび

プログラム運転制御系は、可変利得型比例制御系を構成することによって最適設計が行なえるという結論を得た。最適フィードバックゲインは、線形系の最適制御理論を用いて、あらかじめ off-line で求めておくことができる。

## 第3章 鉄道輸送制御系のスケジュール制御

### 3.1 まえがき

第2章では、鉄道輸送における個々の列車運転の自動化の一方式を検討した。本章では、複数回の列車を対象にした列車群制御の問題を検討する。個々の列車および列車群の制御を完全集中地上制御システムで構成することを仮定した場合、すなわち、走行中の列車と地上の制御用計算機との情報交換が常時可能であると仮定した場合の列車群制御方式<sup>26,42</sup>はいくつか提案されているが、これらを実現するための情報伝送装置の規模、地上の制御用計算機の規模が膨大になり、経済性の観点から問題がある。

この章で述べる列車群のスケジュール制御方式は、運転ヘッドの小さい通勤列車のスケジュール制御に適した方法で、基本的な考え方として、  
a) 運転ダイヤが乱れた時に、まず全列車の総合的な運転時隔の変動を小さくする。

b) 回復余力の範囲内で、個々の列車の時間偏差の制御を行う。

をあげることができる。a)の為の制御は、地上に設置する制御用計算機が担当し、計算機から各列車への制御指令は駅のみで与える。b)の為の制御は、車上に積む制御器（たとえば第2章で議論したプログラム運転装置）が担当し、個々の列車単位で行う。

ここで述べるスケジュール制御方式は、ダイヤ遅延を招いた原因（たとえば車両故障など）を取り除いた後に適用できる方式であり、故障状態が除去されていない時や、回復運転を実施している最中に再び別の故障が発生した場合に対する解決方法は与えていない。

### 3.2 地上の制御用計算機の制御機能

列車遅延が生じると駅での乗客が増加し、さらに遅延を増大させ、列車ダイヤをますます混乱させる結果になるので、次駅までに遅れ時分を回復できない遅延列車が検出されたときに、先行列車を故意に駅で待避させ、運転時隔の変動をそうえる。先行列車の駅出発に関する指令を中央の制御用計算機が担当するが、これに必要な計算機の入出力情報は次のとおりである。

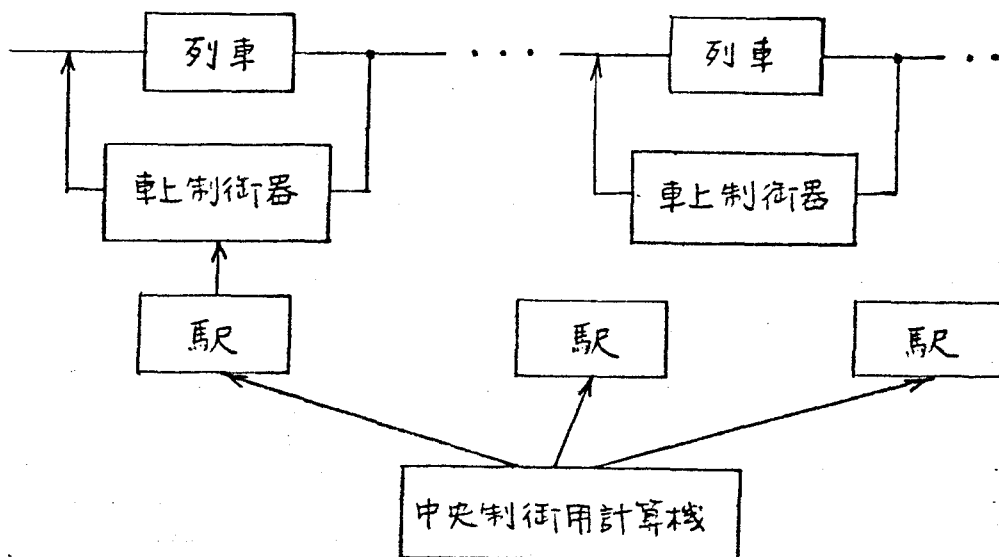
#### a) 入力情報

- 遅延発生場所
- 遅延列車名
- 遅延の度合い
- 遅延発生地点と、各先方駅の向を走行している列車台数

#### b) 出力情報

遅延列車の先方を走行している各列車に対して、どの駅で何分待避させるかに関する情報。

ここで、中央の制御用計算機から、各列車への制御指令は駅のみで与える。



第3.1図 列車群制御システム

個々の列車の運転を制御する車上制御機能と、中央の制御用計算機による中央制御機能の関係をあらわすブロック図を第3.1図に示す。

### 3.3 中央制御のアルゴリズム

中央制御のアルゴリズムとしては、中央の制御用計算機の情報処理量ができるだけ少なく、地上設備（情報伝送装置、計算機）の規模ができるだけ小さくてすみ、かつ有効な制御機能を果たすことが要求される。

今、 $s$  を位置、 $s_j$  を列車遅延発生地点  $s_0$  から前方駅に順次番号をつけたときの  $j$  駅の位置とする。また  $x(s_0, s_j)$  を列車遅延発生地点  $s_0$  と  $j$  駅の間を走行している列車台数とする。

A.  $x(s_0, s_j) \geq j, j = 1, 2, \dots$  を満足するとき

列車遅延発生地点  $s_0$  と  $j$  駅  $s_j$  の間にある列車台数  $x(s_0, s_j)$  が、すべての  $j$  に対して  $j$  に等しいかあるいは  $j$  を上回っているとき、中央制御のアルゴリズムはもっとも簡単になる。今、 $f_j(s)$  を地点  $s$  における  $j$  列車（最初に遅延した列車を 0 列車といて、先行列車に手前から順番に番号をつける）の遅延時分とする。また  $\text{Max}(A, B)$  を  $A$  と  $B$  の大きいほうの値とすると、中央制御のアルゴリズムとしては、先行列車に対して、

$$f_j(s_j) = \text{Max}(0, f_{j-1}(s_j)), \quad j = 1, 2, \dots \quad (3.1)$$

で与えられる待避時分を順次与えて行く。たゞし、

$$f_0(s_0) = \Delta T_0 \quad (3.2)$$

$$f_0(s) = \text{Max}\{0, \alpha_k \cdot (s - s_k) + f_0(s_k)\}, \quad (3.3)$$

$$s_k < s \leq s_{k+1}, \quad k = 0, 1, 2, \dots$$

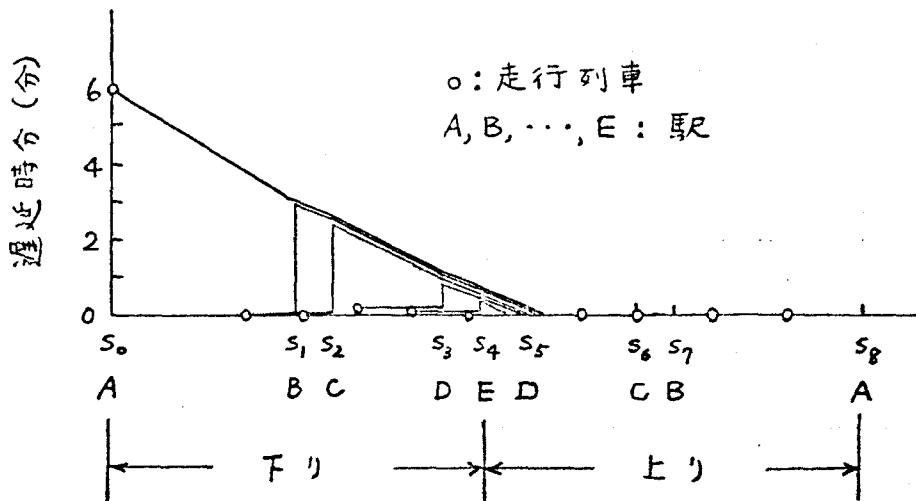
$$f_j(s) = \begin{cases} 0, & s < s_j \\ \text{Max} \{ 0, f_{j-1}(s_j) \}, & s = s_j \\ \text{Max} \{ 0, -\alpha_{j+k}(s - s_{j+k}) + f_j(s_{j+k}) \}, & s_{j+k} < s \leq s_{j+k+1}, k=0, 1, 2, \dots \end{cases} \quad (3.4)$$

$$j = 1, 2, 3, \dots$$

ここで  $\Delta T_0$  は初期遅延発生時分,  $\alpha_j$  は  $j$  駅と  $(j+1)$  駅区間の単位距離走行中に回復し得る平均回復時分 (回復能力) を示す。(3.3), (3.4) 式を用いて (3.1) 式を順次計算し,  $j$  駅における  $j$  列車の待避時分を求めらる。

$$f_j(s_j) = 0 \quad (3.5)$$

を得た後計算を停止する。第3.2図に列車群のスケジュール回復動作の一例を示す。本計算を実施するために, 第3.1表に示すデータを用いた。



第3.2図 列車群のスケジュール回復動作 その1,  
 $x(s_0, s_j) \geq j, \forall j$

第3.1表 中央制御用計算機の入力情報 その1

駅名	$j$	$s_j$ (km)	$x(s_0, s_j)$	$\alpha_j$ (min/km)
A	0	0	/	0.06
B	1	50		1
C	2	60	2	0.05
D	3	90	4	0.04
E	4	100	5	0.04
D	5	110	5	0.05
C	6	140	7	0.04
B	7	150	8	0.06
A	8	200	10	0.04

上記に示すアルゴリズムは、各列車の各区間における平均回復余力  $\alpha_j$  を用いて、遅延列車の各駅到着時刻を予測計算し、各列車の駅における待避時分を計算するものである。

B.  $x(s_0, s_j) \geq j$ ,  $\forall j$  を満足しないとき

( $m-1$ ) 駅までは

$$x(s_0, s_j) \geq j, \quad j=1, 2, \dots, m-1$$

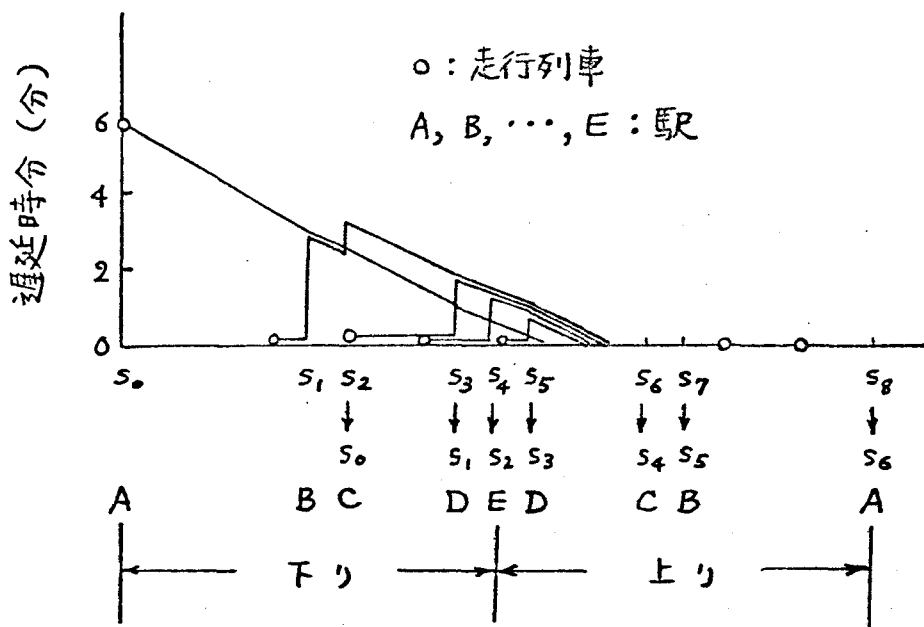
を満足し、 $m$  駅ではじめて

$$x(s_0, s_j) < m$$

となった場合、すなわち  $m$  駅において待避させる列車が存在しない場合には、 $(m-1)$  駅まで A. の手順に従って各列車の各駅における待避時分を求め、 $(m-1)$  列車は、 $m$  駅において、乗客増加による遅延増大をまわく。そこで新たに、

$$\Delta T_0 = (1 + \beta_m) f_{m-1}(s_m) \quad (3.6)$$

に設定  $i$ ,  $m$  駅を初期遅延発生地点とみなし、 $(m-1)$  列車を 0 列車に設定して A. に戻る。ここで  $\beta_m$  は、 $j$  駅における乗客増加による遅延時分の増加率を示す。第 3.3 図に一例を示す。ここでは、列車群のスケジュール回復完了までに、 $x(s_0, s_j) < j$  が 1 回起った場合の例を示している。本計算を実施するのに第 3.2 表に示すデータを用いた。



第 3.3 図 列車群のスケジュール回復動作 その 2,

$$x(s_0, s_2) = 1 < 2$$

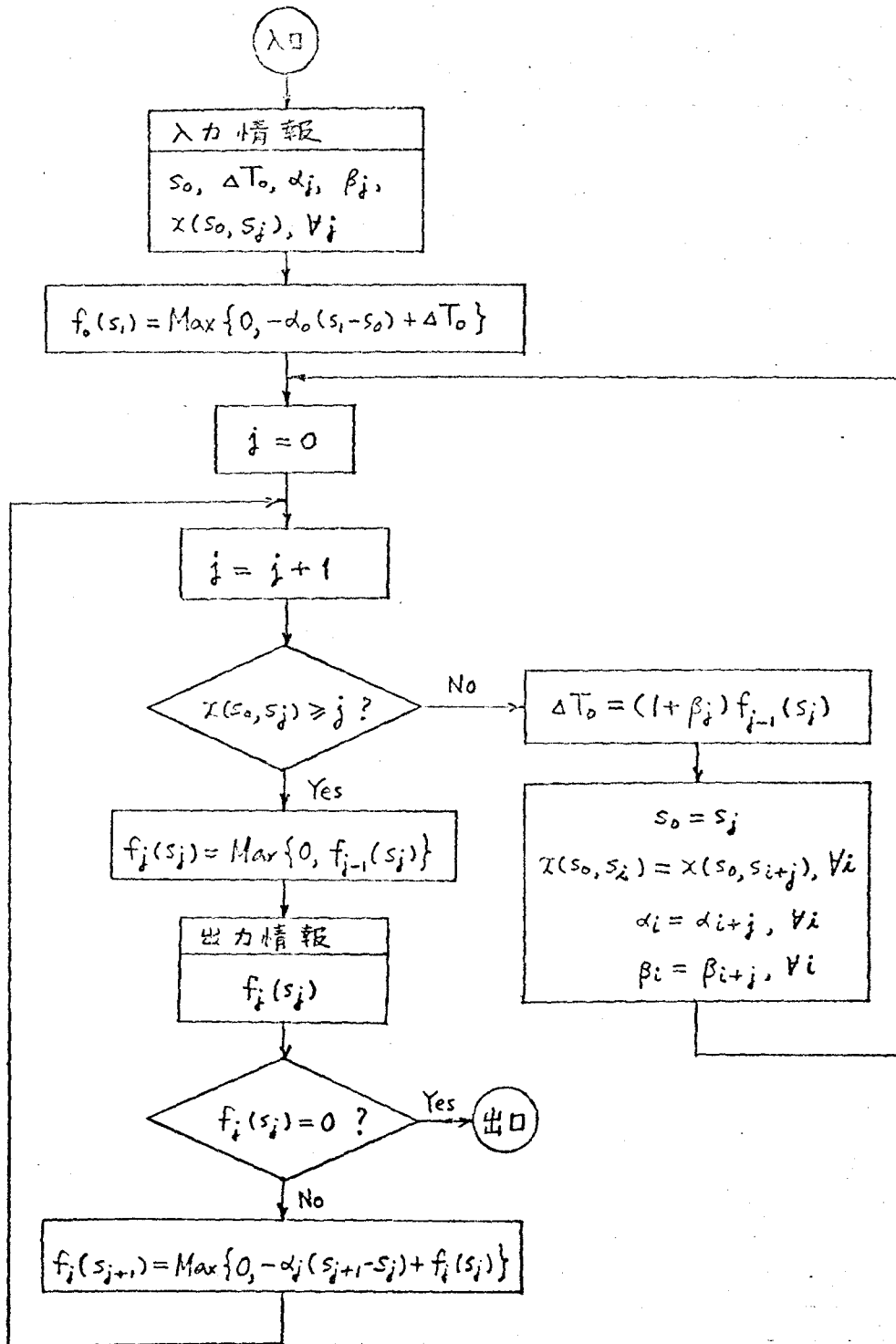
第 3. 2 表 中央制御用計算機の入力情報 その 2

駅名	$i$	$s_j$ (km)	$x(s_0, s_j)$	$\alpha_j$ (min/km)	$\beta_j$
A	0	0	/	0.06	0.3
B	1	50	1	0.04	0.2
C	2→0	60	1(<2)	0.05	0.3
D	1	90	2	0.04	0.2
E	2	100	2	0.04	0.3
D	3	110	3	0.05	0.2
C	4	140	4	0.04	0.3
B	5	150	5	0.06	0.2
A	6	200	7	0.04	0.3

上記に示すアルゴリズムは、遅延列車の先行列車を先方駅において順次待避させる過程において待避させる列車が存在しないときに、その駅における乗客増加による遅延時分の増加率  $\beta_j$  を用いて列車遅延時分を予測計算し、この駅を新たに初期遅れ発生地点とみなして A. に戻すものである。

A. および B. からなる中央制御のアルゴリズムをフローチャートにまとめて第 3. 4 図に示す。複数回の列車遅延が同時に検出された場合には、遅延時分の最大値をとる地点を基準にして、上に示したアルゴリズムをそのまま使うことができる。





第3.4回 中央制御のための計算のフローチャート

### 3.4 特徴

前節に示した中央制御のアルゴリズムは、中央の制御用計算機の負担をできるだけ軽くし、中央制御のアルゴリズムを簡略化するという立場で開発されたものである。ここで車上の制御器は、常時個々の列車の遅延時分を検出し走行速度を制御することによって回復動作をとる。遅延時分を次の停車駅到着までに回復できないうちに、中央計算機のプログラムが動作し、先行列車の各駅における待避時分を計算し各駅に指令を送る。

本方式の特徴を列挙すると次のようになる。

- a) 中央制御機能と車上制御機能を合理的に配分することによって、中央制御のアルゴリズムが簡略にまとめられた。
- b) 中央の制御用計算機から各列車への情報伝達が駅のみで行なわれるので、地上設備（情報伝送装置、中央計算機、端末機器など）の規模が小さくて済む。
- c) 運転ハットの小さい列車群のスケジュール制御に適している。

### 3.5 むすび

本章で提案した列車群制御方式は、個々の遅延列車の回復動作を車上制御器が担当し、地上の制御用計算機は全列車の総合的な運転時隔の変動をそろえるための指令値を計算し、駅を介して各列車に制御指令を送るという簡単な方式である。地上計算機から列車への情報伝送を駅向で行なうと先行列車を制御すると、駅における列車待避時分を減らすことができ、乗客の心理状態を緩和することができるが、回復性能は改善されない。

Levine と Athans は、列車群を連続的に制御し、スケジュールを回復させる方式を発表しているが<sup>26</sup>、中央と列車の情報伝送の見地から、実用的でない。

## 第 II 部

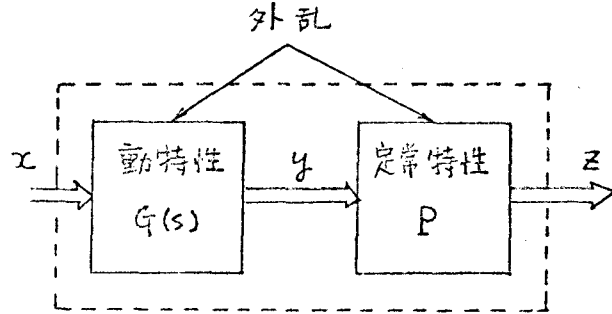
生産工程のシステム制御に関する研究

$$\mathcal{L}[y(t)] = G(s) \mathcal{L}[x(t)], \quad x(t) \geq 0 \quad (4.1)$$

$$z(t) = z(y(t)) \leq k \quad (4.2)$$

$$g(z) = A \cdot z(t) \leq b \quad (4.3)$$

ここで  $\mathcal{L}[x(t)]$  は  $x(t)$  のラプラス変換をあらわしており、 $k$ 、 $b$  は各々  $m$  次元および  $l$  次元定数ベクトル、 $A$  は  $l \times m$  係数行列である。



第4.1図 制御対象のブロック図

(4.1) 式はプロセスの動特性をあらわし、(4.

2) 式はプロセスの定常特性と出力変数に対する制限条件を、また(4.3) 式は外部から出力変数に加えられる制限条件を示している。一方プロセスの定常的な評価関数は、 $c$ 、 $d$  を各々  $n$ 、 $m$  次元定数ベクトルとして

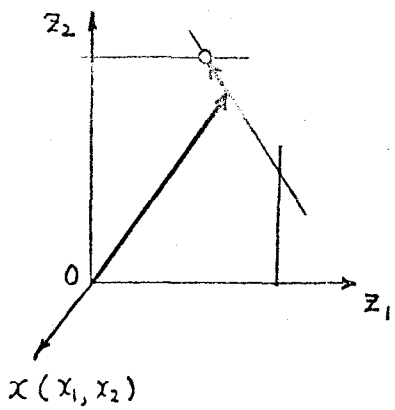
$$F(x, z) = c \cdot x + d \cdot z \quad (4.4)$$

に示すように入力および出力変数の線形結合で表わせるものとする。

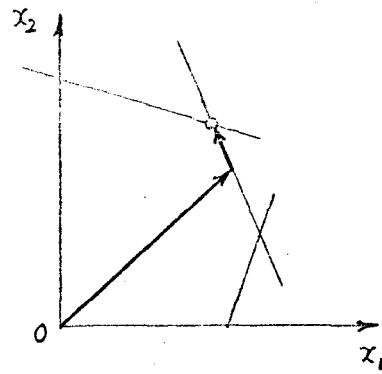
線形系を仮定しているので(4.2) 式は一般に

$$z(t) = z(y(t)) = P y(t) \quad (4.5)$$

とかけるが、 $m \times r$  係数行列  $P$  は未知である。したがって制御系は  $P$  に関する情報を用いなくて構成しなければならない。動特性を考慮しないで、かつ(4.5) 式における  $P$  が陽に与えられておれば上記の問題は通常の LP の問題である。第4.2図に本章で取り扱っている問題と通常の LP<sup>6,12</sup> の問題の幾何学的解釈を示す。ここで取り扱っている問題では、系の数学モデルが未知であるため  $(z_1, z_2)$  と  $(x_1, x_2)$  の間の関係を幾何学的に適



a) 数学モデル未知の場合



b) 通常のLPの場合

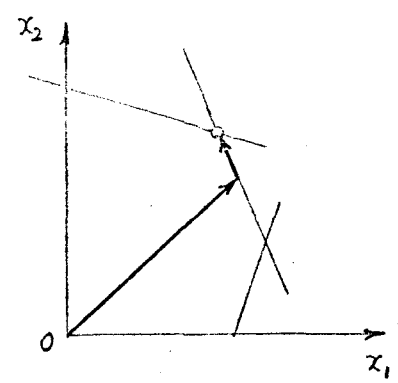
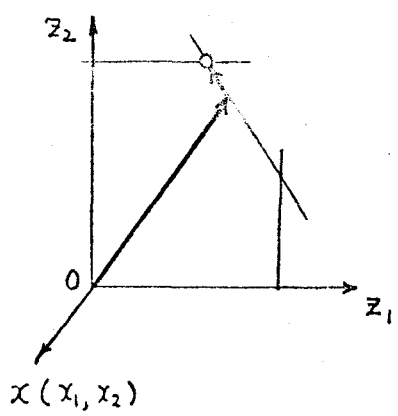
#### 第4.2図 幾何学的解釈

確に表現することができないが、第4.2図a)において  $(x_1, x_2)$  の一組の値を与えると  $(z_1, z_2)$  平面上に一点がきまり、 $(x_1, x_2)$  を制御すると動点は  $(z_1, z_2)$  平面上を図に示したように移行する。zに関する制限条件(4.2)(4.3)式をみたし(4.4)の評価関数を最大または最小にする最適点(図中の○印)  $(z_1, z_2)$  に対応して一組の  $(x_1, x_2)$  が存在し、この最適操作条件  $(x_1, x_2)$  を求めることが本章の目的である。

#### 4.3 最適化制御系の構成

数学モデル未知のプロセスの最適化を行う最も強力な手法は最急勾配法である。ここでは系が線形であることを仮定しているので、最急勾配の方向は一度何等かの方法で決定すれば線形近似が妥当な範囲で変化しない。まず(4.4)式を、直接操作可能な変数  $x$  のみの関数におきかえろ。これは、入力ベクトル  $x$  の各要素が評価関数に及ぼす影響を実験により求めておけばあらかじめ決定することができる。その結果(4.4)式は

$$F(x, z(x)) = F(x) = c^T x \quad (4.6)$$



a) 数学モデル未知の場合

b) 通常の LP の場合

第 4.2 図 幾何学的解釈

確に表現することができないが、第 4.2 図 a) において  $(x_1, x_2)$  の一組の値を与えると  $(z_1, z_2)$  平面上に一点がきまり、 $(x_1, x_2)$  を制御すると動点は  $(z_1, z_2)$  平面上を図に示したように移行する。z に関する制限条件 (4.2) (4.3) 式をみたし (4.4) の評価関数を最大または最小にする最適点 (図中の O 印)  $(z_1, z_2)$  に対応して一組の  $(x_1, x_2)$  が存在し、この最適操作条件  $(x_1, x_2)$  を求めることが本章の目的である。

4.3 最適化制御系の構成

数学モデル未知のプロセスの最適化を行う最も強力な手法は最急勾配法である。ここでは系が線形であることを仮定しているので、最急勾配の方向は一度何等かの方法で決定すれば線形近似が妥当な範囲で変化しない。まず (4.4) 式を、直接操作可能な変数  $x$  のみの関数におきかえろ。これは、入力ベクトル  $x$  の各要素が評価関数に及ぼす影響を実験により求めておけばあらかじめ決定することができる。その結果 (4.4) 式は

$$F(x, z(x)) = F(x) = e'x \quad (4.6)$$

の形におきかわる。ただし  $e$  は  $m$  次元定数ベクトルである。

(4.1) ~ (4.3) 式を満足する領域内で (4.6) 式を最大または最小にたもつ  $x$  を求めるために、任意の初期値から出発し

$$\frac{dx_j}{dt} = K_1 e_j - K_2 \left\{ \sum_{i=1}^m \xi_i - \eta_j + \sum_{i=1}^l \zeta_i \right\} \quad (4.7)$$

$$(j = 1, 2, \dots, n)$$

ただし

$$\xi_i = \begin{cases} 1, & z_i > k_i \\ 0, & z_i \leq k_i \end{cases} \quad (i = 1, 2, \dots, m)$$

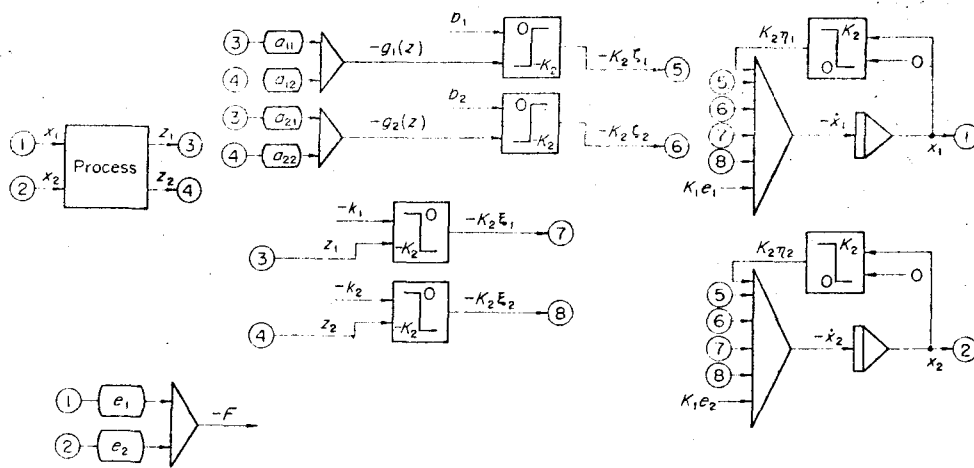
$$\eta_j = \begin{cases} 1, & x_j < 0 \\ 0, & x_j \geq 0 \end{cases} \quad (j = 1, 2, \dots, n)$$

$$\zeta_i = \begin{cases} 1, & g_i(z) > b_i \\ 0, & g_i(z) \leq b_i \end{cases} \quad (i = 1, 2, \dots, l)$$

に従って  $x$  の各要素  $x_j$  を変化させる。ただし  $K_1, K_2$  は或る定数であるが、この値の決定法に関しては後で論じる。 $z_i, k_i, g_i(z), b_i$  は各々ベクトル  $z, k, g(z), b$  の  $i$  番目の要素である。(4.7) 式の第1項は評価関数 (4.6) 式の最急勾配をあらわしており、第2項は  $z_i \leq k_i$  ( $i = 1, 2, \dots, m$ )、第3項は  $x_j \geq 0$  ( $j = 1, 2, \dots, n$ )、第4項は  $g_i(z) \leq b_i$  なる制限条件をはみ出したときに、動点を制限領域内に引きもどすための帰還量である。

LP の幾何学的意味から考えて、最適解は制限領域の端点に存在することから、上記のように動点を最急勾配の方向 (一定) に走らせ、動点が制限領域からはみ出したときに制限領域内へ引きもどすというアルゴリズムは妥当である。ここで引き戻す方向を厳密に規定する必要はない。

2変数の簡単な問題に対して制御対象および制御系をアナログ計算機で模擬したときの構成図を第4.3図に示す。ここでプロセスの動特性によ



第4.3図 アナログ計算機による最適化制御系の構成

るおくれが介入するために  $z$ ,  $g(z)$  に関して, プロセスの時定数だけ遅れた信号を検出して, 動点が制限領域内にあるか否かを判別している。したがって必ず制限領域外へのオーバーシュートが起る。オーバーシュートの大きさは, プロセスの時定数と (4.7) 式の  $K_1$  の選び方によって異なる。  $K_1$  を大きく送れば最適点へ到達するに要する時間は短縮できるが, 制限領域外へのオーバーシュートが大きくなり, 最適点近傍における振動の振幅が大きくなる。一方  $K_1$  を小さく送れば最適解へ到達するに要する時間が大きくなる。すなわち, プロセスが与えられれば, これに対応してその系に適した  $K_1$  が存在する。  $K_2$  は, 動点が制限領域外へとび出したときに, 領域内へ引き戻すに十分な大きさに送ればよい。

#### 4.4 アナログ計算機によるシミュレーション

ここでは第4.3図に示した制御系の模擬回路をアナログ計算機で構成し, 制御方式の妥当性を検討した結果を示す。制御対象の動特性として, 1次遅れ系, 2次系, 逆応答系を仮定した。とりあげた例題は, 次に示す2変数系の問題である。



$$\mathcal{L}[y_1(t)] = G_1(s) \mathcal{L}[x_1(t)] \quad (4.8)$$

$$\mathcal{L}[y_2(t)] = G_2(s) \mathcal{L}[x_2(t)]$$

$$z_1 (= y_1 - 0.5 y_2) \leq 50 \quad (4.9)$$

$$z_2 (= y_1 + 0.25 y_2) \leq 80$$

$$g_1(z_1, z_2) = 0.25 z_1 + 0.5 z_2 \leq 67.5 \quad (4.10)$$

$$g_2(z_1, z_2) = -z_1 + z_2 \leq 60$$

の制限条件のもとに、

$$\begin{aligned} F &= -10 x_1 - x_2 + z_1 + 10 z_2 \\ &= x_1 + x_2 \end{aligned} \quad (4.11)$$

を最大にせよ。

ここで、(4.8) (4.9) 式は、プロセス自体の制限条件であり、(4.10) 式は、外部から加えられる制限条件である。制御系を構成する際、(4.9) 式にあらわれる  $y_1$ ,  $y_2$  の係数の値に関する情報を使うことはできない。

定常状態において、(4.11) 式で与えられる評価関数を最大にする ( $x_1, x_2$ ), ( $z_1, z_2$ ) およびその時の  $F$  の値は

$$(x_1, x_2) = (60, 80)$$

$$(z_1, z_2) = (20, 80)$$

$$F_{\max} = 140$$

で与えられる。

#### 4.4.1 1次遅れ系

プロセスの動特性として、1次遅れ系

$$G_1(s) = G_2(s) = \frac{1}{1 + Ts} \quad (4.12)$$

を仮定して (4.8) ~ (4.11) 式で与えられる例題を解いた結果を第4.4図に示す。

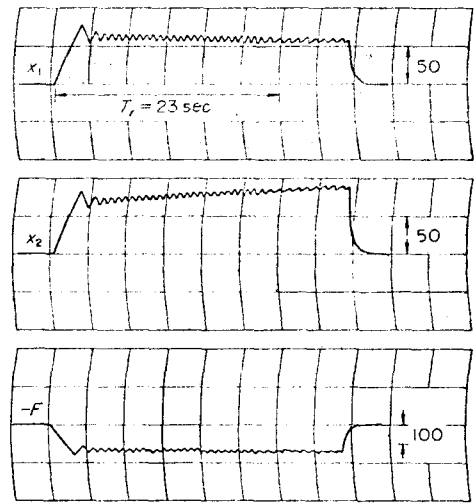
最適点近傍における振動の振幅  $A$  および最適点へ到達するに要する時間  $T_r$  は  $K_1$  の値の選び方に支配される。プロセスの時間定数  $T$  をパラメータとして、 $A$  と  $K_1$ 、および  $T_r$  と  $K_1$  の関係を実験的に求めた結果を第4.5図に示す。この図が示しているように、 $K_1$  はできるだけ大きな値に選ぶ方が最適点へ到達するに要する時間  $T_r$  は短くて済む。プロセスが与えられれば、どの程度の振動を許容し得るかを検討し、許容範囲内で最大の  $K_1$  を選ぶ。次に動点が最適点近傍へ近づくと  $K_1$  を小さくする。この結果、最適点近傍における振動現象を縮小することができる。

この制御系はプロセスの特性方程式を用いずに構成しているので、特性変化によって最適点の位置が多少変動しても容易に追従する。

#### 4.4.2 2次系

プロセスの動特性として、2次系

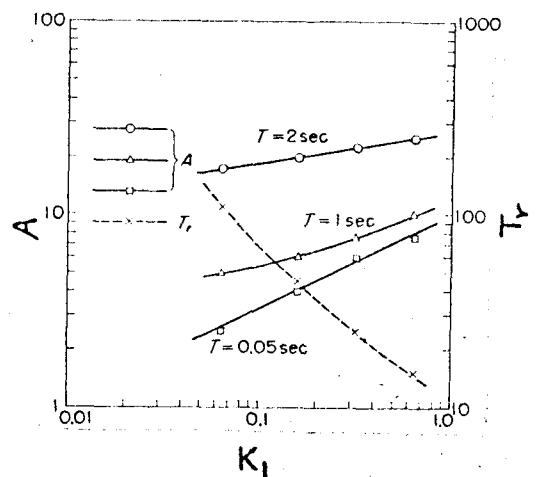
$$G_1(s) = G_2(s) = \frac{1}{1 + qs + ps^2} \quad (4.13)$$



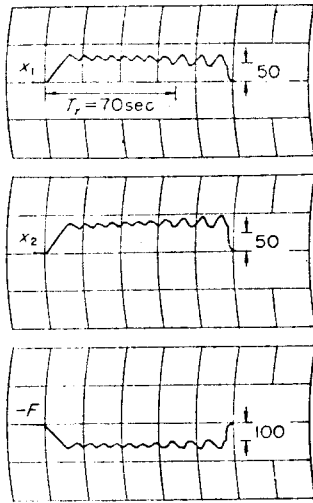
$$G(s) = \frac{1}{1 + 0.5s}$$

$$K_1 = 0.32$$

第4.4図 数学モデル未知のLPの解,  $G(s) = \frac{1}{1 + Ts}$



第4.5図  $A$  と  $K_1$  および  $T_r$  と  $K_1$  の関係,  $G(s) = \frac{1}{1 + Ts}$

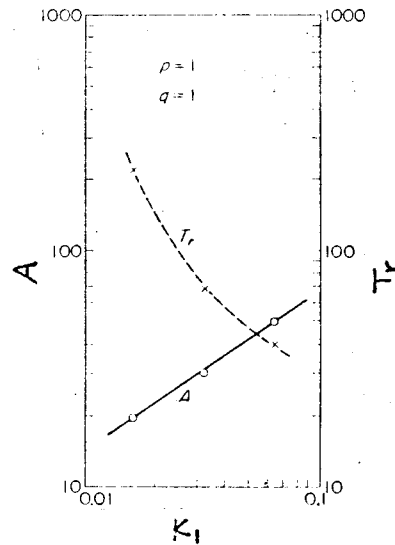


$$G(s) = \frac{1}{1+s+s^2}$$

$$K_1 = 0.032$$

第4.6図 数学モデル未知

のLPの解,  $G(s) = \frac{1}{1+qs+ps^2}$



第4.7図 AとK<sub>1</sub> および Trと

K<sub>1</sub> の関係,  $G(s) = \frac{1}{1+qs+ps^2}$

を仮定して同じ例題を解いた結果を第4.6図に示す。また、AとK<sub>1</sub> および TrとK<sub>1</sub> の関係を実験的に求めた結果を第4.7図に示す。

K<sub>1</sub> の選び方に関する評価規準は、1次遅れ系と同様である。第4.5図と第4.7図の比較から明らかなように、同じK<sub>1</sub>の値に対して、最適点へ到達するに要する時間は、2次系の方が短い。

#### 4.4.3 逆応答系

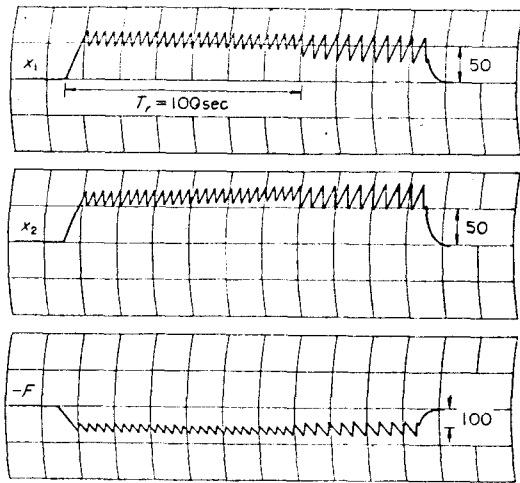
プロセスの動特性として、逆応答系

$$G_1(s) = G_2(s) = \frac{\alpha}{1+T_1s} - \frac{\beta}{1+T_2s}$$

$$\alpha - \beta = 1$$

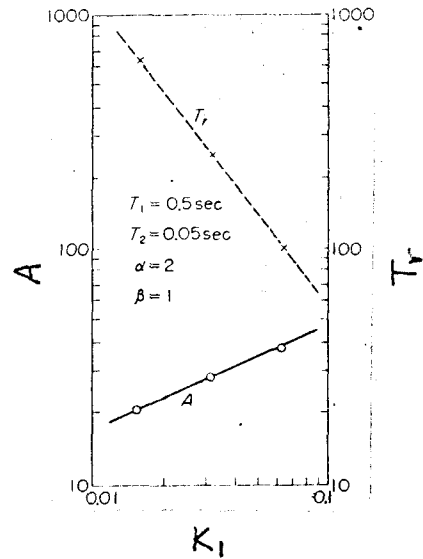
$$T_1 > T_2$$

を仮定して同じ例題を解いた結果を第4.8図に示す。また、AとK<sub>1</sub> および TrとK<sub>1</sub> の関係を実験的に求めた結果を第4.9図に示す。



$$G(s) = \frac{2}{1+0.5s} - \frac{1}{1+0.05s}$$

$$K_1 = 0.064$$



第4.8図 数学モデル未知のLP

第4.9図 AとK1 およびTrと

の解,  $G(s) = \frac{\alpha}{1+T_1s} - \frac{\beta}{1+T_2s}$

K1の関係,  $G(s) = \frac{\alpha}{1+T_1s} - \frac{\beta}{1+T_2s}$

逆応答系では、動点が制限領域外へとび出した時に比較器から出る帰還信号に対して、逆方向に応答するため振動の振幅Aは、1次系のとときに比べて大きくなる。しかし、同じK1の値に対して、最適点へ到達するに要する時間はほとんど変わらない。

一般に、試行法では変数の数αの増大とともに、その試行回数が増大する。αの増大に伴って、多変数化が極めて困難であるが、上記のように線形近似ができる場合には最急勾配の方向を実験により一度決めてやれば、動点が制限領域をとび出すまで一定の方向にそって連続的に変化させ得るので、多変数化が容易である。

動点を動かす速度を規定するパラメータK1の決定はプロセスの動特性の概略をあらかじめ検討しておけば上記の評価規準にしたがって容易に決定できる。また、入力変数の中で、出力に対して大きく効くものが何か

かっている場合には、動点が制限領域からはみ出した時にかかる帰還量を加減してやればさらに高速化がはかれる。たとえば出力  $z_i$  に対して  $x_j$  が大きく効く場合、動点が  $z_i \leq k_i$  をはみ出した時に、 $x_j$  にかかる帰還量を他の変数にかかる帰還量よりも大きく設定してやれば良い。

#### 4.5 むすび

線形近似のできる、特性方程式未知のプロセスの多変数最適化制御問題を、LPの変形としてとらえ、簡単なアナログ要素で on-line 最適化制御を行う方式を示し、その妥当性を検討した。

従来の制御系では、出力値がある値を越えると警報を出し、警報が出ない範囲で適当な設定値を選んで駆動するという警報システムをとっていた。このような系に対して、利潤最大、効率最大、コスト最小といった評価規準を設けて、簡単なアナログ装置で、最適化制御を実施することができれば、これによるメリットの増大は大きいものと思う。

## 第 5 章 シミュレーションによる生産スケジューリング

### 5.1 まえがき

第 4 章では、一つのプロセスを制御対象にした問題を取扱った。本章では、制御対象を一つの生産工程に拡大して、生産量を日ごとに、工場ごとに、あるいは機械ごとに割り当てる生産スケジューリングの問題を取上げる。生産スケジューリングの問題は、いいかえれば順序づけの問題とみることができ、その目的は、与えられた生産命令を最短時間で完了することであり、生産効率を向上させる上で極めて重要な問題である。<sup>2, 38</sup>

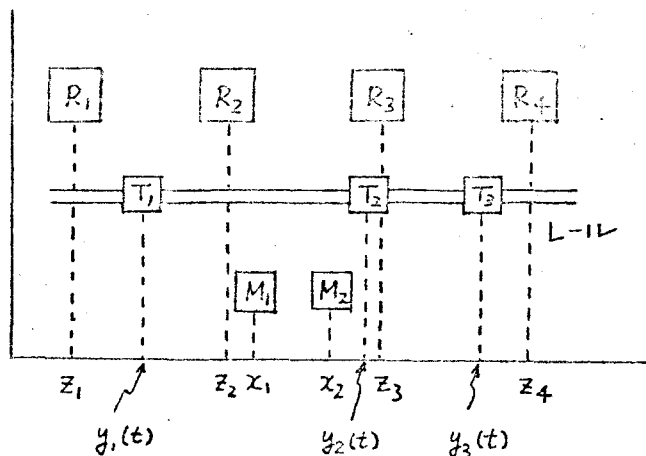
スケジューリングの問題は、本質的には多段決定過程であり、動的計画法（以下 DP と略す）で解こうとしている例<sup>17</sup>もあるが、モデルが余程簡単でないと計算量が莫大になり、手がつけられない。もちろん、すべての組合せについて検討することは天文学的数字になり、不可能である。問題の焦点は、いかにして計算回数を減らし、現実問題に適用し得る、より合理的な手法を見い出すかにある。ここでは機械加工の一モデルを想定し、考え得るすべての解の中の最適解を得ようという試みは行なわないで、大型計算機で生産工程を模擬することにより、できるだけ少ない計算回数で、より良い実行可能解を得ようと試みている。

### 5.2 生産工程とそのシミュレーション

#### 5.2.1 機械加工工程

第 5.1 図に示す機械加工工程を想定する。ここでは 2 台の機械  $M_1$ ,  $M_2$  を並列運転し、item  $S_n$  ( $n=1, 2, \dots, N$ ) を  $M_1$  または  $M_2$  で

機械加工する。機械加工の終わった item は、機械から取りはずし、 $T_1 \sim T_3$  の3台の運搬車のいずれかで、貯蔵所  $R_1 \sim R_4$  のどこかへ格納する。本工程において、機械加工の終わった item は、たゞちに運搬車で運搬し、貯蔵所へ格納しなければならぬ。



第5.1回 機械加工工程

item  $S_n$  の機械加工において、各工程で費される所要時間を

- $a_{n1}$  : 機械へ取りつけるのに要する時間
  - $a_{n2}$  : 機械加工するのに要する時間
  - $a_{n3}$  : 機械から取りはずすのに要する時間
  - $a_{n4}$  : 貯蔵所へ格納するのに要する時間
- であらわす。

### 5.2.2 問題の設定

問題は、item  $S_n$  ( $n=1, 2, \dots, N$ ) が与えられ、この全 item を機械加工するのに要する時間を最小にするよう  $S_n$  ( $n=1, 2, \dots, N$ ) の順列を決定せよ。いいかえれば、2台の機械  $M_1, M_2$  の稼働率が最大になるようにスケジュールを組み、という形で与えられる。以下において順列  $\{S_1, S_2, \dots, S_n\}$  を  $\{S_n\}_{n=1, 2, \dots, N}$  であらわす。

スケジュールを決定して行く過程において、制限条件

- a) 機械  $M_1, M_2$  が同時に機械加工の状態にある overlap 時間  $T_L$  は、

$$T_L \leq T_{L \max} \quad (5.1)$$

でなければならない。ただし  $T_{L \max}$  は許容最大 overlap 時間をあらわす。

- b) 3台の運搬車  $T_1, T_2, T_3$  は、一本のレール上し動かさないのをおのおのが衝突しないこと、を考慮しなければならない。

### 5.2.3 工程のシミュレーション

item  $S_n$  を機械加工する工程を模擬するのに、以下に示す記号を用いる。

$$I_n = i \quad (i = 1 \text{ または } 2) \quad (5.2)$$

$$J_n = j \quad (j = 1, 2 \text{ または } 3) \quad (5.3)$$

$$K_n = k \quad (k = 1, 2, 3 \text{ または } 4) \quad (5.4)$$

(5.2) 式は、item  $S_n$  を機械  $M_i$  ( $i = 1$  または  $2$ ) で機械加工すること、(5.3) 式は item  $S_n$  を運搬車  $T_j$  ( $j = 1, 2$  または  $3$ ) で運搬すること、(5.4) 式は item  $S_n$  を貯蔵所  $R_k$  ( $k = 1, 2, 3$  または  $4$ ) へ格納することをあらわす。これらの記号を用いると、順列  $\{S_n, I_n, J_n, K_n\}_{n=1, 2, \dots, N}$  を決定すると、全工程のスケジュールが完全に決められることになる。

次に、時刻  $t$  における機械  $M_i$  ( $i = 1, 2$ ) の状態を記述するのに

$$F_i(t) = \begin{cases} 0, & \text{機械 } M_i \text{ がアキの状態} \\ 1, & \text{機械へ取りつけの状態} \\ 2, & \text{機械加工の状態} \\ 3, & \text{機械から取りはずしの状態} \end{cases} \quad (5.5)$$

$$i = 1, 2$$



なる記号を用いる。

機械および貯蔵所は一定の位置に固定されており、第5.1図に示すごとく適当に原点を定め、機械  $M_i$  の位置座標を  $x_i$  ( $i=1, 2$ )、貯蔵所  $R_k$  の位置座標を  $z_k$  ( $k=1, 2, 3, 4$ ) と定める。運搬車的位置は、時間の関数であるから、時刻  $t$  における運搬車  $T_j$  の位置座標を  $y_j(t)$  とし、次のように定義する。

$$y_j(t) = \begin{cases} 0, & \text{運搬車が可動状態} \\ x_i, & \text{運搬車が機械 } M_i \text{ の前に固定されている状態} \\ z_k, & \text{運搬車が貯蔵所 } R_k \text{ の前に固定されている状態} \end{cases} \quad (5.6)$$

$j = 1, 2, 3$

与えられた問題は、 $\{S_n I_n J_n K_n\}_{n=1, 2, \dots, N}$  で示される順列を決定することであるが、今、この順列が何らかの方法で決められた時に、全工程をいかにして模擬し、機械の待ち時間をいかにして評価するかについて述べる。

A. item  $S_n$  の、機械への取りつけ開始時刻  $t_n$  の決定

a) 機械加工工程のみから決まる  $t_n$

$$\begin{aligned} t_1 &= \text{Max} \{0, a_{02} - (T_{Lmax} + a_{11})\} \\ t_2 &= \text{Max} \{a_{02} + a_{03}, t_1 + \sum_{j=1}^2 a_{1j} - (a_{21} + T_{Lmax})\} \\ t_n &= \text{Max} \left\{ t_{n-2} + \sum_{j=1}^3 a_{n-2,j}, t_{n-1} + \sum_{j=1}^2 a_{n-1,j} - (a_{n1} + T_{Lmax}) \right\} \\ &\triangleq \text{Max} \{A, B\} \quad n=3, 4, \dots, N \end{aligned} \quad (5.7)$$

から、運搬工程を無視した時の機械加工工程のタイムチャートを第5.2図のように描くことができる。ここで  $a_{02}, a_{03}$  は工程の初期状態を示すパラメータである。(5.7)式において  $\text{Max}\{A, B\}$  は、 $A, B$  のう

ち大きい方の値を意味する。

$A > B$  のときは、機械に待ち時間が生じないが、 $B > A$  のときには

$$u_n = B - A \quad (5.8)$$

$$= \left\{ t_{n-1} + \sum_{j=1}^2 a_{n-1,j} - (a_{n1} + T_{Lmax}) \right\} - \left( t_{n-2} + \sum_{j=1}^3 a_{n-2,j} \right)$$

の待ち時間が生じる。

b) 運搬車が使用中か否かの検討

item  $S_n$  が、機械からの取りはずしを完了する時刻を  $p_n$  とすると

$$p_n = t_n + \sum_{j=1}^3 a_{nj}. \quad (5.9)$$

また、運搬車は、item が機械から取りはずされる  $w$  分前に、機械の前で待機していなければならないものとする。もし、

$$y_j(p_n - w + 1) = 0 \quad (5.10)$$

ならば、所要の運搬車  $T_j$  はたゞちに使用可能である。もし、

$$y_j(t) \neq 0, \quad p_n - w + 1 \leq t \leq p_n - w + r_n \quad (5.11)$$

$$y_j(t) = 0, \quad t = p_n - w + r_n + 1$$

ならば  $r_n$  分の待ちが生じる。この時には、item  $S_n$  の、機械  $M_i$  への取りつけ開始時刻  $t_n$  を  $r_n$  分遅らせ

$$t_n + r_n \rightarrow t_n \quad (5.12)$$

$$p_n + r_n \rightarrow p_n$$

にしなければならない。

c) 隣接運搬車との衝突が起こらないか否かの検討

(i)  $J_n = 1$  (所要運搬車が  $T_1$ ) のとき

運搬車  $T_2$  との衝突が起こらないか否かのチェックを行う。

$$y_2(p_n - w + 1) = 0 \quad \text{または} \quad y_2(p_n - w + 1) > x_i \quad (5.13)$$

をみたしておれば、衝突は起こらないで、たゞちに所要の貯蔵所  $R_k$  へ運搬可能である。(5.13)式をみたしていないとき、すなわち

$$0 < y_2(t) \leq x_i, \quad p_n - w + 1 \leq t \leq p_n - w + \Delta_n \quad (5.14)$$

$$y_2(t) = 0 \quad \text{または} \quad y_2(t) > x_i, \quad t = p_n - w + \Delta_n + 1$$

ならば  $\Delta_n$  分の待ちが生じる。このときには、 $t_n$  をさらに  $\Delta_n$  分遅らせ

$$\begin{aligned} t_n + \Delta_n &\rightarrow t_n \\ p_n + \Delta_n &\rightarrow p_n \end{aligned} \quad (5.15)$$

にしなければならない。

(ii)  $J_n = 2$  (所要運搬車が  $T_2$ ) のとき

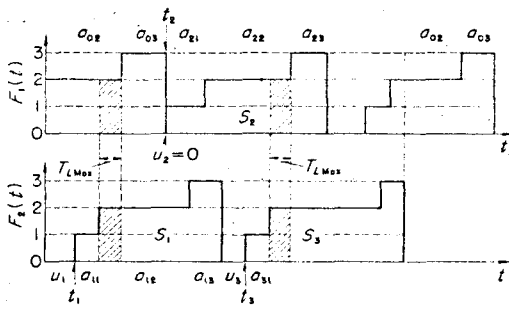
$J_n = 1$  のときと同様の手順により、 $T_1$  および  $T_3$  との衝突が起こらないか否かのチェックを行う。

(iii)  $J_n = 3$  (所要運搬車が  $T_3$ ) のとき

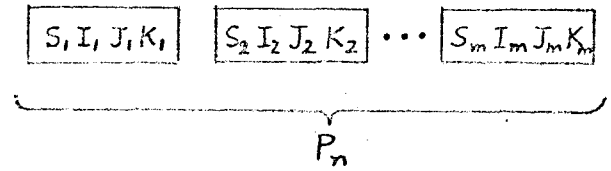
$J_n = 1$  のときと同様の手順により、 $T_2$  との衝突が起こらないか否かのチェックを行う。

## B. 各工程の時間割りつけ

上に示した手順を経て得られる  $t_n$  の最終値が、item  $S_n$  の、機械へのとりつけ開始時刻である。これをもとにして各工程の時間割りつけを行い、運搬、格納工程をも含めた全工程のタイムチャートを描く。すなわち、



第5.2図 機械加工工程の  
タイムチャート



a) 基本形



b) 基本形の合成

第5.3図 基本形(サブシステム)と全システムの関係

$$F_i(t) = \begin{cases} 1, & t_n \leq t \leq t_n + a_{n1} \\ 2, & t_n + a_{n1} + 1 \leq t \leq t_n + \sum_{j=1}^2 a_{nj} \\ 3, & t_n + \sum_{j=1}^2 a_{nj} + 1 \leq t \leq t_n + \sum_{j=1}^3 a_{nj} \end{cases} \quad (5.16)$$

$$y_j(t) = \begin{cases} x_i, & t_n + \sum_{j=1}^3 a_{nj} - w + 1 \leq t \leq t_n + \sum_{j=1}^3 a_{nj} \\ z_k, & t_n + \sum_{j=1}^3 a_{nj} + 1 \leq t \leq t_n + \sum_{j=1}^4 a_{nj} \end{cases} \quad (5.17)$$

で与えられる。

以上を示す工程のシミュレーションを全 item について実行する。この結果、機械がアキになっている待ち時間の総和  $T_w$  は

$$T_w = \sum_{n=1}^N (u_n + r_n + \delta_n) \quad (5.18)$$

で与えられる。

### 5.3 逐次スケジュールを改良する方式

item の総数  $N$  が小さいときには、全順列 ( $N!$  個) に対して、シミュレーションにより待ち時間を計算し、待ち時間が最小になる順列をさがし求めることもできるが、 $N$  の増大と共に困難となる。 $N$  が大きい時には、これをまずいくつかのサブシステムに分割する。サブシステム内で最適化

をほかつたものを基本形と呼ぶ。あらかじめ幾種類かの基本形を用意しておき、与えられた生産命令に対して、これらの基本形の組合せを考慮することにより、全体のスケジュールを逐次改良して行くという手法を採用する。基本形と全システムとの関係を第5.3図に示す。

基本形  $P_n$  を

$$P_n = \{S_k I_k J_k K_k\}_{k=1,2,\dots,m} \quad (5.19)$$

$$n=1, 2, 3, \dots, 2^M$$

たゞし

$$N = 2^M \cdot m \quad (5.20)$$

であらわす。また全システムを  $\{P_n\}_{n=1,2,\dots,2^M}$  であらわす。

### 5.3.1 基本形の作成

順列  $\{S_n I_n J_n K_n\}_{n=1,2,\dots,m}$  を決定すると、 $m$ 個の要素から成る基本形が得られるが、その決定の方法は次の手順に従う。

a) item  $S_n$  の順列  $\{S_n\}_{n=1,2,\dots,m}$  の決定

$m!$ 通りについて検討する。

b) 機械の選択

$$I_n = \begin{cases} 1, & n \text{ が奇数} \\ 2, & n \text{ が偶数} \end{cases} \quad n=1, 2, \dots, m \quad (5.21)$$

により一意的に定める。すなわち、これは2台の機械  $M_1, M_2$  を交互に使用することを意味する。

c) 運搬車の選択

運搬車は、同一のもの連続して使用しないので

$$J_n = 1, 2 \text{ または } 3, \quad n=1, 2, \dots, m \quad (5.22)$$

から選ぶ。このときには  $3 \times 2^{m-1}$  通りの使用法が考えられる。

d) 貯蔵所の選択

$$K_n = \begin{cases} 1, & J_n = 1 \\ 2, & J_n = 2, J_{n-1} = 3 \\ 3, & J_n = 2, J_{n-1} = 1 \\ 4, & J_n = 3 \end{cases} \quad (5.23)$$

により定める。すなわち、使用する運搬車が定められると貯蔵所は一意的に決まる。

a) ~ d) の手順により、順列  $\{S_n I_n J_n K_n\}_{n=1,2,\dots,m}$  を逐次発生させ、

$$T_w = \sum_{n=1}^m (u_n + r_n + d_n) \rightarrow \text{Min}$$

なる順列を求める。

5.3.2 基本形の配列法の決定

基本形を適当に組合せ、合成することにより、与えられた生産命令をこなして行く。 $2^M$ 個の基本形の合成により全システムを構成するが、 $2^M!$ 個のすべての順列を考慮しないで、計算機による Sorting 法の応用として Merging 法<sup>43</sup>を少し変形して、基本形の配列を決定する。したがって、この結果得られるスケジュールが、考え得るすべてのスケジュール中の最適なものになっていることは稀である。

Sorting の方法は、以下の手順による。今、最初に与えられる順列を

$$\{P_n\}_{n=1,2,\dots,2^M} = \{P_1, P_2, \dots, P_{2^M}\}$$

とする。まず  $(P_1, P_2), (P_3, P_4), \dots, (P_{2^M-1}, P_{2^M})$  のごとく、相隣り合う pair を抽出し

$$\left. \begin{matrix} (P_1, P_2) \\ (P_2, P_1) \end{matrix} \right\}, \left. \begin{matrix} (P_3, P_4) \\ (P_4, P_3) \end{matrix} \right\}, \dots, \left. \begin{matrix} (P_{2^M-1}, P_{2^M}) \\ (P_{2^M}, P_{2^M-1}) \end{matrix} \right\}$$

のように、pairの中を入れかえてみる。おのおのについてシミュレーションにより待ち時間を計算し、待ち時間の小さい pair を採用する。例えば、 $\{P_1, P_2, P_4, P_3, \dots, P_{2M-3}, P_{2M-2}, P_{2M}, P_{2M-1}\}$  となるとする。

次に、各 pair を固定し、quartet のシリーズを作り、入れかえてみる。すなわち

$$\left[ \begin{array}{l} \{(P_1, P_2), (P_4, P_3)\} \\ \{(P_4, P_3), (P_1, P_2)\} \end{array} \right], \dots, \left[ \begin{array}{l} \{(P_{2M-3}, P_{2M-2}), (P_{2M}, P_{2M-1})\} \\ \{(P_{2M}, P_{2M-1}), (P_{2M-3}, P_{2M-2})\} \end{array} \right]$$

について、待ち時間の小さい quartet を採用する。

以下同様にしてこの手順を繰返す、最後には

$$\left[ \begin{array}{l} \{(\dots, P_4, P_3, P_1, P_2, \dots), (\dots, P_{2M-3}, P_{2M-2}, P_{2M}, P_{2M-1}, \dots)\} \\ \{(\dots, P_{2M-3}, P_{2M-2}, P_{2M}, P_{2M-1}, \dots), (\dots, P_4, P_3, P_1, P_2, \dots)\} \end{array} \right]$$

を比較し、待ち時間の小さいシーケンスがボめる順列である。

#### 5.4 シミュレーションの結果と検討

6種類の item および各 item の各工程における所要時間が第5.1表に示すように与えられたものとする。S<sub>1</sub> を 50%、S<sub>2</sub>~S<sub>6</sub> を残りの 50% の割合で生産するものとし、S<sub>2</sub>~S<sub>6</sub> の配分は日毎に異なるものとする。基本形として、第5.2表に示すものを採用するものとする。

第5.1表 タイムテーブル

S <sub>n</sub>	a <sub>n1</sub>	a <sub>n2</sub>	a <sub>n3</sub>	a <sub>n4</sub>
	-	7	5	15
S <sub>1</sub>	2	10	5	15
S <sub>2</sub>	2	12	5	20
S <sub>3</sub>	2	14	5	25
S <sub>4</sub>	2	16	5	30
S <sub>5</sub>	2	18	5	35
S <sub>6</sub>	2	20	5	40

$$T_{Lmax} = 5, \quad w = 2$$

各基本形は、5.3.1で述べた手法によりボめたものである。表中に示すごとく、各基本形 P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub> 内での待ち時間は、おのおの 8, 10,

第 5.2 表 基本形の例

$P_1$	$P_2$	$P_3$	$P_4$
S...1 3 2 1	S...1 1 3 4	S...1 1 4 5	S...1 6 5 1
I...1 2 1 2	I...1 2 1 2	I...1 2 1 2	I...1 2 1 2
J...1 3 1 2	J...1 2 3 2	J...1 2 1 3	J...1 3 1 2
K...1 4 1 3	K...1 2 4 3	K...1 3 1 4	K...1 4 1 3
$T_w = 8$	$T_w = 10$	$T_w = 28$	$T_w = 44$

28, 44分である。基本形  $P_1 \sim P_4$  を各々一回ずつ使用して、基本形の配列を行う。  $M=2$  の場合について、シミュレーションを行った結果を第 5.3 表に示す。この結果  $\{P_1, P_2, P_3, P_4\}$  の初期配列から出発して、最後に得られた基本形の配列は  $\{P_1, P_2, P_4, P_3\}$  である。また、この時の待ち時間の合計は 100 分であり、

$100 - (8 + 10 + 28 + 44) = 10$  分の待ち時間は、各基本形のつなぎめで生じたものである。

すべての組合せを考慮して、この問題を計算した結果  $2^2! = 24$

通りのスケジュールに対して  $T_w < 100$  となったのは 4 通りであった。このうち、 $T_w = 90$  および 96 のスケジュールが各々 2 通りずつであり他の 20 通りのスケジュールは、すべて  $T_w \geq 100$  という結果を得た。

本方式における計算回数  $f_a$  は、基本形の数を  $2^M$  とする

$$f_a = 2^M + 2^{M-2} + \dots + 2$$

一方、すべての組合せを考慮したときの計算回数  $f_b$  は

$$f_b = 2^M!$$

となり、 $M$  の増大と共に天文学的数字にのぼる。  $f_a$  と  $f_b$  を比較したものを第 5.4 表に示す。

第 5.3 表 シミュレーションの結果

$i$	$P$	$T_w$
1	$(P_1, P_2)$	18 ←
	$(P_2, P_1)$	34
	$(P_3, P_4)$	78
	$(P_4, P_3)$	72 ←
2	$\{(P_1, P_2), (P_4, P_3)\}$	100 ←
	$\{(P_4, P_3), (P_1, P_2)\}$	102



本方式によって真の最適解(待ち時間最小のスケジュール)が得られることは稀であるが、計算量の点から考えて、itemの数が大、かつモデルが複雑な現実の問題に適用すると効果的である。

第5.4表  $f_a$  と  $f_b$  の比較

M	$f_a$	$f_b$
1	2	2
2	6	$2.4 \times 10$
3	14	$4.0 \times 10^4$
4	30	$2.1 \times 10^{13}$
5	62	$2.6 \times 10^{35}$
⋮	⋮	⋮

### 5.5 むすび

機械加工工程を想定して、現実問題に適用し得るという観点から、生産スケジュールリングを行う一方式を検討した。

E. S. Page が Chain Monte Carlo 法を発表している。<sup>34</sup> Page の方法では、全 item を乱数によってグループわけし、各グループ内の item の配列は考えないで、グループ間の配列に関して順列を考慮して解を求めている。一方、この章で述べている手法は、乱数によってグループわけするのではなく、システムの大きさに応じて全システムを一定の item 数から成るサブシステムに分割して、最適化された基本形を作成し、この基本形の配列を考慮することによって解を求めている。

## 第6章 双対原理による大規模生産システムのスケジューリング

### 6.1 まえがき

本章では、時間的に serial に関連したいくつかのサブプロジェクトの結合から成る大規模な生産工程のスケジューリングの問題を取り扱う。各サブプロジェクトを完成させるのに必要な仕事量および納期が与えられているものとする。スケジューリングの問題は、全体のプロジェクトを完成させるためのコスト、すなわち必要な仕事量（人・月）にかかるコストと、或るプロジェクトから他のプロジェクトへ人、資源等と移動する時に生じる損失にかかるコストの和を最小にする問題として与えられるものとする。

問題の数式モデルは、多変数の二次計画法、すなわち線形の制限条件のもとで、二次形式の評価関数を最小にする問題として定式化できる。これを基本問題と呼ぶ。次に、できるだけ少ない計算量で最適スケジュールを得る方法を求めるために、Fenchelの双対原理<sup>29,30</sup>を適用し、基本問題の双対問題を定式化する。双対問題の数値解を求めるのに、もとの問題の階層構造を利用して、小規模な連立方程式を多数回繰返して解く問題に帰着して、最適スケジュールを求める。

### 6.2 システムの数式モデル

生産の全工程 $P$ は、 $P_1, P_2, \dots, P_N$ から成る $N$ 個のサブプロジェクトで構成され、プロジェクト $P_i$ と $P_{i+1}$  ( $i=1, 2, \dots, N-1$ )は時間的に serial に繋がっているものとする。すなわち、 $P_i$ の仕事が完了すると同時に $P_{i+1}$ の仕事を開始する。

プロジェクト $P_i$ を完成する納期を $m_i$  (ヶ月)とし、 $k$ ヶ月目の最初にプ

プロジェクト  $P_i$  に投入 (または削減) される人間, 資源等の量を  $x_i(k)$  とする。また, プロジェクト  $P_i$  を完成させるのに必要な仕事量 (人・月) を  $W_i$  とすると,  $n_i, x_i(k), W_i$  の間には

$$\sum_{k=1}^{n_i} \sum_{j=1}^k x_i(j) = W_i, \quad i=1, 2, \dots, N \quad (6.1)$$

の関係がある。  $n_i$  ヶ月経過後には, プロジェクト  $P_i$  に人間, 資源等を残せないで,

$$\sum_{k=1}^{n_i+1} x_i(k) = 0, \quad i=1, 2, \dots, N \quad (6.2)$$

を満たさなければならない。たゞ  $i, x_i(k)$  が正の時は人間, 資源等をプロジェクトへ投入することを意味し, 負の時は削減することを意味する。  $-x_i(n_i+1)$  は  $n_i$  ヶ月の終りにプロジェクト  $P_i$  から削減する人間, 資源等の量をあらわし, この人間, 資源等を, 次のプロジェクト  $P_{i+1}$  を開始する時にそのまま投入するものとする,

$$x_{i+1}(1) = -x_i(n_i+1), \quad i=1, 2, \dots, N-1 \quad (6.3)$$

を満たさなければならない。これは, 各プロジェクトが時間的に serial につながっていることによる制限条件である。

プロジェクト  $P_i$  を完成するためにかかるコストは,  $P_i$  を完成するのに必要な仕事量 (人・月) にかかるコスト, および人間, 資源等を或るプロジェクトから他のプロジェクトへ移しかえる時に生じる損失によるコストの和であらわされるものとする。今,  $x_i(k)$  なる人間, 資源等を動かすことによって生じる損失は  $x_i^2(k)$  に比例するものと考え。何故ならば, 一時に多くの人間, 資源等を或るプロジェクトへ投入すればする程, その中で非生産的な役割をか果せない人間, 資源等の割合が増す可能性が増大するからである。以上からコスト関数は,

$$J = \sum_{i=1}^N \left\{ \frac{1}{2} \sum_{k=1}^{n_i+1} \phi_i(k) x_i^2(k) + \sum_{k=1}^{n_i} \gamma_i(k) \sum_{j=1}^k x_i(j) \right\} \quad (6.4)$$

であらわせる。ただし、 $\phi_i(k)$ 、 $\gamma_i(k)$  は各々の評価量にかゝる重みを示す。

スケジューリングの問題は、(6.1)~(6.3)式で示される制限条件のもとで、(6.4)式のコスト関数を最小にする  $x_i(k)$  ( $k=1, 2, \dots, n_i+1$ ,  $i=1, 2, \dots, N$ ) を求める問題として定式化される。これは、線形の等式制限条件のもとで、二次形式の評価関数を最小にする二次計画法の問題である。ここで問題になるのは、変数の数が  $\sum_{i=1}^N (n_i+1)$ 、制限条件の数が  $3N-1$  で、 $N$  および  $n_i$  の増大と共に問題の次元が急激に増すという点である。

(6.1)~(6.4)式で示したスケジューリング問題は、次に示す二次計画法の問題<sup>15</sup>として一般化される。

$$A_i x_i = b_i, \quad i = 1, 2, \dots, N \quad (6.5)$$

$$\sum_{i=1}^N C_i x_i = d \quad (6.6)$$

のもとで

$$\text{Minimize}_{x_i} \sum_{i=1}^N \left\{ \frac{1}{2} x_i' \Phi_i x_i + \Gamma_i' x_i \right\} \quad (6.7)$$

ただし

$x_i$  :  $n_i$  次元列ベクトル

$A_i$  :  $m_i \times n_i$  係数行列,  $m_i < n_i$ ,  $\text{Rank } A_i = m_i$

$b_i$  :  $m_i$  次元係数列ベクトル

$C_i$  :  $M \times n_i$  係数行列

$d$  :  $M$  次元係数列ベクトル

$\Phi_i$  :  $n_i \times n_i$  正定値対称行列

$\Gamma_i$  :  $n_i$  次元係数列ベクトル

$x'$  :  $x$  の転置

第 6.1 表に (6.1) ~ (6.4) 式によってモデル化したスケジューリング問題と, (6.5) ~ (6.7) 式に一般化した二次計画法の問題の関係を示す。

第 6.1 表 二次計画法の問題とスケジューリング問題の関係

二次計画法の問題	スケジューリング問題
$N$	$N$
$n_i$	$n_i + 1$
$x_i$	$\begin{bmatrix} x_i(1) \\ x_i(2) \\ \vdots \\ x_i(n_i+1) \end{bmatrix}$
$\Phi_i$	$\begin{bmatrix} \phi_i(1) & 0 & \dots & 0 \\ 0 & \phi_i(2) & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \dots & 0 & \phi_i(n_i+1) \end{bmatrix}$
$\Gamma_i$	$\begin{bmatrix} \delta_i(1) + \delta_i(2) + \dots + \delta_i(n_i) \\ \delta_i(2) + \dots + \delta_i(n_i) \\ \vdots \\ \delta_i(n_i) \\ 0 \end{bmatrix}$
$m_i$	2
$A_i$	$\begin{bmatrix} n_i & n_i - 1 & \dots & 1 & 0 \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix}$
$b_i$	$\begin{bmatrix} W_i \\ 0 \end{bmatrix}$
$M$	$N - 1$
$C_1$	$\begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & \vdots & 0 \\ \vdots & \dots & \vdots & 0 \end{bmatrix}$
$C_2$	$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 \\ 0 & \dots & \vdots & 0 \\ \vdots & \dots & \vdots & 0 \end{bmatrix}$
$\vdots$	$\vdots$
$C_{N-1}$	$\begin{bmatrix} 0 & \dots & 0 \\ \vdots & \dots & \vdots \\ 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}$
$d$	$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

### 6.3 Fenchel の双対原理<sup>29, 30</sup>

本節では、第6章の理論的背景になっている Fenchel の双対原理を簡単に説明しておく。

有界な線形汎関数:  $X$  をノルム空間<sup>27</sup>とし、汎関数  $x^*: X \rightarrow \mathbb{R}$  のうち

$$x^*(\alpha x_1 + \beta x_2) = \alpha x^*(x_1) + \beta x^*(x_2), \quad \forall x_1, x_2 \in X, \quad \forall \alpha, \beta \text{ スカラー}$$

を満たし

$$|x^*(x)| \leq M \cdot \|x\|, \quad \forall x$$

を満たす定数  $M$  が存在する時、 $x^*$  を有界な線形汎関数という。

双対空間: ノルム空間  $X$  の上に定義されるすべての有界な線形汎関数  $x^*$  の集合を  $X^*$  であらわし、これを  $X$  の双対空間という。ただし  $x^* \in X^*$  のノルムを

$$\|x^*\| \triangleq \sup_{\|x\| \leq 1} |x^*(x)|$$

で定義する。

共役凸汎関数:  $X$  をノルム空間、 $C$  をその凸部分空間、 $f: C \rightarrow \mathbb{R}$  を凸汎関数とし、共役集合  $C^*$  および共役凸汎関数  $f^*(x^*)$  を次のように定義する。

$$C^* \triangleq \{x^* \in X^* : \sup_{x \in C} [x^*(x) - f(x)] < \infty\} \quad (6.8)$$

$$f^*(x^*) \triangleq \sup_{x \in C} [x^*(x) - f(x)] \quad (f^*: C^* \rightarrow \mathbb{R})$$

共役凹汎関数:  $X$  をノルム空間、 $D$  をその凸部分空間、 $g: D \rightarrow \mathbb{R}$  を凹汎関数とし、共役集合  $D^*$  および共役凹汎関数  $g^*(x^*)$  を次のように定義する:

$$D^* \triangleq \{x^* \in X^* : \inf_{x \in D} [x^*(x) - g(x)] > -\infty\} \quad (6.9)$$

$$g^*(x^*) \triangleq \inf_{x \in D} [x^*(x) - g(x)] \quad (g^*: D^* \rightarrow \mathbb{R})$$

Fenchel の双対原理:  $X$  をノルム空間、 $C, D$  をその凸部分空間とし、 $C \cap D$  が内点をもち、 $f: C \rightarrow \mathbb{R}$ ,  $g: D \rightarrow \mathbb{R}$  を各々凸および凹汎関数とする。もし  $\mu = \inf_{x \in C \cap D} \{f(x) - g(x)\}$  が有限であるとすると

$$\mu = \inf_{x \in C \cap D} \{f(x) - g(x)\} = \max_{x^* \in C^* \cap D^*} \{g^*(x^*) - f^*(x^*)\} \quad (6.10)$$

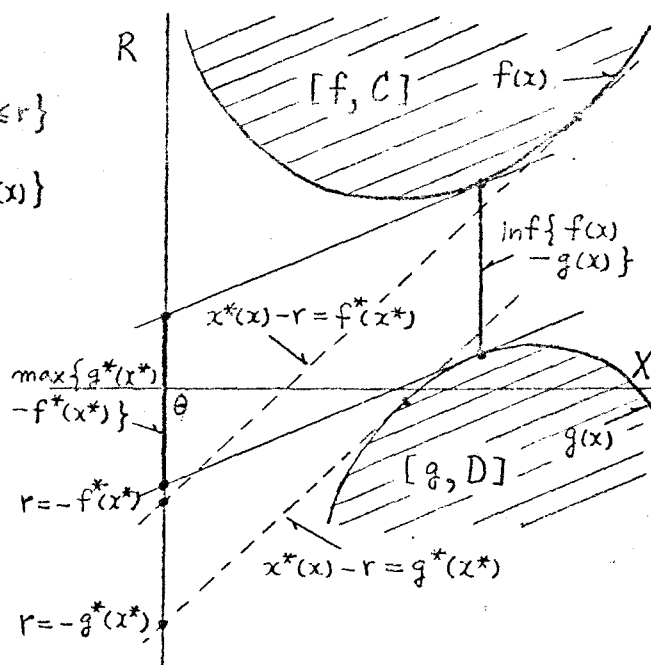
が成り立つ。

[幾何学的解釈]

$$[f, C] \triangleq \{(r, x) \in R \times X : x \in C, f(x) \leq r\}$$

$$[g, D] \triangleq \{(r, x) \in R \times X : x \in D, r \leq g(x)\}$$

で定義すると, (6.10)式の左辺は凸集合  $[f, C]$  と  $[g, D]$  の間の垂直距離の最小値をあらわし, 右辺は  $[f, C]$  と  $[g, D]$  を分離する二つの平行な超平面の間の垂直距離の最大値をあらわす。(第6.1図参照) (6.10)式の右辺を満たす  $x^*$  を求める問題を, 左辺を満たす  $x$  を求める問題の双対問題という。



第6.1図 Fenchelの双対原理の幾何学的な解釈

6.4 双対問題の定式化

Fenchelの双対原理を用いて, (6.5) ~ (6.7)式であらわされる基本問題の双対問題を定式化する。

$$\begin{aligned} f(x) &= \sum_{i=1}^N \left\{ \frac{1}{2} x_i^T \Xi_i x_i + \Gamma_i' x_i \right\} \\ g(x) &= 0 \\ C &= E_n \end{aligned} \tag{6.11}$$

$$D = \left\{ x \in E_n : A_i x_i = b_i, i = 1, 2, \dots, N, \sum_{i=1}^N C_i x_i = d \right\}$$

において, 共役集合および共役汎関数を求める。たゞ  $E_n$  は  $n$ 次元ユークリッド空間を示し,

$$n = \sum_{i=1}^N m_i$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

をあらわす。

$E_n$  の双対空間は  $E_n$  であるので,  $x^*$  を  $y$  であらわす。ただし

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

で,  $y_i$  は  $m_i$  次元列ベクトルである。

$$g^*(y) = \inf_{x_i} \left\{ y_1' x_1 + y_2' x_2 + \cdots + y_N' x_N : A_i x_i = b_i, i=1, 2, \dots, N \right. \\ \left. \sum_{i=1}^N C_i x_i = d \right\} \quad (6.12)$$

が有限であるためには,

$$y_i = A_i' \lambda_i + C_i' \lambda_{N+1}, \quad i=1, 2, \dots, N \quad (6.13)$$

でなければならない。ただし,  $\lambda_i$  は  $m_i$  次元未知列ベクトル,  $\lambda_{N+1}$  は  $M$  次元未知列ベクトルである。このとき

$$g^*(y) = g^*(\lambda) = \lambda_1' b_1 + \lambda_2' b_2 + \cdots + \lambda_N' b_N + \lambda_{N+1}' d \quad (6.14)$$

となる。

一方,

$$f^*(y) = \sup_x \{ y' x - f(x) \} \\ = \sup_{x_i} \sum_{i=1}^N \left\{ -\frac{1}{2} x_i' \Phi_i x_i - (\Gamma_i' - y_i') x_i \right\} \quad (6.15)$$

sup は

$$x_i = -\Phi_i^{-1} (\Gamma_i - y_i) \\ = -\Phi_i^{-1} (\Gamma_i - A_i' \lambda_i - C_i' \lambda_{N+1}), \quad i=1, 2, \dots, N \quad (6.16)$$

において得られる。(6.13)(6.16) 式を (6.15) 式に代入して整理すると;

$$f^*(y) = f^*(\lambda) = \sum_{i=1}^N \left\{ \frac{1}{2} (\lambda_i' A_i + \lambda_{N+1}' C_i) \Phi_i^{-1} (A_i' \lambda_i + C_i' \lambda_{N+1}) \right. \\ \left. - \Gamma_i' \Phi_i^{-1} (A_i' \lambda_i + C_i' \lambda_{N+1}) + \frac{1}{2} \Gamma_i' \Phi_i^{-1} \Gamma_i \right\} \quad (6.17)$$

となる。(6.14)(6.17) 式から

$$g^*(y) - f^*(y) = g^*(\lambda) - f^*(\lambda) \\ = \sum_{i=1}^N \left\{ -\frac{1}{2} (\lambda_i' A_i + \lambda_{N+1}' C_i) \Phi_i^{-1} (A_i' \lambda_i + C_i' \lambda_{N+1}) \right. \\ \left. + (b_i' + \Gamma_i' \Phi_i^{-1} A_i') \lambda_i + (d' + \Gamma_i' \Phi_i^{-1} C_i') \lambda_{N+1} \right\} + \text{const.} \quad (6.18)$$



を得る。結局、双対問題は  $\lambda_1, \lambda_2, \dots, \lambda_{N+1}$  に関する二次形式 (6.18) 式を最大にする問題となり、これは次の連立一次方程式を解く問題と等価である。

$$A_i \Phi_i^{-1} A_i' \lambda_i + A_i \Phi_i^{-1} C_i' \lambda_{N+1} = b_i + A_i \Phi_i^{-1} \Gamma_i$$

$$i = 1, 2, \dots, N \quad (6.19)$$

$$\sum_{i=1}^N C_i \Phi_i^{-1} A_i' \lambda_i + \left( \sum_{i=1}^N C_i \Phi_i^{-1} C_i' \right) \lambda_{N+1} = d + \sum_{i=1}^N C_i \Phi_i^{-1} \Gamma_i$$

(6.19) 式の解が双対問題の解であり、これを  $\lambda_1^0, \lambda_2^0, \dots, \lambda_N^0$  とすると、基本問題の解  $x_1^0, x_2^0, \dots, x_N^0$  は (6.16) 式より

$$x_i^0 = -\Phi_i^{-1} (\Gamma_i - A_i' \lambda_i^0 - C_i' \lambda_{N+1}^0)$$

$$i = 1, 2, \dots, N \quad (6.20)$$

で与えられる。

## 6.5 双対問題を解くアルゴリズム

双対問題は (6.19) 式に示す  $\lambda_1, \lambda_2, \dots, \lambda_{N+1}$  に関する  $m (= \sum_{i=1}^N m_i + M)$  元連立一次方程式を解く問題に帰着された。(6.19) 式を数値的に解くアルゴリズムとして次の手順を採用する。

Step 1  $\lambda_{N+1}$  の初期値を  $\lambda_{N+1}^1$  に設定する。

Step 2  $N$  個の  $\lambda_i$  に関する  $m_i$  次元連立方程式を解く

$$A_i \Phi_i^{-1} A_i' \lambda_i = b_i + A_i \Phi_i^{-1} \Gamma_i - A_i \Phi_i^{-1} C_i' \lambda_{N+1}^k \quad (6.21)$$

$$i = 1, 2, \dots, N$$

この数値解を  $\lambda_i^k$  ( $i = 1, 2, \dots, N$ ) とする。肩書き文字  $k$  は  $k$  回目の繰返しを意味する。

Step 3 設定値  $\lambda_{N+1}^k$  および Step 2 で求めた  $\lambda_i^k$  より

$$h(\lambda^k) \triangleq \left\| \sum_{i=1}^N C_i \Phi_i^{-1} A_i' \lambda_i^k + \left( \sum_{i=1}^N C_i \Phi_i^{-1} C_i' \right) \lambda_{N+1}^k - d - \sum_{i=1}^N C_i \Phi_i^{-1} \Gamma_i \right\| < \varepsilon \quad (6.22)$$

をみたすか否かを調べる。たゞ  $\| \cdot \|$  は通常のユークリッドノルムを意味する。また  $\varepsilon$  は許容誤差を示す。(6.22)式が満たれていれば繰返し計算を停止する。満たれていなければ Step 4 に従って  $\lambda_{N+1}$  を  $\lambda_{N+1}^{k+1}$  に設定しなおし、 $k+1 \rightarrow k$  にして Step 2 へもどる。

#### Step 4

$$\lambda_{N+1}^{k+1} = \lambda_{N+1}^k + \omega^k \left\{ \sum_{i=1}^N C_i \Xi_i^{-1} A_i' \lambda_i^k + \left( \sum_{i=1}^N C_i \Xi_i^{-1} C_i' \right) \lambda_{N+1}^k - d - \sum_{i=1}^N C_i \Xi_i^{-1} \Gamma_i \right\} \quad (6.23)$$

たゞし、 $\omega^k$  は適当な定数であり、 $\omega^k$  のきめ方は後で述べる。

Step 1 から Step 4 に示したアルゴリズムは (6.19) 式の形の特殊性を利用して作ったものであり、 $m (= \sum_{i=1}^N m_i + M)$  次元連立方程式を解く問題を、 $M$  個のパラメータを介して  $N$  個の  $m_i$  次元連立方程式を繰り返し解く問題におきかえている。これは、もとの問題が (6.5) ~ (6.7) 式に示すような階層構造とないているために可能となったものである。この方法には、Gauss-Seidel 法の<sup>8</sup>思想がおりこまれており、 $m$  次元連立方程式の係数行列に含まれている多くの 0 要素が有効に生かされている。

#### Step 4 中 $\omega^k$ のきめ方

まず最初に、小さい  $\omega^k$  の値に対して

$$f(\lambda^{k+1}) < f(\lambda^k) \quad (6.24)$$

となる様に、もし必要ならば (6.23) 式の右 2 項に  $-1$  を掛けておく。理想的には、 $f(\lambda^{k+1})$  が  $\omega^k$  に関して最小となる様にステップ中  $\omega^k$  を選択するのが良いが、 $f(\lambda^{k+1})$  を最小にする  $\omega^k$  を見つけることは容易でない。また、Goldstein によれば<sup>14</sup>、 $\omega^k$  を各ステップ  $k$  ごとにかえずに、 $0 < \omega \leq \delta$ ,  $\delta > 0$  を満たす一定値  $\omega$  を採用することによって、 $\lambda$  に関する二次形式  $f(\lambda)$  は、 $f(\lambda^{k+1}) \leq \mu f(\lambda^k)$ ,  $0 < \mu < 1$  に従って幾何級数的に収束するような  $\delta$  が存在するはずである。しかし、この  $\delta$  をみつけることも容易でない。そこで、各繰返しにおける  $\omega^k$  の値として、最初大きめの値  $\omega$  を選び、(6.24) 式を満たすか否かを調べる。もし満たさないならば、 $\omega$  を  $\frac{1}{2}\omega$  に減らしてふた

たび (6.24) 式を満たすか否かを調べる。はじめに (6.24) 式を満たす  $\omega$  を  $\omega^k$  とし採用する。

## 6.6 数値例と検討

双対原理を二つのスケジューリング問題に適用して得た結果を示す。

例1: 最も簡単な場合、すなわち全体が一つのプロジェクトで構成されていて ( $N=1$ )、プロジェクトの納期が 12ヶ月 ( $n=12$ )、仕事量が 200人・月 ( $W=200$ )、

$$\phi(k) = 1.0, \quad k=1, 2, \dots, 13$$

$$\delta(k) = 1.0, \quad k=1, 2, \dots, 12$$

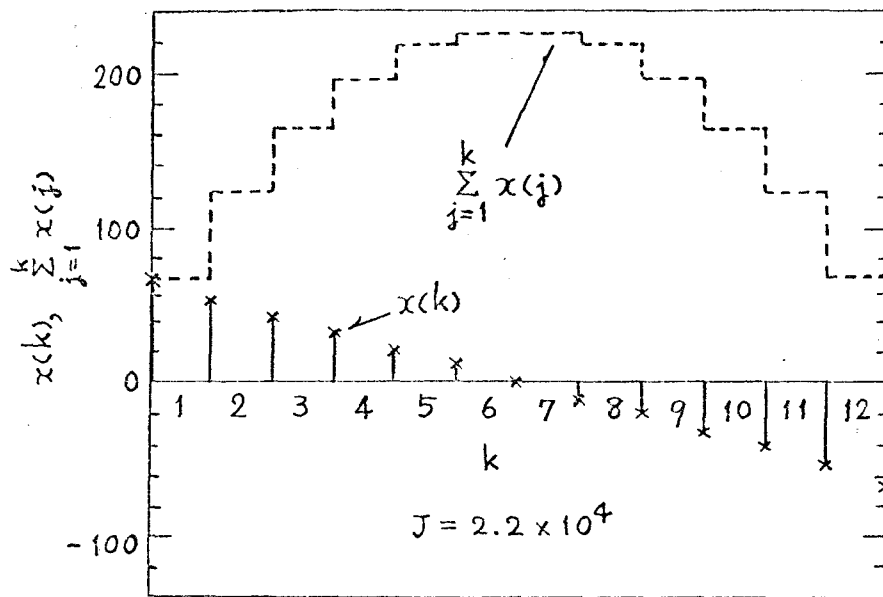
で与えられる場合の最適スケジュールを第6.2図 a) に示す。このプロジェクトへ投入する人間・資源等  $\alpha$  量を  $k=1, 2, \dots, 6$  まで  $\Gamma$  に減らし、 $k=7, 8, \dots, 13$  においてこのプロジェクトから削減する人間・資源等  $\alpha$  量を  $\Gamma$  に増やすスケジュールが、コスト最小を与えることがわかる。第6.2図 b) には、通常常識的に行うスケジュールを示す。ここでは  $W/n=16.7$  の人間・資源等を最初に投入し、12ヶ月経過後 16.7 の人間・資源等を削減する。a) と b) を比較すればわかるように、a) に示す最適スケジュールを採用することによって、コストを通常  $\alpha$  スケジュールの約半分に減らすことができる。

例2: 全体のプロジェクトが 10 個のサブプロジェクトから成り、これらは時間的に serial につながっているものとする。各プロジェクトの納期  $n_i$ 、および各プロジェクトを完成するのに必要な仕事量  $W_i$  は、第6.2表のように与えられているものとする。また、

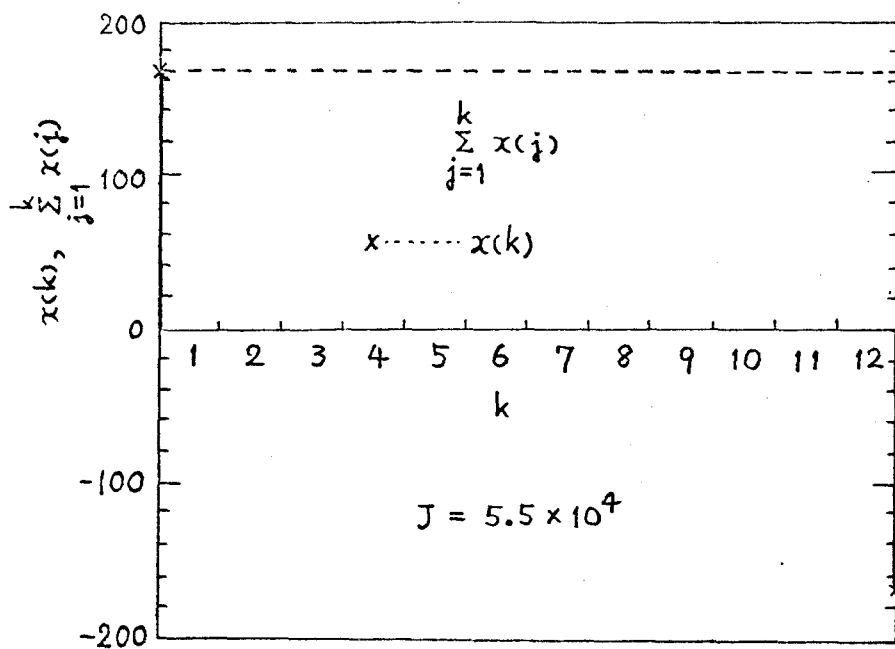
$$\phi_i(k) = 1.0, \quad k=1, 2, \dots, 7$$

$$\delta_i(k) = 1.0, \quad k=1, 2, \dots, 6$$

$$i = 1, 2, \dots, 10$$



a) 最適スケジュール



b) 常識によるスケジュール

第6.2図 単一プロジェクトのスケジュール

とする。これはサブプロジェクトが半期(6ヶ月)毎に構成され、下半期は上半期の2倍の仕事量があり、毎年2割ずつ生産高を増す大プロジェクトの5ヶ年にわたるスケジュールリングを意味する。

この問題は、70変数、29個の制限条件という多変数の二次計画法の問題であるが、双対原理およびもとの問題の階層構造を有効に利用することによって、10個の2次元連立方程式を繰返して解く問題におきかえて解いている。この問題に対する最適スケジュールを第6.3図に示す。また、第6.4図には、 $\lambda_{II}$ の収束の様子を示す。9つの要素から成るベクトル $\lambda_{II}$ の初期値はすべて0に送った。

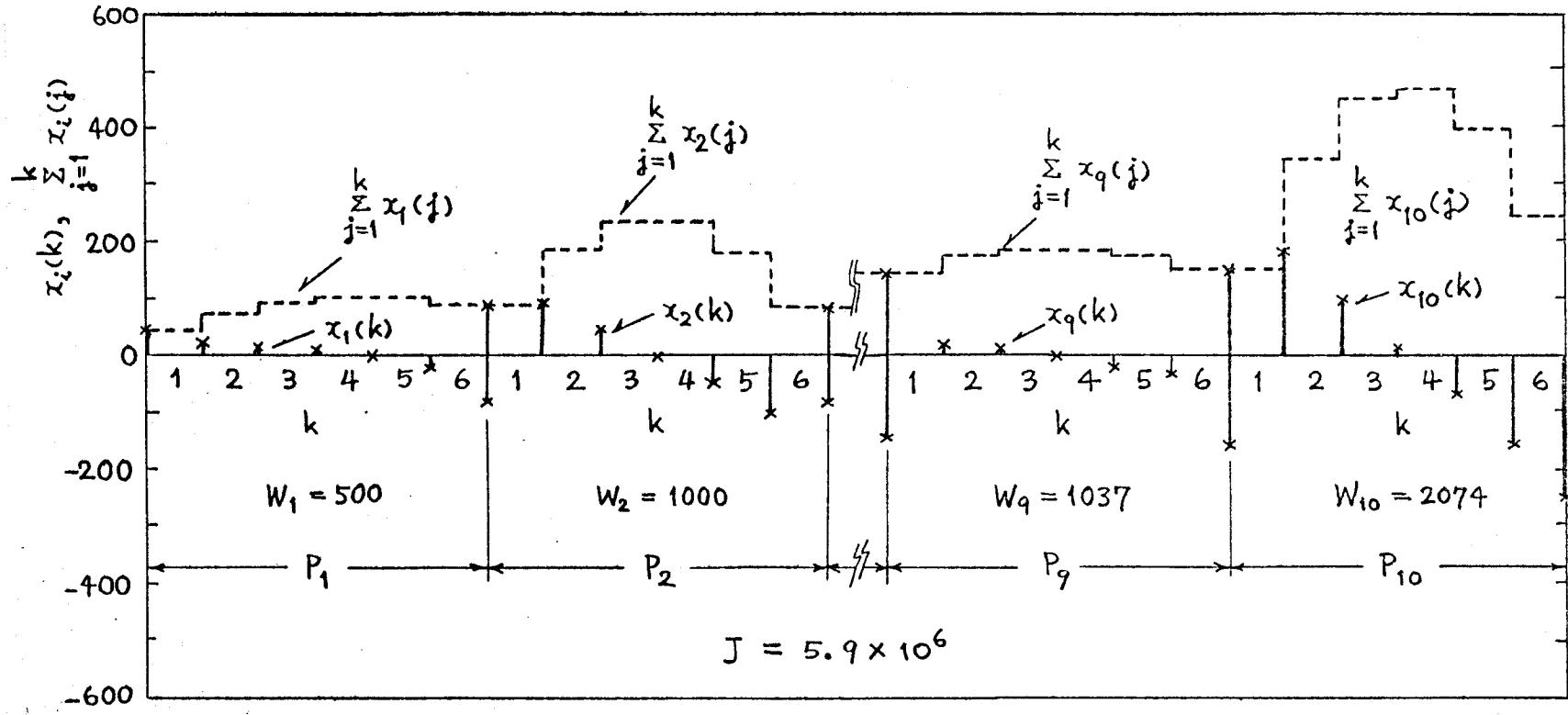
例1にあげた単一プロジェクトの問題と、このマルチプロジェクトの問題の最適スケジュールを比較すると、マルチプロジェクトの解は単一プロジェクトの解に見られた線形性から相当ずれている。これは各サブプロジェクトの間に介在する相互依存性によるものである。

## 6.7 むすび

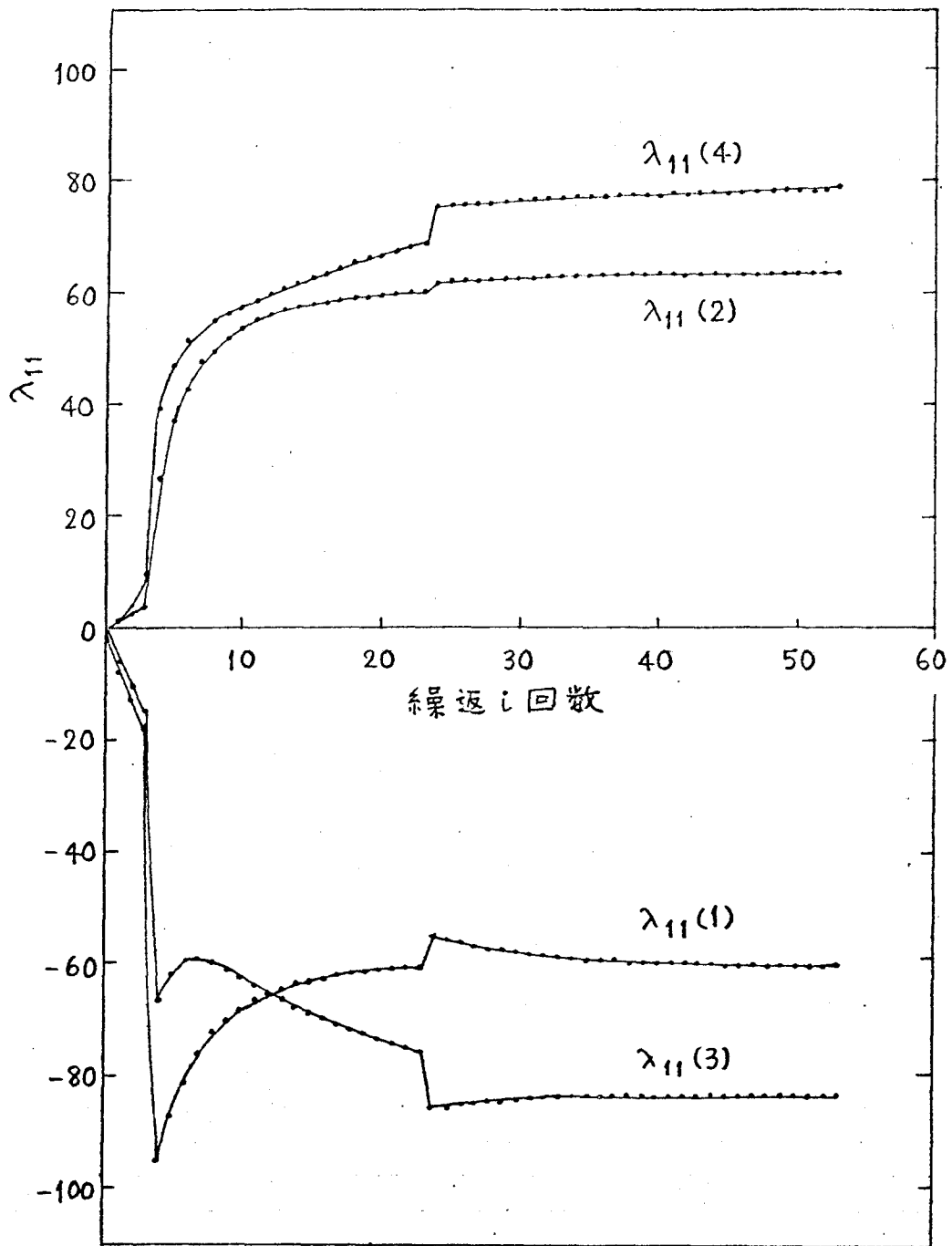
ここに示したスケジュールリングのモデルは、現実の問題に照らし合わせると、まだかなり荒っぽいものであり、修正の余地はあるが、今後、さらに複雑なシステムのモデルを作成する上での出発点になり得るものと考えられる。何故ならば、大きな生産プロジェクトの大半は、大なり小なり階層構造を持ち、全体の複合プロジェクトのプランニングあるいはスケジュールリングの最適化を迫られるからである。

第6.2表  $n_i$  および  $W_i$  の値

$i$	$n_i$	$W_i$
1	6	500
2	6	1000
3	6	600
4	6	1200
5	6	720
6	6	1440
7	6	864
8	6	1728
9	6	1037
10	6	2074



第6.3図 マルチプロジェクトの最適スケジュール



第6.4図 パラメータ  $\lambda_{11}$  の収束

## 結 言

以上の「輸送網および生産工程のシステム制御に関する研究」において得られた研究成果の要点を各章ごとにまとめておく。

### 第1章

- 1) 将来の、自動化された陸上輸送システムを解析するためのモデルを作成した。
- 2) システムの評価規準として、輸送所要時分の荷重平均値を取りあげ、Kuhn-Tucker の定理を用いてシステム運用の最適化の条件を導いた。
- 3) 重要な変数、パラメータの間の trade-off 関係を明らかにした。

### 第2章

- 1) 列車のプログラム自動運転制御系の最適設計方式を確立した。
- 2) 設計方式に、直接法と分解原理の二つの方法を取り入れて比較した結果、簡単な直接法で充分満足な最適制御系を設計できることを示した。
- 3) 制御系は、可変利得型の比例制御系で構成され、簡単な論理装置で実現できることを示した。

### 第3章

- 1) 列車群のスケジュールを、中央の制御用計算機で制御するためのアルゴリズムを確立し、その妥当性の検討を行った。

### 第4章

- 1) LP の問題をアナログ計算機で解く手法を修正して、線形系で近似できる、パラメータの値の未知なプロセスの多変数最適化制御方式を示した。
- 2) 制御方式を、一次系、二次系、逆応答系に関してアナログ・シミュレーションし、方式の妥当性を示した。



## 第5章

- 1) 機械加工工程のスケジューリングを行うためのモデルを作成した。
- 2) シミュレーション手法によって、スケジュールを逐次改良する方式を確立した。

## 第6章

- 1) 時間的に serial に関連した多数個のサブプロジェクトで構成される大規模生産システムのスケジューリングを行うためのモデルを作成した。
- 2) Fenchel の双対原理を応用して、階層構造を持つ多変数二次計画法の問題を、小規模な連立方程式を多数回繰返して解くアルゴリズムに帰着させた。
- 3) 通常、常識的にとられるスケジュールに比べて、最適スケジュールではコストを約半分に減らすことができた。

終りに、本論文で取り扱っている研究分野が、今後指向すべき方向について述べる。

輸送網のシステム制御に関する研究は、現在、活発な議論が進められている環境システム制御の問題等とともに、将来、都市問題の一環としてとらえ、都市全体を最適に設計し、運用するためのシステム工学を目指すべきであろう。一方、生産工程のシステム制御に関する研究は、一工場、あるいは一企業の経営情報システム(MIS)につながるべき問題である。

これらの所謂、大規模システムのシステム制御に関する問題は、従来の狭い意味での制御工学の研究のみでは解決できない。生産工学、経営工学、数理経済学、社会工学等、非常に広範囲にわたる隣接分野との境界領域を探求することが重要な意味をもってくるものと思われる。

## 謝 辞

この研究は、主として著者が所属する三菱電機(株)中央研究所で行なわれ、一部、米国スタンフォード大学に留学中になされたものを、大阪大学基礎工学部藤沢俊男教授の有益な御指導、御教授のもとにまとめられたものである。

大阪大学基礎工学部桜井良文教授には、著者の研究生生活の頭初より終始懇篤な御指導と御鞭撻を賜り、さらに、論文提出の機会を与えて戴いた。

本研究の初期の段階では、パデュー大学福永圭之介準教授(元三菱電機(株)中央研究所)に終始有益な御指導を頂いた。

大阪大学基礎工学部坂和愛幸教授、辻三郎教授、須田信英助教授には、本論文の作成に關して、種々御指導、御援助を頂いた。

本論文の第1章の研究は、著者が昭和41年~43年にフルブライト交換留学生として、スタンフォード大学およびスタンフォード研究所に滞在中に行なわれたものであり、スタンフォード大学準教授 Dr. D. G. Luenberger, Systems Control, Inc. の Mr. L. P. Hajdu, Dr. R. E. Larson (元スタンフォード研究所), スタンフォード研究所 Mr. K. W. Gardiner に有益な討議と示唆を頂いた。また、スタンフォード研究所勤務の機会に Systems Control, Inc. の Dr. P. Merritt (元スタンフォード研究所) によって与えられた。

なお、三菱電機(株)中央研究所馬場準一博士、大野栄一博士には御厚情あふる御激励と御援助をいただき、特に、馬場準一博士には、新しいシステム概念を御教示いただいた。

以上の方々に対して、衷心よりの感謝の意を捧げるものである。

## 参 考 文 献

1. Arrow, K. J., L. Hurwicz, and H. Uzawa, "Studies in Linear and Nonlinear Programming," Stanford Univ. Press, Stanford, Calif., 1958.
2. Arrow, K. J., S. Karlin, and H. Scarf, "Studies in the Mathematical Theory of Inventory and Production," Stanford Univ. Press, Stanford, Calif., 1958.
3. Athans, M., "Optimal Control," McGraw-Hill, New York, 1966.
4. 馬場準一, "システム工学の考へ方," 制御工学, Vol. 10, No. 10, pp. 439-443, 1966.
5. Chilton, E. G., ed., "Future Urban Transportation Systems: Technological Assessment," Technical Report, Stanford Research Institute, Menlo Park, Calif., 1967.
6. Dantzig, G. B., "Linear Programming and Extensions," Princeton Univ. Press, Princeton, N. J., 1963.
7. Dantzig, G. B. and P. Wolfe, "Decomposition Principle for Linear Programming," Operations Research, Vol. 8, No. 1, 1960.
8. Faddeeva, V. N., "Computational Methods of Linear Algebra," Translated from Russian to English by C. D. Benster, Dover Publications, New York, 1959.
9. 福永圭之介, "試行法による多変数最適化制御," 制御工学, Vol. 5, No. 1, pp. 46-52, 1961.
10. 福永圭之介, 田村坦之, "線形系における多変数最適化制御-数学モデル未知の Linear Programming," 制御工学, Vol. 10, No. 1, 1966.
11. 福永圭之介, 田村坦之, 羽根田博正, "シミュレーションによる生産スケジューリング制御," 制御工学, Vol. 10, No. 7, 1966.
12. 福永圭之介, 田村坦之, "Mathematical Programmingの動向, その1," 三菱電機技報, Vol. 39, No. 11, pp. 1412-1419, 1965.
13. Geoffrion, A. M., "Elements of Large-Scale Mathematical Programming," RAND Report R-481-PR, RAND Corp., Santa Monica, Calif., 1969.
14. Goldstein, A. A., "Constructive Real Analysis," Harper and Row, New York, 1967.

15. Hadley, G., "Nonlinear and Dynamic Programming," Addison-Wesley, Palo Alto, Calif., 1964.
16. Hajdu, L. P., K. W. Gardiner, H. Tamura, and G. L. Pressman, "Design and Control Considerations for Automated Ground Transportation Systems," Proceedings of IEEE, Vol. 56, No. 4, pp. 493-513, 1968.
17. Held, M. and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems," J. SIAM, Vol. 10, No. 1, pp. 196-210, 1962.
18. Kalman, R. E. and R. W. Koepcke, "Optimal Synthesis of Linear Sampling Control Systems Using Generalized Performance Indexes," Trans. ASME, Vol. 80, No. 6, pp. 1820-1826, 1958.
19. Kalman, R. E., P. L. Falb, and M. A. Arbib, "Topics in Mathematical System Theory," McGraw-Hill, New York, 1969.
20. 児玉慎三, 馬場鉄一, "Varaiya氏の分解原理による線形系の最適制御," 電気学会雑誌, Vol. 88-7, No. 958, pp. 1329-1338, 1968.
21. Korn, G. A. and T. M. Korn, "Electronic Analog and Hybrid Computers," McGraw-Hill, New York, 1964.
22. Kuhn, H. W. and A. W. Tucker, "Nonlinear Programming," Proc. 2nd Berkeley Symp. on Math. Stat. and Prob., Univ. of Calif. Press, Berkeley, Calif., 1962.
23. Lasdon, L. S., "Duality and Decomposition in Mathematical Programming," IEEE Trans. on System Sci. and Cybernetics, Vol. SSC-4, No. 2, pp. 86-100, 1968.
24. Lasdon, L. S. and J. D. Shoeffler, "A Multi-level Techniques for Optimization," Proc. JACC, pp. 85-91, 1965.
25. Larson, R. E., "Computational Aspects of Dynamic Programming," IEEE Int'l Convention Record, Part 3, Automatic Control, pp. 15-26, 1967.
26. Levine, W. S. and M. Athans, "On the Optimal Error Regulation of a String of Moving Vehicles," IEEE Trans. on Automatic Control, Vol. AC-11, pp. 355-361, 1966.
27. Liusternik, L. and V. Sobolev, "Elements of Functional Analysis," Translated from Russian to English by A. E. Labarre, et al., Frederic Unger, New York, 1961.
28. Longley, L., "Analog Computer Application in Operations

- Research," *Electronic Engineering*, Vol 36, pp. 378-381, 1964.
29. Luenberger, D. G., "Optimization by Vector Space Methods," John Wiley, New York, 1969.
  30. Luenberger, D. G., "Convex Programming and Duality in Normed space," *IEEE Trans. on System Sci. and Cybernetics*, Vol. SSC-4, No. 2, pp. 182-188, 1968.
  31. Mond, B., "Duality for Variational Problems," *J. Math. Analysis*, Vol. 18, No. 2, pp. 355-364, 1967.
  32. 小原太郎, 鳥居健太, 他, "プログラムによる列車の自動運転," *三菱電機技報*, Vol. 37, No. 9, pp. 1171-1177, 1963.
  33. Page, E. S., "On the Scheduling of Job by Computer," *Computer J.*, Vol. 5, No. 3, pp. 214-220, 1962.
  34. Page, E. S., "On Monte Carlo Methods in Congestion Problems; I. Searching for an Optimum in Discrete Situations," *Operations Research*, Vol. 13, No. 2, pp. 291-299, 1965.
  35. Pyne, I. B., "Linear Programming on an Electronic Analog Computer," *AIEE Trans. Communications and Electronics*, No. 24, pp. 139-143, 1956.
  36. "プログラム制御による列車自動運転に関する基本的研究報告書," *日本鉄道サイバネティクス協議会*, 1967.
  37. Ross, H. R., "New Transportation Technology," *Int'l Sci. and Technology*, pp.26-37, November, 1966.
  38. 岡根智明, 他, "スケジューリング問題," *日科技連数学計画シンポジウム, 報文シリーズ* No. 8, 1963.
  39. Tamura, H., "Optimal Design of a Programmed Train Control System and Computer Control of Traffic Schedule Keeping," *Preprints IFAC Symposium on Systems Eng. Approach to Computer Control*, pp. 515-520, Kyoto, Aug. 11-14, 1970.
  40. 田村坦之, "双対原理による大規模生産システムのスケジューリング," *システムと制御* (投稿中)
  41. Varaiya, P., "Decomposition of Large-Scale Systems," in "Systems Theory," edited by L. A. Zadeh and E. Polak, McGraw-Hill, New York, 1969.
  42. Yamazaki, S., "Train Operation Systems," *Preprints IFAC Symposium on Systems Eng. for Control Systems Design*, Tokyo, Aug. 25-28, 1965.
  43. 米田, "互換による順列の逐次発生," *情報処理*, Vol. 3, No. 4, 1962.

## 目 次

第 1.1 図	ループ状の案内路線と交差点を持つ案内路線	8
第 1.2 図	路線容量と速度の関係	12
第 1.3 図	輸送所要時分の荷重平均値 $T(v, p)$ (2 駅向題)	17
第 1.4 図	輸送所要時分の荷重平均値 $T(v, p)$ (5 駅向題)	18
第 1.1 表	システムの最適運用条件	16
第 1.2 表	単位時間あたりの乗客輸送量 ( $\lambda/H$ )	18
第 2.1 図	時向偏差と速度指令変化値の関係	24
第 2.2 図	プログラム自動運転: 東海道新幹線, その 1 $\Delta S = 10 \text{ km}$	28
第 2.3 図	プログラム自動運転: 東海道新幹線, その 2 $\Delta S = 5 \text{ km}$	28
第 2.4 図	プログラム自動運転制御系のブロック図	29
第 2.1 表	直接法および分解原理によって得た評価関数値の比較, $\Delta S = 10 \text{ km}$	29
第 3.1 図	列車群制御システム	32
第 3.2 図	列車群のスケジュール回復動作, その 1 $x(s_0, s_j) \geq j, \forall j$	34
第 3.3 図	列車群のスケジュール回復動作, その 2 $x(s_0, s_2) = 1 < 2$	36
第 3.4 図	中央制御のための計算のフローチャート	38
第 3.1 表	中央制御用計算機の入力情報, その 1	35
第 3.2 表	中央制御用計算機の入力情報, その 2	37
第 4.1 図	制御対象のブロック図	42
第 4.2 図	幾何学的解釈	43
第 4.3 図	アナログ計算機による最適化制御系の構成	45
第 4.4 図	数学モデル未知の LP の解, $G(s) = \frac{1}{1+Ts}$	47

第4.5図	Aと $K_1$ および $T_r$ と $K_1$ の関係, $G(s) = \frac{1}{1+Ts}$	47
第4.6図	数学モデル未知のLPの解, $G(s) = \frac{1}{1+qs+ps^2}$	48
第4.7図	Aと $K_1$ および $T_r$ と $K_1$ の関係, $G(s) = \frac{1}{1+qs+ps^2}$	48
第4.8図	数学モデル未知のLPの解, $G(s) = \frac{\alpha}{1+T_1s} - \frac{\beta}{1+T_2s}$	49
第4.9図	Aと $K_1$ および $T_r$ と $K_1$ の関係, $G(s) = \frac{\alpha}{1+T_1s} - \frac{\beta}{1+T_2s}$	49
第5.1図	機械加工工程	52
第5.2図	機械加工工程のタイムチャート	57
第5.3図	基本形(サブシステム)と全システムの関係	57
第5.1表	タイムテーブル	60
第5.2表	基本形の例	61
第5.3表	シミュレーションの結果	61
第5.4表	$f_a$ と $f_b$ の比較	62
第6.1図	Fenchelの双対原理の幾何学的解釈	68
第6.2図	単一プロジェクトのスケジュール	73
第6.3図	マルチプロジェクトの最適スケジュール	75
第6.4図	パラメータ $\lambda_{ij}$ の収束	76
第6.1表	二次計画法の問題とスケジューリング問題の関係	66
第6.2表	$n_i$ および $W_i$ の値	74