IFAC

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

WARSZAWA 1969

Control of Large Scale Systems

Fourth Congress of the International Federation of Automatic Control
Warszawa 16–21 June 1969

TECHNICAL SESSION 42

Organized by
Naczelna Organizacja Techniczna w Polsce

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

Control of Large Scale Systems

TECHNICAL SESSION No 42

FOURTH CONGRESS OF THE INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL
WARSZAWA 16 – 21 JUNE 1969

Organized by
Naczelna Organizacja Techniczna w Polsce

Contents

Paper No			Page
42.1	USA	- W.Findeisen, I.Lefkowitz - Design and applications of Multilayer Control	3
42.2	USA	- R.K.Bhattacharyya, E.S.Bahary - Multistage Optimization of a Production System Using a Discrete Version of Pontryagin's Maximum Principle	23
42.3	GB	- H.H.Johnson - The Occurence and Effect of Data Loss in a Hierarchical Digital Computer System Providing On-Line Process Control	38
42.4	SU	- A.P.Kopielowitch, A.A.Bielostocki, B.A.Wlasiuk, G.I.Nikitin, W.M.Hrapkin - Systems and Control Algorithms for a Complicated Steel Mill Complex	54
42.5	GB	- G.Hayhurst - An Approach to Automatic Adaptive Batch Production Systems	70
42.6	GB	- B.B.Hickling - The Use of Computer Techniques in the Management Control of a Steelworks	81

Wydawnictwa Czasopism Technicznych NOT - Polska

Zakład Poligraficzny WCT NOT. Zam. 51/69.

DESIGN AND APPLICATIONS OF MULTILAYER CONTROL

W. Findeisen
Katedra Automatyki I Telemechaniki
Warszawa, Poland

I. Lefkowitz
Case Western Reserve University
Cleveland, Ohio, U.S.A.

I. Introduction

This paper is directed to the problem of control of complex industrial systems where the measure of performance has an economic base and where the cost of implementing the control is imbedded in the problem formulation and solution. The overall motivation may be maximizing profit or minimizing cost subject to constraints and boundary conditions induced by product specifications, environmental interactions, etc. The cost of implementation should generally include, in addition to the hardware requirements, the inputs of time and effort for system modeling, design of control algorithms, programming and related tasks.

It is assumed at the outset that the overall problem cannot be solved analytically and that the answer lies in a judicious (and perhaps inspired) application of approximations, empiricisms and heuristics. It is assumed further that the computer must play a key role in any meaningful attack on the problem; in particular, the computer must be effectively utilized at every stage of development of the control system ranging from the initial analysis and design stage to the on-line implementation and final evaluation stages.

A general approach to the control system design conditioned to the above considerations is described here. The approach is structured as follows:

1. Multilevel control hierarchy

   The complex system is decomposed into simpler subsystems, each with its own controller. The controller is designed according to a local criterion and a local model. Higher level controllers then integrate the actions of the local (first-level) controllers so that the overall system objective is best served. A simple two-level hierarchy is shown in Fig.1.

2. Multilayer control hierarchy

   The control problem is decomposed into simpler subproblems of a form readily solved and implemented by available techniques. Higher layer control functions then serve to integrate the individual subproblems, again so that the overall problem is adequately handled. The basic block diagram representation of the multilayer hierarchy appears in Fig.2.

3. Information feedback

Periodic measurements of the state of the controlled system provide a feedback of information through which compensation for disturbances, model approximations and simplified control algorithms is realized.

4. Computer-aided design and information processing

The computer serves as an essential tool at the design stage and as the means of implementing control at the operating stage.

It should be noted that the components of the approach listed above are interdependent. Thus, each subsystem controller of the multilevel hierarchy might itself be realized in terms of a multilayer structure. The multilayer hierarchy, in turn, is based on the processing and feedback of information by which simple models are rendered effective agents for achieving on-line control algorithms. Finally, the computer as a simulator, information processor and decision maker renders the whole procedure feasible.

The multilevel and multilayer hierarchies have been previously described and the motivations given.[1,2,3,4] Two important attributes are mentioned as they relate to the control system design problem:

1. The large complex problem is replaced by a number of relatively simple and more readily handled subproblems.

2. Control actions are performed roughly in proportion to the mean frequency of need. Thus, as we proceed up the hierarchy, the decision-making process becomes generally more complex but it need be carried out much less frequently.

The Control Problem

We consider the following system description in order to develop certain aspects of the control problem. We denote the time interval [0,T] as the base period over which control is to be applied and process performance P is to be evaluated. We assume P is generally dependent on the inputs and outputs over the interval [0,T], hence write

$$P = f(c_{[0,T]}, m_{[0,T]}, u_{[0,T]}) \tag{1}$$

where c denotes the set of output variables, m the set of manipulated inputs and u the set of disturbance inputs. More specifically, we assume that u characterizes the influence of environment and other systems on the performance of the controlled system, m represents the set of available control actions or decision variables and c identifies those variables of the process which (i) depend deterministically on the inputs, (ii) are significantly related to the control problem, and (iii) can be determined on-line at discrete time instants either through direct measurement or inferentially.

Assuming a causal system,

$$c(t) = g(x(0), m_{[0,t]}, u_{[0,t]}, t) \quad (2)$$

where $x(0)$ denotes initial state and $c(t)$ is the output at time $t \in [0,T]$. In general, we restrict our attention to physical systems so that the variables are continuous in time; however, in particular implementations of computer control, $m_{[0,T]}$ may be piecewise constant and $c(t)$ is generally determined only at discrete time instants.

A class of systems of particular interest is the continuous-type industrial process designed for quasi-steady-state operation. In the steady-state, we assume Eq.(2) reduces to the static relationship

$$c(t) = g(m(t), u(t)) \quad (3)$$

An important attribute of the multilayer approach is, in fact, the design of the direct or first-layer controller so that the plant may be approximated by its steady-state model when considering the higher layers of the control hierarchy.

The goal of the control system may be stated as

$$\max_{m \in M} E[P] \quad (4)$$

where P is given by Eq.(1), E is the expectation operator, m denotes $m_{[0,T]}$, and

$$M = \{m: h(c(t), m(t), u(t)) \geq 0 \text{ for all } t \in [0,T]\} \quad (5)$$

It is assumed in expressing Eq.(4) that the expectation is based on an implied distribution of u over the set U of possible realizations of $u_{[0,T]}$.

It might be pointed out that other formulations of the goal might be considered and indeed may be more appropriate. In particular, because of the uncertainty in u and the practical difficulty of determining a meaningful distribution, we might use a "satisfaction" approach e.g., choose $m \in M$ so that $P \geq \theta$ for all $u \in U$, where θ is a suitably defined lower limit on acceptable performance.

The ultimate objective of the control system design is to develop appropriate algorithms for information processing and decision-making in pursuit of the goal expressed by Eq.(4) but taking into account the cost of implementation. Thus, in principle, we want to maximize

$$E[P(c,m,u)] - C \quad (6)$$

where C denotes the overall cost of implementing the control. The maximization of Eq.(6) is made with respect to the set of design decisions, e.g., the structures of the control algorithms employed, the information used at various levels of the hierarchy, the periods for control actions, etc.

A Guideline for System Design

The direct implementation of Eq.(6) is generally not feasible because the detailed assessment of process performance and implementation costs in relation to control system "complexity" is extremely difficult. The system designer must make a number of a priori decisions based in part on experience, heuristics and the results of computer simulations. Referring to the system description given previously, and in relation to the multilevel and multilayer control hierarchies, these decisions comprise the following:

A. The overall system is decomposed into subsystems for first-level control. We then associate with each subsystem a set of "local" relations of the form of Eq.(1) through (5). Now the set of disturbances for say the k^{th} subsystem includes not only a subset of the overall system disturbance vector but also the effects of interactions with other subsystems. Another distinction is that the functionals f,g and h may be modified by actions of second and higher-level controllers concerned with the overall system performance. Design considerations of the higher-level controllers are not treated here except as part of the illustrative example discussed later.

The system is partitioned along lines usually dictated by the kinds of processing units used, their relative proximity, their identification with a specific product or service and related considerations. These factors are often relevant to costs and constraints associated with data transmission, reliability and emergency actions. They may also be important if independent operation of the processing units is required under certain conditions. Additional factors that should be considered in structuring the subsystems, however, are:

a) reduction of complexity of the control algorithm.
b) minimization of the interaction effects between subsystems.

Both factors serve to reduce the cost of control implementation, either by simplifying the computations or by reducing the required frequency of higher level intervention.

B. Each subsystem control problem is organized into a multilayer control hierarchy. The control actions associated with each layer are described with reference to Fig. 3.

The first, *Direct Control Layer*, applies controlling inputs m to the process, for the output c to follow a desired trajectory r. This is presented in Fig. 3 as a simple feedback structure, although more complex structures with the same objective are, of course, possible. The direct control algorithm is of fixed structure,* with a parameter vector α. The vector α may be adjusted in principle (from the third layer of control), at some interval T_3^α.

The second, *Optimization Layer*, performs determination of the desired trajectory r. It is assumed that this is done based upon some representation u^2 of the disturbance u, and using a fixed structure algorithm with a vector of parameters β. The computation of the desired trajectory is performed at some interval T_2.

The third, *Adaptation Layer*, readjusts the vector β at some interval T_3^β, and the vector α at the interval T_3^α. The adaptations are based upon either an updating of the model used to derive the control algorithm or some kind of hill-climbing performed on a suitable measure of local performance. Whatever the basis, the implementations take the forms of fixed-structure algorithms with a vector of parameters γ.

C. The specific control functions in the multilayer hierarchy are designed; i.e., the following design decisions are to be made:

1. Identification of the variables to be associated with each layer of the control hierarchy and the local criteria of performance. In particular, this implies a partitioning of the set of disturbance inputs according to the mode and frequency of compensation, based on such factors as frequency spectrum characteristics and performance sensitivity.

2. Specification of structure of the control algorithms associated with each layer of the control hierarchy.

3. Determination of the control intervals $T_1, T_2, T_3^\alpha, T_3^\beta$.

The above design decisions may be considered another set of inputs to the system. In particular, we may repeat various aspects of the design process from time to time; i.e., we modify our design decisions based on the feedback of operating experience or new information about the system and its environment. We assign this action to a fourth layer of control, the "*Self-Organizing*" layer. Note that the output of the fourth-layer controller determines the structure of the control system; this is in distinction to the first three layers of control whose outputs are generally numerical.

The multilayer hierarchy suggests a sequential procedure for formulating and evaluating the design decisions. We start with the first layer and choose a set of outputs to be controlled such that the following criterion function is sufficiently small:

$$J_2 = E[P^\circ - P^\circ(T_2)] \qquad \text{(i)}$$

where P° denotes the optimum performance of the process, with no restriction

*The direct control function may be implemented by a discrete algorithm with interval T_1 or by a continuous algorithm.

induced by the choice of c or control algorithm, and $P°(T_2)$ denotes the suboptimal performance resulting from choosing the set of outputs to be controlled and from maintaining $c(t) = c°(kT_2)$ for $kT_2 < t \leq (k+1)T_2, k = 0,1,2,...$ (i.e., m(t) is varied to maintain c(t) at the set-point value). Then, we choose the first-layer controller algorithms and the period T_1 such that a suitable measure of the following error be sufficiently small, e.g.,

$$J_1 = E||r-c|| \tag{ii}$$

Note that criterion (i) specifies that the degradation of performance due to intermittent intervention of the second-layer control be small, while criterion (ii) requires that the first-layer controllers be capable of maintaining the outputs at their desired values. Both criteria, J_1 and J_2 may be reduced at the expense of computational effort (by shortening T_1 and T_2, for example) and tradeoff considerations are indicated.

If the error norm $||r-c||$ is appropriately chosen, it may give an approximate measure of the performance loss resulting from non-ideal first-layer control response. A basis for the tradeoff for the first-layer design is provided then by the expression.

$$C_1 J_1(A_1, T_1) + C_2(A_1)/T_1 \tag{7}$$

where A_1 denotes the set of first-layer control algorithms used, C_1 is the cost coefficient for performance loss and C_2 is the cost of implementing a single control action. Note that both J_1 and C_2 are dependent on A_1.

We may evaluate (7) for various combinations of control algorithms that seem reasonable candidates for application*. It is assumed, in making comparisons that the parameter vector α is determined according to the adaptation criterion used at the third layer. Thus, we may determine, in principle, the choices of A_1 and T_1 yielding a minimum value of criterion (7).

The design of the second-layer control function proceeds on the premise that the first-layer controller is working perfectly; i.e., c is identical to r, the system may be described by a simple steady-state model and a class of disturbances denoted by u^1 is suppressed by the direct control actions (hence need not be considered explicitly by the second-layer controller). With these assumptions, we adopt a structure of the second-layer control algorithm of the form

$$r = r(\beta, u^2) \tag{8}$$

*Most likely restricted to the class of simple linear feedback controllers unless special requirements are indicated.

where u^2 denotes the set of disturbance inputs or their statistical characteristics which are (a) determinable from on-line measurements and (b) significant with regard to the optimization problem. Note that we may choose the set u^2 by a procedure similar to that described for choosing c.

We assume that the first-layer design procedure has produced the tentative decisions A_1^*, T_1^* and define

$$J_2^* = E[P(A_1^*, T_1^*)]\Big|_{\substack{r=r^\circ \\ \alpha=\alpha^\circ}} - E[P(A_2,T_2)]\Big|_{\substack{\beta=\beta^\circ \\ \alpha=\alpha^\circ}} \tag{9}$$

where the first term on the right denotes the performance resulting from the first-layer design decisions where r and α are optimal, the second term denotes the performance resulting from the use of second-layer control algorithm A_2 with period T_2(i.e., $c(t)=r(\beta^\circ,u^2(kT_2))$ for $t \in (kT_2,(k+1)T_2]$, $k=0,1,2,...$, where β° denotes the optimum value of the parameter vector), with A_1^* and T_1^* implied for the first layer. Hence J_2^* represents (approximately) the mean performance loss attributed to the second-layer controller design. Analogous to (7) we have the design criterion for the second-layer control

$$C_3 J_2^*(A_2,T_2) + C_4(A_2)/T_2 \tag{10}$$

where C_3 is the cost coefficient for performance loss and C_4 is the cost of implementing a single control action. We may use (10) to guide the choice of A_2 (out of a set of feasible alternatives) and control interval T_2+.

We may extend the above procedure to define performance loss functions associated with the adaptive control layer; however, we have to distinquish between the α and β updating processes. Labeling the tentative design decisions obtained thus far by A_1^*,T_1^*,A_2^*,T_2^*, we define

$$J_3^\alpha = E[P(A_1^*,T_1^*,A_2^*,T_2^*)]\Big|_{\substack{\alpha=\alpha^\circ \\ \beta=\beta^\circ}} - E[P(A_3^\alpha,T_3^\alpha)]\Big|_{\substack{\beta=\beta^\circ \\ \gamma=\gamma^\circ}}$$

where $P(A_3^\alpha,T_3^\alpha)$ denotes the performance resulting from adaptation algorithm A_3^α (assuming A_1^* and A_2^* are incorporated in the lower control layers). Note that $\{A_3^\alpha\}$ may include the variety of adaptive techniques described in the literature for feedback control of nonlinear or nonstationary dynamic systems.

+Note that when T_2 gets sufficiently small, the performance at the first layer may be affected to the extent that an iteration on the first-layer design is called for.

The choice of A_3^α and T_3^α proceeds as above.

Analogously, we define

$$J_3^\beta = E[P(A_1^*,T_1^*,A_2^*,T_2^*)]\Big|_{\substack{\alpha=\alpha^\circ \\ \beta=\beta^\circ}} - E[P(A_3^\beta,T_3^\beta)]\Big|_{\substack{\alpha=\alpha^\circ \\ \gamma=\gamma^\circ}}$$

The terms in the above expression are defined similarly to the expression for J_3^α. The set $\{A_3^\beta\}$ may include various regression techniques, curve fitting methods and a two-time scale approach.[5] We again choose A_3^β from among the various alternatives based on a tradeoff expression of the form previously discussed; determination of T_3^β then follows.

The merit of designing the system sequentially lies in the fact that we do not have to make all the design decisions at once but can proceed layer by layer, evaluating the influence of each decision almost independently. Of course, there will generally be some interaction of subsequent decisions on those made previously. It is expected, therefore, that some iteration of the structural decisions will be necessary before reaching the final design.

The complexity of the system generally precludes any analytical formulation of the criterion functions defined above, or indeed analytical determination of the optimal conditions indicated. Quantitative results for design purposes, therefore, must rely heavily on computer simulations. In particular, we assume that various alternative algorithms or control configurations may be explored on the fast-time scale of the computer simulation. In like manner, iterative optimization procedures may be used to determine optimum parameter values, etc. It is assumed, further, that some information is available on the nature of the dominant disturbance variables, say in the form of past records, so that approximations to expected values may be obtained based on observed frequency distributions or statistical parameters.

The role of the computer in the design and implementation stages of the control system and the importance attached to the cost tradeoffs puts into evidence the necessity for effective programming languages for simulation, optimization routines and for on-line control. Indeed the hierarchial approach introduces additional needs relative to communications from level to level and layer to layer.

Example

We illustrate the application of the approach outlined above by reference to an example system which was formulated to provide a vehicle for studies of multilevel and hierarchial control concepts.

The system consists of a community and the facilities for supplying its electric power and water needs. The essential features of the system are displayed in the block diagram of Fig. 4.[6,7,8]

The block diagram identifies five subsystems as follows:

1. The community as characterized by its power and water needs.

2. The power generating plant which supplies the major (local) demand for electric power.

3. The power supply network which includes the power distribution grid and other (remote) power sources.

4. The municipal water supply system which includes the natural watershed reservoirs, water treatment plant, distribution network, etc.

5. The desalination plant which supplements the normal (but inadequate) water source by converting brine to potable water.

Several remarks are pertinent at this point:

1. The subsystems are complex dynamic systems which interact with one another; e.g., the energy supply for the desalination plant is a byproduct of the power generation plant, hence the two plants are closely coupled through this common interface.

2. There are many opportunities for optimal decision-making and control because of (a) a large number of degrees of freedom in operating the system and (b) several disturbance inputs with very significant variations (e.g., power demand, water demand, rainfall, weather conditions).

3. Each subsystem may itself be decomposed into a number of subsystems relevant to the systems control problem. For example, the power generation plant will include (in the system considered) a nuclear power source, steam generator, turbine and alternator. In general, there will be a control system associated with each of these subsystems.

4. There are very complex economic, sociological and political factors entering into the modeling of the community subsystem. These have been circumvented in the present study by assuming a control policy which minimizes operating costs subject to the constraint that the community's demands for water and power are continually met (excluding equipment malfunctions, system breakdowns, etc.).

5. For the purpose of the illustration, it is sufficient to consider just the dual-purpose plant consisting of a single desalination unit and a single power generation unit. It is assumed that the power grid supplies a fixed portion of the load; hence the power plant must satisfythe varying component of the load. Similarly, the desalination plant makes up the difference between the scheduled flow from the impounding reservoir and the averaged community demand. An important distinction with respect to the power generation system, however, is that there is a distribution reservoir which effectively decouples the water production system from the instantaneous variations in water demand.

Let us consider some aspects of the multilevel approach with reference to the dual-purpose plant. We assume for the time being the necessary regulatory functions to be installed (e.g., turbine governor control, alternator voltage control) so that the system operates with adequate response to load and disturbance variations and exhibits adequate stability.

We want to operate both power and desalination plants so that we minimize their combined costs of operation subject to satisfying the power and water demands specified for the system. Since the two plants interact, optimizing each independently may not be consistent with the overall optimum. Treating the two plants as one system has the disadvantage of introducing a higher dimensional optimization problem and hence one more costly to implement as an on-line control solution.

In the application of the multilevel approach, each plant is optimized based on an assigned cost (or value) of the thermal energy interchanged between the two subsystems. The role of the first-level controllers then is to determine the optimum values of the manipulated inputs (or operating levels) for its associated subsystem. A second-level controller may now act to coordinate the two first-level controllers by determining a cost factor (for thermal energy) such that the overall costs for the coupled system is minimum.

It is important to note here that the cost of the thermal energy interchange is a function of many variables. However, if the power plant is operated according to criteria of maximum efficiency of energy conversion (subject to a set of contraints), and if the water plant is operated to minimize the thermal energy requirement in satisfying the water production rate specification (and other relevant constraints), then the cost factor may be expressed as a function basically of the power and water production rates.

The second-level control function may be implemented through an iterative scheme such as developed by Lasdon.[9] Here, in effect, the cost factor associated with the coupling term (thermal energy interchange) is determined by a gradient search in the fast-time scale of the computer with the results transmitted to the plant controllers in real time.

The alternative considered here is based on developing an explicit cost relation for the thermal energy cost factor C_Q expressed as an algebraic function of power and water production rates, P_E and Y, respectively.* Thus,

$$C_Q = C_Q(P_E, Y) \qquad (11)$$

*This was obtained by use of simulation and multiple regression.[7,8]

This procedure offers the flexibility of a two-layer structure for the second-level or coordinating function. At the first layer, Eq.(11) is used to determine C_Q from the scheduled value of Y and measurements on P_E, with the result transmitted to the first-level optimizers. Note that this action is necessary only when there is a significant change in C_Q. More specifically, we compute C_Q every T_a units of time where, $E|\Delta C_Q| \leq \varepsilon_1$, where ΔC_Q denotes the maximum change of C_Q within an interval of length T_a. In general, ε_1 will be related to the cost of carrying out the computation of C_Q and implementing the result.

The second-layer component of this second-level control function involves the updating of the approximate algorithm based on Eq.(11). It is assumed that this will be done at a period T_b, where $T_b >> T_a$. The purpose here is to compensate for errors in the approximating function as a result of wide variations of P_F and Y and also to compensate for the many slowly varying disturbances which are not explicit in the relationship but which do have an effect. Again we consider a criterion of the form $E |C_Q - C_Q'| \leq \varepsilon_2$ as a basis for choosing T_b, where C_Q' denotes the corrected value of C_Q resulting from an updated Eq.(11), and ε_2 is related to the average cost of carrying out the updating procedure.

A second aspect of the multilevel approach entering into the optimization problem is that of scheduling the water production. As noted in a previous section, various reservoirs in the water system serve to decouple the required production rate from the instantaneous water demand, thus, in effect introducing still another degree of freedom in the decision making process. We may relegate to a third-level controller the optimum scheduling problem; i.e., determining a sequence $\{Y_{t+i}, i = 0,1,2,\ldots\}$ such that when this information is used by the second-level controllers*, the overall performance is maximized in an appropriate sense. Here Y_k denotes the production rate scheduled for the k^{th} control interval with subscript t denoting the present interval.

Inputs to the scheduling problem are the variations in water demand, river inflow and weather. Because of the coupling between the desalination and power generation plants, the variations in power demand and grid purchase rates also are significant factors. Analysis of these inputs revealed strong

---

*We may consider here that Y is a vector whose components define the scheduled water production rates of the conventional water supply and the desalination plant, respectively. Hence, in addition to the multilevel control system associated with the dual-purpose plant described above, we have one also associated with the conventional water supply.

cyclic components with periods of one day, one week and one year, respectively.* The storage reservoirs with effective time constants of the order of a day and a year, respectively, permit a partitioning of the scheduling problem into short, medium and long period decision processes.[6] This is shown schematically in the block diagram of Fig. 5. The particular point to be brought out here is that this decomposition is based on time scale and provides also the attributes of simplifying the on-line decision-making problem and reducing the costs of implementation by relating the frequency of control action to the need.

The essential features of the multilevel structure just described are displayed in the block diagram of Fig. 6. It is well to point out with reference to this figure that the hierarchial structure provides guidance and motivation for the system design and is not an end in itself; hence, the figure, which represents the structure in concept, may differ from the physical system in some aspects. We note too that the scheduling hierarchy of Fig. 5 is part of the "Dynamic Scheduling" block of Fig. 6.

We further illustrate the multilayer hierarchy with specific reference to the desalination plant**. There were three manipulated variables considered, each a valve position governing the flowrate of a process stream. Conventional feedback loops were provided for each of the process streams (Thereby eliminating disturbances due to varying line pressures, pump heads, etc.).

It was determined that the dominant variable with respect to the second-layer control action was the heat interchange flow Q. In particular,

$$E[\Delta P]_{Q=Q^\circ} < E[\Delta P]_{m_1=m_1^\circ}$$

where ΔP denotes the performance degradation due to variations in electric power load and other relevant disturbances during the control period T_2. The term on the left of the inequality expresses the mean loss of performance when Q is maintained fixed at the value determined by the second-layer controller i.e., $Q(t) = Q^\circ(kT_2)$ for $t \in (kT_2,(k+1)T_2]$, where Q is controlled by manipulating m_1. The term on the right of the inequality expresses the performance loss when m_1 is maintained constant at its optimal value, i.e., $m_1(t)=m_1^\circ(kT_2)$ for $t \in (kT_2,(k+1)T_2]$,(where m_1° is related to Q°). Accordingly, one of the

* Actual records of river flow, water demand and power demand from representative communities were used here.

**The study was carried out on a bench-scale model of the desalination process coupled (in real-time) to a computer simulation of the power generation subsystem. The results described in this section apply, therefore, to the laboratory system.

components of the vector r was the desired Q (determined as a function of current power demand and scheduled water production rate). A feedback loop around Q (determined from inferential measurements) was effected through manipulation of the flow stream* m_1. The choice of m_1, incidently, was based almost entirely on the effectiveness of control of Q to its set-point value.

The remaining components of the vector r corresponded to the optimum flowrate settings for the remaining process streams.

The second-layer control algorithms were of the form

$$r = r(P_E, Y, \beta) \tag{12}$$

Eq.(12) was obtained by the following procedure:[8]

1. The plant (in steady-state) was simulated on the computer using typical mass, energy balance and equilibrium relationships.

2. Some of the parameters of the simulation model (affected by mass and energy transfer rate coefficients known only empirically) were updated by comparing the output of the simulation with that of the physical process at several operating points.

3. A multiple regression model was fitted to the simulated system.

4. The necessary conditions for optimum performance were derived from the regression model, yielding Eq.(12). The parameter vector β is related to the coefficients of the regression model.

The structure of the second-layer control algorithm was arrived at through qualitative considerations of the relative costs of implementing various alternatives**. The distributed nature of the system precluded a direct analytical attack based on the plant model. Direct hill climbing on the physical system was excluded because of noise problems and very unfavorable dynamics. Hill climbing on the simulated system suffered the disadvantage of high computation costs. On the other hand, the plant's response surfaces seemed sufficiently smooth and regular that relatively low order regression models appeared to fit well. The resulting algorithm,Eq.(12), was then of simple form and readily implemented.

The third-layer algorithms have not yet been implemented. some updating is suggested for the elements of α associated with the feedback loop on Q because of the nonlinearity of the system and the wide range for r_Q. Since the operating point changes relatively slowly, this adaptation is easily incorporated.

*Actually, a cascaded control configuration was used here.

**An exhaustive study was not made of this question; many other possibilities were suggested but not explored.

The adaptation on β may be carried out by repeating steps 2 - 4 of the above procedure every T_3^β increments of time.

Summary

Attention is directed to the system design problem in the implementation of control of complex industrial systems. The multilevel and multilayer hierarchial structures are presented as the framework for carrying out the design process following a sequential and iterative procedure. Implicit in the approach is the need for simplifying the control problem by decomposition, approximation and information feedback. Another essential feature is the compromise between cost of implementation and performance loss.

The hierarchial approaches are predicated on the use of the computer for fast-time simulation, information processing and on-line control.

Many of the concepts and guidelines described are illustrated in the context of a specific system application.

References

1. "Multilevel Concept and Systems Engineering", M.D. Mesarovic, Proceedings of the 1965 Systems Engineering Conference, September 1965.

2. "Multilevel Approach Applied to Control System Design", I. Lefkowitz, Trans. ASME, 88, June 1966.

3. "Advances in Multilevel Control", M.D. Mesarovic, I. Lefkowitz, J.D. Pearson, Proceedings of the Tokyo Symposium on Systems Engineering for Control System Design, Tokyo, Japan, August 1965.

4. "Parametric Optimization by Primal Method in Multilevel Systems", W. Findeisen IEEE Trans. on Systems Science & Cybernetics, August 1968.

5. "An Adaptive Technique for On-Line Optimizing Control", J.H. Burghart, I. Lefkowitz, Proceedings 1967 JACC, Philadelphia, Pa.

6. "A Hierarchial Approach to Scheduling a Dual Purpose Plant", J.E. Eickelberg, M.S. Thesis, Case Western Reserve University, 1968.

7. "Simulation of an Electric Power/Desalination Plant Interface", R.G. Wilhelm, M.S. Thesis, Case Western Reserve University, 1968.

8. "A Model-Based Technique for On-Line Static Optimizing Control" H.I. Klee, M.S. Thesis, Case Western Reserve University, 1968.

9. "A Multilevel Technique for Optimization", L.S. Lasdon, J.D. Schoeffler, Proceedings of the Joint Automatic Control Conference, Troy, New York, June 1965.

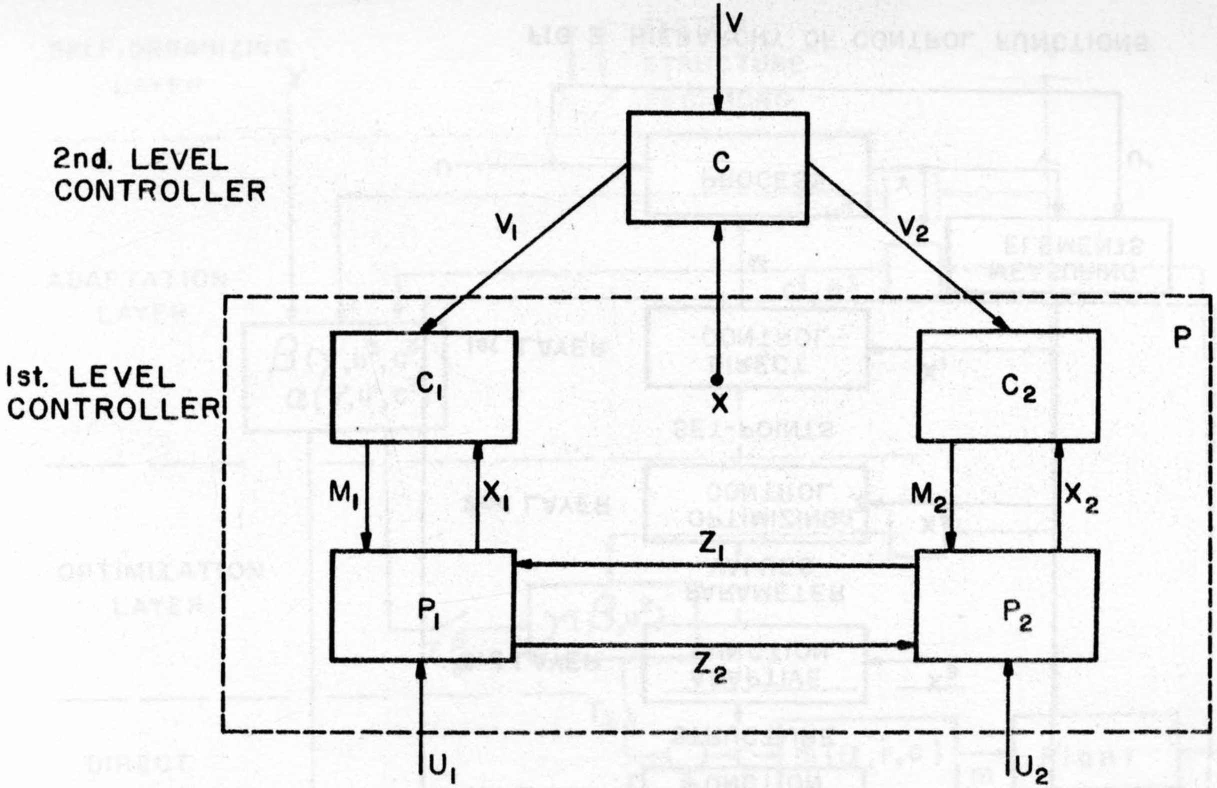


FIG. 1 TWO LEVEL CONTROL OF
INTERACTING SUBSYSTEMS

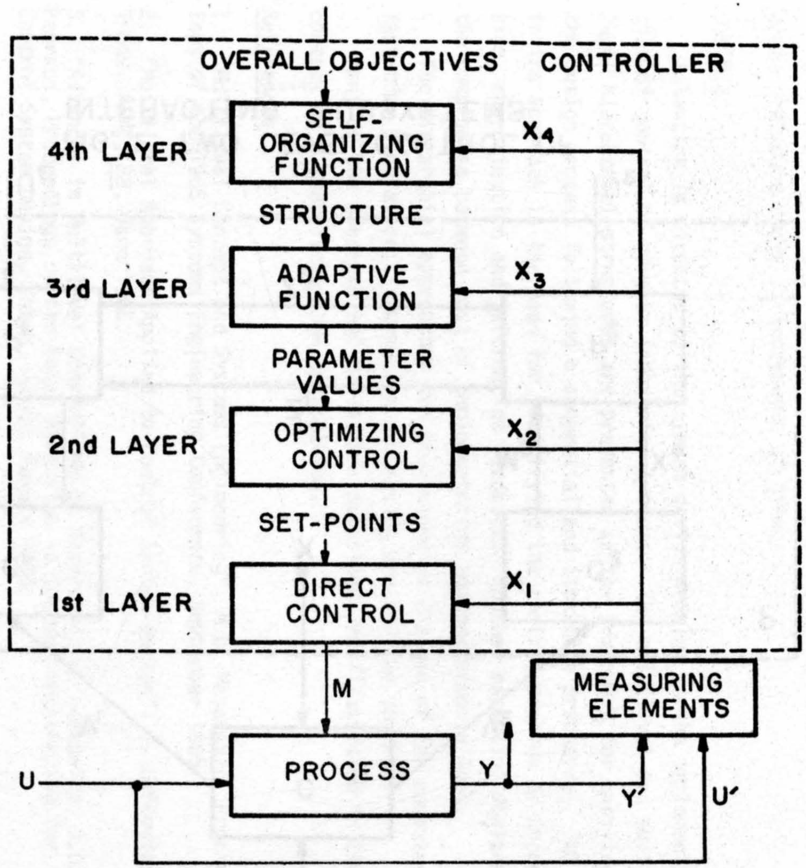


FIG. 2 HIERARCHY OF CONTROL FUNCTIONS

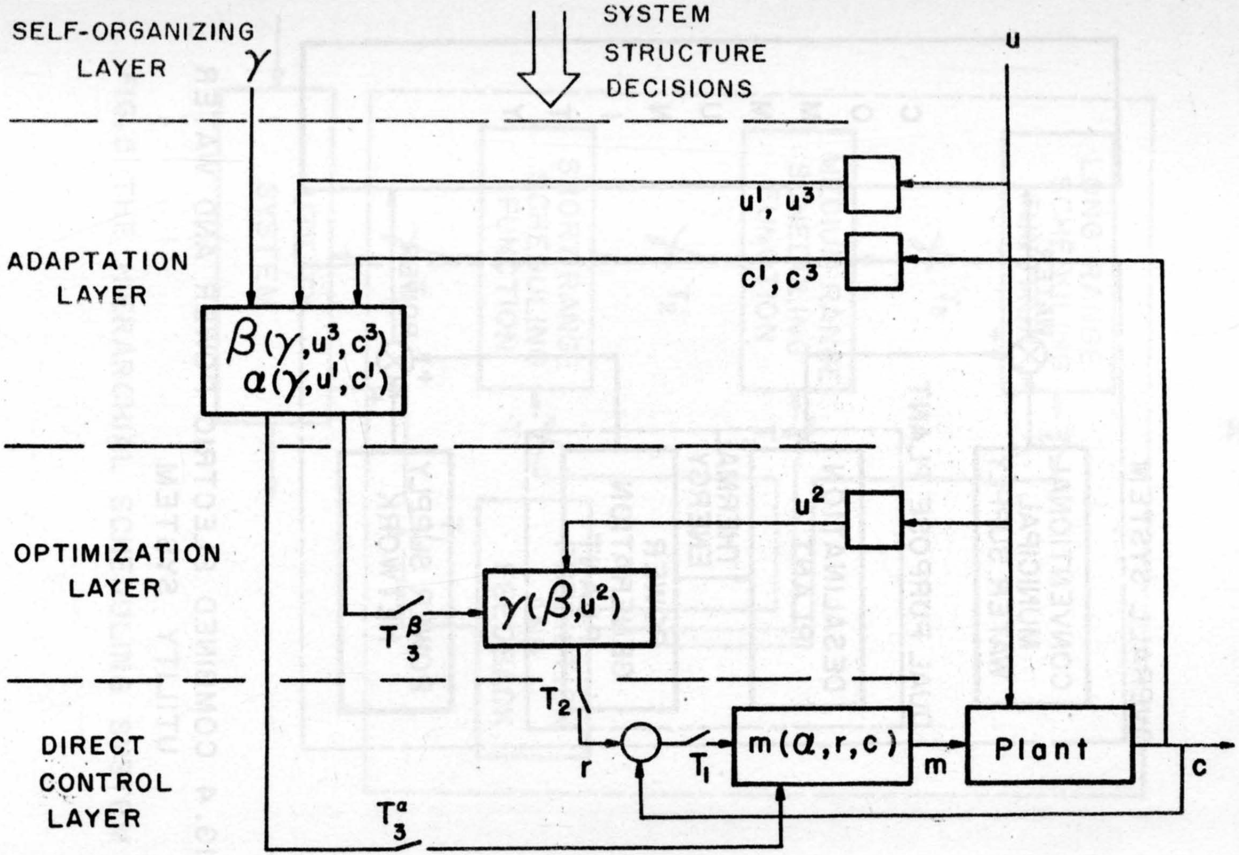


FIG. 3 DIAGRAM OF MULTILAYER CONTROL SYSTEM

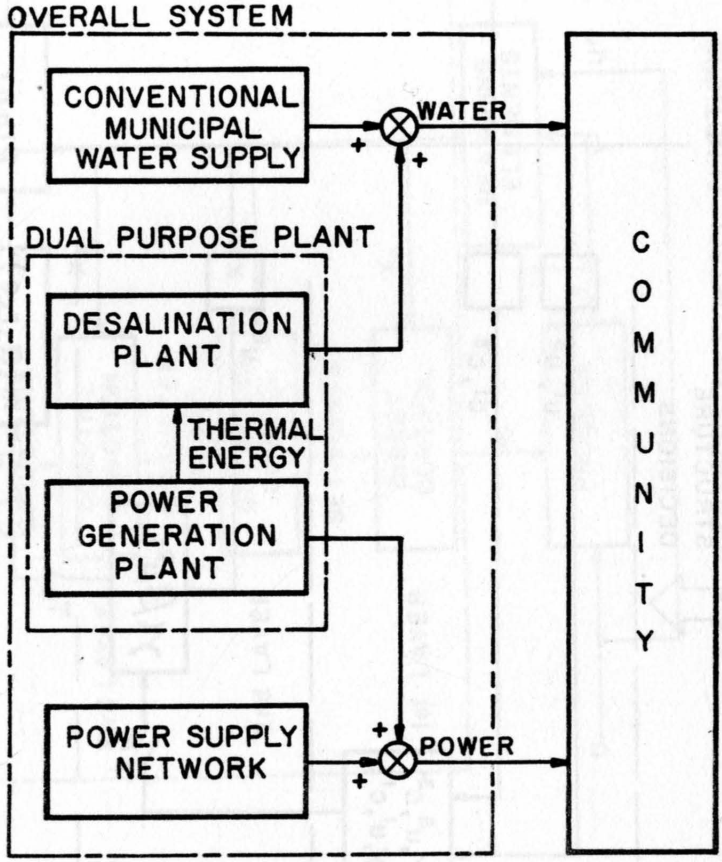


FIG. 4 COMBINED ELECTRIC POWER AND WATER UTILITY SYSTEM

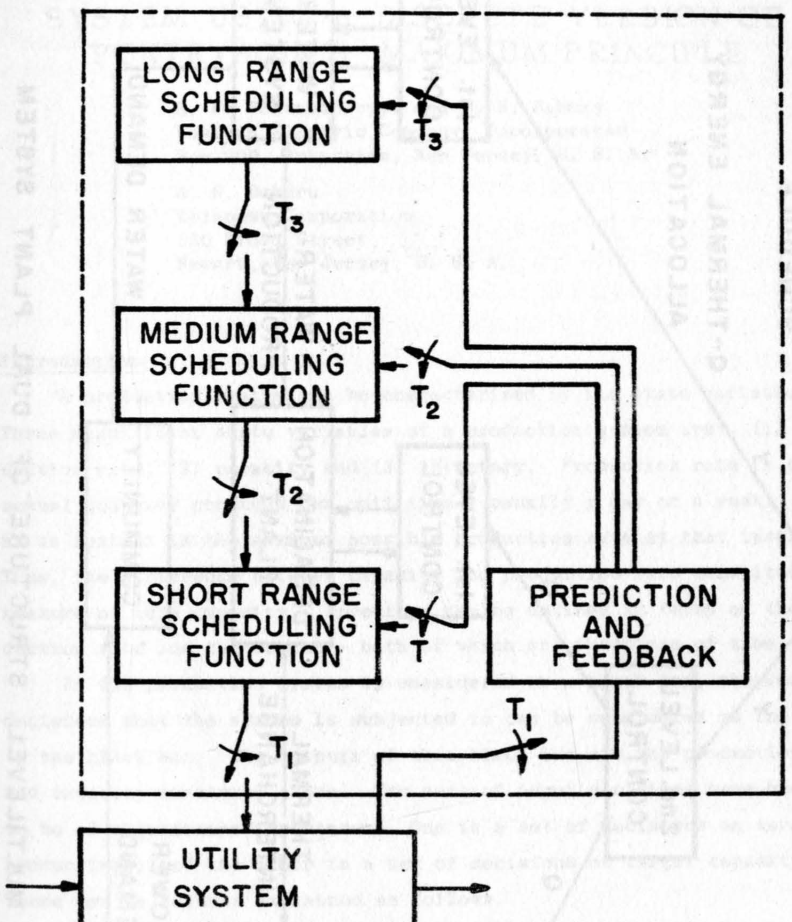


FIG. 5 THE HIERARCHIAL SCHEDULING SYSTEM

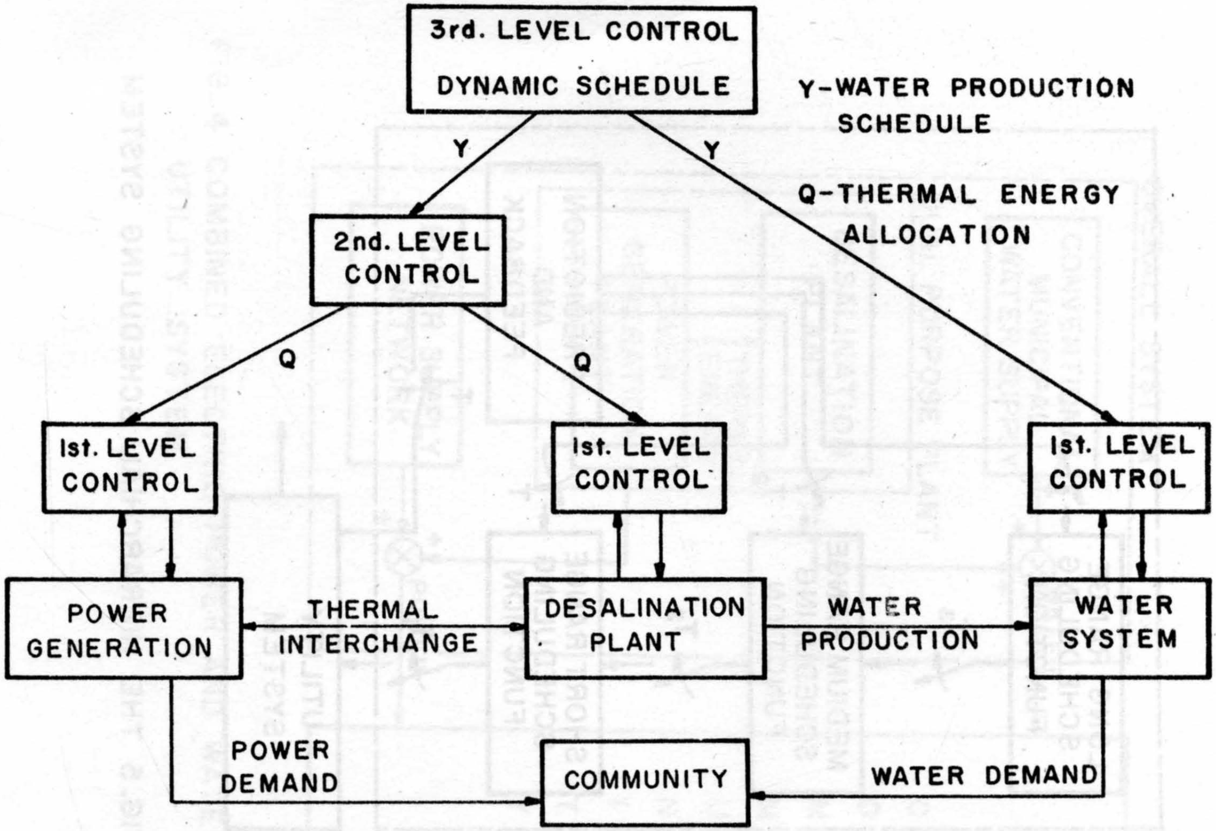


FIG. 6 MULTILEVEL STRUCTURE OF DUAL PLANT SYSTEM

MULTISTAGE OPTIMIZATION OF A PRODUCTION SYSTEM USING A DISCRETE VERSION OF PONTRYAGIN'S MAXIMUM PRINCIPLE

R. K. Bhattacharyya and E. S. Bahary
Western Electric Company, Incorporated
Box 900, Princeton, New Jersey, U. S. A.
A. N. Bakhru
Celanese Corporation
550 Broad Street
Newark, New Jersey, U. S. A.

Introduction

A production system can be characterized by its state variables. Three significant state variables of a production system are: (1) production rate, (2) capacity and (3) inventory. Production rate is the actual quantity produced per unit time - usually a day or a week. Capacity at an instant is the maximum possible production rate at that instant. Thus, the difference between capacity and production rate constitutes a measure of idle capacity. Inventory can be defined in terms of the production rate and the demand - both of which are functions of time.

If the production system is considered as a black box, the various decisions that the system is subjected to can be considered as the inputs to the black box. The outputs of this black box are the production rate and capacity mentioned above. Two sets of input decisions have been found to be of significant importance. One is a set of decisions on target productions, and the other is a set of decisions on target capacities. These can be further explained as follows.

The entire planning horizon of the production system can be divided into several periods of time, such that at the beginning of each period, a decision is taken about the target production during that period. Each such period can be called a production-period. During any particular period, the actual production responds to the decision on the target production of that period according to the system characteristics, and it takes some time to achieve the target production of that period.

Similarly, the entire planning horizon can be divided into several capacity-periods, such that at the beginning of each capacity-period, a decision is taken about the target capacity during that period. The actual capacity during any particular period responds to the decision on the target capacity of that period according to the system characteristics.

Thus, both sets of decisions consist of step functions in the time domain. The duration of a step is equal to the duration of the corresponding period. Experience shows that a production-period is considerably shorter than a capacity-period.

Given the demand forecast over the entire planning horizon, the object is to choose the optimal sequence of the decisions on the various target capacities and the various target production rates of the respective periods comprizing the planning horizon. A sequence of decisions is considered optimal if a certain cost function, to be defined later, is minimized. The optimal solution depends on the system characteristics as well as on the nature of the demand forecast. As for example, for a sluggish system, even in the presence of a highly seasonal demand, the optimal policy may be to maintain the production rate at a more or less constant level throughout the entire planning horizon. On the other hand, for a highly responsive system, which may be more expensive to maintain, it may be possible to vary the production rate in accordance with the seasonal demand - thereby reducing inventory and back order cost. Thus, the cost of improving system characteristics has to be balanced against the added advantage that the improved system can offer. This balance, in general, depends on the nature of demand forecast, and on the cost of improvement.

The model for the production system is structured as a set of difference equations. The optimal solution is obtained by using a discrete version of Pontryagin's Maximum Principle.[1]

Choice of Model

The model for the production system consists of a set of difference equations with the decisions on target productions and target capacities as the forcing functions. The instants these decisions are made, define the initial points of the respective production periods and capacity periods. These periods, instead of being arbitrary, are functions of system parameters. Since a production-period is defined as an interval of time, at the beginning of which a decision on the target production for that period is taken, therefore the length of that production-period should be equal to the time that must elapse before it makes sense to take the next target decision for the next period. Preliminary analysis of some data show that there are a number of production systems where production rate responds almost exponentially to the decision on target production, and asymptotically approaches that target. This exponential response is exemplified by manufacturing progress functions, training curves, etc. In such circumstances, the length of a production period can be defined as the time taken by the system to attain some percentage, say 95% of that target. The exact value of this percentage depends on the noise factor in the production system and also on product complexity, technology, etc.

Following similar arguments, the length of a capacity period is the time taken by the system to attain, for all practical purposes, the target capacity for that period. The capacity of a system usually responds to a decision on target capacity as shown in Figure 1. V_i is a decision on target capacity. $C_{i,0}$ is the initial capacity when the decision is taken. A and B are two critical points in the response. A specifies the point when, after initial delay, lead time etc., the actual growth of capacity acquires momentum. B specifies the point when the target is achieved for all practical purposes, and as such, is the terminal point of the capacity period under consideration.

Thus, the lengths of production periods and capacity periods are characterized by system parameters. Once these parameters are estimated from previous data (to be discussed later), the period lengths can be determined.

Let the entire planning horizon consist of r capacity periods. In general, a capacity period is much longer than a production-period, and as such, overlaps with a number of production periods. For the sake of convenience in describing the optimization scheme, it will be assumed that each capacity period overlaps with a fixed number of production periods denoted by ρ. This assumption is not a necessary condition for the optimization scheme. (In practice, the value of ρ is different for different capacity periods.) Thus, the entire planning horizon consists of $r\rho$ production periods.

The difference equations (1) through (4) representing the model of the production system are discretized in steps of sufficiently small intervals of time - each of length Δt. A typical value of Δt is one day. There are m such intervals in each capacity period, and k such intervals in each production period, where $m = k\rho$.

$$P_{i,j} = P_{i,j-1} + \Delta t \sigma_{i,x}[U_{i,x} - P_{i,j-1}] \tag{1}$$

$$C_{i,j} = C_{i,j-1} + \Delta t \left[\frac{\zeta_i (V_i - C_{i,j-1})}{1 + \xi_i \exp[-(j-1)\Delta t\, \zeta_i]} \right] \quad (2)$$

$$I_{i,j} = I_{i,j-1} + \Delta t \left[P_{i,j-1} - D_{i,j-1} \right] \quad (3)$$

$$Q_{i,j} = Q_{i,j-1} + \Delta t \left[\lambda \sigma^2_{i,x} + F(I_{i,j-1}) + \gamma\, C_{i,j-1} + \eta \left\{ \frac{\zeta_i (V_i - C_{i,j-1})}{1 + \xi_i \exp[-(j-1)\Delta t\, \zeta_i]} \right\}^2 \right] \quad (4)$$

The subscript i,j in the above equations specifies an instant of time that occurs in the i^{th} capacity period. The difference in time between the instant (i,j) and the starting point of the i^{th} capacity period is $j\Delta t$. The starting point of the first capacity period (and as such, of the entire planning horizon) is (0,m). There are m number of intervals - each of length Δt in the first capacity period whose end-points are (1,1), (1,2),...(1,m). The point (1,m) denotes the starting point of the second capacity period. In total, there are r capacity periods. The end point of the r^{th} capacity period (and as such, of the entire planning horizon) is (r,m).

$P_{i,j}$, $C_{i,j}$, $I_{i,j}$ and $D_{i,j}$ in the above equations are respectively the production rate, capacity, inventory and demand rate at the instant (i,j). $Q_{i,j}$ is the total cost starting from the initial point (0,m) up to the instant (i,j). V_i in equation (4) is the decision on target capacity taken at the beginning of the i^{th} capacity period, i.e., at the instant (i-1,m). $U_{i,x}$ in equation (1) is the decision on target production that is in existence at the instant (i,j). Since such decisions are taken at intervals of $k\Delta t$, therefore it follows that, for $x \neq 1$, the decision $U_{i,x}$ is taken at the instant (i,a) where a = (x-1)k, i.e., at the starting point of the x^{th} production period in the i^{th} capacity period. $U_{i,x}$ remains unchanged up to the instant (i,b) where b = xk. For x = 1, the decision $U_{i,1}$ is taken at the instant (i - 1,m). In equations (1) through (4), j runs from 2 to m and i runs from 1 to r. The modifications of these equations are obvious when j = 1. As for example, equation (1) will become:

$$P_{i,1} = P_{i-1,m} + \Delta t\, \sigma_{i,1} [U_{i,1} - P_{i-1,m}]$$

The explanations for equations (1) through (4) are as follows:

Equation (1) is a discrete version of a first order differential equation with $U_{i,x}$ as the input forcing function and $P_{i,j}$ as the output. The steady-state gain of this first order system is unity, and $\sigma_{i,x}$ specifies the time constant. For a given $U_{i,x}$, the greater is the value of $\sigma_{i,x}$, the more responsive is the system, and, as such, generally more expensive to maintain.

The continuous version of equation (2) is given by equation (5).

$$\frac{dC_i}{dt} = \frac{\zeta_i (V_i - C_i)}{1 + \xi_i \exp[-\zeta_i (t - t_0)]} \tag{5}$$

The subscript j has vanished in equation (5). C_i is a function defined in the i^{th} capacity period which starts at (i-1,m) and ends at (i,m). t_0 corresponds to the instant i-1,m. t is a continuous variable which takes the place of $j\Delta t$ of equation (2). Let the initial condition of (5) be given by $C_i = C_{i,0}$ at $t = t_0$. The solution of equation (5) is given by equation (6).

$$C_i = \frac{V_i - [V_i - C_{i,0} - \xi_i C_{i,0}]\exp[-\zeta_i (t - t_0)]}{1 + \xi_i \exp[-\zeta_i (t - t_0)]} \tag{6}$$

Equation (6) is a modified version of Pearl-Reed Curve.[2] Inserting the value of C_i from equation (6) in equation (5) yields equation (7)

$$\frac{dC_i}{dt} = \frac{\zeta_i [V_i - C_{i,0}][1 + \xi_i]\exp[-\zeta_i (t - t_0)]}{\{1 + \xi_i \exp[-\zeta_i (t - t_0)]\}^2} \tag{7}$$

From equations (6) and (7), it can be proved that for $V_i > C_{i,0}$ (Figure 1), there exist a positive real ζ_i and $\xi_i > -1$ such that given any two critical points A and B, satisfying inequality (8), the curve C_i against t will pass through those two points while maintaining the time-derivative of the curve positive everywhere, and asymptotically approaching V_i as $t \to \infty$. If $V_i < C_{i,0}$, the above statement holds with the obvious modification that the time-derivative is negative everywhere. Inequality (8) with reference to Figure 1 can be written as follows:

$$\frac{T_k}{T_j} < \frac{X}{Y} \tag{8}$$

where $X = (V_i - C_{i,j})(C_{i,k} - C_{i,0})$

and $Y = (V_i - C_{i,k})(C_{i,k})(C_{i,j} - C_{i,0})$

T_j and T_k are the abscissae of the points A and B.
To take some typical figures,

$$\text{let } C_{i,j} = C_{i,0} + \frac{5(V_i - C_{i,0})}{100} \tag{9}$$

$$\text{and } C_{i,k} = C_{i,0} + \frac{95(V_i - C_{i,0})}{100} \tag{10}$$

The claim is that some appropriate values of ζ_i and ξ_i can be evaluated which will allow the model to have the usual characteristics of a growth curve (i.e., positive time-derivative and the asymptotic property), provided:

$$\frac{T_k}{T_j} < 361 \quad (\because \text{it follows from equations (9) and (10) that } \frac{X}{Y} = 361)$$

There is a wide class of systems for which the 95% point can be reached in a time less than 361 times the time needed to reach the 5% point. For such systems, equation (2) gives a satisfactory description of the capacity curve.

The proof of the statement following equation (7) can be outlined as follows:

Inserting the values of the coordinates of A and B in equation (6) and solving for ξ_i yields equation (11)

$$\xi_i = \frac{V_i - C_{i,j} - [V_i - C_{i,0}]\exp(-\zeta_i T_j)}{[C_{i,j} - C_{i,0}]\exp(-\zeta_i T_j)}$$

$$= \frac{V_i - C_{i,k} - [V_i - C_{i,0}]\exp(-\zeta_i T_k)}{[C_{i,k} - C_{i,0}]\exp(-\zeta_i T_k)} \tag{11}$$

Rearranging terms of equation (11),

$$X\exp(-\zeta_i T_k) - Y\exp(-\zeta_i T_j) = (X-Y)\exp[-\zeta_i(T_j + T_k)] \tag{12}$$

where X and Y have the above mentioned definitions. Plotting two graphs of the two sides of equation (12) against ζ_i as a variable, it directly follows that if $\frac{T_K}{T_j} < \frac{X}{Y}$, there exists a positive real ζ_i for which the two graphs will intersect, thus ensuring the existence of a solution.

From equation (11) it follows that since ζ_i is real and positive, therefore $\xi_i > -1$. Since $\zeta_i > 0$ and $\xi_i > -1$, therefore it follows from equation (7) that the derivative is positive everywhere for $V_i > C_{i,0}$. (Negative if $V_i < C_{i,0}$). From equation (6), it follows that as $t \to \infty$, $C_i \to V_i$, thus guaranteeing the asymptotic property. The above discussion shows that equation (2) is a satisfactory description for a wide class of systems whose capacity response is as shown in Figure 1.

Equation (3) expresses the inventory at an instant (i,j). Equation (4) expresses the cumulative cost from the beginning of the planning horizon up to the instant (i,j). Thus, $Q_{0,m} = 0$, and $Q_{r,m}$ is the total cost that has to be minimized. F in equation (4) is a function of $I_{i,j-1}$ denoting the cost of carrying the inventory $I_{i,j-1}$. Thus, if $I_{i,j-1}$ is positive, F is the inventory cost. If $I_{i,j-1}$ is negative, F is the backorder cost. If the domain of interest is relatively small, the function F is often approximately linear. Thus

$$F(I_{i,j-1}) = \alpha I_{i,j-1} \text{ if } I_{i,j-1} \text{ is nonnegative}$$

$$= -\beta I_{i,j-1} \text{ if } I_{i,j-1} \text{ is nonpositive}$$

α and β are two positive real constants denoting the coefficients of inventory cost and backorder cost respectively. The derivative of F with respect to $I_{i,j-1}$ does not exist at $I_{i,j-1} = 0$. The following analysis of the optimal policy requires that the derivative of F exists everywhere. For all practical purposes, the cost due to an infinitesimal positive inventory and that due to an infinitesimal backorder are both negligible. Thus $\left.\frac{\partial F}{\partial I_{i,j-1}}\right|_{I_{i,j-1} = 0}$ can be equated to zero without jeopardizing the practical applicability of the analysis.

<u>Restrictions</u>

There are some restrictions that limit the possible range of values of $U_{i,x}$ and V_i

(1) for $j \le k$

$0 \le P_{i,j} \le \text{Min } \{C_{i-1,m} \text{ and } C_{i,k}\}$

(2) for $j > k$

$0 \le P_{i,j} \le \text{Min } \{C_{i,k(x-1)} \text{ and } C_{i,x}\}$

where x has its previous definition

(3) $0 \le C_i \le M$

where M is a positive real constant.

Restrictions (1) and (2) imply that $P_{i,j}$ has to be non-negative. Also, $U_{i,x}$ has to be such that the production rate at any instant in the x^{th} production period cannot be greater than the minimum value of the capacity in the same production period, because capacity is defined as the maximum possible production rate.

The third restriction states that M is the maximum allowable target for capacity due to various physical and economic reasons.

Estimation of Model Parameters

The following is a list of model parameters to be estimated using experience and previous data: $\sigma_{i,x}$, ζ_i, ξ_i, λ, α, β, γ and η. Provision should be made to update these estimations in an adaptive fashion, as new data are acquired. A least square estimation scheme can be stored in the computer as a package. Straightforward application of least square scheme yields the following results:

$$\hat{\sigma}_{i,x} = \frac{\sum\{[P_{i,j} - P_{i,j-1}][U_{i,x} - P_{i,j-1}]\}}{\Delta t \sum[U_{i,x} - P_{i,j-1}]^2} \qquad (13)$$

where $\sum$ is taken over all significant previous data collected for the same value $U_{i,x}$. $\hat{\sigma}_{i,x}$ is the estimated value of $\sigma_{i,x}$.

To evaluate $\hat{\zeta}_i$ and $\hat{\xi}_i$, equations (14) and (15) can be solved by the method of steepest descent

$$\sum\left\{\left[g_p\hat{\zeta}_i - h_p - h_p\hat{\xi}_i \exp(\theta_p\hat{\zeta}_i)\right]\left[h_p \exp(\theta_p\hat{\zeta}_i)\right]\right\} = 0 \qquad (14)$$

$$\sum\left\{\left[g_p\hat{\zeta}_i - h_p - h_p\hat{\xi}_i \exp(\theta_p\hat{\zeta}_i)\right]\left[g_p - h_p\theta_p\hat{\xi}_i \exp(\theta_p\hat{\zeta}_i)\right]\right\} = 0 \qquad (15)$$

where $\sum$ is taken over all significant previous data collected for the same value of V_i, and the subscript p is the running index for the data.

$$g_p = V_i - C_{i,j-1} \;;\; h_p = \frac{C_{i,j} - C_{i,j-1}}{\Delta t},\; \theta_p = -(j-1)\Delta t$$

To evaluate $\hat{\lambda}$, $\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$ and $\hat{\eta}$, the values of $\hat{\sigma}_{i,x}$, $\hat{\zeta}_i$ and $\hat{\xi}_i$ are inserted in equation (4). Relevant data are collected on Q, I, C and V. The least square error function is generated based on equation (4). The partial derivatives of the error function with respect to $\hat{\lambda}$, $\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$ and $\hat{\eta}$ when equated to zero yield five linear equations which can be solved by any standard method.

Optimal Strategy

Given the system as described by equations (1) through (4) with the associated restrictions, the first problem is to find the optimal sequences of $U_{i,x}$ and V_i for a given list of values of $\hat{\sigma}_{i,x}$, and a given sequence of $D_{i,j}$. The same process can be repeated for a few different lists of $\hat{\sigma}_{i,x}$ (each list corresponding to a system) such that the optimal system can be chosen from among the allowable set of systems. A discrete version of Pontryagin's Maximum Principle is used. The state variables $P_{i,j}$, $C_{i,j}$, $I_{i,j}$ and $Q_{i,j}$ shall be resepctively denoted as $Z^q_{i,j}$ where the superscript q goes from 1 to 4. Thus, in general:

$$Z^q_{i,j} = f^q_{i,j}[Z^1_{i,j-1}, Z^2_{i,j-1}, Z^3_{i,j-1}, Z^4_{i,j-1}, U_{i,x}, V_i] \tag{16}$$

where q goes from 1 to 4, i goes from 1 to r, j goes from 2 to m. $f^q_{i,j}$ denotes a function which is continuous in its arguments and whose first partial derivatives with respect to the arguments exist and are continuous in the arguments (as explained earlier). When j = 1, equation (16) becomes:

$$Z^q_{i,1} = f^q_{i,1}[Z^1_{i-1,m}, Z^2_{i-1,m}, Z^3_{i-1,m}, Z^4_{i-1,m}, U_{i,1}, V_i]$$

A hamiltonian is defined as in equation (17).

$$H_{i,j} = \sum_{q=1}^{4}\left[\Psi^q_{i,j}\cdot Z^q_{i,j}\right] = \sum_{q=1}^{4}\left[\Psi^q_{i,j}\cdot f^q_{i,j}\right] \tag{17}$$

where the adjoint variables $\Psi^q_{i,j}$ are defined as:

$$\Psi^q_{i,j-1} = \frac{\partial H_{i,j}}{\partial Z^q_{i,j-1}} \quad \text{as q goes from 1 to 4.} \tag{18}$$

The end conditions of the adjoint variables are given by:

$$\Psi^q_{r,m} = 0 \quad \text{for q = 1,2,3, and } \Psi^4_{r,m} = 1$$

Thus $H_{r,m} \equiv Q_{r,m}$ = total cost over the entire horizon.

Assume an arbitrary sequence of $U_{i,x}$ and V_i in conformance with the previously discussed restrictions. These decisions give rise to a corresponding sequence of $Z^q_{i,j}$. Choose any n^{th} capacity period from among the r periods in the horizon. The decisions for the n^{th} period are V_n and $U_{n,x}$ where $x = 1,2,\ldots,\rho$. Let these decisions be perturbed by the amounts ΔV_n and $\Delta U_{n,x}$ respectively. The consequent change in the overall cost is:

$$\Delta Q_{r,m} \equiv \Delta H_{r,m} = \sum_{q=1}^{4} \left[\psi^q_{r,m} \cdot \Delta Z^q_{r,m} \right]$$

$$= \sum_{q=1}^{4} \left[\psi^q_{r,m} \cdot \Delta f^q_{r,m} \right] \text{[by equation (16)]}$$

$$= \sum_{q=1}^{4} \left[\psi^q_{r,m} \left\{ \frac{\partial f^q_{r,m}}{\partial U_{r,\rho}} \Delta U_{r,\rho} + \frac{\partial f^q_{r,m}}{\partial V_r} \Delta V_r + \sum_{i=1}^{4} \left(\frac{\partial f^q_{r,m}}{\partial Z^i_{r,m-1}} \Delta Z^i_{r,m-1} \right) \right\} \right]$$

$$= \left\{ \sum_{q=1}^{4} \psi^q_{r,m} \frac{\partial f^q_{r,m}}{\partial U_{r,\rho}} \right\} \Delta U_{r,\rho} + \left\{ \sum_{q=1}^{4} \psi^q_{r,m} \frac{\partial f^q_{r,m}}{\partial V_r} \right\} \Delta V_r$$

$$+ \sum_{i=1}^{4} \left[\left\{ \sum_{q=1}^{4} \psi^q_{r,m} \frac{\partial f^q_{r,m}}{\partial Z^i_{r,m-1}} \right\} \Delta Z^i_{r,m-1} \right] \text{(by rearranging terms)}$$

$$= \frac{\partial H_{r,m}}{\partial U_{r,\rho}} \Delta U_{r,\rho} + \frac{\partial H_{r,m}}{\partial V_r} \Delta V_r + \sum_{i=1}^{4} \left[\frac{\partial H_{r,m}}{\partial Z^i_{r,m-1}} \Delta Z^i_{r,m-1} \right] \text{[by equation (17)]}$$

$$= \frac{\partial H_{r,m}}{\partial U_{r,\rho}} \Delta U_{r,\rho} + \frac{\partial H_{r,m}}{\partial V_r} \Delta V_r + \sum_{q=1}^{4} \psi^q_{r,m-1} \Delta Z^q_{r,m-1} \text{[by equation (18)]} \quad (19)$$

By recursion of the above process, equation (19) can finally be expressed as:

$$\Delta Q_{r,m} \equiv \Delta H_{r,m} = \sum_{i=n}^{r} \sum_{j=1}^{m} \left[\frac{\partial H_{i,j}}{\partial U_{i,x}} \Delta U_{i,x} + \frac{\partial H_{i,j}}{\partial V_i} \Delta V_i \right]$$

$$+ \sum_{q=1}^{4} \left[\psi^q_{n-1,m} \Delta Z^q_{n-1,m} \right] \quad (20)$$

where the subscript $x = 1,2,...,\rho$ as explained previously, depending on j.

Since by choice, only V_n and $U_{n,x}$ $(x = 1,2,..,\rho)$ are changed, and since the response of the production system is causal, therefore $\Delta Z^q_{n-1,m} = 0$ for $q = 1,2,3,4$.
Therefore equation (20) reduces to equation (21)

$$\Delta Q_{r,m} \equiv \Delta H_{r,m} = \sum_{j=1}^{m} \left[\frac{\partial H_{n,j}}{\partial U_{n,x}} \Delta U_{n,x} + \frac{\partial H_{n,j}}{\partial V_n} \Delta V_n \right] \quad (21)$$

Equation (21) can be written in an expanded form as equation (22) by assigning the values $1,2,...,\rho$ for x:

$$\Delta Q_{r,m} \equiv \Delta H_{r,m} = \sum_{j=1}^{m} \frac{\partial H_{n,j}}{\partial V_n} \Delta V_n + \sum_{x=1}^{\rho} \sum_{j=(x-1)k+1}^{xk} \frac{\partial H_{n,j}}{\partial U_{n,x}} \Delta U_{n,x} \tag{22}$$

Now, using the usual arguments that appear in literature in connection with the discrete maximum principle (references 1 and 3), it follows that a necessary condition for a V_n^* and a $U_{n,x}^*$ to be optimal is that $\sum_{j=1}^{m} H_{n,j}$ should have a local minimum with respect to V_n at $V_n = V_n^*$ and $\sum_{j=(x-1)k+1}^{xk} H_{n,j}$ should have a local minimum with respect to $U_{n,x}$ at $U_{n,x} = U_{n,x}^*$, where $x = 1,2,\ldots,\rho$

Algorithm[3]

Based on the previous results, the following algorithm can be worked out. Choose a possible value of V_n. For this V_n, choose a $U_{n,\rho}$ and evaluate $\sum_{j=(\rho-1)k+1}^{\rho k} H_{n,j}$. Check if $\sum_{j=(\rho-1)k+1}^{\rho k} H_{n,j}$ has a local minimum at the chosen value of $U_{n,\rho}$. If not, delete this value of $U_{n,\rho}$ and choose a new value of $U_{n,\rho}$ until the local minimum is obtained. Call it $U_{n,\rho}^*$. Keeping $U_{n,\rho}^*$ fixed, find $U_{n,\rho-1}$ that provides local minimum for $\sum_{j=(\rho-2)k+1}^{(\rho-1)k} H_{n,j}$. Call it $U_{n,\rho-1}^*$. Repeating this process, evaluate the U's up to $U_{n,1}^*$ in a backward direction. Now evaluate $\sum_{j=1}^{m} H_{n,j}$, and check if it provides local minimum at the chosen value of V_n. If it does, denote this value of V_n as V_n^*. If not, delete the entire calculation. Choose a new value of V_n, and repeat from the beginning. If an optimal combination of $V_n^*, U_{n,1}^*, U_{n,2}^*, \ldots, U_{n,\rho}^*$ exists, this scheme will eventually detect it.

If the decisions between (n,m) and (r,m) are optimal, then, by principle of optimality, $V_n^*, U_{n,1}^*, \ldots U_{n,\rho}^*$ are optimal in the global sense. Thus, the entire optimal strategy can be worked out sequentially - starting with $n = r$, and working backwards up to $n = 1$.

Results

Some simple results with hypothetical data are shown in Figures 2 and 3. In both Figures 2 and 3, the entire planning horizon consists of 360 days, and the capacity-period consists of 30 days. In each of the two cases, two demand forecasts are considered as denoted by D_1 and D_2. D_1 is constant and equal to 100 over the entire horizon. D_2 fluctuates in a regular fashion between 50 and 150. The system in Figure 2 has a relatively small value of σ and hence is less flexible than the system in Figure 3. The production-period of the sluggish system is 30 days, and that of the more flexible system is 10 days. It is noticed that with the sluggish system, both D_1 and D_2 give rise to the same optimal decisions. However, in the more flexible system, the optimal decisions for D_1 and D_2 are different, provided $\lambda \leq 1000$ (approximately). Thus, with respect to the demand forecast D_2, it pays to change the system from Figure 2 to Figure 3 as long as λ is less than 1000.

References

1. C. L. Hwang and L. T. Fan - A Discrete Version of Pontryagin's Maximum Principle - Operations Research, January-February 1967.

2. H. T. Davis - The Analysis of Economic Time Series - Cowles Commission for Research in Economics, Monograph No. 6 - Principia Press, Inc., 1941.

3. S. S. L. Chang - Computer Optimization of Nonlinear Control Systems by Means of Digitized Maximum Principle - I.R.E. Convention Record 1961.

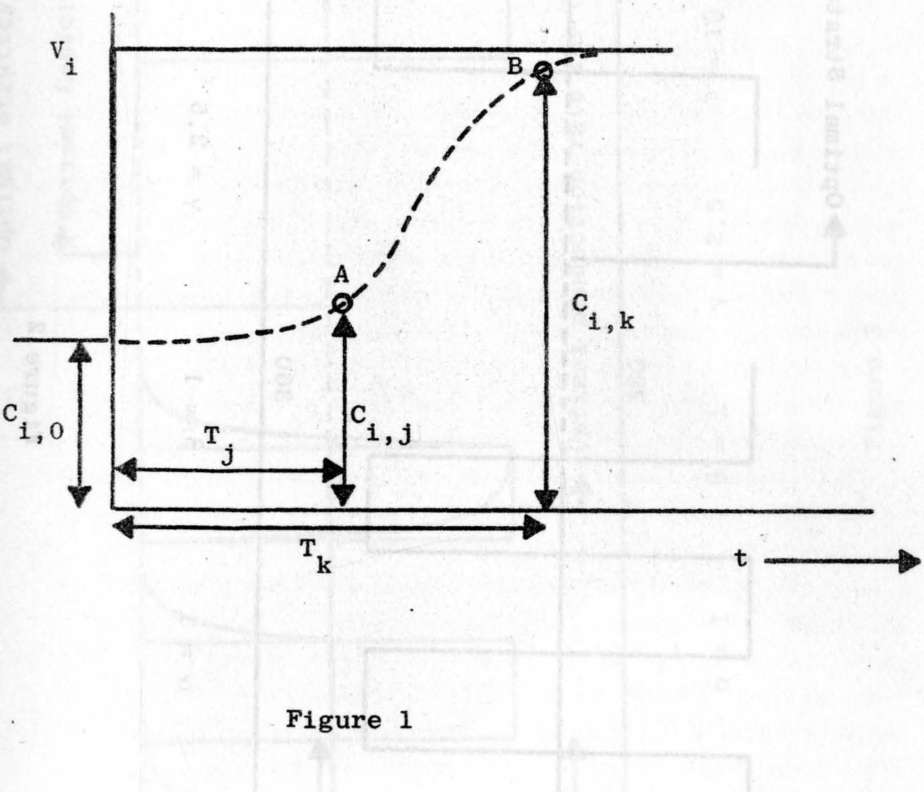


Figure 1

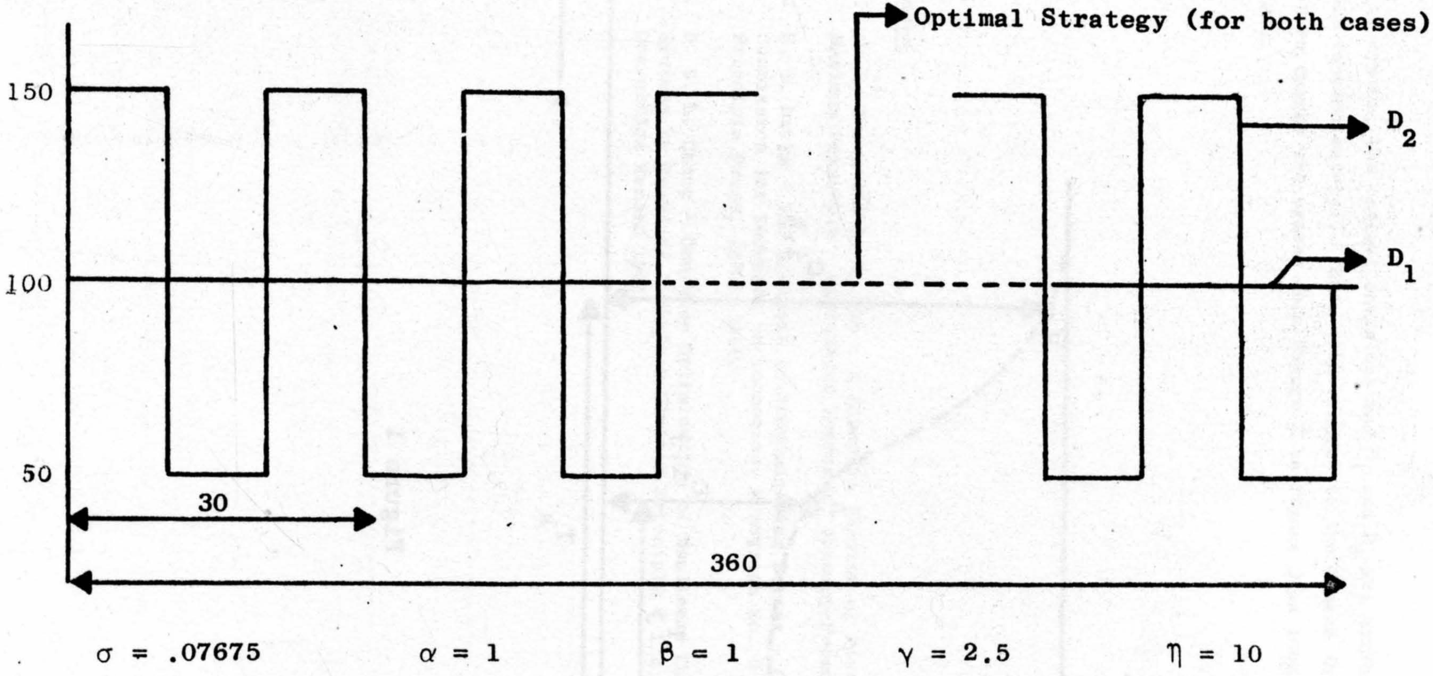
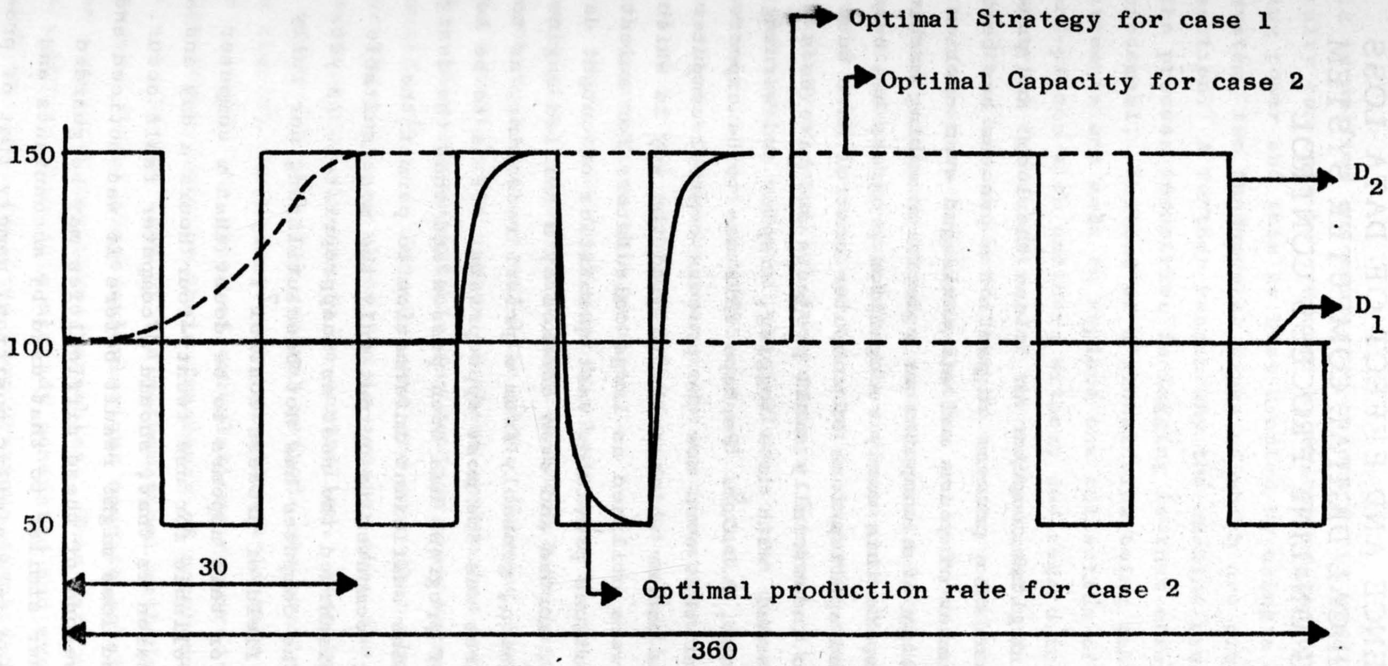

Optimal Strategy (for both cases)
150
100
50
D_2
D_1
30
360
σ = .07675
α = 1
β = 1
γ = 2.5
η = 10

Figure 2



$\sigma = .23025 \qquad \alpha = 1 \qquad \beta = 1 \qquad \gamma = 2.5 \qquad \eta = 10 \qquad \lambda \leq 1000$

Figure 3

THE OCCURRENCE AND EFFECT OF DATA LOSS IN A HIERARCHICAL DIGITAL COMPUTER SYSTEM PROVIDING ON-LINE PROCESS CONTROL

H. H. Johnson,
Imperial College,
London,
United Kingdom.

1. Introduction

The use of a digital computer to 'close the loop' and provide on-line control of a process or part of a process has been practical for a number of years and was envisaged even earlier when the potentiality of a computer as a decision making machin was realised. Despite this early recognition progress has been relatively slow and applications of computer control have been largely limited to industrially small projects and have been particularly concerned with data logging, display and warning systems and, at best, D.D.C.[6]. Perhaps this was to be expected but it would appear that even now the process control computer is being used in a manner which calls to mind the way in which earlier machines were utilised as large calculators for scienti and design calculations providing vast quantities of output dat which was scanned, sorted and even checked by a skilled enginee who made the decision, probably from a priori reasoning, as to which set of figures was the most appropriate. It is to be hop that this stage of progress has been passed and that the design calculation contains sufficient information to permit the 'number-crunching' computer to output only the most suitable figures, having discarded the rest as inappropriate. As yet this ability of the computer has not been utilised, nor fully accepted, in the field of process control[7].

One reason for this appears to be doubt that a computer is sufficiently reliable for use twenty-four hours a day and a closely allied reason is that, should a computer fault occur, serious damage and loss might result before it was noticed and corrected. The second of these difficulties may be guarded against in a manner similar to that used by accountants and bankers when placed in a similar position, namely that of providing checks and rechecks to ensure that incorrect information is identified and corrected before it is used. The first difficulty can only be overcome by the provision of redundant computing power and this is best handled by using a linked computer system, two fundamental types of which are discussed in the next section. A further reason why the on-line use of computers in the process industries is lagging is that effort is being made piecemeal. Instead of a global view being taken of the situation attempts are made to replace one collection of electronic bits-and-pieces with another, without due regard to the infinite superiority of the abilities of the replacement. An underlying cause for this may be that a single computer capable of performing all the necessary data manipulation and control calculations would require enormous storage capacity and an extremely rapid operating speed, possibly unobtainable or even unrealisable. This situation could well be resolved by using a computer system providing the facility of parallel programming. However, it would seem that a particular cause of the failure to use computers to take overall control is that insufficient and imprecise theory is available to allow a would-be user scope to investigate system capabilities and it is one purpose of this paper to establish a basic theoretical approach to this problem.

Much analytical work has been done in establishing algorithms for digital control and many of these have been successfully demonstrated in an off-line mode. However, careful analysis of the results of a number of these approaches indicates clearly that they are unsuitable for on-line work because of timing considerations. Indeed, some are such that the calculation time for the proposed algorithm is greater than the maximum time constant of the process to be controlled. Despite these criticisms a number of powerful general methods are available and in order to clarify subsequent work, an attempt is made here, c.f. Westcott[9], to classify the types of calculations required by various control algorithms, more by their frequency of occurrence and their time of calculation than by the mathematical methods which are employed. First stage calculations operate solely on the data from one sample of monitored variables. This data may, if desired, be combined with data already generated and available directly, which may have resulted from previous sampled-data or from back-up or off-line calculations. The results of these first stage calculations must, in general, be available to the process in the form of control variables within one sampling period. Second stage calculations are those which directly affect the data, other than the sampled-data, used by the first stage calculations In the simplest cases this might involve only the retrieval of data from backing store to make more pertinent values of such variables as physical constants available for direct control calculations. At a higher level new values of this data may require calculation as the process proceeds, either on demand due to the cumulative effect of disturbances or as a continuing development, possibly stimulated by the results of higher stage calculations. Both first and second stage calculations are essentially real time in that response times are known and must be met with some level of statistical certainty.

Higher stage calculations, of which the desired number will be indicated by the construction of the control algorithm, may occasionally be capable of responding to requests for updated information, but more often will initiate the decision to use this data whenever the computation has made it ready for use. Hill-climbing techniques, model improvements, parameter evaluations etc., would commonly be included amongst these stages. These with the former two stages would provide effective control However, a final stage might be considered with the aim of improving the control algorithm itself. This would be fundamental a learning program with the ability to effect and investigate the variation of sampling rates, the alteration of relative emphases on sections of the calculations and, indeed, testing the results of the application of different algorithms.

2. <u>Computer Systems</u>

It has been seen that for a number of reasons a single computer is unlikely to provide the most effective means of controlling a process, although in fairness it should be mentioned that a multiprocessing computer might provide a reasonable answe In general, however, the advantages of linking computers into a comprehensive system outweigh those obtained in using a single complete computer. Particular advantages are that computers may be selected by size and speed to within quite fine limits, thus reducing to a minimum the cost of the equipment. Redundanc may be provided by adding one or two small processors to the system containing numbers of such machines instead of complete duplication necessary with a single machine, again reducing the overall cost and enhancing the chances of the system maintaining at least partial control even under extreme failure conditions.

Two distinct types of computer systems are possible, the unilateral and the hierarchical. In the first of these each computer has access and is accessible to every other machine and commands and data may be passed via a channel, duplicated in case of failure, which is inspected by every computer in the system and action taken if necessary. The advantages of such a system are that should a computer go overdue because of failure or because a calçulation takes an excessively long time to complete, processing on later data may be transferred to another uncommitted machine in the ring. A computer may, without affecting the control, be removed for off-line testing and maintenance and if the computers are identical or similar, engineering problems are simplified. This however raises perhaps the main disadvantage to the unilateral concept, which is that the desirable features of the computer with regard to storage and size and computing speed vary greatly with the type of calculation required, which implies that the unilateral system must contain machines which are dissimilar electronically and use incompatible programming languages.

For a hierarchical system in which a computer at any level is linked to only one computer at a higher level, apart from the case of the hierarchy head, and one or more machines or digital interfaces at a lower level, this lack of compatibility between machines has not such a serious effect, since no overall supervisory system is required and effective action or lack of action becomes dependent on the availability or otherwise of the relevant data. Normal machine executive programs are capable of accepting and transmitting data and the respective interactions of the computers of the system may be serviced on an operation completed basis. This method has the very great advantage that no complicated supervisory program is required and if it assumed that all computers in the system are sufficiently advanced to permit autonomous transfer of data under program initiation, the emphasis for the effective working of the system is on the individual member of the system selecting the appropriate data record for manipulation so as to optimise a function particular to that computer. Furthermore, if it is assumed that data passe down from one level of the hierarchy to the lower level is only of the form of replacement parameters for use in calculations at this lower level immediately reception is complete, then by the Principle of Optimality the optimal processing of data records passed to each individual computer from below will ensur the optimal processing of the system as a whole.

It is worthwhile to note that, although only a true hierarchical system is considered here, certain advantages might ensue by permitting interactions between computers on the same level. This is perhaps particularly true at the lowest level where it might be possible to implement first stage control by means of computers with identical construction, differing only in store size.

3. Selection of a Suitable System

3.1 The Deterministic Case

Clearly the simplest model of a computer system in control of a process is one in which all sampling rates are fixed and all calculation times for any stage are constant. In this case the choice of the optimal distribution of computers in the hierarchy becomes a deterministic problem over a finite time, since for every computer, whatever point of data processing is chosen, this point must repeat at equal intervals of time, although the actual values of the data will, of course, be different. As with every scheduling problem of this type an approach involving a certain amount of empiricism is unavoidable, otherwise the number of possible systems is infinite. Thus it will be assumed here that a reasonable system has been selected and that the amount and type of data processing has been decided for each computer, see Pearson[5]. This involves choice of the appropriate algorithms to produce the desired control, confirming that each algorithm is valid with respect to the timing consideration of speed of calculation to produce adequate response time and selection of which computer in the system is to perform which parts of the calculation, bearing in mind that, in general, the higher the stage of calculation the higher the computer in the hierarch The problem is then to optimise the size and speed and hence minimise the cost of each computer.

The general problem may be stated as follows: for each computer there exist M calculations C(m) ($1 \leq m \leq M$), each with a maximum calculation time TM(m), ($1 \leq m \leq M$) and each calculation is to be repeated N(m) ($1 \leq m \leq M$) times, so that TM(m).N(m) = T, the calculation period for the computer.

If the actual calculation time is taken as a function of the multiply and add times, Tm and Ta viz. TA(m) = A(m).Ta + B(m).Tm and if TE(m,n) is the time earlier than the n^{th} calculation of the type m and TL(m,n) is the time later than the n^{th} calculation of type m so that

$$TE(m,n) + TA(m,n) + TL(m,n) = TM(m,n) = TM(m)$$

since the calculation time is constant, and the corresponding storage requirements are SE(m,n) = SE(m), SA(m,n) = SA(m) and SL(m,n) = SL(m) since the storage requirement for each repeated calculation C(m) is constant.

This may be considered as a problem to find start times, where STE(m,n), STA(m,n) and STL(m,n) are the start times of the pause before the n^{th} calculation of type m, the actual calculation and the wait after the calculation is complete and

$$STE(m,n) = TM(m).n$$

$$TM(m).n \leq STA(m,n) \leq TM(m).(n+1)$$

$$STA(m,n) + A(m).Ta + B(m).Tm \leq STL(m,n) \leq TM(m).(n+1)$$

or concisely to find $XTA(m,n) \geq 0$

$$\text{subject to } XTA(m,n) + A(m).Ta + B(m).Tm \leq TM(m) \quad (1)$$

to minimise the cost function $CA = P.Ta + Q.Tm + R.Smax$ where P, Q and R are constants connecting computer calculating speeds and store size with overall cost and Smax is the maximum store size in use at any time within the calculation period, given the values of Ta, Tm and XTA(m,n). This is a problem of a linear programming form, but is complicated by the fact that only one computer calculation may be carried out at one time and thus the values of XTA(m,n) are mutually interrelated and to vary one may necessitate the variation of others.

Clearly the values of Ta and Tm affect importantly the overall problem, but equally clearly decreases of Ta and Tm cannot increase Smax, so that it is reasonable to select appropriate values, which are sufficiently small to permit all the calculations to be completely performed within the calculation period, to find non-negative values of $XTA(m,n) \leq F(m) = T(m) - A(m).Ta - B(m).Tm$ which are compatible and minimise Smax.

This value of Smax may be used in the equations (1) to improve Ta and Tm and the process repeated until the optimum value of CA is obtained. The value of S at any time t ($0 \leq t \leq T$) is given by the function

$$S = \sum_{m=1}^{M} (\alpha(t,m).SE(m) + \beta(t,m).SA(m) + (t,m).SL(m$$

where one value of $\alpha(t,m)$, $\beta(t,m)$ and $\gamma(t,m)$ is unity and the others are zero for each m and one value of $\beta(t,m)$ is non-zero for all m. Hence in simple cases, for example when SE(m) = SL(m) and M is small, it would be easy to find the value of Smax by inspection. For larger M this is not straightforward and a descent method is given.

It is observed that for fixed Ta and Tm the total calculatio time is fixed and equals

$$\sum_{m=1}^{M} (A(m).Ta + B(m).Tm).N(m) \leq T \qquad (2)$$

which provides a feasibility condition on the values of Ta and Tm and that the calculations fall into three distinct types, those for which SE(m) < SL(m), SE(m) = SL(m) and SE(m) > SL(m). Throughout the time 0 – T of a calculation period the store size varies at times 0, t_1, t_2 .... t_q, t_{q+1} .... t_Q, T, these times being dependent on the start times XTA(m,n) and the store size being constant over each interval (t_q, t_{q+1}). If the scheduling problem of selecting start times is translated into a permutation problem, following Nicholson and Pullen[4], a transformation may be found, which together with the feasibility condition that the n^{th} calculation of type m is performed before the $(n+k)^{th}$ calculation of the same type, for which the optimal solution to the original problem is equivalent to the optimum of the transformed problem. Formally there exists at least one optimal feas-ible solution from the $(\sum_{m=1}^{M} N(m))!$ possible, but not necessarily feasible, permutations; however, it is not often a practical possibility to establish this. The method used by Rule[8] and Nicholson[3] of adjacent interchanges would yield a reasonable level of optimality. However, a better result may be obtained in this case by utilising the fact that only one resource, the computer processor, is considered and that the value of M will, in general, be not too large.

If the value of Smax occurs in the interval (t_p, t_{p+1}), then associated with this interval will be one particular value of XTA(m,n(p)) for each of the M calculations. Over the range $(\min_m(STE(m,n(p)), \max_m(STE(m,n(p)) + TM(m)))$ the single resource, M job and single operation per job scheduling problem is obtained with the conditions

$$TM(m).(n(p) - 1) \leq XTA(m,n(p)) \leq TM(m).n(p)$$

and an objective function CB = Smax. Translating this problem into the corresponding permutation problem it would often be possible to consider all M! permutations and select one which minimises the objective function. Even if this is too large a calculation a cycling procedure is available since the calculations fall into three types, as discussed above, of which one has no direct effect on the store size and the other two are such that an earlier start for one and a later start for the other reduces the store size. Thus, if start time for calculation A is taken to be start time for B, if this reduces Smax and start time for B can be taken as start time for A, if this again reduces Smax, but otherwise start time for B can be taken as start time for C if improving and the process continued until the cycle closes and the start time for Z becomes the start time for A still improving Smax.

This method yields a high level of optimality for each computer of the hierarchy and produces within limits a minimal cost for the complete system. The results are obtained in a straightforward way and at no time do the calculations become long or complicated, or severely restrict the user. Indeed, it would clearly be quite easy to repeat the calculations for a number of varying systems as part of a preliminary investigation.

3.2 <u>The Stochastic Case</u>

In a system which is not fully deterministic a number of changes are noticeable. Firstly, it is not possible to consider the calculation period for any computer because the load will vary statistically with respect to time due to the number of data records being presented for calculation C(m) $(1 \leq m \leq M)$ in any given period not now being constant. Secondly, the time of calculations may vary due, perhaps, to a larger or smaller number of iterations being required to realise acceptable accuracy. Thirdly, the store size, which reflects the fluctuation in the arrival times of data records may be required to hold a number of data records at one time each requiring the same calculation. All of these have a distinct effect on the required size and speed of each computer of the system and although in a number of cases it might be possible to provide a machine large enough and fast enough to handle the worst conceivable case it is certain that the cost of so doing would be very great. Fortunately, however, very few processes are so sensitive that the occasional failure of a controlling system to provide a response to one sample is unlikely to affect them greatly. Similarly for the higher stages of the control calculations, if updated values are not provided for the lower stages, these will continue to provide effective control, although this will perhaps be suboptimal. In a later section the case of critical measurements is considered.

In the discussion that follows, the problem of the queue of data records awaiting processing by one computer is considered initially with respect to the effective loss of data due to the inability of the machine to process the data with sufficient rapidity, the effect of a finite store size is then included in the argument and an illustration is given of the use of Monte Carlo techniques to provide suitable data, so that the method of optimisation for the discrete case may be utilised for the stochastic case. An outline of how the effect of varying calculation times may be handled is given.

3.3 The Queue Model

Consider the following data distribution, the m^{th} data record D(m,n; ta(m,n), tb(m,n); TP(m,n), TR(m,n), w(m,n,TQ(m,n))) which originates from source n at time ta(m,n) and after queuing and processing returns appropriate data to output n, corresponding to source n at time tb(m,n). TR(m,n) is the maximum permissible response time which may elapse between the collection of the record and the return of the corresponding record and the processing time of the record is TP(m,n). It is assumed that each data record is self-contained and may be processed without reference to any other data record. If the situation occurs in which $tb(m,n) - ta(m,n) > TR(m,n)$, then the response signal is too late to serve any useful purpose and the record is deemed to be lost. Further, if a record is queuing and real time t is such that $t - ta(m,n) + TP(m,n) > TR(m,n)$ or $t > TR(m,n) - TP(m,n) + ta(m,n)$, then the record is similarly lost. The problem may then be regarded as that of a queue whose arrival pattern is that arrivals can only occur at a discrete set of time intervals 0, h, 2h .... and the number of arrivals per instant is 0 or 1, a simple version of aggregated arrivals. The service mechanism is initially taken to be one in which the service time of each member of the queue is a known constant regarded as a multiple of the discrete time step, h, the service capacity is unity and the service availability is continuous. The queue discipline is that the head of the queue is serviced first, but that the position in the queue of any member may be changed as specified below. The criterion for success being to minimise

$$\sum_{n=1}^{N} \sum_{m=1}^{M} W(m,n) ,$$

where $W(m,n) = 0$ if the corresponding record is processed and $W(m,n) = w(m,n, TQ(m,n))$ if the record is lost, where $TQ(m,n)$ is the time in the queue before loss occurs.

The generalised notation may be simplified to refer to the members of the queue only, the k^{th} record in the queue $(1 \leq k \leq K)$ is denoted by $D(k,m,n, TP(k), TQ(k), TR(k), w(k,TQ(k)))$ or in the shortened form $D(k)$. It is necessary to make two assumptions at this stage which will clarify the way in which data is handled by the computer. Firstly, at time ph a record is being processed, either processing is well under way or it has just commenced, but in either case a record entering the queue at time ph must wait at least a time h before processing of its data may begin. Secondly, once the processing of a record has commenced it must continue. The following definitions are also important; the free time on any record is given by $TF(k) = TR(k) - TQ(k) - TP(k)$.

A queue member, data record $D(k)$, is said to be valid if, without rearrangement of the queue, it is possible to process that record without loss; this is so if

$$TF(k) \geq \sum_{i=2}^{k-1} TP(i) + TPP \qquad \text{any } k, \ (1 \leq k \leq K) \qquad (3)$$

where TPP is the time required to complete the processing of the record at present being processed, $TPP \leq TP(1)$.

A queue is said to be feasible if every member is valid, that is, if (3) is true for all k, $(1 \leq k \leq K)$. If at least one member is invalid then the queue is said to be infeasible.

An orientated queue is one in which $TF(k) < TF(k+1)$ $(1 \leq k \leq K-1)$ or if $TF(k) = TF(k+1)$ then $TP(k) > TP(k+1)$, note that if both the free time and the processing time are equal the only distinguishing feature is the time in the queue.

Using these definitions a number of theorems may be proved of which the two most important are given below.

<u>Theorem I</u> The positions of the members of any feasible queue may be rearranged so that the resulting queue is feasible and orientated.

Consider the subsequence consisting of the first K_1 members of the queue of K members and assume this is feasible and orientated and that $TF(K_1 + 1) < TF(K_1)$ then there must be a $K_1^*(2 \leq K_1^* < K_1)$ such that $TF(K_1^*) < TF(K_1 + 1) < TF(K_1^* + 1)$, ignoring in this proof the possibility of equal free times, then

$$TF(K_1^* + 1) > TF(K_1 + 1) \geq \sum_{i=2}^{K_1+1} TP(i) + TPP$$

$$> TP(K_1 + 1) + \sum_{i=2}^{K_1^*+1} TP(i) + TPP$$

and therefore $D(K_1^* + 1)$ is still valid with the $D(K_1 + 1)$ record inserted, and, in general, for $K_1^* + j < K_1$; $j \geq 1$

$$TF(K_1^* + j) > TF(K_1 + 1) \geq \sum_{i=2}^{K_1+1} TP(i) + TPP$$

$$> TP(K_1 + 1) + \sum_{i=2}^{K_1^*+j} TP(i) + TPP$$

Hence the member $D(K_1 + 1)$ may be removed from its position and inserted between $D(K_1^* + 1)$ to form an orientated and feasible subsequence of length at least $K_1 + 1$. Note, should no K_1^* exist, then a similar argument may be used to show that the member $D(K_1 + 1)$ may be removed to the head of the queue, again to form a feasible and orientated subsequence of length not less than $K_1 + 1$.

Either the whole queue is now feasible and orientated or there exists a subsequence of length $K_2 \geq K_1 + 1$ which is, and such that $TF(K_2 + 1) < TF(K_2)$ whence by the same argument a feasible orientated subsequence of length $K_2 + 1$ or greater may be generated. Since the queue is of finite length repeated applications yield subsequences of lengths at least $K_3 + 1$, $K_4 + 1$, etc. until a feasible orientated queue is obtained.

Lemma I. An orientated queue is a unique arrangement of any queue. For clearly the size of the queue is fixed and the values of TF(k) and TP(k) are fixed, thus any other arrangement cannot satisfy the conditions specifying orientation; the case for members with identical free times and processing times is not handled.

Theorem II If an orientated queue is infeasible then it may not be rearranged so that the resulting queue is feasible.

Assume that a feasible queue may be found, then by Theorem I this may be arranged as an orientated feasible queue, but this is impossible since the unique orientated queue is known to be infeasible. Hence the result.

Clearly from these two theorems at time ph^+ the position is that if the queue has been orientated and is feasible, processing may be allowed to proceed smoothly, each data record being processed as it reaches the head of the queue until such time as an additional record enters the queue. When this occurs the queue may again be orientated and if the result is feasible processing still continues smoothly. If, however, the result in now infeasible it may be shown (Theorem III, proof not given) that data loss must occur. The timing of the decision to lose one or more records and which records these shall be may obviously be made at any time after the infeasibility state has been attained, but although there are certain advantages in leaving the decision to times (p+1)h, (p+2)h .... so as to see what other records are forthcoming and in the case of variable calculation times additional processor time may become available, it would appear definitely more suitable to make any queue, which becomes infeasible, feasible as soon as possible. The advantages of this are that a decision need only be taken when a new record joins the queue and then only if the queue becomes infeasible. Since the queue is orientated to test for feasibility a simple search method is available to establish which records must be removed from the queue to minimise $\sum_{k=1}^{K} W(k)$ the modified criterion. When the records have been lost the queue may be rearranged in the appropriate feasible form for processing to continue until a further record is received, in many cases it would appear that the orientated form is most appropriate.

A further advantage occurs when a finite store is considered, as in this case when a new record is received it is not always physically possible to maintain all the records in store and therefore a decision must be made immediately. This may be done by considering the extended function w(k,TQ(k),S(k)) where S(k) is the store size required by D(k), but in this case the search method to minimise the criterion becomes two-dimensional and care must be taken that this does not take too long.

3.4 System Design in the Stochastic Case

The problem now is to apply the results of the previous section to the selection of computers of optimal size and speed. It may be possible to specify probability distributions for each type of data record analytically and to develop a full mathematical solution to the problem, but this is unlikely to be of great value since, apart from the complexity of the analysis, it is probable that from the nature of the problem that the distributions do not correspond nicely to the standard ones, but result from overlapping increases in demand for responses to data records with rapidly changing inputs for various calculations. In this case the simplest approach is a Monte Carlo simulation method in which the effect of various overlapping increases in speed are tested and the number of data records for each type of calculation may be recorded and an estimate arrived at for a suitable store size and calculation speed. In doing this simulation it is important that data records that would be lost are actually discarded as each discarding will have a significant effect on the remaining data entering the queue. It would also be possible to simulate the complete hierarchy as a series of queues, see, for example, Cox and Smith[2].

This method would provide a good solution, but it is possible to adapt the discrete model in a way which would give relatively good results with considerably less work. This consists simply of increasing the number of calculations N(m) by a selected factor K(m) so that in the new problem the number of calculations is taken to be $N^*(m) = K(m).N(m).(1 - \emptyset(m))$, where it is assumed that on average a proportion $\emptyset(m)$ of the records are lost; this adjusting factor could obviously be modified to include the standard deviation of the distribution. Since at any time only one data record of these may be processed new values for the storage requirements are given by $SE^*(m) = K(m).SE(m)$, $SA^*(m) = (K(m) - 1).SE(m) + SA(m)$ and $SL^*(m) = 0$ since in the queue model it was assumed that the response data was returned immediately after calculation. For any type of calculation T(m) is selected to be the same value as in the discrete case, but over each time interval of this size $K(m).(1 - \emptyset(m))$ calculations must be performed, or if $[K(m).(1 - \emptyset(m))] = P(m) - 1$ denotes the greatest integer less than the value in square brackets, P(m) calculations must be performed. In this case, there must be P(m) start times STA(m,n,p) over each interval where, as before,

$$XTA(m,n,p+1) > XTA(m,n,p) \geq 0 \quad (1 \leq p \leq P(m))$$

$$XTA(m,n,p) + A(m).Ta + B(m).Tm \leq XTA(m,n,p+1) < TM(m)$$

to minimise the cost CC = P.Ta + Q.Tm + R.Smax.

This problem may be solved in a manner similar to that applied in the discrete time case, but if it is felt that the increased number of variables from $\sum_{m=1}^{M} N(m)$ to $\sum_{m=1}^{M} N(m).P(m)$ makes the effort too great, it would be reasonable to reduce the size of the calculation by assuming that in any interval T(m) once a calculation of type C(m) has commenced all the calculations of this type are completed.

Thus $XTA(m,n,p) + A(m).Ta + B(m).Tm = XTA(m,n,p+1)$

and $XTA(m,n,1) + P(m).A(m).Ta + B(m).Tm) \leq TM(m)$

In either case an additional calculation must be carried out to ensure that the calculated start times do not cause a greater proportion of the data records to be lost than was originally acceptable.

3.5 <u>Essential Data Records</u>

In certain cases it may happen that the inability of a computer to store or process all records emanating from a particular source might have a critical effect on the control of the process. Using the model suggested it may be possible to guarantee that no loss occurs by choosing the values of $w(m,n^*, TQ(m,n^*), S(m,n^*))$, where n^* is the pertinent source, to be sufficiently great to ensure that these records are never dispensed with in preference to records from any other source. Even so it might happen that only records from this source were in the queue and still data loss was demanded. In this case it is suggested that a different model should be constructed, and clearly modifications to fit other situations could be based on this, on the assumption of a two-stream system similar to that proposed by Anis and El-Naggar[1] and akin to the two-bin inventory system discussed by Whitin[10]. This consists of two queues; on the one of lesser importance is a store of size s, records from this are allowed to proceed into the main stream store of size S when conditions are suitable. The value of S and the ability of the computer processor must be sufficient to cope with the worst case of the essential records and the size of the subsidiary store will vary depending on how much use is being made of the main stream store; thus if its minimum size is s and the main store actually contains T the effective size of the subsidiary store is $t = S + s - T$. Thus two feasible queues may be obtained with one fed from the other when conditions permit.

4. Conclusions

It has been seen that it is eminently possible to establish and use optimal design criteria for a hierarchical computer system controlling on-line a process for which the type and number of calculations has been fully determined.

Further it has been shown that by using a queue model for the flow of data records to each computer of the system the method may be adapted to handle the case when sampling rates and requests for updated parameters vary with time and effective data loss may occur. In this case the criterion used to decide which data records must be discarded may be selected so that the overall effect on the control of the process is minimised.

References

1. ANIS, A.A. & EL-NAGGER, A.S.T.: The Storage-Stationary Distribution in the Case of Two Streams; Jour. I.M.A., Vol. 4, No. 2, (1968)
2. COX, D.R. & SMITH, W.L.: Queues; Methuen (1961)
3. NICHOLSON, T.A.J.: A Sequential Method for Discrete Optimisation Problems etc; Jour. I.M.A. (1967)
4. NICHOLSON, T.A.J. & PULLEN, A.D.: A Perturbation Procedure for Job-Shop Scheduling; Comp. Jour. Vol. 11, No. 1, (1968)
5. PEARSON, J.D.: Multi-level Control Systems; Proc. IFAC (Teddington) Symp. (1965)

6. ROBERTS, J.P. et al: How much less are computers doing than they could?; Control (March 1965)
7. ROSENBROCK, H.H. & YOUNG, J.A.: Real-Time On-Line Digital Computers; 3rd IFAC Congress (1966)
8. RULE, B.F.: Optimal Lecture Room Allocation by Digital Computer; M.Sc. Thesis, Loughborough University of Technology (1966)
9. WESTCOTT, J.H.: Application of Optimal Methods to Control of Industrial Processes; Proc. I.B.M. Symp. (1964)
10. WHITIN, T.M.: Inventory Control Research: A Survey, Management Science 1, (1954)

СИСТЕМЫ И АЛГОРИТМЫ УПРАВЛЕНИЯ ДЛЯ СЛОЖНОГО КОМПЛЕКСА МЕТАЛЛУРГИЧЕСКОГО ПРОИЗВОДСТВА

А.П.Копелович, А.А.Белостоцкий,
Б.А.Власюк, Г.И.Никитин, В.М.Храпкин
(ЦНИИКА, Москва, СССР).

Развитие вычислительной техники позволило рассмотреть задачу автоматизации управления сложными производственными комплексами - такими, например, как крупные цехи и заводы.

Для современных крупных металлургических заводов характерны большие объемы производства, широкий ассортимент продукции, тесная взаимосвязь работы разных цехов между собой (обусловленная, в частности, потоком горячего металла), высокий уровень возмущающих воздействий.

Эти и другие факторы вызывают повышенные требования к системам управления и планирования для металлургического завода: сложные и большие по размерности задачи должны решаться за короткие отрезки времени, а ошибки в управлении влекут за собой крупные производственные потери.

Улучшение качества управления металлургическим производством можно получить при использовании современных средств переработки информации - электронных вычислительных машин (ЭВМ).

Одним из наиболее экономически перспективных направлений является автоматизация оперативного планирования и управления производством комплекса "сталь-прокат" (сталеплавильное - прокатное производство)[I].

На рис.I представлена в качестве примера структурная схема комплекса с одним мартеновским цехом, одним обжимным станом и двумя сортовыми прокатными станами (рельсобалочным и крупносортным).

Все основные агрегаты комплекса "сталь-прокат" отличаются циклическим характером производственного процесса. Каждый цикл работы агрегата представляет собой процесс обработки "детали" - дискретной порции металла (плавки, слитка, блюмса).

В рамках доклада не представляется возможным привести подробную математическую модель работы завода. Отметим лишь, что она состоит из трех основных взаимосвязанных частей: совокупности моделей процессов, протекающих на каждом цикле работы каждого агрегата; условий, определяющих технологический маршрут и ограничения на последовательность обработки "деталей"на агрегатах завода; и условий, обеспечивающих выполнение заданного объема и ассортимента продукции.

Общая задача управления производством состоит в том , чтобы выбрать управляющие воздействия в каждом процессе; составить график работы каждого агрегата (определить характеристики обрабатываемой продукции, моменты начала и конца каждого цикла работы агрегатов) и определить размеры дискретных порций для каждого цикла работы каждого агрегата.

Общее число "машин", которые участвуют в обработке металла на рассматриваемом участке - около двухсот, число "деталей" - плавок, обрабатываемых в течение месяца на участке, - достигает 2000. Завод работает по заказам, которые поступают из центрального планирующего органа; число заказов на квартал составляет несколько тысяч.

Постановка и метод решения задачи управления

В качестве критерия оптимальности управления принят доход, полученный заводом при выполнении месячных заказов:

$$\sum_i C_i(t_i)\gamma_i + y_T - KT - \sum_j b_j x_i \rightarrow max \qquad (I)$$

где: $C_i(t_i)$ - функция стоимости i-ого заказа в зависимости от срока его выполнения t_i;

y_T - стоимость незавершенного производства;

γ_i - объем i-ого заказа;

T - момент выполнения всего объема заказов;

K - стоимость единицы времени работы завода;

x_j - количество потребляемого ресурса j;

b_j - цена ресурса j.

Первый член критерия (I) представляет собой стоимость всех заказов с учетом снижения ее при невыполнении заказов в срок; второй член критерия стимулирует производство на предыдущих участках технологической цепочки при заданном объеме конечных для завода изделий; третий и четвертый - требуют минимизации простоев оборудования завода и потребяемых ресурсов.

Анализ задачи управления заводом показывает, что она не принадлежит к какому-либо известному классу экстремальных задач; в ней сочетаются комбинаторика, элементы вариационных задач и задач математического программирования. Сложность самой природы задачи и ее необозримо большой размер требуют нереальных вычислительных средств для получения точного решения и его коррекции по возмущениям.

Основными путями преодоления "проклятия размерности" подобных задач являются: максимальное использование специфики конкретной задачи, т.е. упрощающих особенностей, которые дают структура объекта и критерий оптимальности; применение приближенных методов решения, в которых используется уже накопленный опыт управления данным производством (задание приоритетов, разумное ограничение множества вариантов и пр.).

Упрощающие особенности задачи возникают из-за того, что связи между отдельными частями крупного производственного комплекса оставляют определенную свободу выбора при выполнении отдельных операций. Некоторая независимость в управлении имеется также в отдельные интервалы времени общего периода планирования.

Особенности задачи управления математически выражаются в специальной структуре системы уравнений и неравенств, составляющих описание объекта управления, например, в блочности этой системы, что позволяет привлечь к решению задачи управления специальные вычислительные алгоритмы, которые экономичнее общих методов решения: методы декомпозиции решения блочных задач линейного программирования[2,3], локальные алгоритмы решения целочисленных задач[4], динамическое программирование для объектов с марковским свойством[5], схему последовательного анализа вариантов[6] и т.д.

Для применения таких вычислительных алгоритмов требуется представить структуру модели объекта и выбрать для конкретной структуры объекта экономную структуру решения задачи.

Определение структуры модели и структуры решения целесообразно проводить в несколько приемов[7]. На первом этапе весь объект управления разбивается на сравнительно небольшое число частей - блоков. Совокупность блоков и их взаимосвязи (условия, объединяющие переменные различных блоков) представляют собой структуру объекта на первом этапе. На основании анализа этой структуры выбирается экономная структура решения задачи: некоторое множество подзадач управления и их взаимосвязь в процессе решения.

На втором (и последующих) этапах рассматриваются уже сформулированные на предыдущих этапах подзадачи управления, причем для каждой подзадачи детализируется структура ее объекта и выбирается структура решения.

Работу металлургического завода в течение месяца представим как сложный многошаговый процесс. Разбиение общей модели работы завода на блоки осуществляют по отдельным участкам производства и по интервалам общего периода времени планирования. Отдельный блок общей структуры представляет собой описание работы одного участка завода на одном подинтервале времени.

Размеры участков производства, которые рассматриваются в одном блоке, и величины подинтервалов планирования желательно выбирать, с одной стороны, так, чтобы общее число блоков было не очень большим и позволило найти более экономичную последовательность решения формулируемых подзадач управления;с другой стороны, так, чтобы число параметров, связывающих различные блоки между собой, было бы по возможности меньше.

Общая сформулированная структура объекта, рассматриваемая на первом этапе разложения, представлена на рис.2, где блок характеризуется индексами κ - номер подинтервала времени и ρ - номер участка. Здесь $\rho = 1$ участок мартеновских печей; $\rho = 2$ - участок блюминга; $\rho = 3$ - участок сортового стана № I и $\rho = 4$ - участок сортового стана № 2.

Каждый блок имеет два типа связей: "статические" - связи с другими блоками (участками) в одном подинтервале времени планирования и "динамические" - с блоками, представляющими описание того же участка производства в другие подинтервалы.

Разложение задачи управления состоит в формулировке локальных подзадач управления отдельными блоками и определении способа координации решений отдельных подзадач с целью оптимального решения общей задачи управления.

Локальные подзадачи управления блоками можно получить, используя принцип оптимальности динамического программирования: "любой процесс, происходящий между двумя фиксированными конечными точками, должен управляться оптимально", т.е. при фиксированных входных и выходных координатах для данного блока оптимальными могут быть только те варианты управления им, при которых доход блока максимален.

Локальные подзадачи - задачи оперативного управления; они состоят в определении режимов операций и последовательности операций на агрегатах участка при заданных начальных состояниях всех агрегатов, заданных графиках поступления "деталей" на участке, заданном времени окончания операций на агрегатах.

Координирующие подзадачи управления - задачи оперативного планирования; они должны определить перечисленные выше заданные параметры для локальных подзадач управления. Большая размерность общей задачи координации также требует разложения ее на отдельные подзадачи, в каждой из которых разрешается только часть связей между блоками.

Каждую последовательность разрешения связей между блоками можно характеризовать определенными затратами на решение задачи на ЭВМ (объемом требуемой памяти ЭВМ, временем решения и др.). Для рассматриваемых задач затраты на решение определяются числом параметров. Среди всех порядков разрешения связей между блоками целесообразно выбрать такой, в котором суммарное число параметров во всех параметрических координирующих подзадачах было бы минимально.

Построенная на основе описанных методов структура решения задачи управления металлургическим заводом представлена на рис.3. Блоки I ÷ 9 образуют систему оперативного планирования производства, блоки I0 ÷ I3 - систему оперативного управления мартеновским цехом, блоки I4 ÷ I6 - систему управления участком подачи металла в отделение нагревательных колодцев, блоки I7 ÷ 20 - систему управления обжимным станом и блоки 2I ÷ 24 - систему управления сортовыми станами.

Ниже рассматриваются разрабатываемые в ЦНИИКА основные подсистемы комплекса "сталь-прокат", находящиеся на различных иерархических уровнях.

Система оперативного планирования производства

Одним из экономически перспективных направлений применения современных быстродействующих ЭВМ является автоматизация оперативного планирования работы металлургического предприятия.

Общая задача оперативного планирования металлургического производства состоит в следующем: исходя из совокупности заказов потребителей, текущего состояния оборудования и незавершенного производства составить оптимальные планы-графики работы основных металлургических агрегатов[8].

Под оптимальным графиком работы агрегатов понимается такой график, который, во-первых, согласован с графиками работы других агрегатов и участков и, во-вторых, минимизирует потери завода, связанные с простоями оборудования, невыполнением сроков отгрузки заказов, охлаждением горячего металла и др. Многочисленные возмущающие воздействия требуют достаточно частой коррекции составленных ранее графиков. Поэтому автоматизация планирования рассматривалась как создание системы планирования, включающей систему сбора и передачи производственной информации, системы алгоритмов, по которым производятся расчеты и коррекция оптимальных графиков, и, наконец, системы передачи запланированных графиков производственному персоналу.

Завод работает по месячным заказам, которые поступают на завод раз в квартал из центрального планирующего органа: число заказов на квартал составляет несколько тысяч. Заказы поступают, в основном, на сортовой прокат, однако завод может продавать и промежуточные продукты - слитки стали, блюмсы.

Качество всех планируемых графиков оценивалось по единому для всего завода критерию оптимальности. В качестве критерия оптимальности был принят доход, получаемый заводом при выполнении месячного портфеля заказов.

Общая задача определения оптимальных графиков работы всех агрегатов исходя из месячного портфеля заказов характеризуется очень большим числом переменных, сложными и многочисленными связями между ними. Поэтому централизованное решение такой задачи даже с помощью современных ЭВМ невозможно и необходимо разложение общей задачи на ряд взаимосвязанных подзадач меньшей размерности.

При разложении выбор системы алгоритмов оперативного планирования проводился исходя из следующих принципов: структура алгоритмов должна быть иерархической; оптимальная структура алгоритмов должна минимизировать общие затраты на решение задачи; воздействиями, которые вышестоящий уровень планирования вырабатывает для нижестоящей, являются либо планы работы тех или иных агрегатов, либо цены промежуточных продуктов.

Принятая в результате структура предусматривает следующие основные алгоритмы:

I) формирование множества партий, т.е. совокупности заказов, прокатываемых на сортовых станах, на одном комплекте валков. Задача сформулирована как комбинаторная задача, решаемая с помощью эвристического алгоритма, основанного на правиле приоритетов.

2) Определение последовательности проката партий на сортовых станах. Для ее решения разработан алгоритм, сочетающий динамическое программирование с направленным перебором.

3) Календарно-объемное планирование комплекса "сталь-прокат"; определяются суточные заказы производства стали в мартеновском цехе, проката металла на блюминге и сортовых станах, отгрузки металла. Для решения применен блочный метод линейного программирования.

4) Планирование суточного графика работы мартеновского цеха: распределение марок стали по печам и по номерам плавок на каждой печи внутри суток. Задача решается как задача целочисленного линейного программирования.

5) Планирование суточного графика обработки плавок на участке блюминга. Алгоритм составлен в результате формализации правил, используемых заводским персоналом.

6) Планирование суточного графика работы сортовых станов. Алгоритм основан на эвристических правилах.

Оценка эффективности перечисленных алгоритмов планирования показала, что система позволяет повысить производительность сортовых станов, мартеновского цеха и блюминга на 2,5 - 3%.

Система оперативного планирования для решения задач использует ЭВМ вычислительного центра металлургического завода. Ввод информации о ходе производства осуществляется с телетайпов, установленных в мартеновском цехе, на блюминге и сортовых станах. Вывод информации осуществляется на печатающее устройство вычислительного центра и непосредственно диспетчерскому персоналу производства.

Система оперативного управления мартеновским цехом

Автоматизированная система оперативного управления создается для цеха, имеющего несколько мартеновских печей, шихтовый двор, миксерное отделение, печной и разливочный пролет[I]. Задача управления цехом ставится как задача максимизации условной прибыли цеха:

$$П = \sum_j c_j q_j - \sum_i \kappa_i \tau_i - a \sum_i g_i \to max \qquad (2)$$

где q_j - объем производства j -ой марки стали;
τ_i - затраты времени i-ой печью на производство;
g_i - расход технологического кислорода i-ой печью;
c_j - относительная стоимость j-ой марки стали;
κ_i - относительная стоимость единицы времени работы i-ой печи;
a - условная стоимость технологического кислорода.

Принятая форма критерия управления мартеновским цехом обладает достаточной гибкостью и при соответствующем выборе коэффициентов может быть сведена к различным частным формам.

Линейный характер критерия дает значительные удобства при решении конкретных задач оптимизации.

Система управления реализует следующие алгоритмы:

I) Планирование текущего графика работы мартеновских печей: для каждого периода плавки на интервале планирования устанавливается некоторый режим расхода кислорода; совокупность планируемых периодов плавок на всех печах образует последовательность заявок на обслуживание печей вспомогательным оборудованием. График обслуживания заявок составляется по каждому типу оборудования отдельно; при распределении оборудования учет графика обслуживания печей уже распределенным ранее оборудованием осуществляется с помощью приписываемого каждому периоду плавки числового параметра - "резерва времени" (разность между моментом поступления заявки на обслуживание и временем освобождения выданного для обслуживания этой заявки оборудования).

После распределения всех видов оборудования произво - дится перераспределение кислорода, таким образом, чтобы сум - марный расход кислорода на цех в каждый момент времени не превосходил заданной величины. При этом увеличивается расход кислорода периода плавки с отрицательным резервом времени и уменьшается на период с положительным резервом времени. После перераспределения кислорода вновь производится распределение вспомогательного оборудования. В соответствии с графиком работы печей рассчитывается график подачи к печам шихты и составов под разливку, а также график выполнения прочих технологических операций на печах.

2) Определение оптимальной стратегии "разведения" печей при совпадении завалок на соседних печах, которая состоит в задании для каждого возможного значения величины интервала между началами завалок на соседних печах очередности проведения завалок и веса плавок. Задача формулируется как задача поиска оптимального управления для управляемой марковской цепи и решается при помощи модификации метода Говарда.

3) Управление вспомогательным оборудованием: задача состоит в назначении конкретного оборудования (завалочной машины, разливочной площадки и др.) на выполнение данной операции, а также в определении задания на очередную подачу чугуна к печам. Распределение оборудования производится в соответ - ствии с приоритетами печей, задаваемыми "резервами времени".

4) Контроль хода производства путем сравнения заданных алгоритмов графиков выполнения операций с фактическим их исполнением и оперативный учет производства.

Автоматизированная система предусматривает централизацию управления цехом. На диспетчерском пункте цеха сосредотачивается вся информация о ходе производства. Информация о ходе производства вводится как автоматически, так и вручную.

Управление движением составов и локомотивов будет осуществляться при помощи маршрутно-релейной централизации и устройств поездной информации, обеспечивающих слежение за номерами составов и локомотивов.

Центральной частью системы является вычислительная машина. ЭВМ должна работать в реальном масштабе времени,обеспечивая мультипрограммный режим работы с прерыванием для ввода-вывода информации. Быстродействие ЭВМ - порядка 30.000 коротких операций в секунду, память - около 20.000 24-разрядных слов.

Экономический эффект от внедрения системы достигается за счет увеличения производительности цеха. Кроме того, ожидается дополнительный эффект за счет снижения расходов топлива и уменьшения потерь тепла слитков вследствие более равномерного выхода плавок из цеха.

Система управления участком подачи металла в ОНК

Система оперативного управления подачей металла в отделение нагревательных колодцев предназначена для участка завода, охватывающего пути отстоя горячих плавок, стрипперное отделение, двор изложниц, а также склад холодных слитков. Особенностью участка является тесное переплетение технологических и транспортных операций[9].

В функции рассматриваемой системы управления входит :

I) Организация движения потока составов с плавками в ОНК в соответствии с заданиями центральной системы планирования.

Основным критерием при организации движения является минимизация отклонений от порядка поступления плавок с учетом необходимости своевременной подачи составов под оборудование во двор изложниц. При этом исходят из реальной транспортной ситуации на участке: занятость транзитных путей, возможность подхода локомотива к составу или необходимость маневровой операции, наличие свободного локомотива и т.п.

Критерии решения каждой из частных задач управления - выбор локомотива, оптимального маршрута и т.п. определяются спецификой путевого развития на участке управления, однако во всех случаях система стремится обеспечить передвижение состава за минимальное время.

2) Формирование потока составов, направляемых во двор изложниц для оборудования. Критерием решения этой задачи является минимизация отклонений времени подачи состава под разливку от графика выпусков плавок в мартеновском цехе.

3) Оптимальное распределение работы между локомотивами, которое может быть сведено к транспортной задаче линейного программирования. Критерием решения этой задачи является минимизация времени простоев составов в ожидании локомотива и холостого пробега локомотивов.

По результатам решения этих задач составляется план - график работы участка на некоторый интервал времени. Частичная или полная коррекция графика производится при каждом изменении состояния объекта.

В соответствии с планом-графиком устройствам железнодорожной автоматики на участке управления выдаются в заданное время команды об автоматическом наборе маршрута движения (включении соответствующих стрелок и сигналов, разборке выполненного маршрута) и осуществляется передача команд о движении на локомотив.

Команды об очередности оборудования и стripперования составов выдаются машинистам локомотивов и диспетчерскому персоналу в стripперное отделение и во двор изложниц. На склад холодных слитков поступают команды о подготовке составов с плавками холодного металла определенного типа.

Источником информации о транспортной ситуации на участке управления являются устройства железнодорожной автоматики. Чтобы получить информацию об изменении ситуации на участке, в узловых точках путевого развития устанавливаются датчики,автоматически передающие в устройства железнодорожной автоматики номер проходящего состава или локомотива и направление их движения. С помощью эстафетных схем устройств железнодорожной автоматики прослеживается движение составов и локомотивов по участку, при этом номера составов и локомотивов перемещаются из одного накопителя в другой в соответствии с передвижением локомотива.

Информация о ходе выполнения некоторых технологических операций (начало и конец оборудования составов, стрипперование и т.п.) вводится операторами с пультов ручного ввода.

Как показала оценка эффективности, система в полном объеме позволит повысить температуру металла, поступающего в колодцы, и уменьшить простои блюминга.

ЛИТЕРАТУРА

I. М.Д. Климовицкий, А.П. Копелович, Автоматический контроль и регулирование в черной металлургии, Изд-во "Металлургия", 1967.

2. G.B.Dantzig, Ph.Wolfe, Decomposition principle for linear programms, „Oper.Res" 1960, vol.8, N1.

3. Е.Г. Гольштейн, Д.Б.Юдин, Новые направления в линейном программировании. М., 1966.

4. Ю.М. Журавлев, Ю.Ю. Финкельштейн, Локальные алгоритмы для задач целочисленного программирования, "Проблемы кибернетики". Вып. 14, 1965.

5. Р.Беллман, С.Дрейфус, Прикладные задачи динамического программирования, Изд-во "Наука", М., 1965.

6. В.Михалевич, Б.Шор, Метод последовательного анализа вариантов для численного решения задач оптимизации, "Труды конференции по вопросам применения ЭВМ в народном хозяйстве", г. Горький, 1964.

7. Г.И. Никитин, Выбор структуры автоматизированной системы управления металлургическим заводом, "Труды ЦНИИКА", вып.19, 1968.

8. Б.А. Власюк, А.П. Копелович, Г.Р. Кюсснер, "Применение ЭВМ для оперативного планирования прокатного производства, "Приборы и системы управления", № 6, 1967.

9. А.А. Белостоцкий, Ю.С. Вальденберг, Система оперативно-диспетчерского управления, "Импульс" для участка металлургического комбината, в сб. "Управление производством", Изд-во "Наука", 1967.

ПЕРЕЧЕНЬ РИСУНКОВ И ПОДРИСУНОЧНЫЕ ПОДПИСИ

Рис.I. Структурная схема завода
Рис.2. Структура модели объекта
Рис.3. Структура управления металлургическим заводом.

Подрисуночные подписи к рис. 3.

I - планирование последовательности партий на сортовых станах;
2 - формирование партий на сортовых станах; 3 - месячное планирование производства; 4 - оперативное планирование работы мартеновского цеха; 5 - планирование графика обработки плавок;
6 - планирование последовательности обработки партий-заказов;
7 - планирование последовательности обработки садок;
8 - планирование последовательности обработки слитков;
9 - оперативное планирование сортового стана № I;
I0 - планирование графика периодов работы мартеновских печей;
II - оперативное управление оборудованием шихтового отделения;
I2 - оперативное управление оборудованием печного пролета;
I3 - оперативное управление оборудованием разливочного отделения;
I4 - оперативное управление железнодорожным транспортом;
I5 - оперативное управление складом слитков;
I6 - оперативное управление стрипперным отделением;
I7 - управление ячейками нагревательных колодцев;
I8 - оперативное управление клещевыми кранами;
I9 - оперативное управление обжимным станом;
20 - оперативное управление раскроем;
2I - оперативное управление адъюстажем стана № I ;
22 - оперативное управление методическими печами;
23 - оперативное управление камерными печами;
24 - оперативное управление сортовым станом.

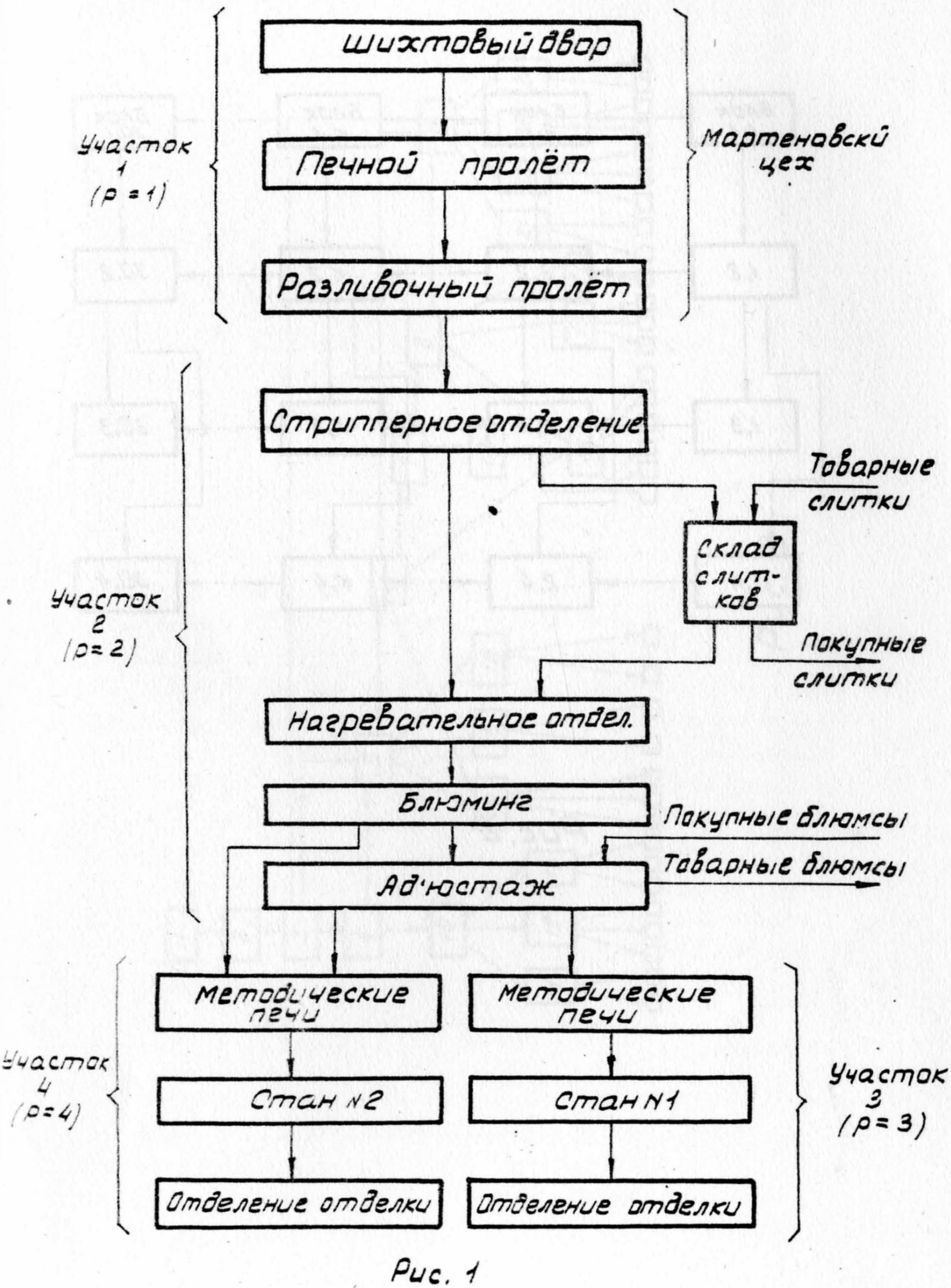

Шихтовый двор
Печной пролёт
Разливочный пролёт
Участок 1 (p =1)
Мартеновский цех
Стрипперное отделение
Товарные слитки
Склад слитков
Покупные слитки
Участок 2 (p= 2)
Нагревательное отдел.
Блюминг
Покупные блюмсы
Товарные блюмсы
Ад'юстаж
Методические печи
Методические печи
Стан №2
Стан №1
Отделение отделки
Отделение отделки
Участок 4 (p=4)
Участок 3 (p= 3)

Рис. 1

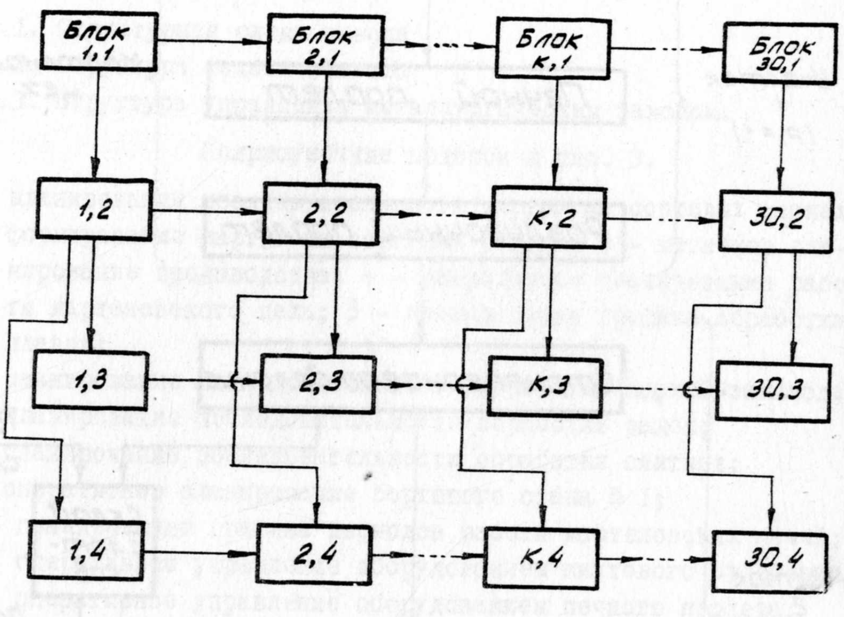

Блок 1,1
Блок 2,1
Блок к,1
Блок 30,1
1,2
2,2
к,2
30,2
1,3
2,3
к,3
30,3
1,4
2,4
к,4
30,4

Рис. 2

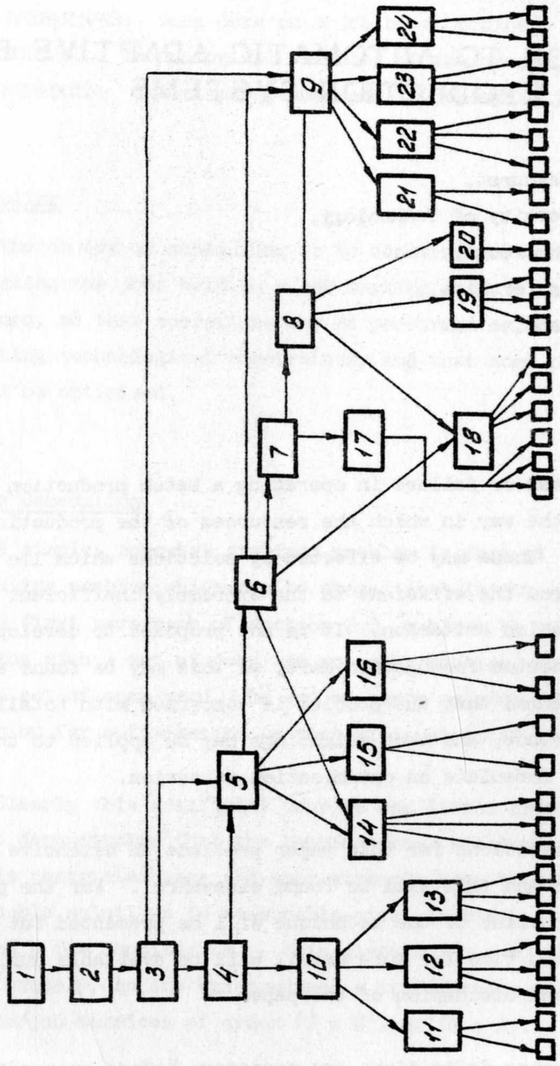


Рис. 3

AN APPROACH TO AUTOMATIC ADAPTIVE BATCH PRODUCTION SYSTEMS

by: G. Hayhurst,
University of Technology,
Loughborough,
Leicestershire,
England.

1.0 Introduction

The success or failure in operating a batch production system is dependent on the way in which the resources of the production units are deployed. These may be effected by solutions which lie throughout a range from the efficient to the extremely inefficient for any given optimisation criterion. It is not proposed to develop the theory of criterion formulation here, as this may be found elsewhere[1,2,3]. Also it is assumed that the problem is concerned with totally decidable logical preference, and that weightings may be applied to the elements objectives to formulate an optimisation criterion.

Space limitations for this paper preclude an extensive treatment of the problem and this will be found elsewhere[7]. For the present, only a brief outline of the technique will be presented but it is anticipated that time for more detail will be available during the presentation and discussion of the paper.

The following definitions are necessary before proceeding further:-

JOBS: Entities which pass through the production system.

MACHINE: Part of the production system which transforms a raw material to a finished output. (which may be raw material for a further machine).

OPERATIONS: Work done on a job by a machine.

ROUTING: Technological ordering of the operations on each job.

SCHEDULE: The strategy employed in deploying jobs to machines.

2.0 Scheduling

The object of scheduling is to construct a strategy or policy allocating the jobs held in a job backlog file to the available machines, so that operations may be performed on them without violating technological constraints, and that some pay-off function should be optimised.

2.1 Static Scheduling

A simple, somewhat stylised problem is that of the Static Scheduling problem which may be stated as being the problem presented in the first paragraph of section 2.0, subject to no updating, i.e. starting with a set of jobs and an empty machine shop, and finishing with a set of completed jobs and an empty machine shop. The usual criterion for optimisation is that of least total time.

Clearly this artificial problem has little practical application, but it demonstrates that the combinational problems are immense even in this restricted case and many attempts have been made to find computable solutions in reasonable-sized problems, i.e. a few jobs (n) and a few machines (m) - this gives an (n x m) sized problem. Unfortunately, no one has produced a satisfactory method[4] beyond machine/job matrices of order (2 x N), or (M x 2).

From the combinatorial aspect of the problem it is seen that the total number of ways in which the N x M matrix may be ordered is $(N!)^M$.

A collection of titles of recent papers on scheduling is given by Eilon and King[5].

2.3 Optimisation of Schedule

It is now necessary to define:-

WASTED TIME: Time when a machine is standing idle before processing the next operation it is to process, when that operation is available for processing.

UNNECESSARY DELAY: A particular schedule may contain some 'gaps' or periods when a machine is idle. These gaps are examined, together with the string of operations on the machine which follow a particular gap, to see whether it is possible to insert an operation from the string into the gap. If this is possible, the schedule contains unnecessary delay.

ACTIVE SCHEDULE: One which does not involve unnecessary delay, or wasted time.

SET OF FEASIBLE SCHEDULES: The ordering of jobs onto machines in all ways compatible with technological constraints, and which do not have wasted time.

Thus it can be seen that:-

Number in set of Active Schedules $\leqslant$ Number in set of Feasible Schedules $\leqslant (N!)^M$.

This concept of an active schedule, suggested by the present author[6,7] is expressed originally somewhat differently for two optimisation criteria by Giffler and Thompson[8].

It will be agreed that any practical optimisation criterion will have totally decidable logical preference and also will place a negative value on idleness; the optimum schedule by these criteria will be found within the active subset.

Furthermore, if the set of feasible schedules is considered in the form of a decision tree with the scheduling decisions at each node, and each branch corresponding to the effect of those decisions to the next decision point, then the graph thus formed may be searched by considering only the branches corresponding to 'active' decisions in both the static and general dynamic cases.

A method of finding the optimum may then be adopted, which is to generate one schedule, and then search the tree by evaluating the pay-off function at each node and abandoning the search of that branch when it becomes apparent that the pay-off function cannot possibly better one of the final solutions already found. This is the branch and bounds method.

This may however be prohibitively large for some ill-conditioned problems. In general a minimum of $\frac{1}{2}$ n(n + 1) nodes must be created with $\frac{1}{2}$ n(n - 1) + 1 listed. Unfortunate cases could need the creation of all 1 + n + n(n - 1) + ...n nodes whilst n! are listed.

The method proposed however is one which balances economy of calculation against quality of optimum required. In this case the 'pay-off' function becomes a 'performance evaluation function' which is chosen such that:

i) the optimisation criterion is defined

ii) the cost of computation is defined.

The method is analagous to that of Doran and Mitchie[9] for their solution to the sliding block puzzle and to Shen-Lin's solution to the travelling salesman puzzle, but the application of these methods would not necessarily reach the best solution. See Fig. 1 for expansion.

A little reflection will show that in the scheduling problem such a simple performance evaluation function could always home directly on the goal chosen by the optimisation criterion and the system would find an answer by pure single step optimisation. This is not necessarily the most 'elegant' (i.e. best schedule by optimising criterion). It would be the most 'economical' of search time, (but not by design), although ideally it would be preferred that the system should obtain the most elegant answer most economically.

The mistake should not be made at this stage of attempting to combine the economic term with the optimisation term to arrive at a combined 'optimisation criterion.

The problem is therefore reduced to the selection of a suitable performance evaluation function which will force the system to abandon the branch which it is at present developing in favour of some other node.

Some criteria will automatically do this, e.g. average total machine idle time.

In general we require:

$$V = \sum_{i=1}^{n} \sum_{j=1}^{m} KD_{ij} * JA_{ij} - g(t,\tau) * h(m,c) + f(t)$$

where KD = matrix of operations outstanding

JA = matrix of weights for particular optimising criterion.

m = number of nodes developed to date

c = cost of computation/node

g(m.c) = Function chosen to develop search tree

t = elapsed schedule time

τ = computation time

V = value of the performance evaluation function ('potential').

This system will also deal with the problem of multi-machine scheduling, i.e. different numbers of machines of the alternative types. The author knows of no other method published to achieve this solution.

In order to test the system on a problem, the problem in Fig. 2(i) had all the active schedules evaluated with the histogram of solutions shown in Fig. 2(ii) for total operation time criterion.

With the p.e.f. $V = \sum_{i=1}^{n} \sum_{j=1}^{m} KD_{ij} * JA_{ij} + B*VB - A*VB^2$

where VB = schedule time elapsed, it was found that even this simple function worked well.

In order to test total operation time criterion a JA_{ij} matrix of unity was taken, together with the linear B*VB term and the 'pulling-out' function A*VB^2 completing the p.e.f.

Results as follows were achieved with constants A,B varied:

i)	A = 0	B = 1	Total Time = 45
ii)	A = 0	B = 2	" " = 36
iii)	A = .1	B = 3.0	" " = 45
iv)	A = 0.2	B = 2.5	" " = 31

Thus for increasingly difficult criteria (increasing B term) it can be seen that the system improved from case (i) to case (ii). Case (iii) however is 'pulled out' too soon by too large an A term, whilst case (iv) achieves an overall optimum.

2.4 The General Case

In this case the problem is one of a multi-stage decision process where a stochastic input may or maynot be present. In this case the states arising from the transformation applied (i.e. state of backlog matrix after machine allocation) are not necessarily those expected because of the input of further jobs, or breakdown of machines. The system becomes a non-deterministic machine.

This stochastic input may be either known, unknown, or partially known. In the case where its probability distribution is known, this has been solved for the specific case of power station allocation by Riorden[10] by dynamic programming, although in large problems it is doubtful whether this method would be so attractive.

In this general scheduling case we may say we have a model of the expected distribution of the disturbances which will be adapted as a result of the effects on the system as it progresses.

Since no information is available at the commencement of the system, it is assumed that all disturbances are equally likely and may be drawn from an infinite square distribution, so it is decided to take the most promising step, as in the static case system. This is the is the general case of the Laplace - Bayes hypothesis for the discrete state problem. We shall, in fact, partition state space, and build up a histogram in the number of dimensions required by the particular problem to estimate the distribution of these disturbances. Also by feeding these in at the 'top' of the histogram and 'decaying' those at the bottom with time, it can be ensured that the histogram is capable of sensing changes in the distribution. Also predictive theory may be used to advantage in exactly analagous theory to time series analysis.

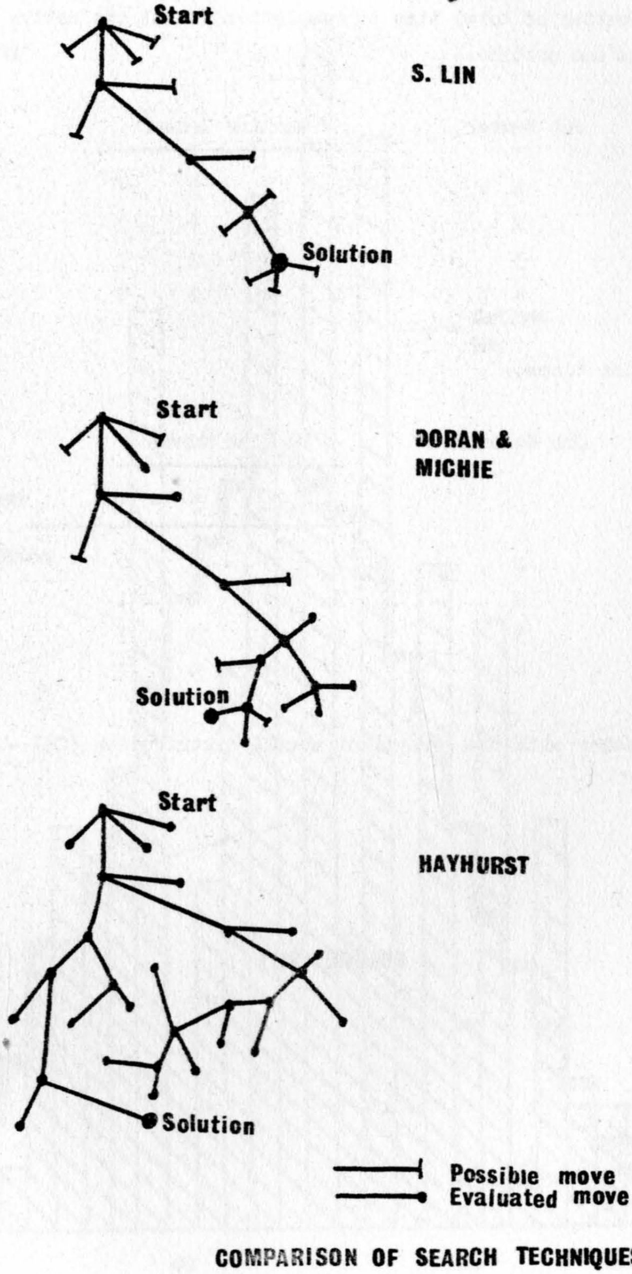
Should the system be operated in an environment against malevolent strategically-minded opponents, a minimax criterion should be adopted to cope with these Von-Neumann type opponents, rather than the Bayesian type. Further details of the philosophically interesting problem to which this line of thought leads will be found elsewhere[7].

This discretisation of state-space has been employed recently by Mitchie[11] for the description of unknown systems and applied successfully to the pole and cart problem.

3.0 Conclusions

The proposed methodology is put forward to provide a solution to the problems which are at present solved intuitively, and which must be solved quantitatively if a fully automatic system of control is to be adopted. It is capable of covering the whole spectrum of solution types from true optimum, with astronomically high calculation cost, to a somewhat worse, (but quantitavely unknown) solution at a reasonable (but loosely defined) cost.

It is a method which will work for any system thus it applies from fully-manual, through partial automation, to fully automatic systems, and in view of the low utilisation level at present employed in industry, the method offers attractive returns to the user. Also, with fully numerically-controlled systems as are at present being installed[12], such a method is essential.



COMPARISON OF SEARCH TECHNIQUES

FIGURE 1

Distribution of total time to completion of all the active set of schedules in the problem:-

Job Number	Machine Order			
1	1	2	3	4
2	2	1	4	3
3	4	3	2	1
4	1	4	2	3

with operation times:-

Job Number	Machine Number			
	1	2	3	4
1	2	9	6	6
2	7	4	2	1
3	9	5	3	7
4	2	4	9	7

plotted together with the equivalent normal distribution (CHI - squared) = 128.485).

FIGURE 2(i)

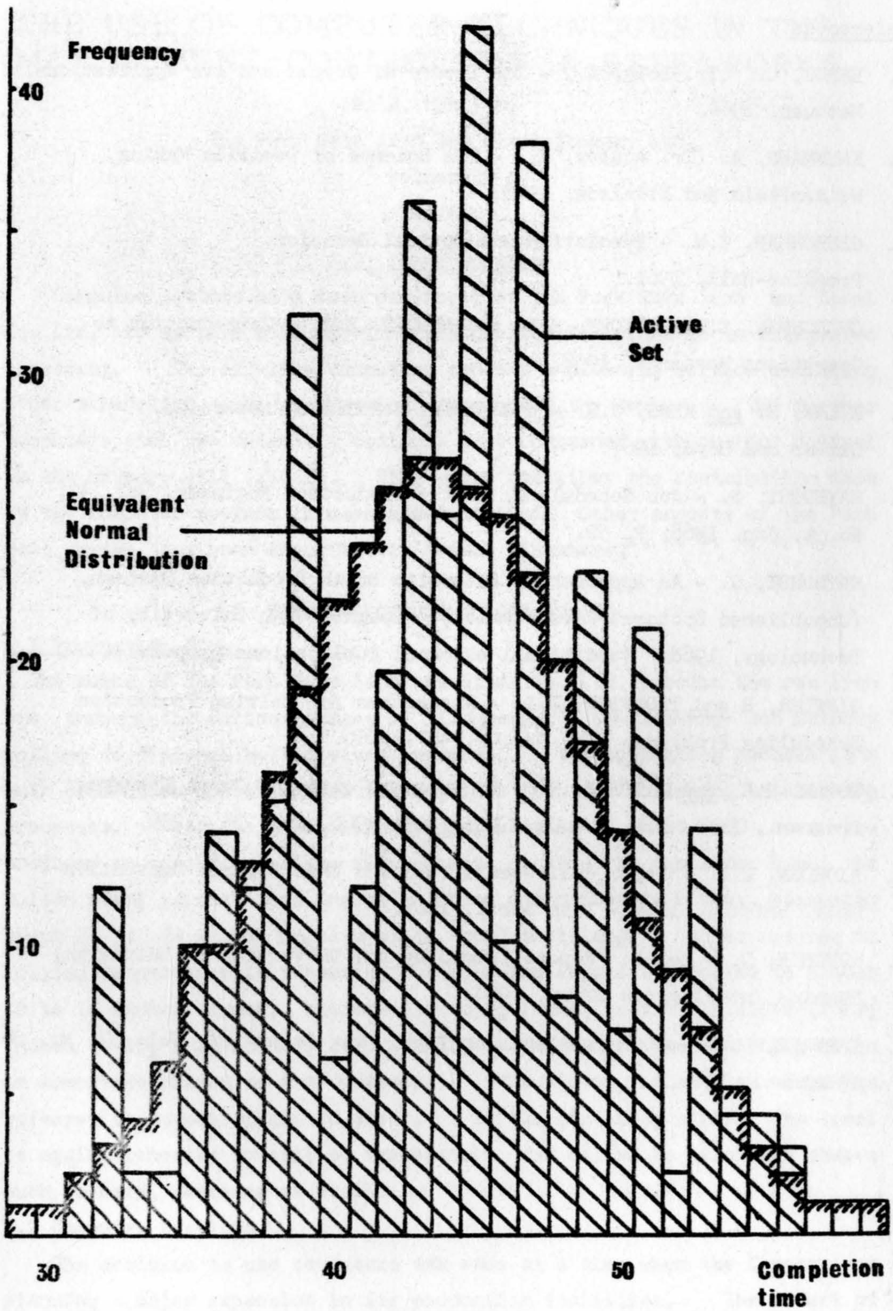

Frequency
40
30
20
10
Active Set
Equivalent Normal Distribution
30
40
50
Completion time

FIGURE 2(ii)

References

1. BERGE, C. (Tr. Doig, A.) - The Theory of Graphs and its Applications. Methuen, 1964.

2. KAUFMANN, A. (Tr. Audley, R.) - The Science of Decision Making. Weidenfield and Nicolson, 1968

3. CHURCHMAN, C.W. - Prediction and Optimal Decision. Prentice-Hall, 1961.

4. CHURCHMAN, C.W., ACKOFF, R.L. and ARNOFF, E.I. - Introduction to Operations Research, 1958.

5. EILON, S. and KING, J.R. - Industrial Scheduling Abstracts. Oliver and Boyd, 1967.

6. HAYHURST, G. - Job Scheduling, in the Production Engineer, Vol. 47, No. 1, Jan. 1968, P. 52.

7. HAYHURST, G. - An Approach to Automatic Batch Production Systems. (Unpublished Doctoral Dissertation). Loughborough University of Technology, 1968. (additional external publications proposed).

8. GIFFLER, B and THOMPSON, G.L. - Algorithms for solving Production Scheduling Problems. Op. Res. 8. 487. 1960.

9. DORAN, J.E. and MITCHIE, D. - Experiments with the Graph Traverser Program., Proc. Roy. Soc. A. Vol. 294, 1966. pp. 235 - 259.

10. RIORDEN, J.S. - Paper delivered at the 2nd UKAC Control Convention - Univ. Bristol, 11th - 14th April, 1967.

11. MITCHIE, D. - Boxes: Report issued by the University of Edinburgh, Machine Intelligence Group, 1968.

12. IREDALE, R. - Putting Artisans on Tape. New Scientist, Vol. 37, No. 584 15th Feb., 1968.

THE USE OF COMPUTER TECHNIQUES IN THE MANAGEMENT CONTROL OF A STEELWORKS

B. B. Hickling

The Park Gate Iron and Steel Company Ltd.

Rotherham

England

1.0 Introduction

Computer systems have been developed at The Park Gate Iron and Steel Co. Ltd. to perform both off-line and on-line functions in an integrated hierarchy. Two off-line computers provide a planning service embracing order scheduling and progressing throughout the works. Two further computers and two automatic controls are concerned with on-line control in the primary mill (fig.1). This paper describes the contribution made by the computer systems to management control. Other aspects of the Park Gate system have been described in detail elsewhere.[1, 2, 3, 4, 5, 6.]

2.0 Background to the System

2.1 Production Flow

The works of The Park Gate Iron and Steel Co. Ltd. process the raw iron ore through the various stages of ironmaking, steelmaking and primary rolling to finished rolled steel products. The ironmaking process is a bulk production operation providing molten iron for following steelmaking processes. At the steelmaking stage, batches of about 100 tons are produced to specific quality requirements dictated by the order book. The molten steel is cast into ingots weighing approximately 6 tons, measuring about 24 x 24 inches cross section and 7 feet long. After heating to rolling temperature in reheating furnaces the ingots are rolled to blooms (5 to 14 inches square), billets ($1\frac{3}{4}$ to $4\frac{3}{4}$ inches square) or slabs ($3 \times 2\frac{3}{4}$ inches to $10\frac{1}{2} \times 2\frac{3}{4}$ inches) before being despatched to the finishing mills or sometimes direct to the customer. The blooms, billets or slabs are referred to as semi-finished steel. In the finishing mills, the steel is again reheated to rolling temperature and rolled to finished shapes such as bars, sections or strip.

2.2 Computer Services

The decision to use computers was made at a time when the Company was planning a major expansion in its production facilities. The effect of the expansion was to increase the production capacity of the works from 450,000 to 850,000 ingot tons per annum and to extend the Company's range of products and steel qualities. Computers were introduced to provide a planning service for the whole works and in one crucial area to provide on-line control. Work on the design stage of the computer systems started in May 1961 and the systems were brought into use progressively with the commissioning of the new plant over the period October 1963 to April 1964

The computer systems are based on three main magnetic tape files:-

(a) The Progress File records the Company's order book and all work in progress.

(b) The Ingot File records all ingots in existence, including rolling instructions and order details for those ingots which have been allocated to orders.

(c) The Stock File records the incidental stocks of semi-finished and finished material.

The main files are updated 20 times each day with information in the form of paper tape either from the primary mill computers or punched from works documents. In addition to routine planning and control documents, a great deal of the information output by the computers is designed to assist managers at various levels to have a better understanding of the activities of the works, and to direct their attention into areas where their efforts should be concentrated. This type of information can be described under three headings:-

(i) Minute by Minute Control

(ii) Day by Day Control

(iii) Longer Term Control

3.0 Minute by Minute Control

3.1 Primary Mill System

All minute by minute control is concentrated in the primary mill, a vital production area of the works. This mill occupies a potentially restrictive position, in that it receives ingots from three steelmaking sources, and distributes semi-finished steel to 6 finishing mills(Fig.2). The mill is about 650 yards long and there are eight successive stages of continuous production (Fig.1).

The computer system is designed to overcome the traditional concept of a 'chain of islands of control' in the mill. It co-ordinates operators and automatic controls to obtain maximum throughput. This is achieved by a computer-controlled display system operated from a production control centre (Fig. 3). Information is given to mill operators on cathode ray screens on which the computer displays instructions for ingots moving through the mill (Fig.4).

3.2 Operation of the System

Every two hours the production controller requests from the central planning computers the rolling and shearing instructions for the ingots which have been charged into the reheating furnaces. These instructions are produced in the form of punched paper tape. The central computers also produce Bank Progress Cards, to be used when the semi-finished steel travels through the examination and stockyard areas.

Before the ingots in each furnace are discharged into the mill, the paper tape is read into the production control computer. As each ingot leaves the furnace, it is identified on rotary switches, enabling the computer to add the item to the displayed list.

The display includes an arrow for each operator, indicating the position of all the items in the mill (Fig.5). As an operator completes an item, he reports to the computer by pressing one of ten message buttons (Fig.6). The computer then moves his arrow down to the next item on the list. The message buttons also report information on routine processing for example the depth of desurfacing, or may report that an item has been delayed, returned to the furnace, diverted or scrapped. The computer uses this information to keep up to date the instructions displayed to operators.

The production control computer receives information directly from the ingot weigh and exchanges information with the shear computer. The system is being extended to include direct measurement from photo-electric cells at the bloom shear.

3.3 Exception Procedures

The production controller supervises the system and gives the computer instructions about size changes, etc. The computer can display a series of "urgent messages" in the form of 3-letter codes, either as a warning to the production controller of a suspected error condition or giving information which will help to reduce delays and misunderstandings. There are 25 urgent messages, and each message demands an immediate answer from the production controller. Two examples of the messages are :

(a) "QID" If the furnaceman sets an ingot identity on the rotary switches which is not in the core store of the production control computer, QID ("query identity") appears on the screens. All the input panels at the furnaces are frozen until the query is answered.

(b) "PLI" When the last ingot for the current billet mill size setting is discharged, PLI ("last ingot") is displayed on the screens, warning all concerned that a size change will follow in about ten minutes.

3.4 Production Log

A production log is maintained on an electric typewriter attached to the computer (Fig.3). All important messages in and out of the control system are timed and recorded. When semi-finished steel arrives on the cooling beds, a one-line summary of the ingot is printed on the log, which includes identity, time, weight, yield and shearing details. The existence of the log enables both mill and metallurgical staff to have an immediate knowledge of production details.

3.5 Identification of the Semi-finished Steel

As material arrives at the cooling beds, the production control computer informs the reception area staff of the identity and destination of the material by printing the information on a teleprinter adjacent to each bed. If the shear computer has left excess lengths on billets in order that samples may be taken, a code on the teleprinter output informs the staff of the location and details of the sample lengths.

4.0 Day by Day Control

4.1 Routine Production Planning

Day by day management control is aided by the computer-based production planning system. The nature of the production planning problems at Park Gate is determined largely by the order book. The function of the computer system could be described as helping to reconcile a mass production plant and a jobbing order book. In round terms 1000 orders a week are received, of an average weight of 12 tons but with nearly half the orders being for less than 6 tons, the weight of an ingot. The orders cover 1000 quality variations and some hundreds of finished shapes and sizes.

The task of the production planning department, aided by the computers, is therefore to find ways of grouping these customers' requirements first into casts by quality and later into rolling batches by size. A major complication is the inherent uncertainty of the processes which precludes rigid planning. Planning must be a dynamic function, constantly reacting to production misfortunes, but still preserving as much order as possible against the ever present threat of chaos.

The central computers maintain the master progress file on magnetic tape, recording the fullest details of the order book and all work in progress. This file contains up to 10,000 order records and 10 million characters of information about the steel allocated to these orders. The file is updated 20 times each day with 2,000 feedback messages reporting the progress of steel through the plant. From this file the computer produces in a day a variety of documents in the form of summaries and instructions for action by planning and production departments as follows :-

(a) Finishing Mill Planning

As orders are received, they are examined and coded by the planner responsible for the finishing mill concerned. A delivery promise is given, based on a daily Forward Loading Summary produced by the computer, showing the tonnage already accepted, analysed by weeks and sizes. From the delivery promise are calculated target dates for steelmaking and primary rolling. From time to time the planner may decide to vary the sequence of planned roll mountings. The computer then alters the target dates for all orders affected. In due course the computer prints out on request lists of orders for rolling in the finishing mills, with details of billets and slabs required and available.

(b) Cast Planning

Each morning the cast planners receive from the computer an up-to-date statement of orders requiring steel, taking into account miscasts and primary rolling difficulties up to 6.00am. The information provided is in the following forms :

Summaries - of tonnages required, analysed vertically by quality and horizontally by target weeks, semi-finished size and finishing mill to assist in establishing priorities and grouping cast quantities.

Order Lists - giving full details of all orders requiring steel to be made within two weeks, supplemented if necessary by specific requests for further orders for unusual qualities, or specified semi-finished sizes or finishing mills.

4.2 Management Control by Exception

In one sense the whole production planning system is a management control tool in that it progresses all orders through the works. The more particular instances of management control, however, take the form of exception reports on various stages of processing :-

(a) Progress Reports

Progress reports on individual orders are issued by the computer when error conditions are detected or customer service is threatened.

(b) Cast Planning

A list is produced each day for the production planning manager which highlights orders due for rolling in the finishing mill within two weeks, for which steel is still required. These are orders or remakes to cover production losses which have failed to find their way into casts through the normal procedures. In order to prevent a delivery promise being broken special efforts can be made by the manager to obtain steel either from unallocated stock, or a specially authorised cast, or by purchasing steel from another company.

(c) Ingot Allocation

When a cast is approved by the ingot allocator, the computer reports any discrepancies between the orders originally planned and those actually allocated to the ingots.

(d) Primary Rolling

A daily Cold Charging List is produced for the production controller, detailing all ingots in cold stock which have been allocated to orders. The list is sorted into semi-finished size and primary rolling week. It tells the controller how long the ingots have been allocated and whether or not they are in the assembly area ready for charging. This document has overcome the natural tendency of the primary mill staff, to prefer charging hot ingots to cold ones despite the urgency of the orders within the ingots.

If an ingot is not reported as rolled or taken out of the furnace within 24 hours after being charged into the furnace, a Not Reported message is sent to the production control centre. These messages are repeated daily until cleared by the production controller. In 12 months, this approach reduced such queries from about ten each day to one each week.

(e) Examination of Semi-finished Steel

Progress reports are sent to the examination areas for all billets which are required in the finishing mill within the next 48 hours. Special efforts are then made to deliver these billets in time to be rolled in the finishing mill. Duplicates of the progress reports are sent to the respective finishing mills as a warning to expect the material and make the necessary arrangements.

(f) Rolling Billets in the Finishing Mills

When the rolling of a particular size has been completed, the finishing mill planner instructs the computer to alter the planned rolling day of any orders which have missed the rolling. The computer produces a progress report for any replanned order where billets missed the rolling. The mill manager then has the responsibility of making sure that these billets do not miss another rolling.

(g) Errors in Shop Floor Recording

At any stage of production, more material may be reported moving out of a department than was previously recorded, creating a negative stock. The computer issues daily progress reports for all orders with negative stock conditions. These reports are given to the respective mill planners, who must solve the queries and prepare documents to correct the master files. Similarly the computer queries all feedback messages which quote material from a cast which has not already been identified with that order.

(h) Completed Orders

If sufficient material has been delivered to an order to satisfy the customer's requirement, but leaving a small amount still in the process route, the computer produces a Query Order Complete report, which is sent to the finishing mill staff. They have to decide whether the remnant is actual material or the result of a previous error in recording. In the first case, they must deliver the material to the customer or deallocate it to stock; in the second case they must input documents to correct the master files. This type of report is reissued every fourteen days. If a third report is issued, it is sent to the production planning manager for personal attention.

5.0 Longer Term Control

Longer term management control is assisted by reports, produced by summarising the main files and a 'dump file' on which all the production messages are accumulated each month.

5.1 Standard Costing

At the end of each month, the production movements for that month are taken from the dump file and inter-process stocks are taken from the main files. Together they form the basis of a series of tabulations which tell the production story for the month, both in terms of tonnages and standard values. The tabulations cover in detail the process route from steelmaking to delivery of finished material in the following stages:-

Production of ingots
Ingot stocks
Primary rolling yields
Examination and dressing of semi-finished steel
Semi-finished stocks
Finishing mill rolling yields
Examination and dressing of finished material
Finished stocks
Deliveries to customers

Performances are compared with standards, and variances are calculated at each stage of the process route.

5.2 Metallurgical Control

A parallel exercise to standard costing is carried out for the metallurgical department. The results of the examination and dressing of semi-finished and finished material are summarised from the dump file, and reports are issued weekly and monthly which show the results analysed by quality and size.

5.3 Stocks

In addition to producing lists for routine stock-taking, lists are produced twice each year showing the age of all steel stocks. These lists are used by the production director when deciding which of the stocks to retain and which to scrap.

5.5 Forecasting

Sales forecasting and market research are helped by statistics of orders received, sales analysis, weight of material on order, etc.

5.6 Progress Lists

An important aspect of the production planning system is the ability to produce progress lists detailing the position of all orders for a particular customer, type of steel or delivery period. Such lists are used by the progress department as the agenda for meetings with customers to discuss production arrangements in the immediate future. Progress lists of overdue orders are provided to help managers to assess delivery performance.

6.0 Future Developments

Most of what the company set out to do in 1961 has been achieved. The central planning computers are now 80% loaded and it will be difficult to undertake any large expansion of the present scheme. Two further limitations of the present computer system are inaccuracies in shop floor recording and delays in obtaining information from all departments except the primary mill.

Although great improvements have been made in the accuracy of input data, the message error rate is still about 4%. During the first year of implementation the systems team was fully engaged in combating errors in input by all means at their disposal.

Because most of the 2,000 production messages each day are processed by a data preparation room which is only staffed during normal office hours, the picture of the works as recorded in the main files may be up to 36 hours out of date. This means, for instance, that it is impossible for the computer to issue detailed working instructions to a foreman at the beginning of his shift.

6.2 Proposed Real Time System

To overcome the present limitations, a computer development scheme, covering the next 5 years, has been planned for Park Gate. The two central computers will be replaced by a large computer with disc storage. Production messages will be input directly to this computer through about 50 teleprinters located around the works. This will enable incoming data to be compared with the latest position of the order recorded on the files. Any inconsistencies will be printed out immediately on the teleprinter concerned.

The aim of the new system will be to provide managers from shop floor supervisor to the Managing Director with immediate access to the up-to-date and accurate information appropriate to each level of management control.

References

1. "A Technical Survey of The Park Gate Iron and Steel Co. Ltd.". Published in 1964 by Steel Times, pp. 114-117.

2. "Computer Control of Steelworks Production". Proceedings of the Institution of Electrical Engineers, Volume III No. 6, June 1964, pp. 1183-1192. (J. T. Jones and N. J. Williams).

3. "Integrated Computer Control of Steelworks Production". Proceedings of MESUCORA II Congress International, Paris, 1963, (G. H. Kelly). (Also reproduced in English Electric Journal, Vol. 19 No. 4 July/August 1964).

4. "Integrated Process Control at The Park Gate Iron and Steel Co. Ltd. Two Years After Commissioning". Institution of Electrical Engineers Process Control Conference September 1966. (J. A. Donoghue).

5. "Information Flow and Communications in Steelworks". Iron and Steel Institute Meeting, November 1966. (B.B.Hickling and J. T. Jones).

6. "Experience with The Park Gate Computer System and Possible Future Developments". International Seminar on Control Systems in Metallurgical Works, Ostrava, Czechoslovakia, May 1967. (N. J. Williams).

The author wishes to express his thanks to the Directors of the Park Gate Iron and Steel Company Ltd. for their support and permission to contribute this paper and to all members of the Information Handling Department for their invaluable help.

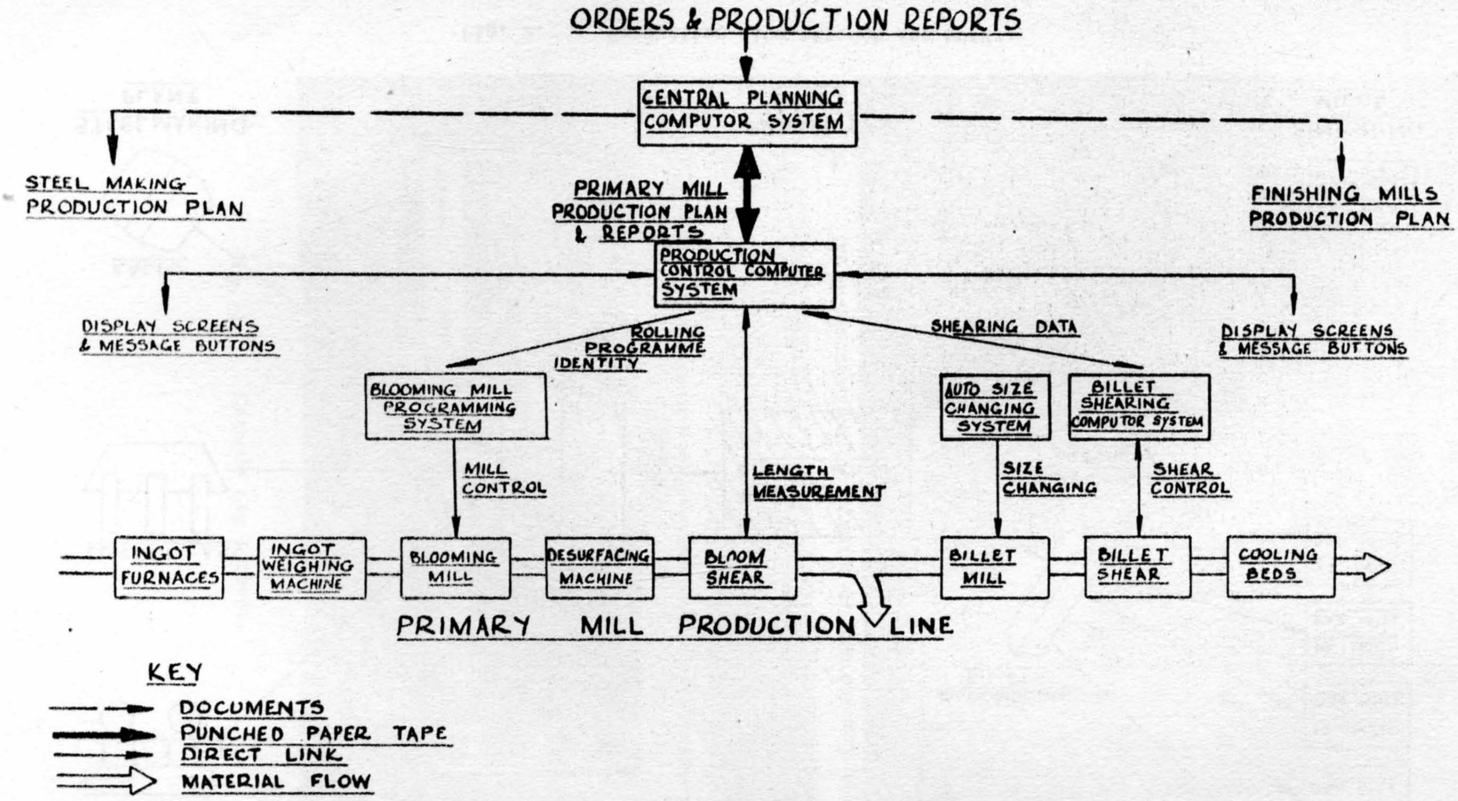


Fig. 1. Computer hierarchy.

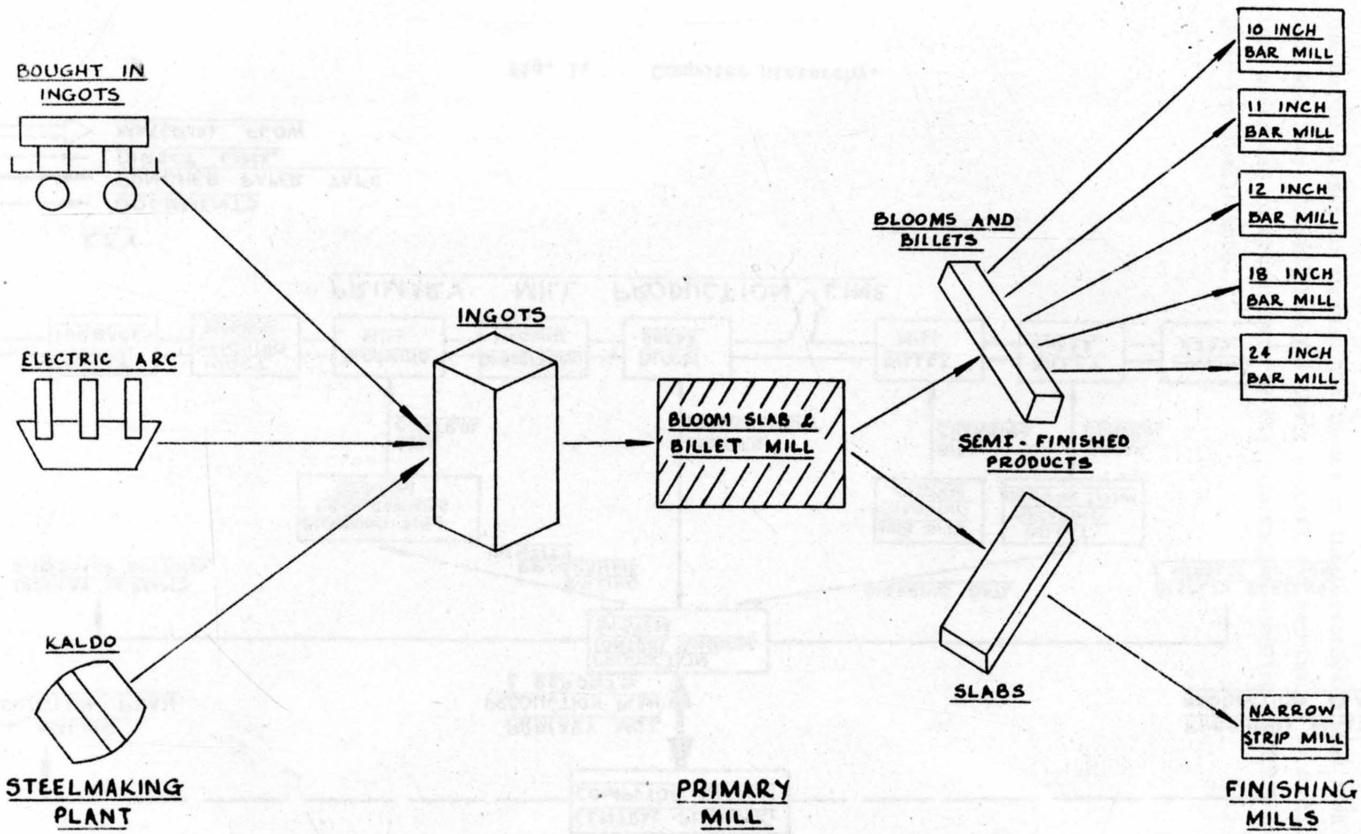


Fig. 2. Production flow through the works.

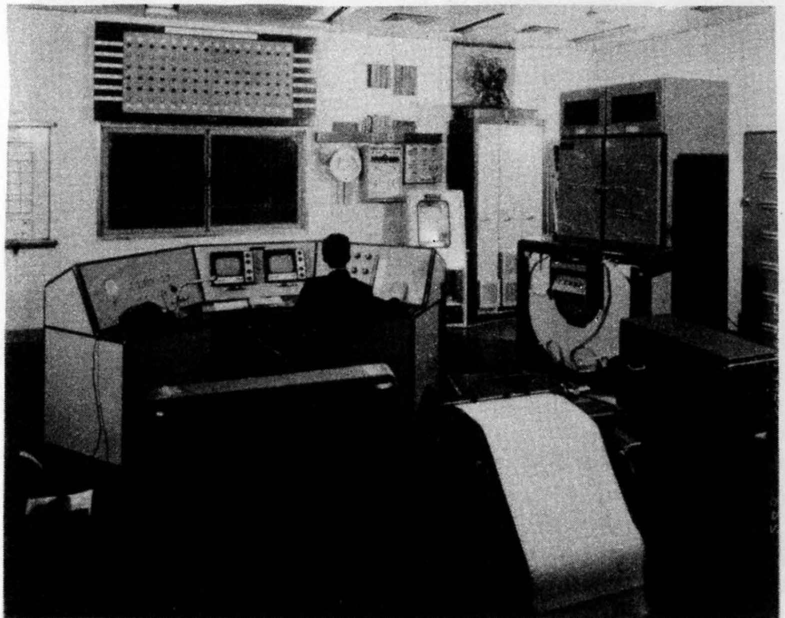


Fig. 3. The production control centre in the primary mill.

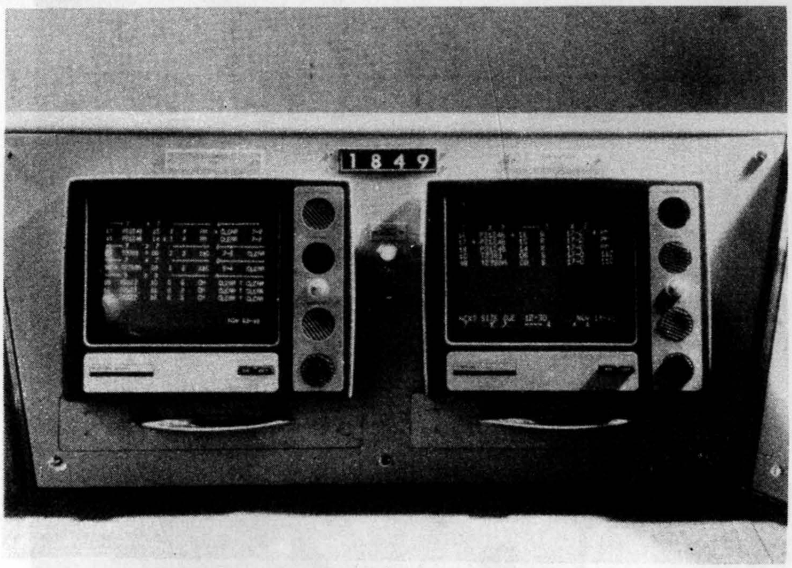


Fig. 4. Cathode ray screens.

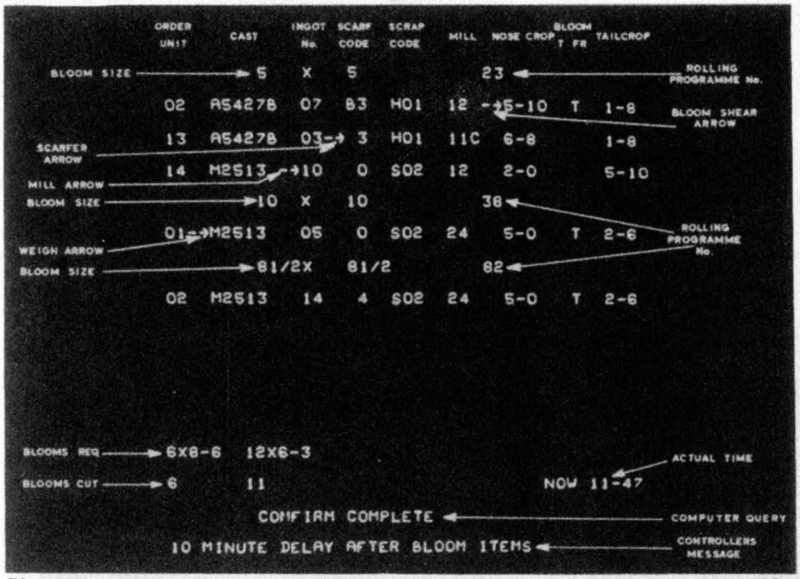


Fig. 5. The format of information on the cathode ray screens.

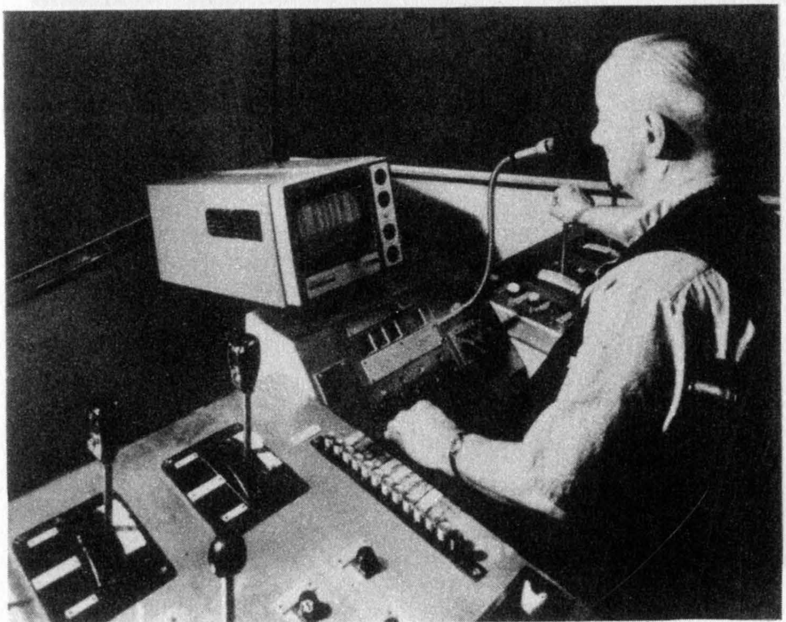


Fig. 6. The operating pulpit for the bloom shear.