MODELLING AND OPTIMISATION OF MIXED VARIABLE PROCESSES

MODELLING AND DYNAMIC OPTIMISATION OF MIXED VARIABLE PROCESSES USING HYBRID SIMULATION TECHNIQUES

By

KURT ANTON STRÖBELE, B.Sc. (Eng)

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Doctor of Philosophy

McMaster University
May, 1973

DOCTOR OF PHILOSOPHY (1973)
(MECHANICAL ENGINEERING)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: Modelling and Dynamic Optimisction of Mixed Variable Processes Using Hybrid Simulation Techniques

AUTHOR: Kurt Anton Ströbele, B. Sc. (Eng.) (Witwatersrand University)

SUPERVISORS: Dr. M.C. de Malherbe, Prof. R. Newcombe.

NUMBER OF PAGES: 287, G-36

TO MY PARENTS

ABSTRACT

An approach to study control problems in complex systems of interacting, mixed variable processes is presented. Discrete event simulation is combined with continuous simulation to provide a versatile method for system modelling and analysis. Provisions are made to include a human model into the model of a real system which allows an accurate reproduction of systems in which human control is an integral part. A special model display unit was built to provide feedback to the model operator, which can be readily connected to a hybrid computer. The display unit permits operation of the model by process people with no detailed knowledge of simulation which is an added advantage.

A copper smelter system was modelled and simulated on a hybrid computer which includes discrete as well as continuous processes and requires human control to integrate the process to achieve a maximum output. The system thus represents a good example for a variety of similar systems encountered in industry. Semi-empirical models of smelting processes were developed based on existing knowledge of the processes. Using the display unit the evolutionary development necessary for some models was possible.

The simulation model was calibrated and validated to accurately reproduce the real system behaviour. Based on simulation experiments on improved operating technique for controlling aspects of the smelting process was tested and an increase

in production was predicted. This operating technique was tested in plant trials and adopted while the predicted production increase was verified.

The validated model was then used to develop heuristic decision and forecasting algorithms for automatic optimal control of model operations. Based on the realistic model different decision strategies could be tested by observing their success in controlling the simulation model. The control algorithms were refined to give a satisfactory control of the simulation model, without the need of human intervention. The smelter model with the heuristic control can thus serve as a useful tool in developing an on-line control system for the real system, which could be used as a decision making aid to operating personnel.

ACKNOWLEDGEMENTS

The sincerest thanks are expressed to my wife Marianne for her encouragement, help and patience during the many months required for the preparation of this thesis.

The help and direction of my supervisor, Professor M.C. de Malherbe, and acting supervisor, Professor R. Newcombe, not only in this work, but throughout my graduate work at McMaster University is sincerely appreciated.

The generous support of the National Research Council, where this project originated and where most of the experimental work was done, and in particular the assistance provided there by Dr. U.P. Graefe and Mr. L.K. Nenonen in this study is gratefully acknowledged.

The cooperation of Noranda Mines in providing data for the model and implementing a plant test, and the assistance provided by personnel without which this study would not have been possible is gratefully acknowledged.

Thanks are expressed to Drs. J.L. Duncan, R.M. Korol and Dr. R.L. Judd for their help as members of the supervisory committee and to other members of staff for their valuable assistance throughout the course of this work.

Finally, the financial assistance by McMaster University and by the National Research Council throughout the course of my studies in the form of Scholarships is appreciated.

TABLE OF CONTENTS

PAGE

CHAPTER 4.0 SMELTER MODEL DEVELOPMENT

CHAPTER 5.0 MODEL VALIDATION

CHAPTER 6.0 MODEL ANALYSIS AND RESULTS BY MANUAL CONTROL

CHAPTER 7.0 DEVELOPMENT OF AUTOMATIC MODEL CONTROL

LIST OF FIGURES

PAGE

LIST OF TABLES

1.0 INTRODUCTION

1.1 PROBLEM FORMULATION

In an industrial environment where a large number of processes interact it is of particular interest to control the overall system in an optimum fashion. In many instances it is possible to control individual units, processes or subsystems in a satisfactory way but the control of an integrated system of processes presents problems. The overall control of a plant is often the responsibility of a foreman who attempts to coordinate processes to maintain a high level of production. The system controller is usually faced with irregular process behaviour, equipment failure and the general problem of achieving the best output within given constraints. This is especially true in situations where the production capacity of an originally small system was increased gradually by expanding capacities of critical processes until a rather complex system emerged. This problem can also arise when units or processes in a system are replaced by newer methods that upset the overall operation and performance. Similar problems are, however, also found in newly developed plants where certain aspects of process interaction had not or could not be accounted for in the design phase. Examples of both situations are frequent and one of each will be discussed in the text.

The object of this thesis is to develop an approach to analyse, study and improve the operation of such systems.

Mathematical models of real systems produce exact results provided there are no discrepancies between the real system behaviour and the mathematical relationships representing it. To produce an accurate model of a complex system a simulation model will allow the best representation of the system especially in situations where exact mathematical relationships of process behaviour do not exist. A simulation approach is therefore used in this study.

In general industrial operations consist of continuous dynamic processes, i.e. processes in which the state is time dependent, and varies continuously over a period of time. In addition to the continuous processes, logical relationships exist that require actions at discrete time intervals. A hybrid simulation approach which allows accurate modelling of both discrete as well as continuous processes will be employed.

Most simulation models are off-line models which generate a simulation run independently producing a final output of statistics and other data. To include the important function of human operators in a system an interactive simulation approach will be taken. This allows the inclusion of human decisions and control functions in a model thereby coming very close to reproduce operations of a real system.

1.2 SYSTEM SELECTION

In this thesis an application of the approach of hybrid simulation of mixed variable processes to a particular, real system is presented to show and verify the technique.

Copper smelter operations form a dynamic system, where both discrete as well as continuous variables govern the system state and serve as a typical example for many industrial processes. Steel and paper making processes are other examples where this approach is applicable. Based on preliminary results of the present work, a similar study was started to study process control in steel making. * In general the technique shown here is applicable to any complex system. Particular benefits are achieved when studying systems in which some form of control is exercised by human operators.

Through the co-operation of the Control Systems and Analysis Laboratories of National Research Council of Canada and Noranda Mines Ltd., the necessary information was obtained on the copper smelting and refining processes of a particular smelter for which a model was then developed.

In addition to regular operating records and other process data that were made available by the smelter, several field studies were necessary to analyse processes and obtain all the data required for developing and validating the model.

* Apart from crane operations every process involved and the appearance of the overall system is entirely different to the copper smelter system. However, the simulation technique developed here could be used with no significant modifications.

1.3 DESCRIPTION OF SYSTEM

It is important to have a clear and thorough understanding of the operation of a system to develop a realistic and accurate model.

It will therefore be useful to describe briefly the processes that are involved in copper smelting. Further details where necessary will appear in Chapter 4.

The following terms are commonly used in connection with copper smelting and refining and are listed here with a brief description:

Fluxing ore - Silicon oxides ($Si0_2$)

Concentrates - Concentrated copper sulphide ores

Calcine - Concentrates after blending and roasting

Matte - Liquid ore in the reverberatory furnace (eutectic mixture of FeS and Cu_2S)

Slag - Liquid mixture of fayalite ($2Fe0.Si0_2$) and gangue

White Metal - Almost pure liquid copper sulphide (CU_2S)

Blister Copper - Almost pure liquid copper

Flue Dust - Precipitates rich in copper from the flue

Converter aisle - The aisle along which the different smelter furnaces are positioned which are serviced by cranes moving along its length (sometimes used to describe the smelter system)

Roasters - Roasting furnace in which excess sulphur and humidity can be removed from concentrates

Reverberatory Furnace Furnace for smelting calcine or concentrated ores.

Converter Furnace - Pierce-Smith Converter furnace for converting matte by oxidisation/reduction reactions.

Anode Furnace - Holding furnace for pyrometallurgical refining of blister copper.

Anode wheel - Casting machine for refined liquid copper.

Pigs - Lumps of solid blister copper cast in sand moulds.

Ladles - Large steel containers for transporting material.

Skulls - Solidified crust of matte or copper in a ladle.

(Liquid phases are given for some materials, as they occur in that form in the smelter).

Figure 1.3.1 is the floor plan of the smelter and shows the location of different furnaces. Smelting and refining of copper can be described by four consecutive pyrometallurgical processes. The first stage considered in the smelter system is that of smelting ore in one of three reverberatory furnaces. Roasting of concentrates and electrolytic refining of the copper precedes and succeeds the above four stages. These processes are not considered as they have no direct bearing on the smelter system operation that is considered here. (This is described in Chapter 4).

There are two hot charged reverberatory furnaces No's. 1 & 2 and one wet charged furnace. Whereas a hot charged furnace is fed with calcine from roasters, the wet or green charged furnace accepts concentrates at ambient temperatures with a certain moisture content.

Matte, molten material from the reverberatory furnaces, is transported in ladles by one of three overhead gantry cranes to the converters.

Both stages2 & 3 of the smelting process take place in the converters which require frequent servicesby cranes during a converting cycle. Blister copper, the final product from the converter process, is removed from converters and transferred to an anode furnace or cast into approximately 5 ton blocks called pigs on the smelter floor, designated as the pig bay. The fourth stage of refining takes place in the anode furnace. Fig. 1.3.2 shows the general material flow for the smelting process from the smelting of raw material to the production of copper.

At any time there are no less than three reverberatory furnaces, four converters, two anode furnaces in operation while the remaining converter or anode furnace is equipped with a new brick lining. The 15 circled numbers referring to material movements by cranes show the variety and number of crane services that are required.

It was not attempted to draw the complete material flow of the actual smelter as it would serve no other purpose than showing its complexity which can be appreciated. At any time only two cranes operate and are busy between 50%-90% of the time.

The crane at the north end is not used on a regular basis unless a breakdown of one of the other two occurs, or if a task of longer duration along the north end is to be performed. Its presence further south causes mutual interference and increases the danger of collisions at speeds of over 300 ft/min. with loads of molten materials at high temperatures. The crane on the north end also has a lower travelling speed which is an added disadvantage.

Activities in the converter aisle are controlled by a converter foreman who attempts to process a certain number of ladles of matte during an 8-hour shift. Operating personnel at converters, anode furnaces, cranes etc. act according to the foreman's overall plan or schedule although detailed decisions are made by individuals. An independent crew controls the reverberatory furnace operation attempting to supply matte uniformly to the converter aisle at the required rate.

The description of the smelter system above excludes some details that add to its complexity. Some of these will be described in Chapter 4 and the Appendices.

Nevertheless,it is evident that efficient control of the overall smelter is a difficult task. The duration of smelting processes in converters and the availability of material from the reverberatory furnaces is not deterministic, that is, no exact forecasting of endpoints of processes is possible. Invariably during several hours of production some processes operate below their capacity or are even idle at times. A buildup of crane service requirements will cause delays if the cranes cannot cope with the tasks fast enough. Under these conditions scheduling of processes is extremely difficult for a foreman,although the degree of perfection obtained through many years of experience is remarkable.

The system presented is a typical situation where performance improvement through better control of the system can be studied on a model of the system.

Furthermore the analysis can provide insight into a variety of other areas such as the effects of expansions to the smelter or the introduction of new processes. This aspect will be discussed in Chapter 6 together with other applications of the simulation model.

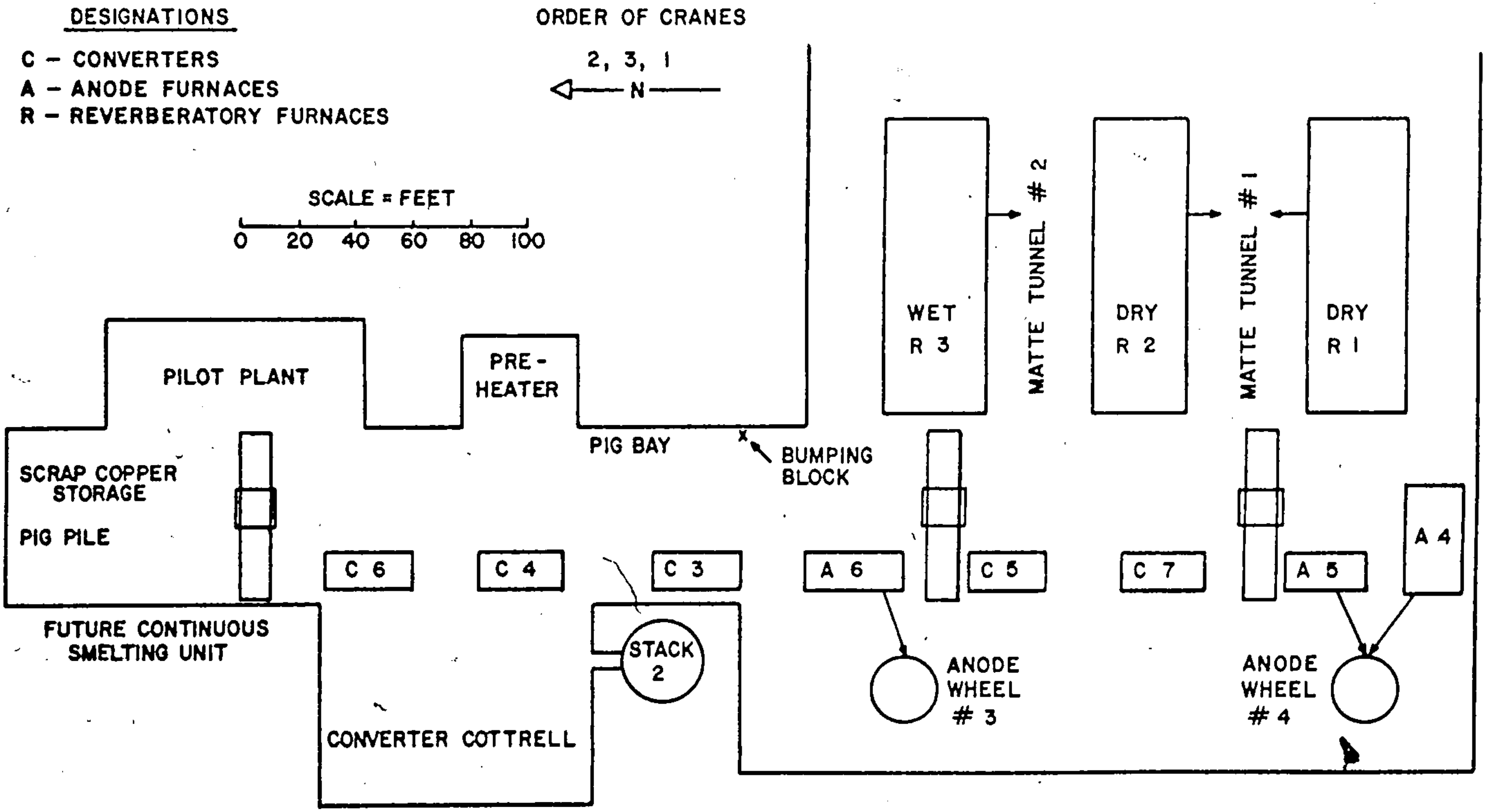


FIG.1.3.1 SMELTER LAYOUT

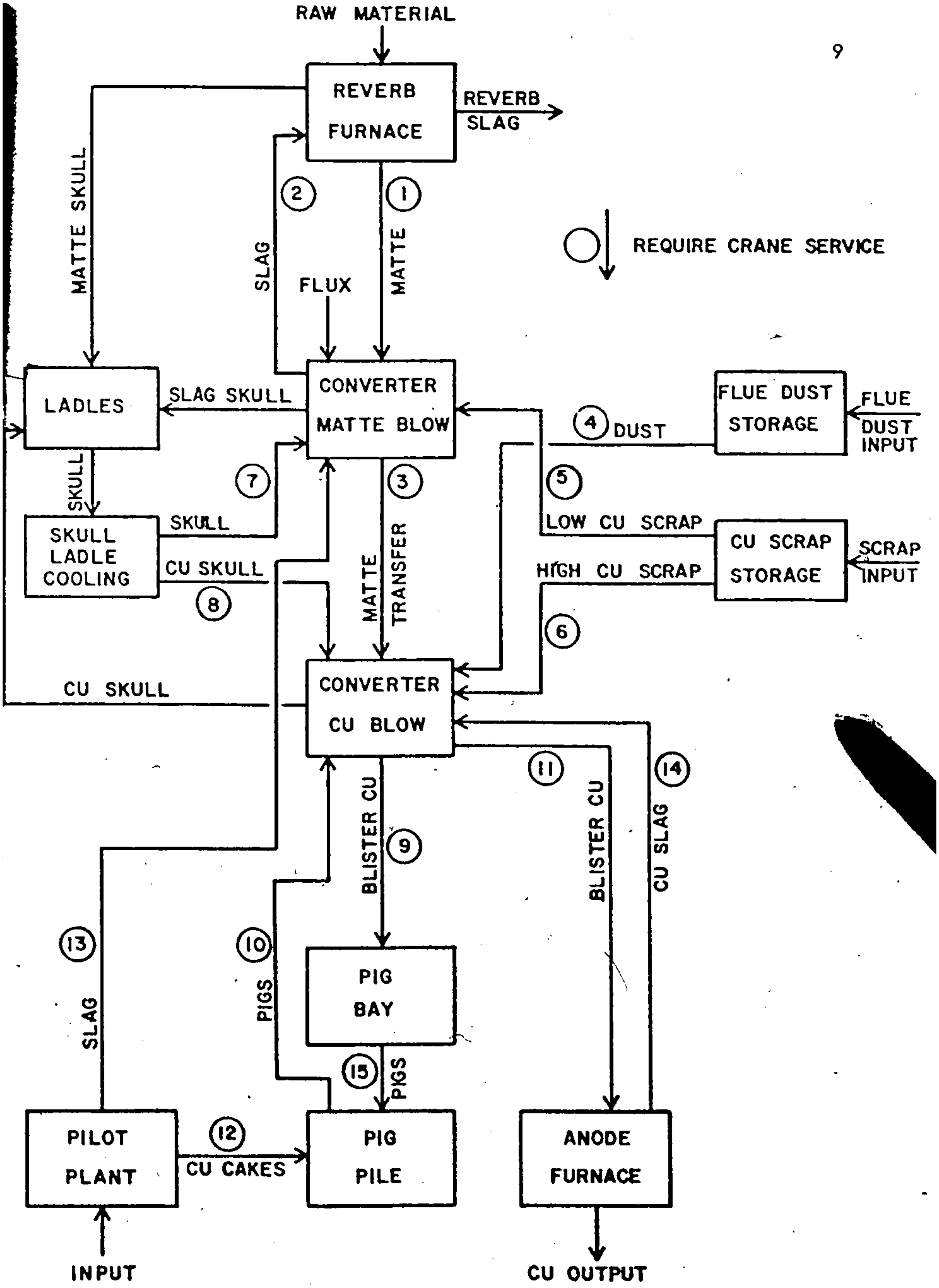


FIG. 2 MATERIAL FLOW CHART

2.0 LITERATURE REVIEW AND REVIEW OF THEORY

2.1 MODELLING OF COPPER SMELTER OPERATIONS

Control problems in many copper smelters have prompted several researchers to analyse and study these problems. There are numerous examples of studies directed at isolated processes of the copper smelting process, but only a limited number are reported in which the overall smelting process was studied.

Templeton and Hankley (T-1) viewed the converter aisle as a total system consisting of typically one reverberatory furnace, three converter furnaces, one anode furnace and one crane. Both discrete and continuous decision variables are included in the model and approaches to system optimisation are presented. However, for the given size of the system the problem becomes enormous and solutions for the entire problem are impossible. Sectioning of the problem produced a number of smaller sub-problems that could be solved more readily. A suggestion is made to combine the subsystems to produce results for the total system which is no longer guaranteed to be an optimal solution. Although the results of the simplified models are not directly applicable to a real situation the work represents a valuable contribution to the study of better control and optimisation procedures for the converter aisle problem.

Davies, (D-1) and Davies and Thixton (D-2) employed an empirical approach to study converter aisle operations. They report several improvements of smelter operations after implementing the results obtained from digital simulation models based on the GPSS simulation language (G-1).

Two models were developed, a macro and a micro model. While the latter was used to simulate smelting operations over a short period of time in great detail, the other provided also long term effects of variations of input variables such as converter overhauls and changes to plant layout and equipment. The simulation model was based on operating rules and statistical data from actual plant performance.

Foreman (F-1) reports of a large scale attempt of on-line process control of a converter furnace operation with plans to extend the programme to reverberatory combustion and overall smelter control.

At the time of reporting the on-line control of the converter was run in parallel to manual control, while refinement of the converter model was anticipated. Results showed that predictions of the on line model at the time did not match actual operating figures close enough, suggesting that more accurate on-line measurements would have to be obtained with the possible inclusion of further process variables. While this approach seemed very promising the entire programme was called off after a 10 month trial period.

The above studies are the most closely related to the work to be described here. Far more attention in research has been given to individual processes in copper smelting than to their integrated behaviour which is the aim of this work. When the development of submodels for particular processes is presented in Chapter 4 a literature review of work related to these processes will be given then.

Because of the general nature of the problem some work in other areas will be discussed in the next section before the major difference between previous work and this study is given.

2.2. MODELS OF OTHER INDUSTRIAL PROCESSES

The steelmaking and petrochemical industries have used computer techniques for process control extensively. Because a high degree of success has been achieved in some cases selected studies will be reported here. It would, however, be beyond the scope of this thesis to go into the detailed applications, but those studies should be mentioned that are more closely related to the work presented.

Schuerger and Tiskus (S-1) have classified automatic control of steelmaking processes according to their level of sophistication. In the basic application the control is indirect and a process model provides information on the process only, leaving control to the operator. On the other end of the scale the most complex applications involve direct control of process variables to achieve optimum performance. Numerous applications of process control of Blast Furnace, Basic Oxygen furnace (L-1) and Electric Furnace operation are found with some applications in continuous casting. Probably the most significant contribution of computerised process control is in the area of hot rolling of steel. Control is exercised over the entire process from the time hot billets enter the mill to weighing the finished coils.

Although success was not overwhelming in early applications the reliability of computer controlled systems has improved steadily and their use is becoming very common. However, even in these applications only individual units are controlled and the authors predict that "the next few years will see a concerted effort to utilise the computer to coordinate the production from many individual processes to maximise the profitability of the entire plant."

In the chemical industries some application of this kind exists, which indicates potential increases of the overall productivity of the plant from 2%-6%.

Turning away from on site applications of industrial process control, there are several theoretical studies worth mentioning.

Ashlour and Bindingnavle (A-1) formulated a soaking pit/rolling mill complex as a queuing system and produced a discrete event simulation model. The model is able to predict improvements by increasing the capacity of the pit model, while the system performance is measured using several different criteria.

In a similar model (A-2) of an entirely different system the authors investigated job-shop scheduling of a hypothetical machine shop. Effects of different loading rules of the system performance are discussed.

Cox and Ralston (C-1) report of a shop control system implemented in a manufacturing environment with the primary objective of obtaining optimum control over the level of work in progress. A major problem, the lack of full understanding of scheduling and priority rules are pointed out which are a prerequisite for an accurate control system.

A survey of system control studies shows that it is difficult to successfully implement a control system in a real environment without pilot studies. It is for this reason that in most cases the introduction of on-line process controls is centered around particular processes and will not be extended until these control systems are perfected. The overall control of the smelter system described by Foreman (F-1) probably failed for this reason. In the steel industry individual processes are by now understood and measurements are accurate enough that reliable control systems could be developed for them. However, the integration of these still lies ahead.

It is difficult or sometimes impossible to achieve reliable measurement of state variables under the harsh conditions encountered in practice. This difficulty combined with the lack of knowledge of some detailed process dynamics has so far prevented the development of reliable process models. In a recent report (0-1) on Operations, Analysis, Process Design and Process Control in the Mineral Recovery and Metal Production Industries, the urgent need for accurate models of processes and for reliable sensors for continuous process state measurement was stressed. The highest priority was assigned to these areas for future research.

The aim of this study will thus be the development of the most accurate possible model for the complete smelter system based only on the existing knowledge of the processes involved. In this way the problems of process measurement can be avoided.

While a simulation approach will be used it will differ from those described in Section 2.1 in three main respects:

1. The use of a hybrid simulation model which allows accurate and efficient representation of both discrete as well as continuous processes. This approach has so far not been used.
2. The inclusion of a human decision model in the overall smelter model which does not exist in the work of Davies (D-2).
3. An attempt is made to develop automatic heuristic control procedures for the model with the aim of overall system optimisation, in contrast to the approach taken by Tempelton (T-1).

2.3 MODELLING OF SYSTEMS

In the previous section reports of different approaches to analyse systems are discussed, all of which were aimed at the overall improvements or optimisation of performance.

The term "system" has so far been used freely and should be defined. In the context of an industrial environment a system can be defined as being a combination of a number of processes, some or all of which might interact. Associated with a system is a clearly defined boundary to it. Outside influences on the system are effected through variations in defined continuous or discrete decision variables. Internal variables or state variables define the status of the system at any time, that is they define the output of the system at a given time.

To distinguish between a physical system and a modelled system, which in turn is a system of its own, the modelled system will be referred to as the model. Subsections of the model will be referred to as submodels.

Careful analysis of the smelter system showed that the problem is too complex for an analytical solution. If the problem was reduced to a smaller one for which a feasible solution by techniques such as dynamic programming could be obtained, then the results would no longer be applicable to the real system. (Ref T-1) Drastic assumptions to simplify the system behaviour would on the other hand not produce a model that gives a detailed representation of the actual system.

In general the aims are to formulate a problem so that it can be solved analytically. Usually the more realistic the model becomes the more difficult it is to obtain an analytical solution. In this particular case the difficulty of modelling processes that are not all deterministic adds to the problem of developing a realistic model. Based on studies of the smelter operations it was decided that a simulation model of the entire smelting system would be the best tool for obtaining useful information about the actual system. Usually control problems in a system originate at a particular location or a process. Attempts to solve the problem are often directed at the area where a defect or problem shows up, thereby neglecting interferences from seemingly unrelated processes. Emphasis in this work is placed on studying complete systems rather than isolating a process from a larger system which might not be the cause of the problem that is investigated.

Mathematical models may be classified in several fashions.

A model can be deterministic or stochastic. In a stochastic model of a physical system variables are random functions of time whereas by definition the value of a variable in a deterministic model will be unique at any instant.

Another classification distinguishes between static or dynamic models. In a static model the state of the system does not vary with time whereas in a dynamic model time drives the system through different states.

In dynamic models the state of the models may change continuously or at discrete time intervals leading to another means of classifying models, namely discrete or continuous models. Discrete models are represented by difference equations or by logical state transition equations, whereas continuous models are based on differential equations. The time rate of change of variables is given continuously by these relationships.

A further means of classifying models is to specify the degree of the response of a system to changes in the input. A linear model of a system is one in which change in input produces an equivalent change in the output.

By stating the means used to derive the model behaviour a further classification into empirical or mechanistic models is possible.

The model of the smelter system developed here is a mixture of some of the idealised modes above and could be said to contain mixed variable, dynamic, semi-empirical, non-linear models.

2.4 SIMULATION

Simulation is the immitation of the behaviour of a real system through the use of a model. A more precise definition is given by Sayre and Crosson (S-6) by distinguishing it from mathematical models: "A simulation model is a symbolic (as opposed to a physical or material) representation of a phenomenon or system, yet in contradistinction to mathematical models, the symbols of simulation are not all manipulated by means of a well-formed discipline, such as algebra, the integral calculus, numerical analysis or mathematical logic." Simulation models will have many different appearances but the approach in general to develop them can be summarised as suggested by Ackoff and Sasieni (A-3):

1. Problem definition.
2. System Analysis
3. Model construction
4. Model Validation.

A stage beyond this point is the analysis of the model and implementation of results in the real system.

The nature of the smelter system required special attention in order to simulate it effectively. Usually dynamic, continuous models are easily modelled on an analog computer,while discrete processes lead themselves to be dealt with on digital computing equipment.

Shapiro (S-2) analyses the problem of choosing a simulation system and discusses the pros and cons of purely digital, purely analog and of hybrid systems.

The conclusion, that there are particular problems that fit each type of computing system is obvious. It is more difficult to recognise them as such. Svorcek (S-4) reports of a particular selection of a simulation tool for dynamic simulations. He points out the particular advantages of a hybrid system for real-time-interactive simulation studies.

By the time the second stage in developing a simulation model, the system analysis stage, is reached, a thorough understanding of the processes will allow a better decision to be made regarding the choice of a simulation system.

Hybrid simulation was chosen for this particular study and is recommended for studies of similar systems for the following reasons:

1. Possibility of real-time simulation.
2. Both discrete and continuous processes can be modelled efficiently and accurately.
3. Interactive simulation or man-machine communication is easily facilitated on a hybrid system. Input through a typewriter and output via lineprinter and a number of peripheral devices is easily facilitated. In addition special display units can be readily connected to display continuous as well as discrete variable states.
4. Parallel processes can be simulated easily on the analog computer, which is a very important aspect in modelling complex systems.
5. In addition to these points a hybrid system naturally has the advantage of not suffering from short-comings of either an

analog or a digital system if the other makes up for it.(e.g. the lack of memory in analog computation is overcome in a hybrid system - or the tedious task of initialising an analog circuit is removed in a hybrid system by automatic setup programs from the digital computer.)

Hybrid simulation certainly has its disadvantages. Program preparation and setup time is longer on the hybrid system compared to the fast use of many of the available digital computer languages.

Even more time consuming and complicated is the debugging of a hybrid computer program. It is not as easy to pinpoint a problem since processes operate in parallel, and errors can occur in the analog or digital system or in the conversion of data between them.

In this application the advantages outweighed the disadvantages simply because the only disadvantages that arose, occurred in the initial stages when the model was built, and thus occurred only once. The advantages however can be appreciated every time the simulation model is used and in part added to its successful application.

2.5 MODEL DEVELOPMENT

In the previous two sections some background on simulation and on classification of models was given.

This chapter deals with the approach taken to obtain all necessary information about the system and how the information was analysed to produce a model of the system. Whereas a wealth of literature on modelling and simulation of systems is available (S-3, M-1, A-3, F-3, M-2, C-3, M-6) it is difficult to extract from their useful guidelines on how to go about simulating such systems in practice. For this reason a short summary of the approach to develop the simulation model is given here.

The obvious first step after recognizing and defining the problem of a complex system is to start on a detailed literature review. Apart from finding out about other work done in the particular area, the review will provide insight into the processes involved and give a better appreciation of the problems that are likely to be encountered. Very useful are publications or reports on the particular system itself if these are available.

At this stage a field study of the physical system will prove very useful. It will give the analyst an opportunity to talk to people involved in the system to obtain first hand information on the operations. In trying to get some information on problem areas he will invariably get a number of suggestions as how to solve them.

If results of the simulation model are to be interpreted by other people it is important to determine what information they will require from it so that it can be provided.

With this background it will be possible to set up a preliminary model and decide on the variables that need to be included to represent the system satisfactorily. The next step is to establish how the model can be best simulated. The reasons for choosing a hybrid simulation model were given in the previous section and need not be repeated here. A very important decision, however, is the choice of a simulation language.

Problem oriented languages were developed in the past 15 years when the use of digital simulation became more predominant. The task of modelling details of systems is tedious if a procedural language such as FORTRAN is used.

Discrete event simulation languages make use of such terms as events, attributes, and files to describe quantities commonly encountered in most simulation models. The use of a simulation language is certainly recommended although it requires learning the new language. Any simulation model could be produced from first principles, that is writing a program in one of the procedural programming languages if there was no other possibility. The effort, however, is considerably reduced when using a simulation language where in some cases the completion of a check list is all the programming required. Some of the more widely used languages are GPSS (G-1), SIMSCRIPT (K-1) and GASP (P-1). These languages are referred to in most textbooks on simulation where useful comparisons are made that could aid in selecting a particular one. (S-3, M-1)

Tocher (T-1) and Buxton (B-1) discuss a number of simulation languages and compare them on several points.

The GASP (P-1) discrete event simulation language was chosen for the hybrid simulation study for basically three reasons:

1. It was available immediately.
2. It is a FORTRAN based language which can be accommodated on a small computer.
3. It is a flexible language that can be modified to suit a particular problem. (This may not be true for all other simulation languages).

More details of its operation will be given in Chapter 4 where the development of a discrete-continuous simulation system is described.

Having decided on an approach to simulate the system the field study has to be defined. All data necessary should be listed together with the means of obtaining it. This means that a good understanding of the processes involved is required by this stage.

Much of the required data can be extracted from regular operating data, but it will have to be supplemented by on site data collection if not sufficient data is available. The accumulated data should then be ordered since use will be made of it throughout the development of the model.

It is important to delay any attempts to start on the actual model building as long as possible. Early attempts tend to become drawbacks later when numerous changes have to be introduced that were not accounted for initially.

Even when the model development has started it will frequently be necessary to go back to the analysis stage if it is found that additional information is required. Developing a good model is a long process and should be carefully undertaken to pay off with reliable results.

The inclusion of human decisions into a simulation model of the kind described in Section 1.3 was given special attention, since the system analysis showed a significant effect of these decision variables on the overall performance of the system.

The model will gradually grow and change until an operational version is ready for testing. Modular design is always recommended in large models. Any corrections or additions can thereby be handled efficiently without upsetting the overall progress of the model.

The final version of the simulation model will then be tested for validity. A simulation can be considered valid if it satisfies preset conditions of accuracy, it can and should not duplicate a real system exactly unless the system is completely deterministic.

Maisel and Gnugnoli (M-1) suggest three ways in which validity can be assured.

1. Built in checks of validity in the simulation.
2. Expert reaction to simulated results.
3. Comparison of results with standards or operating records.

All three possibilities should be made use of to ensure proper validation of a simulation model. A final test, if this is possible in practice, would be to use the model to predict a given response to a change in input which can then be verified by the same test on physical system.

All four means of checking the validity were used in the simulation model and are described in Chapter 5.

3.0 SYSTEM ANALYSIS

3.1. COPPER REFINING PROCESSES

Copper sulphide concentrates used by the smelter contain amounts of copper sulphide (Cu_2S), silicon oxide (SiO_2), iron oxide (FeO), iron sulphide (FeS), and a certain percentage of moisture. Traces of silver and gold are valuable metals and are removed during the final electrolytic process, but play no important part in the smelter refining processes. The exact composition of the ore varies and depending on it, the grade of matte produced by the reverberatory furnaces will change. Fluxing ore, consisting essentially of silicon (SiO_2) is added to the concentrates before they are added to the wet charged furnace. (Furnace #3 on Figure 1.3.1)

The dry charged furnaces are fed with calcine, that is, concentrates with proportions of flux added that have passed through roasters. Calcine has usually got a lower sulphur content and contains no moisture. It is transferred to the hot charged furnace at elevated temperatures to conserve heat.

The ore to be smelted is entered along the two longest sides of the reverberatory furnaces and forms banks on either side leaving a gulley in the middle along which the molten ore runs to form a liquid bath at the east end. Heat is supplied by the burners in the west end of the furnace. The furnace is fired by oil and a forced air flow along the length of the furnace melts the material below it. A slag launder is situated between the burners and is used to return converter slag into the reverberatory furnace.

There are two tapping holes for removing two phases of molten ore from the furnace. The slag hole is situated in the east wall and is located higher than the matte hole on one of the long side walls also near the east side. Figure 3.1.1 gives a schematic representation of a reverberatory furnace with its most important features.

While the basic function of the reverberatory furnace is that of smelting the ore,a chemical reaction taking place to a limited degree is the same as the one described later in the matte phase of the converter, except that the amount of flux is limited to just remove all iron oxides.

$$SiO_2 + 2FeO = 2FeO\ .SiO_2 \qquad - \qquad (3.1)$$

Matte, which is essentially a liquid eutectic mixture of copper and iron sulphides (Cu_2S, FeS) from the hot charged furnaces is in general of higher copper grade than that from the wet charged furnace but in both cases the specific gravity is higher than that of reverberatory slag. Reverberatory slag contains fayalite, silicon oxide bonded to ferrous oxide, and most of the gangue and forms the upper layer of the liquid bath in the furnace. Slag floating on top of the matte can be tapped from the furnace through a hole located above the matte hole. The layers are not distinctly separated and when this region is near any tapping hole no material is removed until the bath level has changed sufficiently that only slag or matte can be tapped from the respective holes.

It is attempted to run the reverberatory furnaces at a fixed output and supply matte at constant intervals. Ladles are transported along a track in the two matte tunnels and are picked up by a crane at the converted aisle (Figure 1.3.1.)

As are all smelter processes, the reverberatory furnaces are run on a 24-hour basis with repairs to the brick linings being done continuously. This has only been possible since the introduction of the suspended roof, in which each brick is supported from a framework above the furnace.

Matte is picked up by a crane and poured into a converter. Figure 3.1.2 is a schematic diagram of a typical converter showing its principal components. The reactions taking place during the matte phase have been a subject of considerable interest and the following reactions are considered possible:

$$3FeS + 5O_2 = Fe_3O_4 + 3SO_2 \quad (3.2)$$

$$3Fe_3O_4 + FeS = 10FeO + SO_2 \quad (3.3)$$

$$2FeS + 3O_2 = 2FeO + 2SO_2 \quad (3.4)$$

$$4Cu + 2FeS + O_2 = 2Cu_2S + 2FeO \quad (3.5)$$

$$2Fe + O_2 = 2FeO \quad (3.6)$$

$$2FeO + SiO_2 = 2FeO \, . \, SiO_2 \quad (3.6)$$

$$6FeO + O_2 = 2Fe_3O_4 \quad (3.7)$$

It is difficult to determine exactly which intermediate reactions take place to remove the iron sulphides in the slag stage but it is considered that the following two reactions represent the process accurately (N-2):

$$2FeS + 3O_2 = 2FeO + 2SO_2 \quad (3.8)$$

$$2FeO + SiO_2 = 2FeO.SiO_2 \quad (3.9)$$

Both reactions are exothermic and provide the necessary heat to sustain the reaction while cold flux (SiO_2) is added. If not enough flux is present magnetite will be formed according to equation (3.7). Magnetite if returned in the converter slag to the reverberatory furnace settles on the floor causing an unwanted buildup of material. The production of magnetite is thus avoided by adding enough flux in the matte phase. During the matte phase several ladles of matte are poured into the converter, and while correct proportions of flux are added, compressed air is blown through the bath via tuyère holes. When most of the iron sulphides have been oxidised they are removed by skimming converter slag from the bath. More matte is added and the process is repeated until between twelve and eighteen ladles of matte are contained in a converter. Transfer of material at this stage from one converter to another is possible and will be discussed in the next section. Once all iron sulphides are carefully oxidised (the converter bath temperature is an indicator) the converter is said to go on a copper blow. In this phase the remaining copper sulphide (Cu_2 S) or white metal is oxidised according to:

$$Cu_2S + 0_2 = 2Cu + S0_2 \qquad (3.10)$$

This reaction is again exothermic and natural heat losses have to be supported by artificial cooling to maintain a constant bath temperature. This is accomplished by adding cold copper lumps of several tons weight to the bath at regular intervals during the copper blow phase. Towards the end of the stage samples are taken to indicate the end point when further blowing would oxidise the accumulated copper.

The copper, now called blister copper is then poured into ladles and transported by cranes to one of the anode furnaces. Blister copper is about 99% pure and refining in the anode furnaces increases the purity up to 99.9%. Traces of sulphur are oxidised or in case of an overblown charge the copper oxides that were formed are reduced.

The refined copper is then cast on one of the two available anode wheels and anode plates are shipped for electrolytic refining.

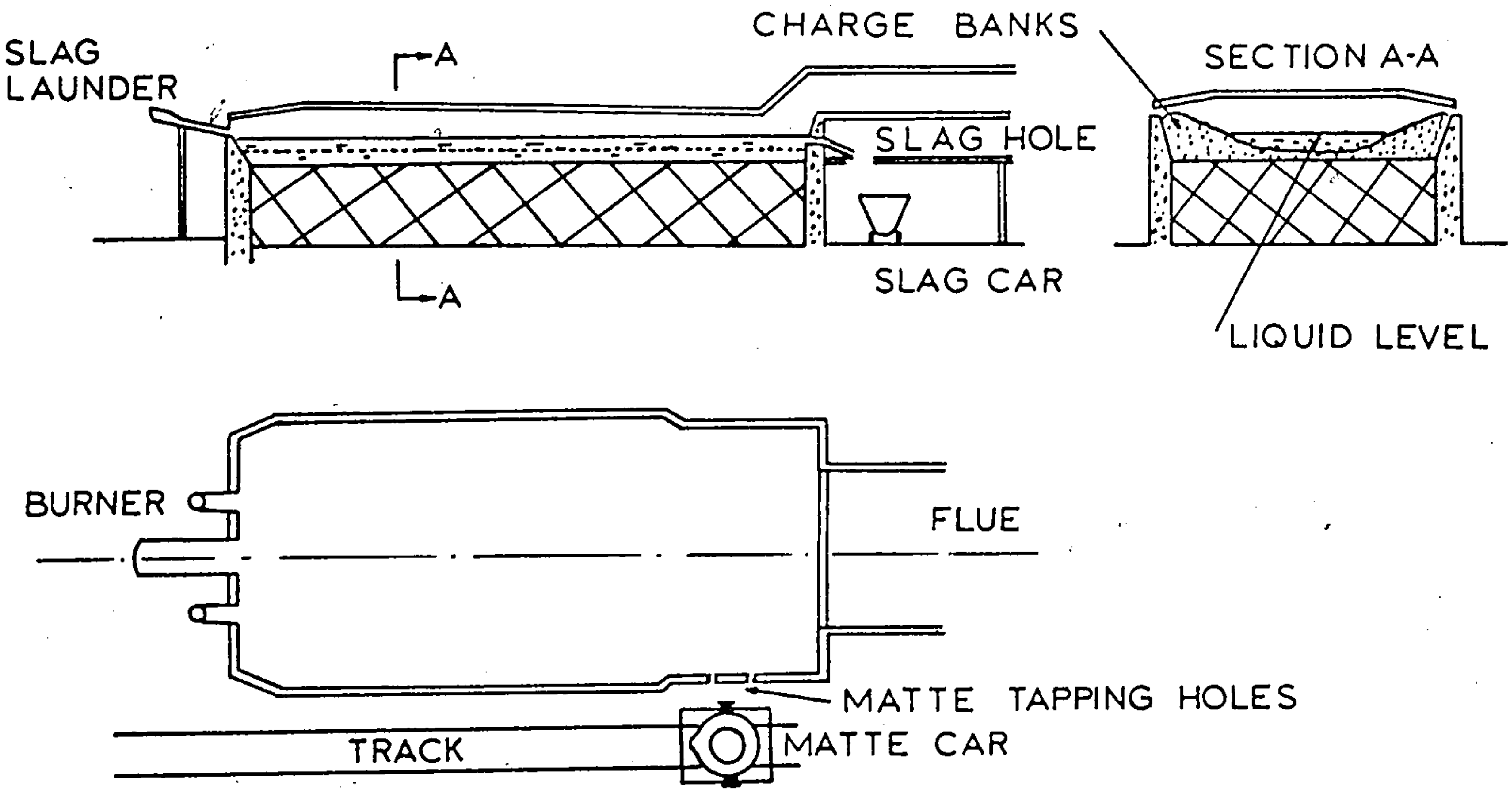


FIGURE 3.1.1 TYPICAL REVERBERATORY FURNACE

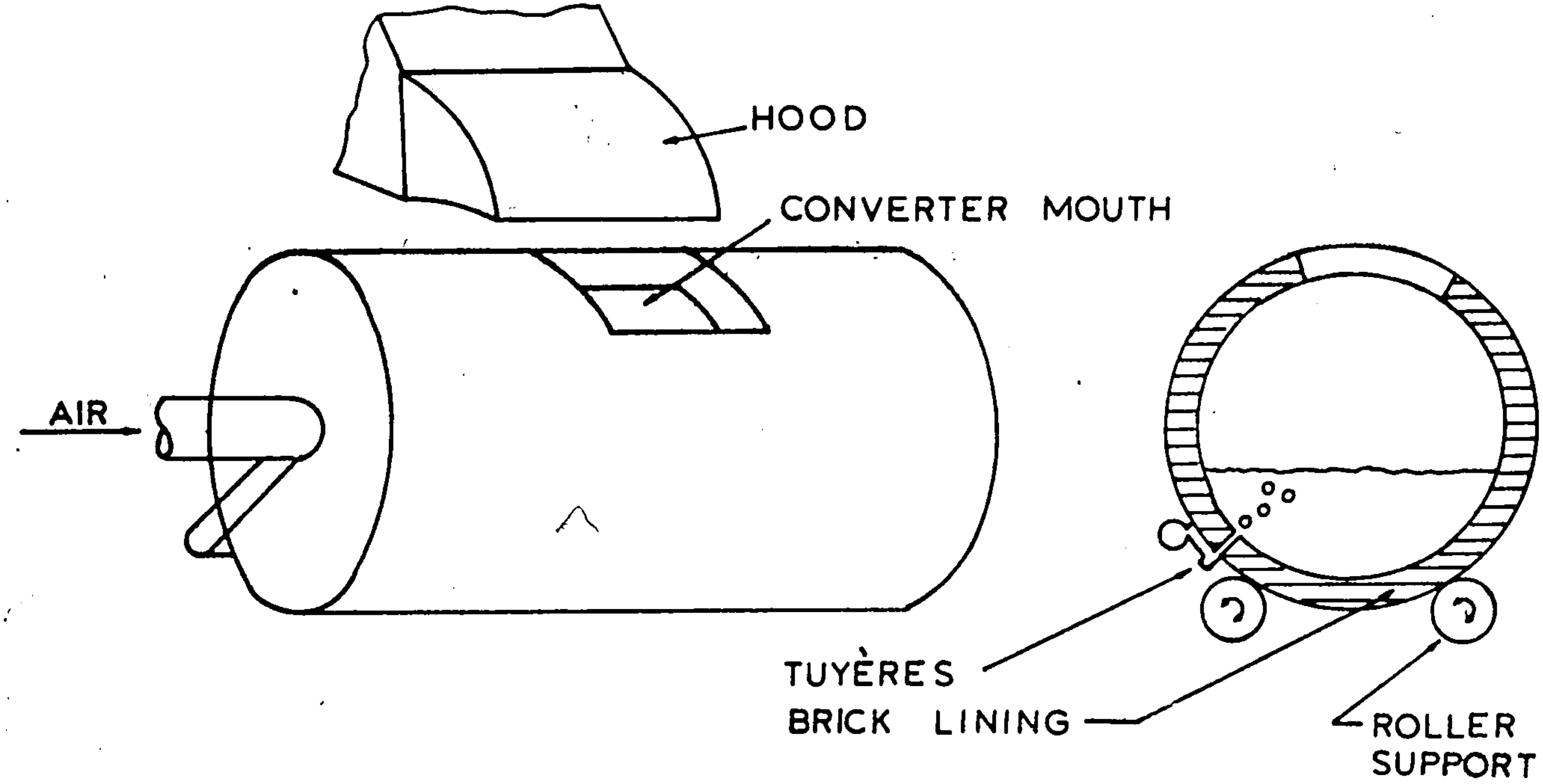


FIGURE 3.1.2 COPPER CONVERTER FURNACE

3.2. MATERIAL TRANSPORTATION BY CRANES

The reverberatory furnace smelting process described in the previous section is a continuous process, while material removal from it, matte and slag tapping occur at irregular intervals. Whenever a copper blow is completed in a converter a skull is dropped into the converter and the copper is removed from the furnace. The furnace is then almost completely empty and is said to start a new refining cycle. During a cycle a crane adds several ladles of matte and skims slag after the matte blowing process is completed. It has been the practice to schedule converters such that at any time matte is blown in three converters while the fourth is in stage three of the smelting process, the copper blow. One of the three converters in the matte blow will proceed with the copper blow around the time when the other copper blow is completed.

During the matte blow material from one converter can be transferred to another to maintain this schedule. This process is usually called "matte transfer" or "white metal transfer". The converter receiving transfer matte will then be ready to proceed with a copper blow sooner, while the supplying converter could be empty again to start a new cycle.

Scheduling of material transfers is critical since matte should only be transferred when it has been completely refined in a matte blow. To do this efficiently any two converters involved in a material transfer have to be ready to receive or supply transfer matte at more or less the same time to avoid any idle time. The availability of the receiving furnace can be delayed by adding more matte to it.

All additions of matte, transfer matte, anode slag, reactor slag, copper pigs, flue dust, slag skulls and copper scrap to the converter and material removals of converter slag and blister copper are performed by the overhead cranes. Collar pulling on a converter is a very time consuming but necessary service also performed by a crane, when a built up collar of material has to be removed from the converter mouth.

The reverberatory furnace receives converter slag by the cranes who also remove matte from the matte cars whenever required.

The pilot plant shown on the north end of the floorplan produces certain quantities of slag and blister copper that also require a crane to be moved.

The position marked as Future Continuous Smelting Process is reserved for a new process that will be included in the model later.

Blister copper is added to the anode furnaces while anode slag is skimmed from them before the refining process commences.

Solidified crusts in ladles are loosened by a crane in bumping the ladles against a large block, and cold copper pigs often need to be moved from the pouring bays to a pig pile.

All the listed crane tasks are performed by the two cranes near the south end of the converter aisle independently or in conjunction depending on their availability and the urgence of a task. Crane services have to be scheduled carefully especially when the times of service requirements can not be exactly determined in advance.

The task of controlling all smelter processes in an optimum way thus presents itself to the converter foreman continuously even if a smooth schedule is drawn up, any unexpected delay, breakdown or irregularity upsets the plan and calls for a new assessment of the overall shift plan.

3.3 SMELTER MODEL

The system analysis showed that the simulation model has to provide enough detail to accurately reproduce all activities in the smelter as seen from the converter aisle. That is all material inputs and outputs from processes together with all related activities have to be provided by the model at the same rates and subject to the same constraints as the system.

Since an off-line model cannot make use of continuously measured process state variables, the model should be based only on variables that can be readily measured. It was thus necessary to obtain a satisfactory dynamic representation of the system operation based only on present knowledge of smelter processes.

In Chapter 4 individual submodels will be described. The models were designed to produce a smelter system responsive to varying material input and output rates, while moving of materials or scheduling of processes is directly controlled by the operator of the simulation model. He thus takes the place of the smelter foreman in the actual plant.

In the real system the influence of the converter foreman on the running of the smelter is very evident and any model excluding this effect could not be expected to reproduce the system satisfactorily. The facilities necessary to include the human model will be discussed in the next chapter.

4.0 SMELTER MODEL DEVELOPMENT

4.1 DEFINITION AND OVERVIEW OF THE SYSTEM MODEL

The boundaries of the system were defined and are shown in Figure 4.1.1. By excluding processes outside the boundary the assumption is made that the state of these processes has no direct bearing on the operation of the system. Their dynamic behaviour can thus be represented by input variables, while output variables have no feedback to the system.

This is in close correspondence with reality and allows a definition of the boundaries such that their presence does not affect the operation of the model.

The smelter model was divided into several submodels as they were logically provided by the system. These submodels are:

Reverberatory Furnaces

Converter Furnaces

Anode Furnaces

Cranes

Miscellaneous Processes and Units.

Model of the foreman (human, for which only interactive facilities have to be provided)

Before discussing these processes and their interactions the essential feedback from the different processes to the model operator should be given some attention.

A model display unit Figure 4.1.2 was built at the National Research

Council which provides light indicators for various process states and position indicators for the crane models. This display unit proved to be of great value for both modelling and operating the simulation model. During the validation phase of the model, operating personnel found it very easy to understand and follow the simulation process through the feedback obtained from the display.

It consists of three servo driven motors that position a pointer (crane) according to a voltage supplied by the analog model of the cranes. On the pointer two light indicators provide information on the state of the crane. The rest of the display resembles the floor plan of the smelter where 77 light indicators of different colour show the state of the modelled processes. Whereas the crane position is derived directly from the analog models of the cranes the light indicators are set in the digital submodels of respective processes and serve as coded signals. By calling a subroutine LIGHT with the corresponding light number as an argument, lights are set and reset through signals from control lines. Figure 4.1.3 shows the information that each indicator provides on the state of the three types of furnaces.

A speedup factor of 20:1 was chosen for simulation and the display unit is scaled down 500:3 to allow the fastest possible simulation of the smelter which still allows the operator to follow the operations. A line printer, stripchart recorders and digital voltmeters provide all the remaining information on the model. Figure 4.1.4 shows the EAI 680 analog computer with the model display unit connected to it.

Fig. 4.1.1 only shows the fundamental processes that have to be included in the model, it does not show the individual units, and how these interact. The function of the foreman is also not included. The model that was developed for the entire smelter system is presented, again simplified~~, on Fig. 4.1.5~~. It shows the function of each of the six major submodels and how these interact. It should be pointed out that material and information flow exists between all units of each type of process. This complicates the function of the operator and leaves wide room for error and sub-optimal operation.

To be able to fulfill his control function efficiently the foreman has to obtain complete and up to date information on the progress of all units in the system ~~as~~ a basis on which to formulate his decisions and issue commands.

Table 4.1.1 lists the type of actions he takes if alerted by any condition in the model. Some of the actions are simple but all require planning and decision making so as to ensure an efficient operation of the overall system. So far the discussion of the model has been di vorced from the implementation. The need for efficient information display of the model will, however, be related to some extent to the particular implementation. Fig. 4.1.6 shows the peripheral devices that were used to relay information from the model to the operator. Also shown are facilities that allow on-line control of the model. All of the triggers and actions listed on Table 4.1.1 are communicated by the outputs and inputs as shown.

All output signals but one are of visual form. A bell ringing on the typewriter is, however, very successfully used to alarm the operator when an important action is required and in principle any other device that can be interpreted by human senses could be used.

The display unit described above is the central output device which will be observed almost continuously by an operator. Some of the processes will be represented by digital models, others by partly digital and partly analog models. Analog variables if given by stripcharts, voltmeters or servo driven motors will show continuous variations. Output from digital models will change at discrete intervals when state variables and statistics are updated. At this time the coded light signals will change if the corresponding process reaches or leaves a given state. Information on any process is only provided if requested except for messages on the typewriter to alert the operator or request action from him.

As a reference to Chapter 7 where a completely automatic control of the model is described Fig. 4.1.5 also shows that the control function of the operator could be replaced by a non human control program.

Following the description of the overall model and the particular attention given to the operator submodel the next section will deal with the development of the hybrid simulation mechanism which produced a realistic operating model. The last sections in this chapter are then devoted to the development of the 5 major models and their implementation.

Because of the size of the problem not all submodels were developed by the author. The crane submodels which were to be mixed variable models, and the reverberatory furnace and matte tunnel submodels, which were to be represented by purely digital discrete event type models, are representative models for the system and were developed by the author. The mixed variable converter furnace submodel was developed by L.K. Nenonen of the N.R.C. who has had valuable previous experience in the converter control studies.

The anode furnace and miscellaneous process submodels were to be discrete event type models and were developed by Dr. U.P. Graefe also of the N.R.C. . Dr. Graefe also provided a subprogram that decodes the input commands to the model and translates these into corresponding events.

After describing the overall operation of the simulation a brief description of each submodel will follow with further details on each presented in the Appendices.

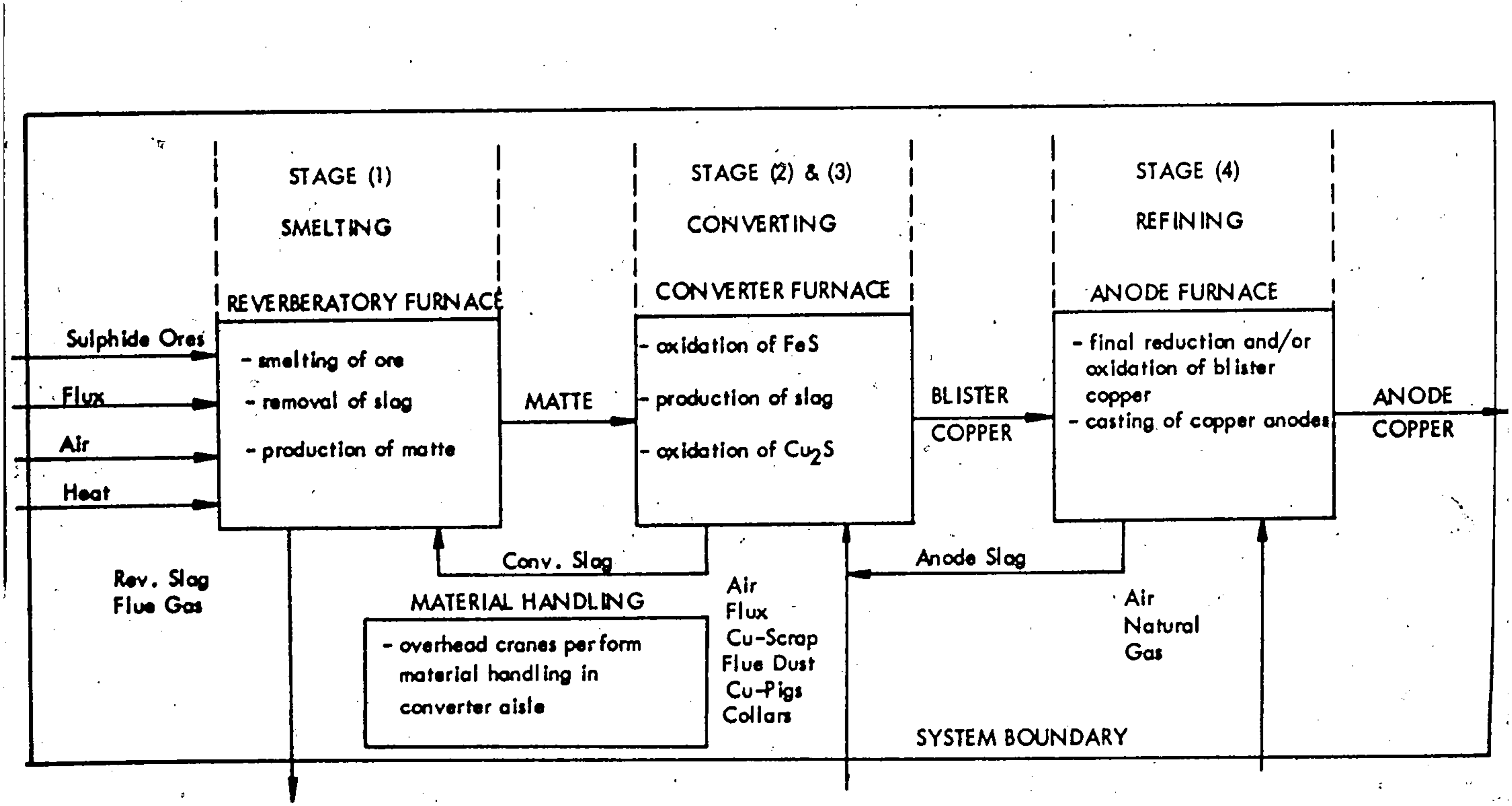


FIGURE 4.1.1 COPPER SMELTING PROCESSES WITHIN SYSTEM BOUNDARY

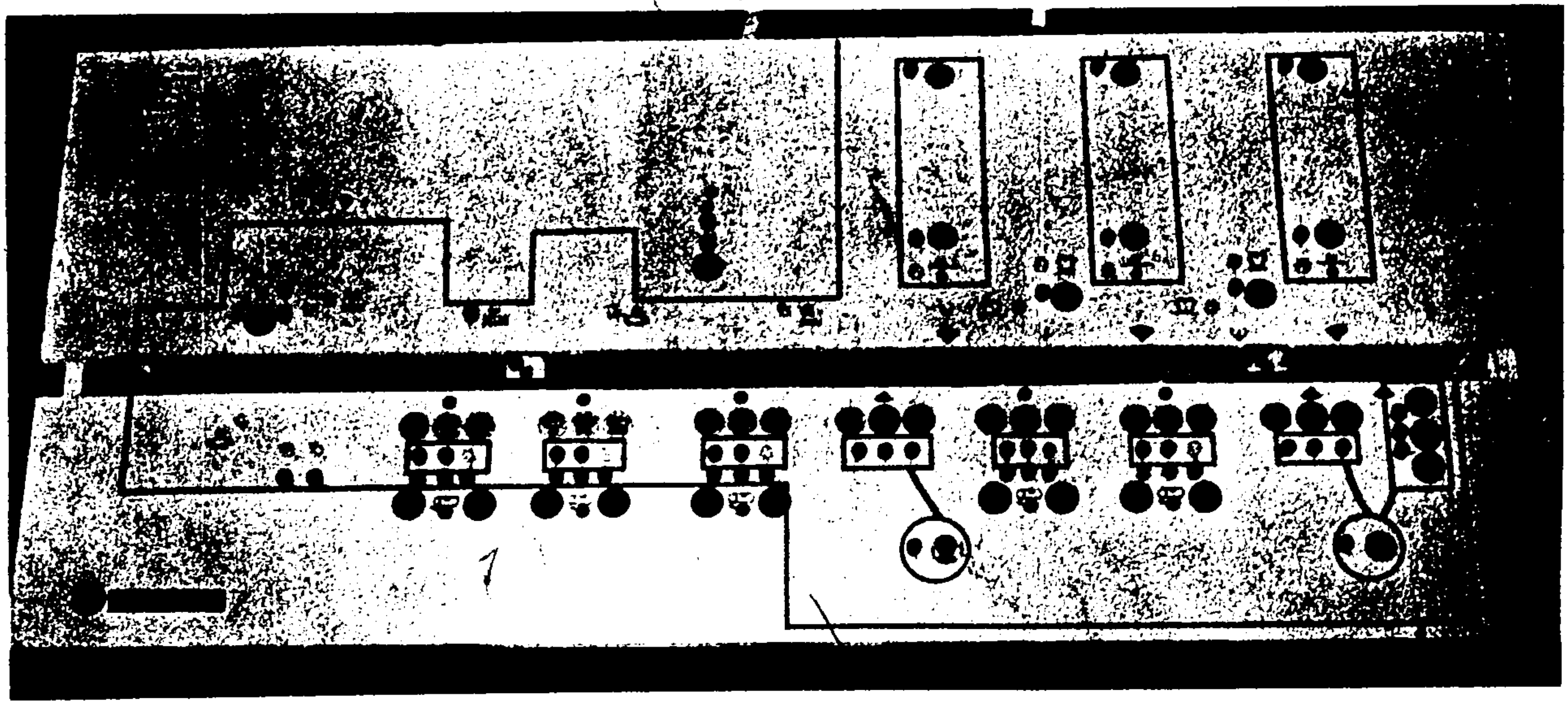
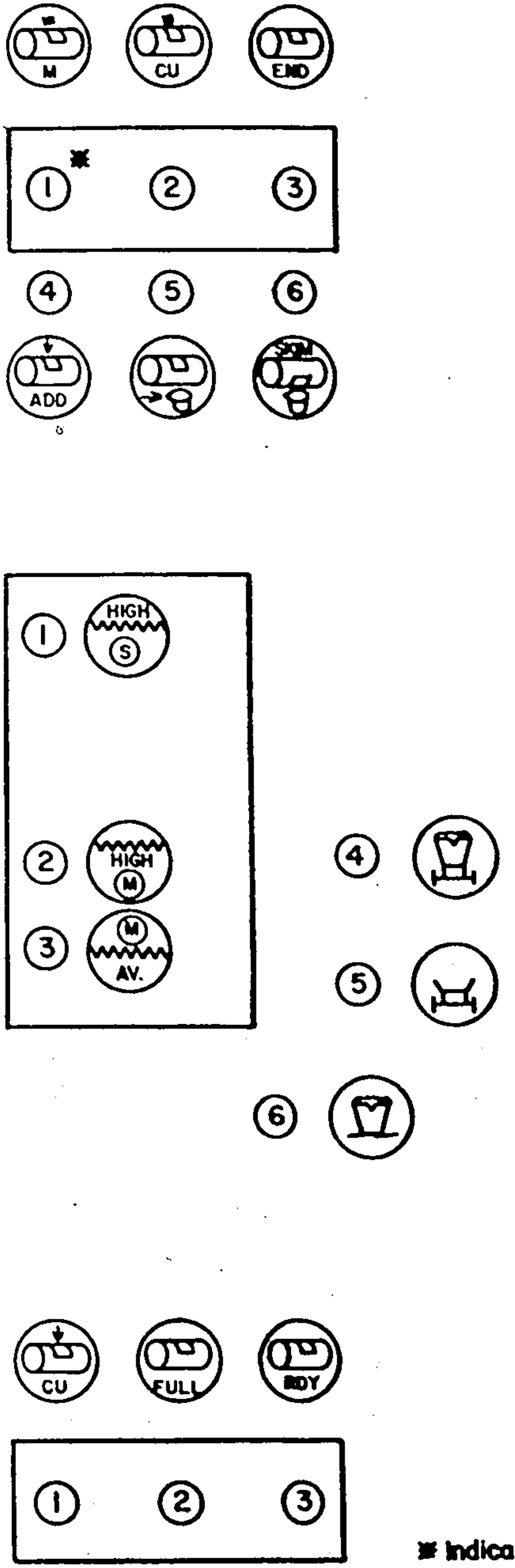


Figure 4.1.2 Smelter Display Unit



CONVERTERS

(1) Matte stage

(2) Copper stage

(3) At an end point, not blowing

(4) Material to be added

(5) Empty ladle required

(6) Material to be removed

REVERBERATORY FURNACES AND MATTE TUNNELS

(1) High slag level

(2) High matte level

(3) Matte available

(4) Matte on matte car

(5) Empty ladle needed

(6) Full ladle of matte on floor

ANODE FURNACES

(1) Blister copper required

(2) Full, ready for refining

(3) Refined, ready for tapping

* Indicator lights

FIG 4.1.3 SMELTER DISPLAY UNIT : INFORMATION FORMAT

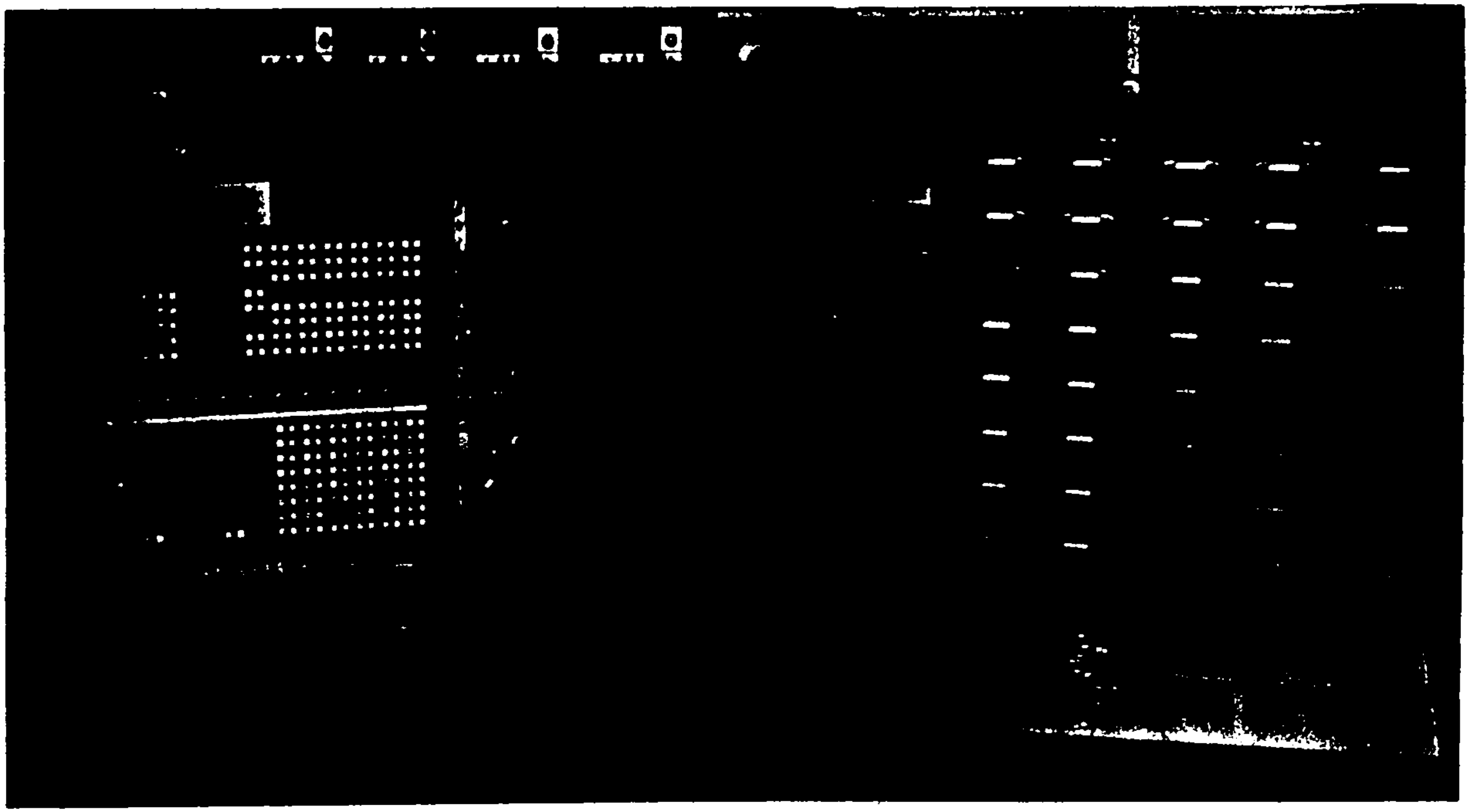


Figure 4.1.4 Analog Computer and Display Unit

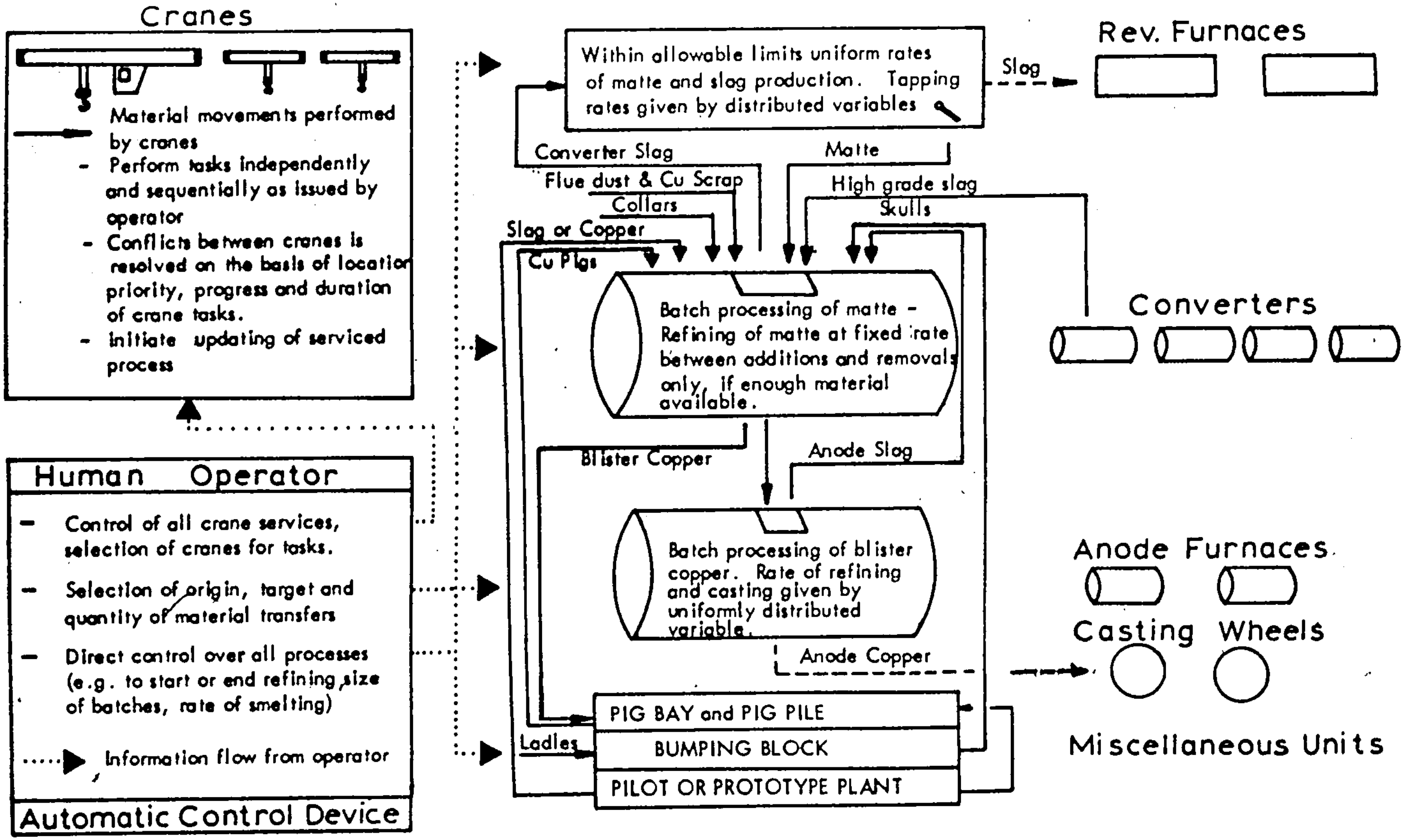

Cranes
Material movements performed by cranes
- Perform tasks independently and sequentially as issued by operator
- Conflicts between cranes is resolved on the basis of location priority, progress and duration of crane tasks.
- Initiate updating of serviced process
Human Operator
- Control of all crane services, selection of cranes for tasks.
- Selection of origin, target and quantity of material transfers
- Direct control over all processes (e.g. to start or end refining, size of batches, rate of smelting)
Information flow from operator
Automatic Control Device
Within allowable limits uniform rates of matte and slag production. Tapping rates given by distributed variables
Slag
Rev. Furnaces
Converter Slag
Flue dust & Cu Scrap
Collars
Slag or Copper
Cu Pigs
Matte
High grade slag
Skulls
Converters
Batch processing of matte - Refining of matte at fixed rate between additions and removals only, if enough material available.
Blister Copper
Anode Slag
Batch processing of blister copper. Rate of refining and casting given by uniformly distributed variable.
Anode Copper
Anode Furnaces
Casting Wheels
Miscellaneous Units
Ladles
PIG BAY and PIG PILE
BUMPING BLOCK
PILOT OR PROTOTYPE PLANT

FIGURE 4.1.5 Representation of Overall Model showing Material and Information Flow

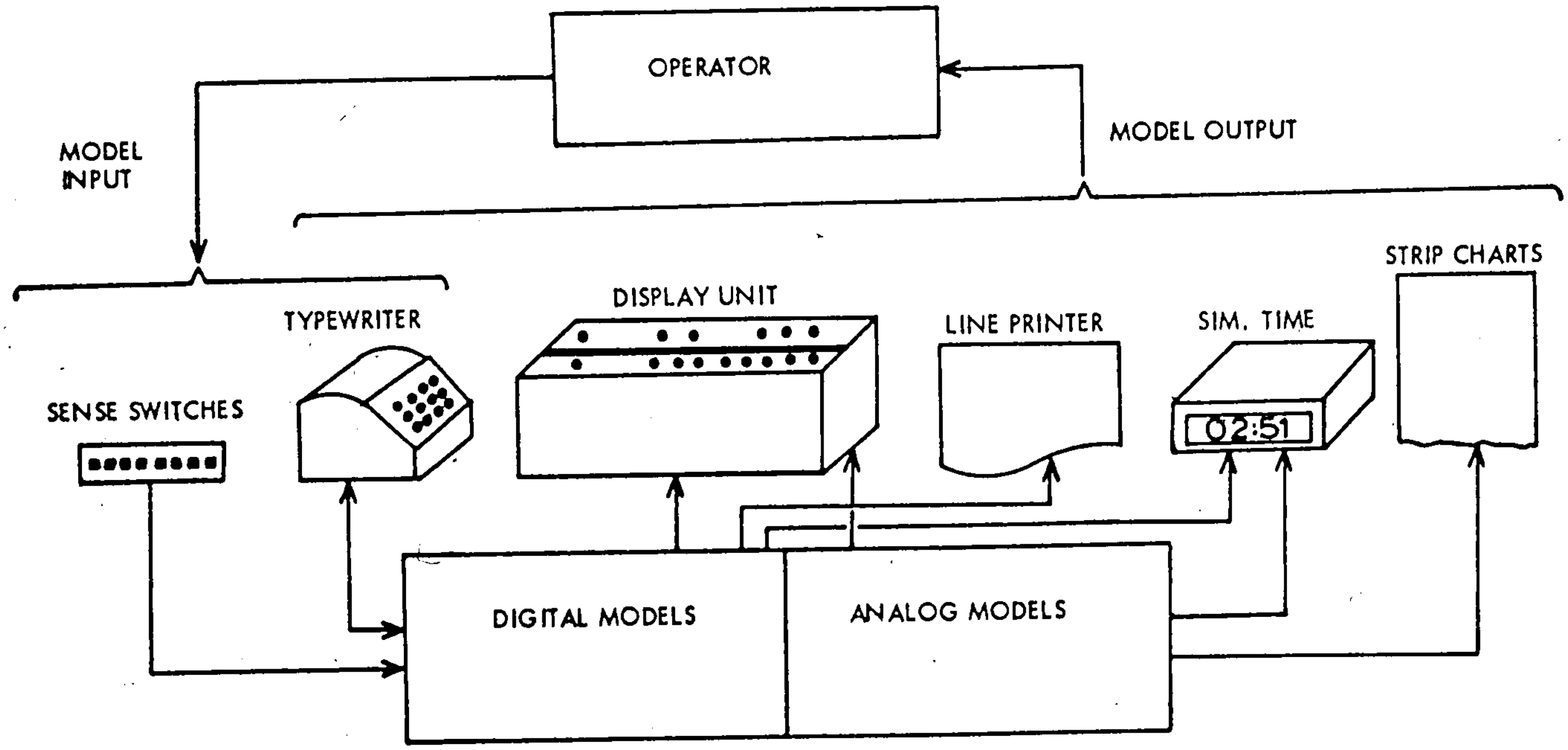


FIGURE 4.1.6 Operator and Model Interaction via various Input and Output Devices

Trigger (status of process or event)	Response (resulting command or plan)	Critical Decision (selection of . . .)
Arrival of matte at matte tunnel	direct to a converter	Converter, crane
End of matte blow in converter	cause skimming, new matte addition	Crane
Start of copper blow in converter	cause addition of flue dust	Crane
End of copper blow in converter	cause addition of skull, skimming of copper	Crane, anode furnace
Stock of skulls low	cause ladle bumping	Crane
Cu-pig required by converter	direct pig to converter	Crane
Stock of pigs low	plan for casting of pigs	Converter
Matte required by one or more	plan distribution of matte as it becomes available	Converters
Material available at pilot or prototype plant	direct to a converter or pig pile	Converter, crane
Anode furnace to start refining when full, or when decided by operator	cause skimming of anode slag then cause start of refining	Crane
Anode furnace completed refining	cause start of casting to fit scheduling of anode wheels	
Converter requires collar pulling	cause collar pulling	Crane
Crane requires maintenance	cause start of crane service	
Converter is to be stopped	cause interruption of converter operation	

TABLE 4.1.1 CRITICAL CONTROL FUNCTIONS OF A HUMAN OPERATOR IN THE RUNNING OF THE SMELTER

4.2 CONTROL OF THE SIMULATION MODEL

The reasons for choosing a hybrid computer to perform the simulation of the smelter system were given on Section 2.4 and the decision to use the GASP discrete event simulation language is presented in Section 2.5.

In this section an overview of the hybrid simulation approach to represent the smelter model is given. While some parts of the implementation will be discussed, details of the simulation control appear in Appendix A.

As all simulation languages GASP (P-1) deals with a few defined quantities that allow a simple representation of a real system.

Events are considered to be the operations and/or processes taking place in the system at discrete intervals of time, since simulation is performed at discrete steps and not continuously on a digital computing machine. At these intervals the state of the system will change. Characteristics of an event are given by attributes associated with it. (i.e. the duration, height and comfort of a flight are considered attributes of a flight, if these characteristics were important to a customer in deciding to fly on a certain airline.)

An entity represents an object or person in a system which acts or is acted upon during a simulation. An entity is also characterised by attributes. The following examples show how a simulation is performed by moving a system from one state to another.

A ladle arrives (event) at a certain time (attribute of the event) at the converter aisle (attribute of the event). If a crane (entity) is available (attribute of crane) and a converter (entity) requires material (attribute of converter) then the transfer (event) will be performed immediately (attribute of the event).

This example could be continued to perform all activities in the converter aisle and would represent a purely discrete simulation. Since the crane submodels are analog, continuous models, certain modifications are necessary that will be described later.

Events can have any number of different attributes, only limited by programming requirements. Events are contained in sets or files; the latter name will be used here and will refer to the area in the computer memory where a group of events are stored. The number of files are determined by the system and in each file at least one attribute is to be of the same kind, usually the first.

There is always one time file, the master file which contains all "time events" ordered chronologically. The controlling programme GASP removes and executes time events sequentially while updating the simulation time. When an event is removed from the time-file the event code characterises the type of event it represents and execution is performed by branching to a respective subroutine or section of the programme.

After initialisation at least one time event has to exist in the time file. It will be removed by the controlling programme and executed. The execution of events can create further events which are filed in the time file. These are endogenous events in contrast to exogenous events such as commands from the model operator which will be discussed in the next section. As long as there are events in the time file the simulation continues until any preset condition (number of events, elapsed time, etc.)

for termination is reached.

In this work the basic structure of the event filing system of the GASP - simulation language was used. It required modifications to produce an integrated mixed variable simulation language capable of handling both discrete event-type as well as purely analog models.

On a hybrid computer there are three modes of operation of the analog computer, INITIAL CONDITION, HOLD and OPERATE. The OPERATE-mode refers to the state of the analog computer when differential and logical equations are being solved. The HOLD-mode refers to a state where all analog values are frozen. The IC-mode is selected to set up circuits before proceeding to the OPERATE mode for execution.

The modes of operation can be selected from the digital computer which provides a means of synchronising digital and analog simulation models on the hybrid computer. While discrete event simulation is not a form of real time simulation, the analog computer solves equations at a uniform rate given by the selected time scale. This rate determines the rate of simulation. To ensure that discrete events are not executed consecutively while instantly updating the simulation time,a timing circuit was developed. It is shown in Appendix A (Fig.A-3) where the detailed operation is described.

The link between the timing circuit and the digital simulation control program is the output of comparator C39. Whenever a time event is executed the one comparator input is set to the scaled time interval to the next time event, TREL. This occurs in the HOLD-mode when other analog models might be given new input or target conditions. When placed in the OPERATE mode the integrator A35 will operate until the time to the next event has elapsed when the comparator C39 changes its logic state which will be sensed by the digital control program. During any OPERATE period the control programme continuously samples logic outputs of the timing circuit and other analog circuits. The simulation is then frozen by selecting the HOLD mode and the event is executed after the scaled time interval. The next time event is then selected, the timing circuit having been reset and this process repeats itself.

As mentioned, during the OPERATE mode analog circuits are solved while some of their logic outputs are continuously tested by the simulation control programme. This is the case if any target (e.g. a crane reaching and waiting at a location as described in the next section) of an analog model is reached. Again the simulation is frozen by selecting the HOLD mode to reset the particular model or schedule an endogenous event. If an endogenous event is scheduled to occur before the previous next event the target of the timing circuit is reset. This condition can also arise when an exogenous event, issued from an outside source is to occur before the previous next time event. If the HOLD mode is not caused by the timing circuit itself the intermediate simulation time is provided by the previous event time to which the rescaled value of the timing integrator C35 is added. It can be obtained by reading the output of an analog to digital converter.

While more detail can be found in Appendix A,Fig.4.2.2 gives a simplified representation of the process of discrete-continuous event simulation on a hybrid computer.

It allows for the parallel operation of several analog models which means that the simulation control program not only executes the time events but also triggers the start and registers the endpoints of analog models. Because these other models can create new time events (as can be done by manual input) at times other than during the execution of time events, it is necessary to test for new events and reset the timing circuit wheneversuch a condition arises. This is done by two subroutines as explained in Appendix A, where the controlling program GASP is also flowcharted and explained in more detail.

The need for interactive control of the simulation model was pointed out in the previous section. An input instruction will either cause a model to change immediately or else it can be an event type instruction which will be placed in a file to be executed at a later time.

To use the example given earlier, the arrival of a ladle filled with material will alert the operator by a light signal. He can then direct the ladle to a destination that he selects by issuing an appropriate command to a crane. In this way human decisions control certain areas of the simulation, in the same way as a foreman controls processes and operations in the smelter.

To get the controlling program to accept and interpret an input from the typewriter a sense switch isalsocontinuously sampled by the program in the OPERATE mode. When pressing the switch, the HOLD-mode is selected and a command translation subprogram accepts and processes any typed input. This is explained in more detail in Appendix A and was also omitted for simplication on Figure 4.2.2. Fig. 4.2.3 attempts to show the information flow in the simulation model. Because of the two operating modes HOLD and OPERATE it is difficult to relate it to Figure 4.2.2. It might be helpful to remember that any execution in the digital section of the program (including inputs to analog models) occurs when the analog section is in the HOLD-mode. During the OPERATE mode the only activity in the digital section is that of sampling logic signals of analog models and of the sense switch. (This is not shown on Fig. 4.2.3).

The information flow does therefore not represent one path through different stations, at any time, but it rather shows several paths as they can occur at different times during simulation. The following examples relating to Fig. 4.2.3 might help to gain an understanding of the operation of the model:

1. At the start of simulation a programme section is executed which initialises all analog and as digital models, reads data cards and as well as initialising simulation variables. Control is then turned to SIMULATION CONTROL programme and simulation as shown in Fig. 4.2.2 commences.
2. If the operator presses a sense switch (No.1) the statement ENTER appears on the typewriter. He can respond by entering a command (Format is given in Appendix A) whfch could be one of three types. If it is not filed in one of the files an event is executed directly which might affect any digital or analog model. (A process could be triggered to start in which case light signals might be reset i.e. information flow all the way to the display unit).
3. During any simulation run various statistics are continually created and updated for all processes of the model. These are discussed in the next sections and can be requested at any time on a single basis or as a summary of the simulation status of the entire model. A REPORT GENERATING programme produces this typeout which can be seen on Figure 6.3.4 in Chapter 6 where results of the model are presented.
4. By entering ## simulation continues and control goes to the SIMULATION CONTROL section. Events are removed from files and executed. In

the case of the crane models this involves setting inputs and targets of analog models which will cause physical crane movements that can be observed on the display unit during the OPERATE mode.

5. By pressing another sense switch (No.7) the area marked as AUTOMATIC CONTROL is activated. Its operation is explained in Chapter 7 and included here only to show that it fulfills the function of the human operator by generating internally whatever inputs might otherwise have been provided to the model.

Figure 4.2.4 shows the implementation of the simulation model on the EAI690 Hybrid System at the National Research Council of Canada.

The core requirement of about 50K memory locations for the entire model, of which about 2K are variables in DIMENSION or COMMON statements, was well in excess of the 16K provided by the EAI640 digital system. A primary overlay system was used to execute the program of the model which was split into twelve submodules. It utilises a magnetic disc for storing sections of the program not required in core at any time during execution. This overlay system was developed by Dr. U.P. Graefe. The event structure lent itself well to split up the programme of the model in sections. No cross-reference is necessary between the submodels and the constraint that overlay modules are to be independent of each other could thus be met.

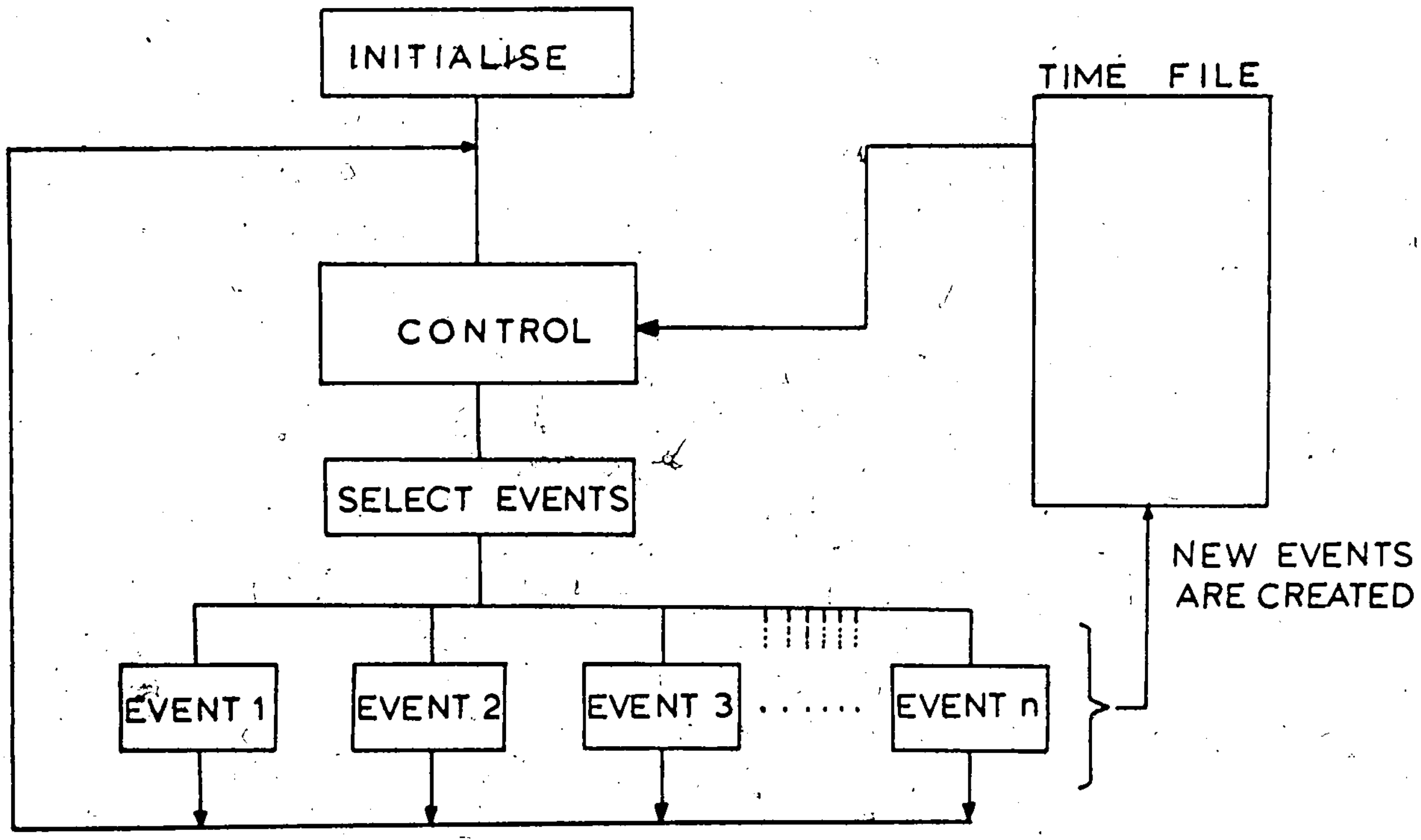


FIGURE 4.2.1 DISCRETE EVENT SIMULATION

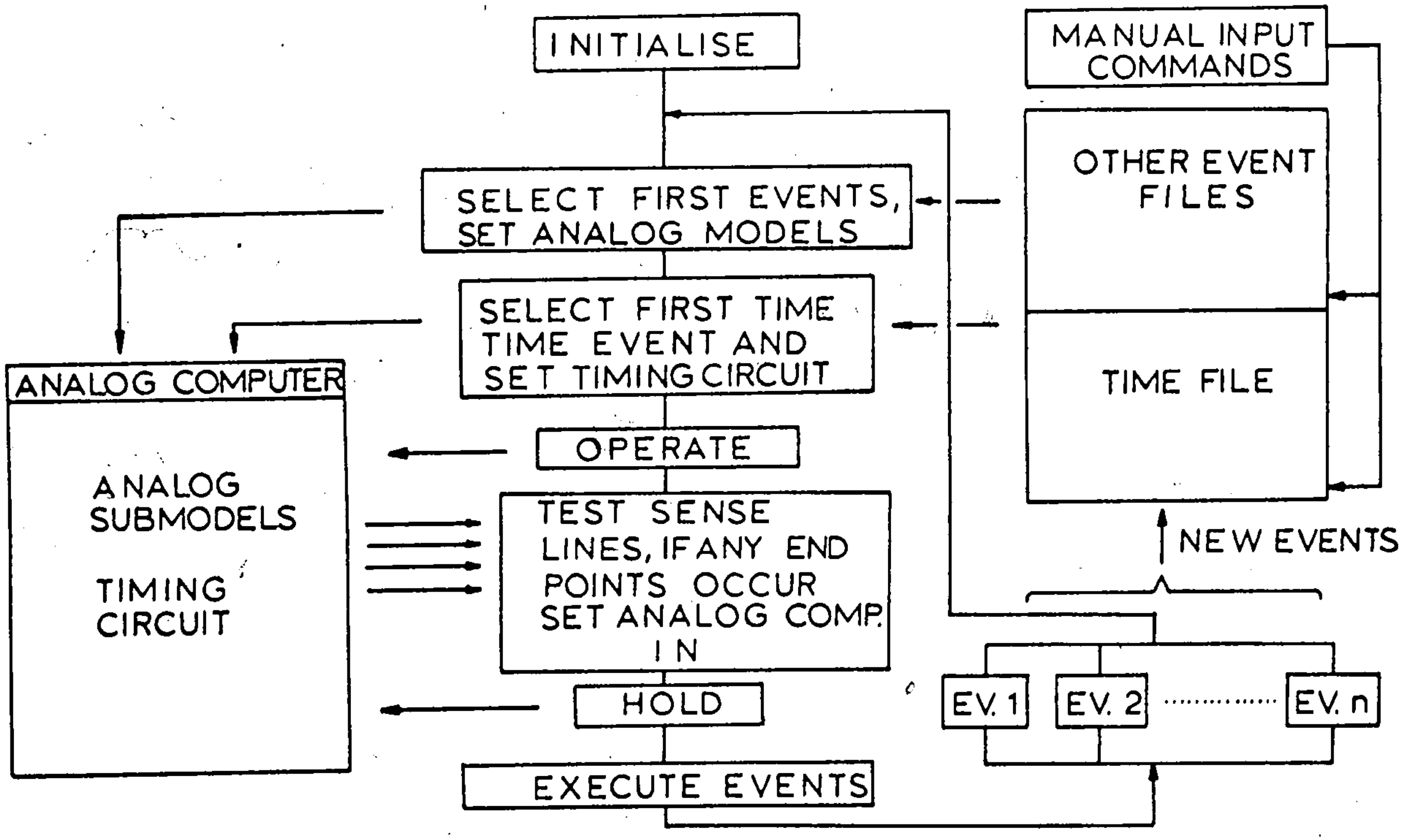


FIGURE 4.2.2 MIXED VARIABLE PROCES SIMULATION

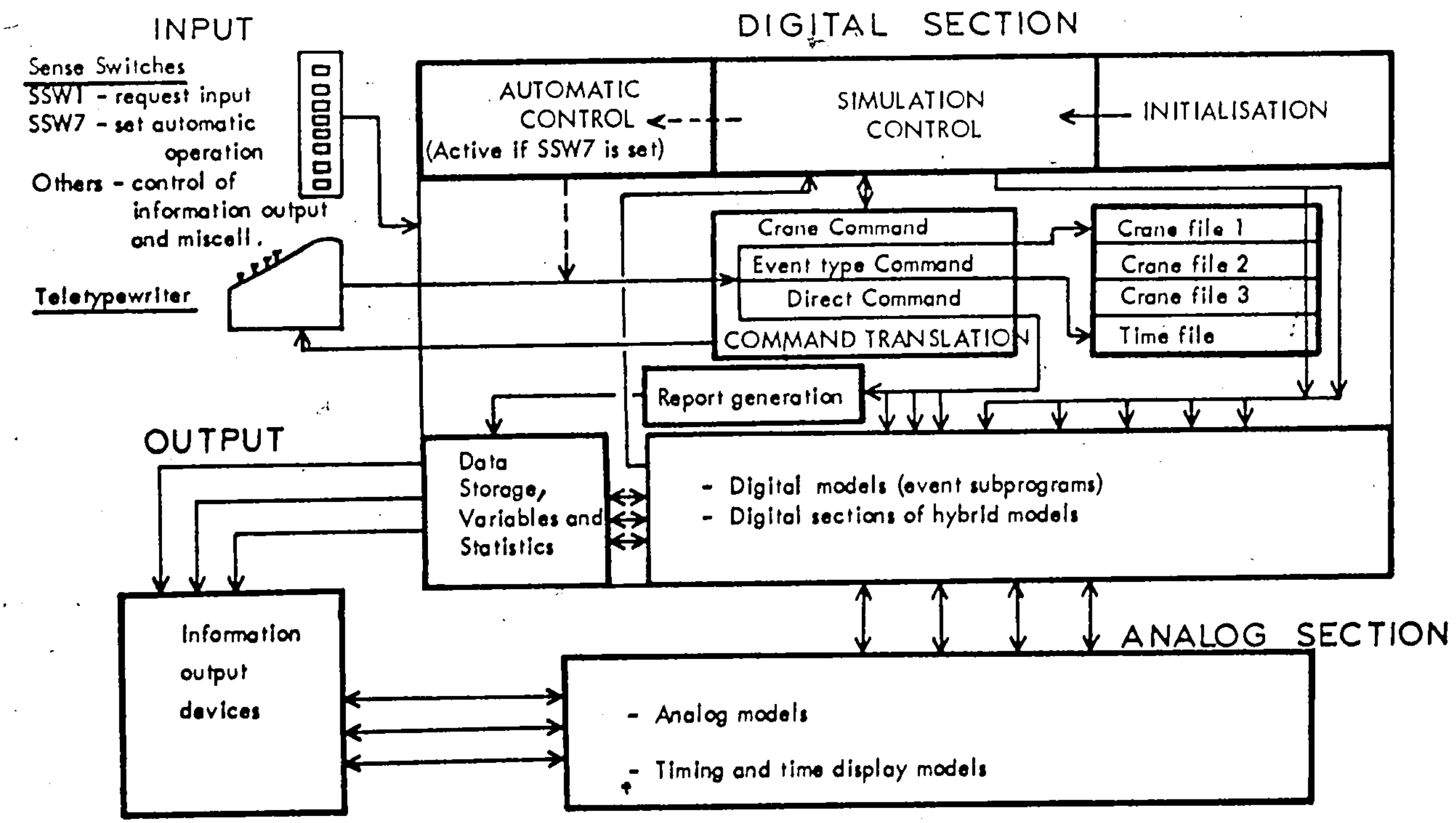

INPUT
Sense Switches
SSW1 - request input
SSW7 - set automatic operation
Others - control of information output and miscell.
Teletypewriter
DIGITAL SECTION
AUTOMATIC CONTROL (Active if SSW7 is set)
SIMULATION CONTROL
INITIALISATION
Crane Command
Event type Command
Direct Command
COMMAND TRANSLATION
Crane file 1
Crane file 2
Crane file 3
Time file
Report generation
OUTPUT
Data Storage, Variables and Statistics
- Digital models (event subprograms)
- Digital sections of hybrid models
Information output devices
ANALOG SECTION
- Analog models
- Timing and time display models

FIGURE 4.2.3 Simplified Information Flow in the Simulation Model

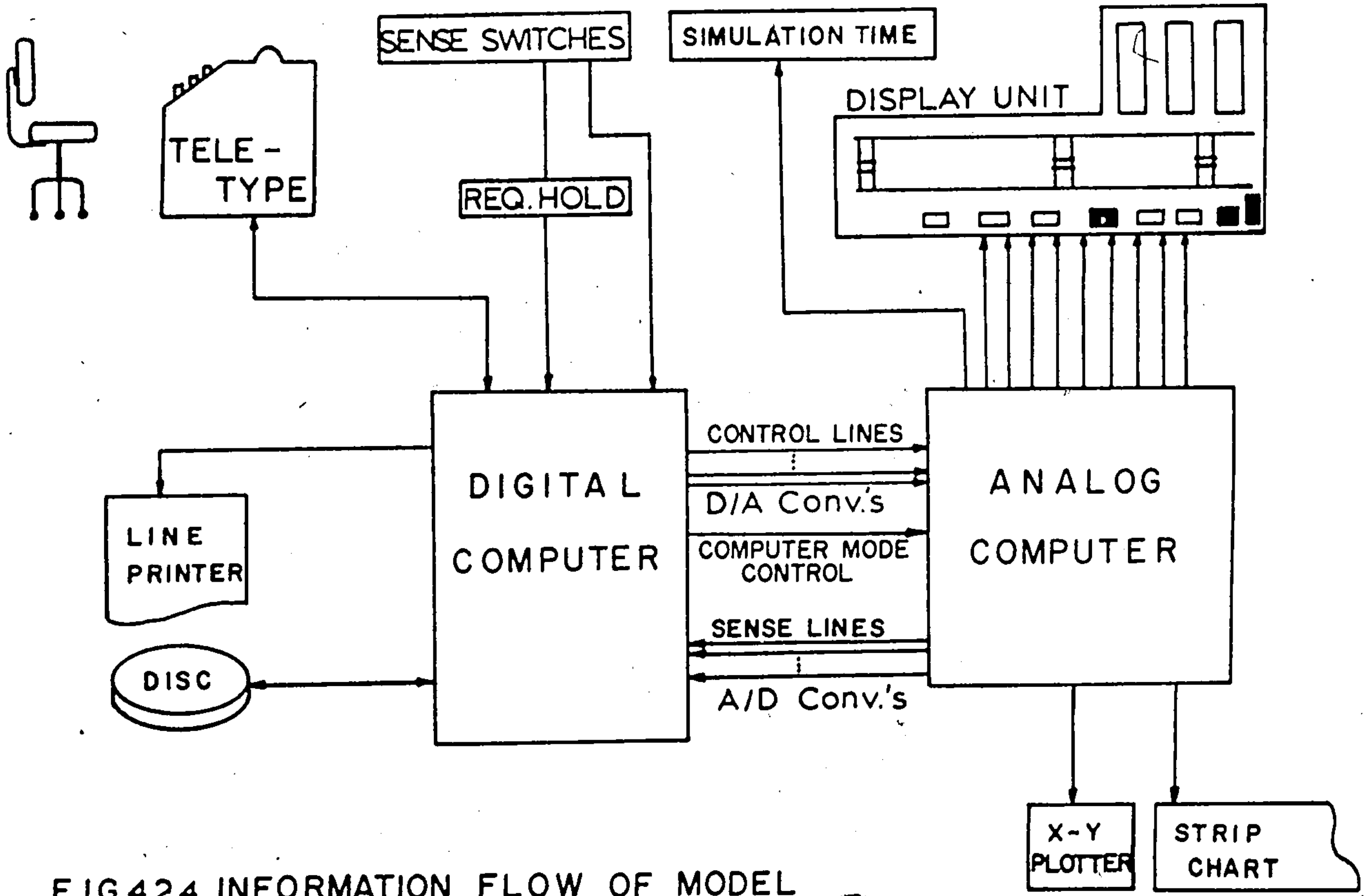


FIG.4.2.4 INFORMATION FLOW OF MODEL

4.3 CRANE SUBMODELS

The main tasks performed by any one of the three overhead cranes are those of transporting ladles or other items from one point to another. Less frequently cranes are being used to "pull a collar" from a converter furnace, when by means of a special pulling hook the built-up material around the converter mouth is removed. Other tasks are the bumping of a ladle against a block to break the solidified crust inside a ladle called skull, or a floor cleaning operation by means of a shovel.

All crane operations can be reduced to two simple tasks, one of moving from one destination to another, and one of being occupied for a certain length of time at a fixed position.

Furthermore each crane executes its tasks independently as long as its occupation at any time does not conflict with that of another crane. This conflict is possible since all three cranes use the same tracks and cannot pass each other. If such a conflict occurs, i.e. if one is to move south and the other north at the same time, then preference is given to the task of higher importance or urgency. It is also possible that a sequence of tasks is performed in conjunction by two cranes. Regular maintenance times are scheduled weekly during which time the cranes are inoperative. This does not eliminate unexpected breakdowns of varying length which still occur at irregular intervals.

In many digital simulation models of material handling systems the activity of cranes has to be accounted for. The following studies (L-2, A-1, D-2, T-1) included overhead cranes as submodels of larger discrete event simulation models.

In general the state of cranes is described by several characteristics such as:

availability	x_1
position	x_2
duration of present task	x_3
priority of present task	x_4
	$\vdots$
nth description of state	x_n

If these characteristics are combined they represent the state of the crane model:

state of model $= f(x_1, x_2, x_3 \ldots x_n)$

During the execution of a crane event the state will change according to the task that is performed, the nature of the state change being dependent on the states of all other cranes and processes.

For each problem the equations governing the transformation of state are different depending on the assumptions and rules according to which a sequence of events are to occur.

In this study the following state variables are used to describe a crane activity uniquely at any time. The characteristics are represented by variables, which are shown in Appendix B

present position

previous position

present waiting time

previous waiting time

priority of present task

priority of previous task

rank of last event in crane file

rank of present crane task

event triggered by crane

quantity of material on crane

operational state of crane

availability of crane

state of crane activity (delay)

state of task sequencing (waiting)

condition of crane (maintenance)

In the actual system a crane operator receives the order to perform a sequence of tasks which he will then execute in that order, at the end of which he will have received further instructions or else be idle. The functions to be performed by a crane model are shown on Figure 4.3.1. The model consists of a logical part representing the crane operator and a mechanical part representing the crane which executes the physical task. Below follows a description of the implementation of the crane models which represents the actual process very accurately. Crane commands are issued by the model operator and are placed sequentially in one of three crane files according to a ranking number. Execution of tasks by the crane models occurs according to the first-in-first out rule and tasks are executed continuously until no tasks remain in the respective crane file when an busy indicator light on the display unit goes off. As long as no breakdown occurs an operational light signal will light up.

Each crane model consists of a digital and an analog section which executes tasks by causing corresponding changes in other submodels. (i.e. addition of material removal of material, etc.) The circuit diagram of the analog and flowcharts of the digital models are given in Appendix B.

The controlling digital program GASP exercises control over all crane tasks.

The simulation is in the HOLD mode while a task is removed from the file and the respective digital crane-submodel routine is called.

In this section of the crane model the variables that make up a crane command are used to start the execution of the corresponding task. Interferences, priorities, duration of the task etc. are checked to establish whether in fact the required task can be executed immediately. If this is not the case the task is delayed while the crane remains idle. If execution of the task is possible the analog model is set up accordingly, that is the position of the target, direction of travel, speed of travel and the waiting time at the destination are set on the analog components by suitable calling programs. If another crane causes an interference and has to be moved out of the way, then the analog and digital models of the interfering crane are set to respond accordingly. Thereafter the controlling program returns the analog computer to the OPERATE mode and the crane will move to its destination and perform its task by waiting for the duration of time required for the particular task.

Time studies for all typical operations provided average times for these tasks. A sense line will be reset at the end of the waiting period at the destination and the computer is returned to the HOLD mode.

To complete the crane task it might in some cases be necessary to cause further action at that time, i.e. cause another event to be executed. An example where no such action is required would be the moving of an empty ladle from one point to another. If however a full ladle was transported from one furnace and poured into a converter this task would trigger the event "material was filled into converter," which will update the state of the converter. The code for this event is checked and if non-zero the respective event is executed by the controlling program. This completes one typical crane task and the next task will thereafter be executed in a similar fashion.

The active tasks of moving and waiting at a position of the cranes are thus represented by analog models. Since these can operate in parallel the execution of crane activities is very close to the real operation. During the non-operating periods when submodels are updated or new tasks selected the simulation time is frozen and does affect the crane models.

To allow for the inoperative periods in the crane models, statistics of breakdown frequencies and repair times are used. The digital models generate these periods by random sampling from frequency distributions. If a breakdown occurs the "operational" light indicator will be turned off and the crane accepts and executes no tasks until a repair period has elapsed. The same applies if the preventative maintenance is performed.

To prevent the bumping of the two faster cranes into the slower crane #2 a delay mechanism is included in the analog models of the faster cranes that will prevent this from happening. In Chapter 7 it will be shown how a series of synchronised tasks can be performed by any two cranes. This feature allows one crane to operate in conjunction with another, rather than independently.

By issuing several commands to the cranes it is thus possible to direct material flow and thus the operations in the smelter in any desired fashion. Tasks are executed in parallel by the cranes and their movements and operational states can be observed on the model display unit.

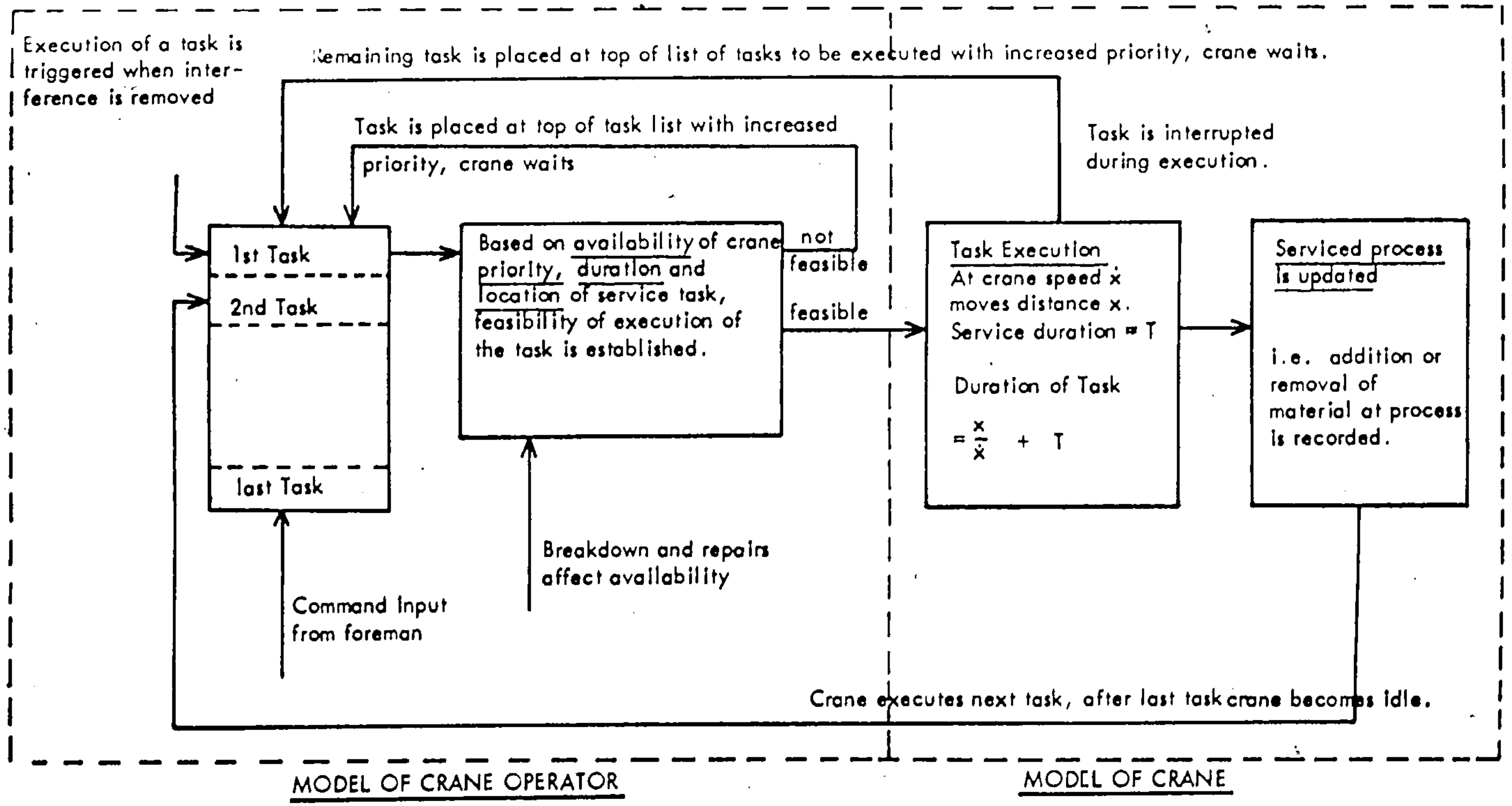

Execution of a task is triggered when interference is removed
Remaining task is placed at top of list of tasks to be executed with increased priority, crane waits.
Task is placed at top of task list with increased priority, crane waits
Task is interrupted during execution.
1st Task
2nd Task
last Task
Based on availability of crane priority, duration and location of service task, feasibility of execution of the task is established.
not feasible
feasible
Task Execution
At crane speed ẋ moves distance x.
Service duration = T
Duration of Task
= x/ẋ + T
Serviced process is updated
i.e. addition or removal of material at process is recorded.
Command Input from foreman
Breakdown and repairs affect availability
Crane executes next task, after last task crane becomes idle.
MODEL OF CRANE OPERATOR
MODEL OF CRANE

FIGURE 4.3.1 Schematic Representation of Crane Model

4.4. REVERBERATORY FURNACE SUBMODELS

A schematic representation of the reverberatory furnace is given in Figure 3.1.1. In Chapter 3 the process of dry and wet reverberatory smelting was described briefly. Anderson (A-4, A-5) describes the operating practice at Noranda in detail and shows how the capacity was increased over the years. Although production figures to day are different, the basic operations have remained the same. (e.g. change from firing with pulverised coal to oil.) While no theoretical analysis is given these two papers present a good background on reverberatory furnace operation indicating the effects many variables have on the furnace operation. McKerrow (M-3) describes a similar smelter operation at Gaspe.

Harris (H-1) has developed a mathematical model for the heat transfer mechanism in a reverberatory furnace. The model is based on an existing furnace but is idealised by simplifying assumptions. It was used to investigate the smelting capabilities of a particular reverberatory furnace and several recommendations were made for improvements. The model didnot however account for metallurgical aspects of smelting which are expected to be affected by varying the smelting conditions.

While the work described by Harris provides insight into the thermal response of a furnace, the model does not represent the entire smelting operation in the reverberatory furnace. In particular the problems encountered in controlling a side charged furnace by temperature measurements are difficult. Pyrometer measurements taken at different points along the furnace read the bank temperature which fluctuates during charging and is not a representative quantity by itself.

Allen (A-6) et al. present a preliminary report on the development of empirical mathematical models expressing aspects of reverberatory furnace operation. The aim of the study was to determine all variables affecting the performance of the furnace. It was hoped to improve the operation by eliminating fluctuations of those variables that affect the optimum smelting conditions. In contrast to the mechanistic model of Harris, Allen et al. produced a multiple regression model based on data of 86 operating days. Because several variables proved to be dependent a non-linear model was used to predict the smelting rate. (The equation for the smelting rate alone fills 12 typed pages.) Predictions of this model were close to the actual performance of the furnace, but simpler, linear models failed to produce satisfactory results.

Until a final report on this work provides evidence to the contrary it appears that accurate reverberatory furnace models have to be based on the large number of interdependent variables involved. Such a model would be very impractical for day-to day operation because of the effort involved in measuring all variables frequently and accurately. Furthermore their evaluation necessitates a digital computing machine. Such a model would probably be useful only for a particular furnace while any other furnace might behave quite differently.

In this study it was attempted to limit control variables only to those that are available on a regular basis at the smelter. Since inputs and outputs of materials are the only quantities that directly affect the operation of the converter aisle, it was possible to develop a relatively simple empirical model based on material balances.

There are two important aspects of the reverberatory furnace that have to be represented by the model. Firstly, matte of a given grade of copper is supplied by the furnace. Since this is the primary source of material to the converter aisle it has an important influence on the overall performance of the smelter and had to be modelled as accurately as possible. Secondly the reverberatory furnaces receive slag from the converter aisle. This is a more flexible but necessary process in the smelter. Figure 4.4.1 shows the model of the reverberatory which supplies matte and produces slag. The rate equations shown are developed in Appendix C.

The reverberatory furnaces models are implemented on a purely digital basis. Since both the rate equations for slag and matte production are uniform and since the availability of matte to the converters is given by a stochastic variable, there was no need for an analog model. All three furnaces are treated in a similar fashion. A number of discrete events were programmed that simulate the behaviour of the model if they are executed in a specific sequence.

Typically the event of an arrival of an empty matte ladle will trigger an event which indicates when the ladle will have been filled again. If any event that requires an operator response occurs there will be a feedback to the operator informing him of the occurrence of such an event. This is either done by a light signal on the display board or by a typed message on the teletype-writer or the line printer. The meaning of the light indicators is given on Figure 4.1.3. Light (4) indicates that matte is available, and when picked up light (5) will come on while light (4) is turned off. Light (5) is turned off only when a crane is instructed to deposit an empty ladle at the matte tunnel. If no converter was ready to receive matte, the ladle can be placed on the floor when light (6) is turned on. Lights (1) to (3) give information on the material levels in each furnace.

When no operator or other action is necessary during a time span it is possible to bring a sequence of minor events together into one larger event. This event will cover a longer period of elapsed simulation time and economises on space and time in the programme. A small example will explain this:

1. The matte car starts moving to the tapping hole with an empty ladle
2. The matte car arrives at the tapping hole
3. Matte tapping starts
4. Matte tapping ends
5. The matte car starts moving to the converter aisle
6. The matte car arrives at the converter aisle with a filled ladle.

These six events can be combined into one event, namely event (6) which is scheduled at the time that event (1) is just about to occur. It would not be possible to combine these events if it is considered necessary for example to specify the time that matte tapping starts.

The availability of matte and slag in the furnace are the two variables that determine the input and output of the model. The availability of material is represented by the level instead of a volume measurement to conform with the actual readings in the smelter. These readings are taken at intervals during smelter operation and provide a measurement for controlling the smelting rate.

It would go too far to point out all the implications involved here, but it should be mentioned that these readings are difficult to take simply because the separation of the slag and matte layers is not distinct. Furthermore the bath is surrounded partly by the charge banks causing the size of the bath to vary. Estimation of the total quantity of molten material in the furnace is therefore subject to some error.

All assumptions that were made together with a detailed description of all events and variables used to model the reverberatory furnaces are given in Appendix C.

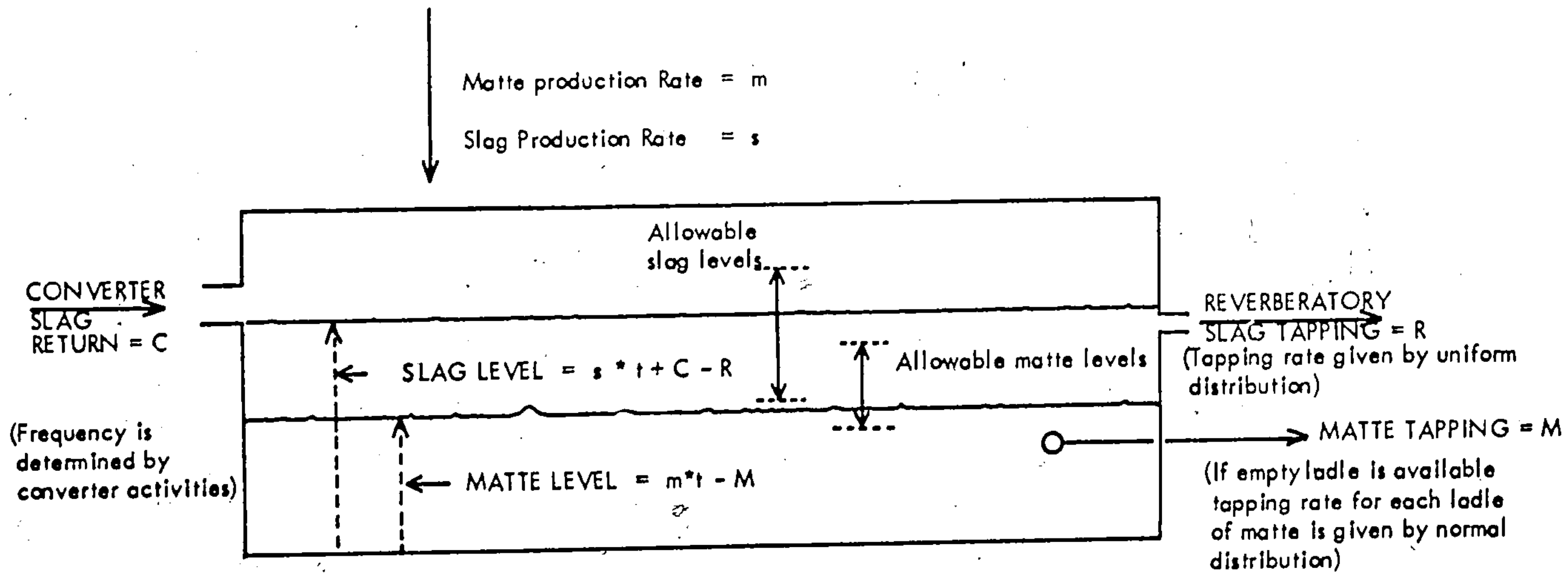

Matte production Rate = m
Slag Production Rate = s
Allowable slag levels
CONVERTER SLAG RETURN = C
(Frequency is determined by converter activities)
SLAG LEVEL = s * t + C - R
Allowable matte levels
REVERBERATORY SLAG TAPPING = R
(Tapping rate given by uniform distribution)
MATTE TAPPING = M
(If empty ladle is available tapping rate for each ladle of matte is given by normal distribution)
MATTE LEVEL = m*t - M

FIGURE 4.4.1 Schematic Representation of the Reverberatory Furnace Model

4.5 CONVERTER FURNACE SUBMODELS

The converting process was discussed in Section 3.1 where its functions of converting matte (Cu_2S, FeS) to white metal (Cu_2S) by oxidising the iron and sulphur was described. The latter product escapes as SO_2 gas while the iron is removed in the slag.

Mathematical modelling of the converter process is reported by Dudgeon, Nenonen et al. (D-3), Niemi (N-3) and Nenonen (N-2). In each case models based on heat and material balances are presented. Krause et al. (K-2) have proposed a theoretical model to control the converter process by an on-line computer but describe its implementation as economically not feasible. Foreman (F-2) has reported the implementation of a similar control study based on a mathematical model of the converter processes, which did however not prove to be reliable enough for continuous on-line control because of process measurement problems.

Converter models can be based on heat and material balances to represent the chemical process of converting. For the purpose of this study the converter submodels should represent the chemical process of converting in terms of reaction rates and material inputs and outputs. This means that the model should respond to controllable variables such as matte additions, slag removals and provide end points of blowing periods based on heat and material balances. The functions of the converter model are shown on Figure 4.5.1. The reaction rate equations shown are developed in Appendix D.

The converter submodel used in the smelter model is a mixed variable model having analog as well as digital content. It was developed by L.K. Nenonen of the National Research Council.

Only a brief description of the implementation will be given here, a detailed description of all functions being given in Appendix D.

There are five converters each being represented by the same model. There are thus five identical analog models and a number of digital events representing the behaviour of converters. At any time only four of the five converters are operational, the other being down for repairs to the brick lining.

Converters are represented on the display unit by six light indicators for each of the five furnaces. The functions of the indicators are given by Figure 4.1.3.

Based on the information relayed by the indicators the model operator can control the converting process whenever his action is required.

Referring to Figure 4.1.3 light indicator (1) henceforth referred to as light, signifies that the converter is in a matte stage if the light is on. The model allows a certain number of matte ladles to be added to the converter at any time during one of eight possible matte blows. Light (4) is on as long as more matte can be added. Once the limit is reached light (4) is off while (1) stays on. In addition to the light colours of white, green and red, there are information tags as shown which remind the operator of the indicator meaning. At the beginning of the first blowing period a minimum quantity of matte has to be added to the converter before it can start the process of converting which is termed "blowing", when air is blown through the metal bath. Once the critical limit of matte is reached the analog model representing the process dynamics of the converter will start the converting process, and light (3) indicating the end of a blowing period is turned off. At the end of a matte blow the slag that was produced has to be removed and returned to the reverberatory furnaces by a crane. This state is indicated by light (6) at which time the converting process stops, indicated by light (3). To remove the top layer of slag from the metal bath, to skim slag, an empty ladle is placed below the furnace which turns and pours material into the ladle. If such a ladle is required light (5) is on.

At the end of a matte stage the model operator will instruct a crane to pick up a ladle of slag, pour it into a reverberatory furnace and return the empty ladle to the converter. The same command is repeated until all the slag is removed which will be indicated by light (6) being turned off while light (4) comes on, i.e. more matte can be added to proceed with the next matte blow.

When discussing the crane models it was pointed out that at the end of the waiting time at the target the respective command is executed to update the model affected by the operation.

In this particular case the task of skimming a ladle of matte will trigger the event "matte has been skimmed from converter No. 1".

In executing this converter event the quantity of slag removed will be deducted from the variable representing slag and any other operation regarding the converter model will be performed then. Similar events exist for every other addition or removal of material.

After adding one ladle of matte to the converter in any blow other than the first, light (3) is turned off and the converter blows until a given fraction of the available iron in the bath is slagged off. Matte can, but does not have to be added until light (4) is turned off. As are all other submodels the converter models are protected from incorrect commands such as material additions beyond the allowable limit. Commands are illegal if the type of command is not compatible with one of 15 possible converter states and is simply ignored.

Whenever the copper content in a converter reaches a given minimum quantity after a material addition, a message on the typewriter gives the operator the option to end the particular last matte blow (the going-high blow during which a higher bath temperature is obtained) and proceed with a copper blow. Material removal from a converter is allowable at the end of any matte blow after skimming slag. The converter receiving the transfer-matte can be in the same state or blowing in a matte stage.

Slag can be skimmed during the going high blow which is indicated by light (6) without reaching the end of a matte stage which is indicated by light (3). Skimming of slag, once the end point is reached, is performed as in other matte blows except that the last ladle of slag is poured in another converter in the matte blow. This is because of the high percentage of copper in the slag.

Following this, lights(6) and (1) will be turned off and lights (2) and (4) turned on. At this point the model requires the addition of flue dust before the end point light (3) is turned off when blowing in the copper stage commences. During the copper blow the converter model requests addition of cold copper lumps (pigs), indicated by light (4). At the end of a copper blow (light (3) on), a skull has to be added (light (4) on) before the blister copper produced can be removed (light (6) on). After the copper is removed and transferred to an anode furnace or poured in the pig bay a new converter cycle starts.

Other than the materials already mentioned copper scrap and reactor slag can be added when available to converters in the matte stage.

The response and demands of the converter models thus correspond very closely to the actual converter operation while the duration of blowing periods are provided by the process dynamics in the analog converter model. After addition or removal of material the analog model is reset to process the given quantity in the furnace at a rate determined by the rate of air flow through the converter bath.

Strip chart recorders are used to show the quantity of FeS in the furnace during the matte stage and CU_2S during the copper stage. Examples of these will be shown in Chapter 6.

End points of blowing periods are relayed to the digital computer model in the same way as the end points of crane tasks namely by the change of the logic signal of a comparator.

Digital models provide converter breakdown events and indicate the need for collar pulling after a given quantity of material has been blown.

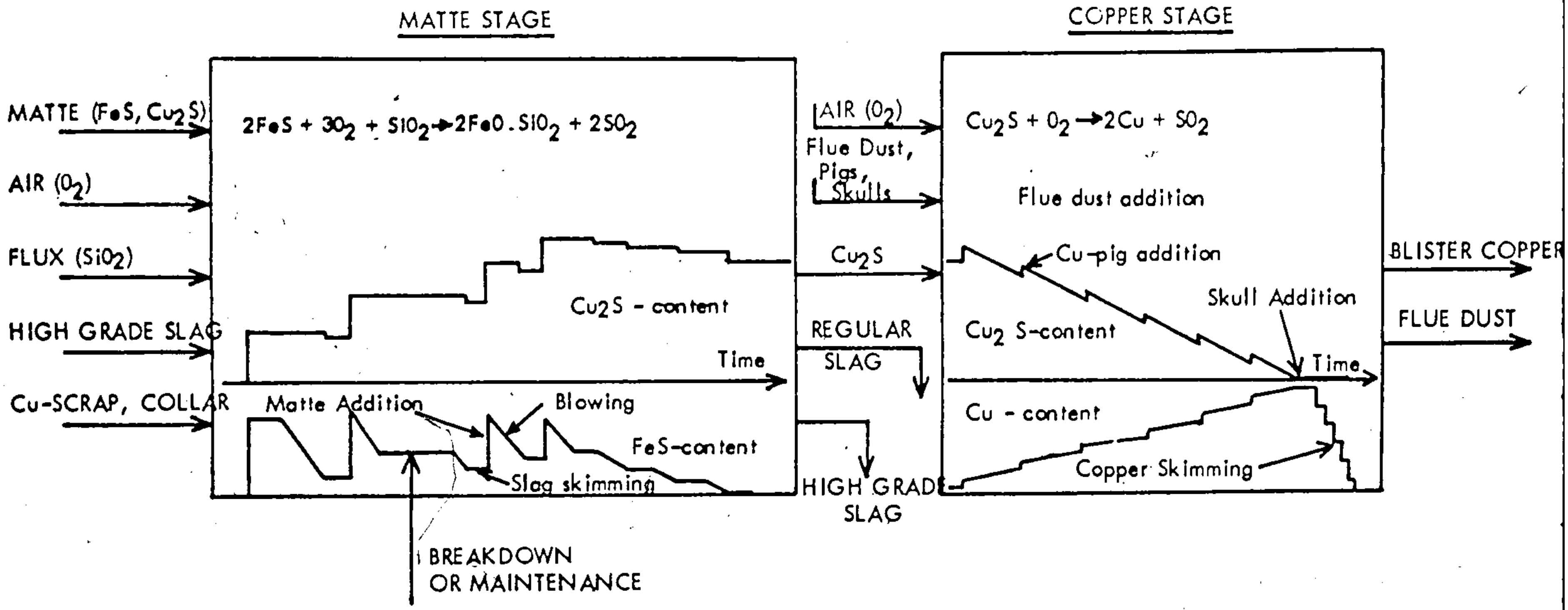


FIGURE 4.5.1 Converter Model showing Batch Processing of Matte

4.6 ANODE FURNACE SUBMODELS

The final process of refining blister copper in the anode furnace is one of oxidation or reduction. This is accomplished by blowing either air or natural gas through the anode furnace, which is a large cylindrical vessel, holding 15 or more ladles of blister copper. There are three anode furnaces, one of which (#4) employs the method of poling when refining. Batches of blister copper from converters are poured into the furnace until it is full or near full. A refining period is preceded by skimming of slag from the furnace. The decision to start refining before the furnace is filled can be made, which is usually only done if a long delay is expected before a further batch of copper becomes available. Refined copper flows out of the back of the furnace and is cast into anode plates on one of two anode wheels. Some time after the end of casting the furnace is again ready to accept copper. A short description of the implementation follows.

There are three light indicators on the display unit for each anode furnace; the state each one represents is given by Figure 4.1.3.

The anode furnace models request addition of copper until they are filled. After every addition respective events update the quantity of copper. No action will take place once a furnace is filled until a command for removing anode slag and one to start refining is given. These commands can also be issued before the furnace is filled to capacity. The end of the refining process is indicated by the "ready" indicator which will be turned off when a command to start casting is issued. Some time after casting is completed the indicator "copper can be added" lights up again, and a new refining and casting cycle begins. The duration of the refining and casting processes is given by the furnace content and uniformly distributed rates for each process. Again illegal commands occur if commands concerning the anode furnace are not compatible with its state and are ignored. Further details on this submodel can be found in Appendix E.

4.7. MISCELLANEOUS SUBMODELS

Apart from the three main furnaces and the overhead cranes for material handling there are several other units, processes and locations that have to be accounted for in the model. These models are purely digital models and are briefly described below. Further details can be found in Appendix E.

Pilot Plant Submodel

Located at the north end of the converter aisle is a pilot plant for a continuous copper smelting process. It produces so called reactor slag and copper cakes which are lumps of copper cast in molds. Slag is picked up by a crane and poured into a converter in the matte blowing stage. Copper cakes are used for the same purpose as pigs are in the copper stage of converters, for cooling purposes.

The model utilises three light signals to indicate the availability of slag, copper cakes and the need for an empty ladle. Slag and cakes are produced in the model at varying intervals, and when available an appropriate command to a crane will allow the removal.

Prototype Plant

This process is shown as a future continuous smelting unit on the floor plan of the smelter, Figure 1.3.1. As the name suggests it is an advanced version of the continuous smelting process and began operating during the later stages of this study. A digital model makes provision for including its operation in the overall model. Its effect was, however, not included in the results described in this thesis.

Pig Bay

Blister copper is poured into moulds in the area marked pig bay to produce lumps of cold copper required for cooling in the converter copper blows.

The model of the pig bay records the number of pigs at any time. If fresh pigs are cast a light indicator shows when these have cooled sufficiently to be removed.

Pig Pile

There is an area at the north end of the aisle where copper pigs are stored, if removed from the pig bay. Pigs are usually only taken from the pig pile, a safety stock, if there are no pigs left in the pig bay. The model of the pig bay records the number of pigs and allows additions and removals.

Bumping Block - Ladles - Skulls

After a ladle has been used several times a solidified layer deposits and is removed by using a crane to bump the ladle against a large block. Before this is done the ladle has to cool for some time. The loose crust is called the "skull". The submodel for the ladles provides a counter for the usage of ladles. A light at the bumping block signals that bumping is required if the usage of ladles reaches a given value.

By a suitable command the operator can cause "bumping". If there was no ladle at the bumping block a signal showing that there is a "ladle to be bumped" would be off. The command for bumping could not be executed in that case. The number of skulls are updated whenever one is added by bumping, or removed by a crane.

There are no models describing scrap or flue dust availability. It is assumed that there is flue dust available when required at the beginning of the copper blow, and scrap can be picked up at the north end when required. The amount of scrap processed in reality depends on its availability which varies and is not significant in the overall model.

5:0 MODEL VALIDATION AND CALIBRATION

5.1 VALIDATION OF SIMULATION MODELS

Although the stage of model validation is treated in a separate section the processes of model building and verification are closely related. Every step during the model building phase was tested and verified against the postulates on which the model was based.

Once the overall model was operating, validation and modifications to improve the model formed an iterative process until a satisfactory performance was obtained. The isolated treatment here merely serves to identify the distinct process of validation.

Van Horn (V-1) stresses the importance of this stage and summarises the reasons for the difficulty of validating simulation models by saying that "simulation offers the most flexible and realistic representation for complex problems of any quantitative technique". He goes on to say: "Thus, many of the aspects that make validation difficult for simulation also give validation a great deal of importance."

Much attention is given to the problem of testing a simulation model in all texts on Modelling and Simulation of Systems or Processes. While no general procedure for validation of all types of models can be given most authors propose a similar approach. It should be stressed that a valid model is not one that duplicates reality but one, that meets specified design requirements.

Wigan (W-1) proposes the following hierarchy of stages involved in developing a valid model:

1. Postulates
2. Fitting
3. Calibration
4. Identification
5. Validation

He points out that as the model is developed each stage can compensate for any errors of the previous stages and consequently the process of validation becomes more difficult.

Identification ensures that no deductions are made from the model that it is not able to reliably produce. In other words the detail of the model should be consistent with the historic data available. This stage is of particular importance in developing regression models.

Naylor et al. (N-5) place the emphasis on the third of three phases in verification of computer simulation experiments, namely on forecasting. The first two stages, the formulation of postulates and testing of the postulates are essential in building a model but do not guarantee valid results. The ability of a model to predict the behaviour of the system under study is considered the ultimate test of a simulation model.

Mihram (M-4, M-5) makes a clear distinction between verification and validation stages and includes both in an iterative process of model development. This was pointed out earlier. Verification ensures the validity of the programmed logic whereas validation determines the correspondence of the real system and its model.

To ensure validity of the smelter model four stages of validation were considered.

1. Verification, or testing of the validity of the model.
2. Calibration of individual submodels.
3. Validation of the model using operating records.
4. Predictions of model compared to response of actual system.

The separate stages of validating the model will be discussed in the next sections.

Before leaving this section mention should be made of the generation of stochastic processes.

Random variables are used extensively in the model and have a significant influence on the validity of the model. Both uniform and normal distributions are used to represent process behaviour. During a simulation run random samples are drawn from the distributions to produce typical values of stochastic variables.

The genaration of pseudo random numbers is treated in most texts on simulation of stochastic processes (M-4,N-3) and will not be discussed in detail. In this study a random number generator tailored to the EAI 640 is used. It is based on a multiplicative congruential relationship and produces a random number in the interval 0.0 to 1.0. The function routine used (I-1) is shown below where 32767 corresponds to the maximum word size of the digital computer. It is guaranteed to produce a sequence of 2^{n-2} numbers without repetition, where n = 16 is the number binary bits in each word on the EAI640 digital computer.

```
FUNCTION DRAND (IY)
IY = IY * 259
```

```
      IF(IY)5,6,6
5       IY = IY + 32767 + 1
6       YFL = IY
      DRAND = YFL/32767
      RETURN
      END
```

The density function of a uniformly distributed variable y in the interval (a, b) is given by

$$f(y) = \frac{1}{b-a} \quad a < y < b \qquad ....(1)$$

$$= 0 \quad \text{otherwise}$$

Values of the variable y are generated by drawing a random number r in the range of 0. to 1. and converting it according to:

$$y = r(b-a) + a \qquad ....(2)$$

This operation is performed in the simulation model by a function routine with two arguments, the upper limit of the distribution B and the lower limit A.

```
      FUNCTION UNFRM (A, B)
      COMMON ISEED
      UNFRM = A+(B-A) * DRAND (ISEED)
      RETURN
      END
```

In a similar way a normally distributed variable can be generated provided the mean and the standard deviation of the distribution are known.

The density function of a normally distributed variable y is given by:

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} \quad e^{-\frac{1}{2}\left[\frac{y-\mu}{\sigma}\right]^2} \qquad \ldots (3)$$

where μ and σ are the mean and the variance of variable y.

A method suggested by Box and Muller (B-2) is used to generate a normal deviate y directly by using two random numbers. This is suitably performed by the following function routine RNORM. Arguments being the mean of a normal distribution and the standard deviation.

```
      FUNCTION RNORM (XMEAN, STDV)
      COMMON ISEED
      RA  =  DRAND (ISEED)
      RB  =  DRAND (ISEED)
      V   =  (-2.0 * ALOG(RA))  ** .5 * COS(6.283 * RB)
      RNORM =  V * STDV
      IF(RNORM + 3. * STDV) 6,6,8
    6 RNORM  =  3. * STDV
    7 RNORM  =  RNORM + XMEAN
      RETURN
    8  IF(RNORM - 3. * STDV)7,7,9
    9  RNORM  =  3. * STDV
      GO TO 7
      END
```

To conserve memory space however, a shorter and faster routine is used which is based on the Central Limit Theorem. A detailed discussion of this method is given by Naylor (N-3). It consists of finding the mean of a sample of twelve uniformly distributed numbers which is then transformed to give a normally distributed value. This is performed by the following function routine:

```
    FUNCTION  RNORM (XMEAN, STDV)
    COMMON  ISEED
    DO 1  I = 1, 12
    S  =  DRAND (ISEED)
1   A= A+S
    RNORM  =  (A-6) * STDV + XMEAN
    RETURN
    END
```

At any time during the simulation a distributed variable is thus obtained simply by calling the respective function subroutine generating a uniform or normal variate and specifying the two characteristics for each distribution.

5.2 VERIFICATION OF THE MODEL

This stage is discussed by some authors under the topic of validation. It includes checking of the programmed logic, i.e. computations performed in different submodels are checked as it is done when debugging a regular computer programme. Debugging of analog models is done conveniently by observing process variables on display facilities during a simulation run rather than analysing results at the end.

Due to the interactive nature of this model, however, a completely verified model could still produce erroneous results which are not due to unsatisfactory calibration.

The operator controls aspects of the simulation by issuing commands to which the model responds. Incorrect commands due to a typing error or a decision error can thus also produce erroneous results if they are not detected and corrected before execution by the model.

To avoid these errors from occurring built in checks ensure that every command entered is legal, e.g. material can only be picked up at a location if it is available and can only be transferred to a furnace if the furnace is ready to receive it. In a similar way a broken down crane will not accept commands from an operator until it is operational again.

These checks at all submodels thus ensure that the logic of the operator, who is part of the simulation, is correct.

During the process of verification various assumptions had to be modified since they prevented the model from operating in a realistic manner.

A good example of this is the logic of resolving conflicting crane operations. Initially certain postulates and assumptions were made as how to represent the operation of the cranes realistically. When running the simulation model it was observed that the crane submodels performed tasks in a different way than observed in reality. Changes in the logic had to be made several times until a satisfactory operation was obtained.

Thus verification included not only checking whether the model performed what it was programmed to do but it also included a verification of the postulates put forward.

The extensive display facilities of the simolation model proved to be very useful at this stage. Most of the processes could be observed during simulation and compared to the behaviour of the system (e.g. crane movements as described above). In addition to the visual display a line printer provides information on any one or all submodels at any stage during a simulation run. This information is obtained by the command INFO followed by a component code. (The type of information provided is discussed in the detailed description of the submodels in the Appendices.) While this information is used frequently during simulation it was used extensively during verification and validation. In addition there are typeouts in most submodels that can be controlled by setting one of eight sense switches on the digital console. The information is thus provided every time a state change occurs in a particular submodel. Appendix F lists the functions of the sense switches. Apart from consulting operating people constantly during the validation period experts from the smelter were invited twice to observe and criticise the logic and accuracy of the model. This aspect is particularly stressed by Maisel and Gnugnoli (M-1). The involvement of operating experts was useful in particular here where the model of the operator was part of the overall simulation model.

5.3 CALIBRATION OF THE MODEL

At this stage the model was verified and assumed correct. This statement does not mean that from here on no changes to the programme logic were made. When calibrating the model, variables were assigned values based on the available data. If it was found that the model did not produce the required results further data was used to improve the model. If the additional data still did not provide a satisfactory model, the model had to be reexamined and improved in an iterative fashion as described before.

However, for the purpose of describing the process of calibration, this aspect will be treated separately, assuming that the model was completely verified.

Operating data from the plant was received in different forms which are listed below:

(1) Daily Reverberatory furnace shift reports

(2) Reverberatory furnace No.3 control reports

(3) Reverberatory furnace log book

(4) Converter shift reports

(5) Individual converter reports

(6) Converter log book

(7) Converter air flow charts

(8) Anode furnace reports

(9) Daily smelter reports

(10) Weekly smelter control reports

In addition to these records three field studies were conducted during which operating data was recorded for all crane activities, for the operation and performance of several shifts and for particular processes such as matte tapping.

Other data such as break down statistics, weights, locations, durations of tasks etc. was supplied by Noranda upon request.

Details on the data required for particular submodels can be found in the Appendices where the individual models are described.

The extensive amount of data available was analysed and fitted to the models. Care had to be taken in analysing statistics of some processes to ensure compatibility with the mechanics of the respective submodel. An example will illustrate this point. Statistics on matte tapping times showed a normal distribution of tapping times except for some values that fell outside the range of all other variables.

These long delays in matte tapping are, however, taken into account already in the model and had to be filtered out of the distribution which was then used in the model to generate intervals for matte tapping. Calibration of the converter models was given most attention and involved variations of input variables such as air flow and oxygen efficiency for each unit while the blowing rates, and slag formation could then be compared against converter operating statistics.

Other models were calibrated in a similar way by comparing output variables of the models against operating data. If no correspondence existed, control variables were changed within allowable limits to produce the required agreement.

In some cases the interaction between processes prevented isolated testing of processes which was then done as part of the overall validation of the model as described in the next section.

5.4. VALIDATION OF OVERALL MODEL PERFORMANCE

While the validity of individual submodels is essential, it does not ensure the validity of the overall model. Interactions occur between processes which cannot be predicted. In addition the decisions of the simulation operator produce results that will not resemble operating data unless the decisions are based on the same policy that is employed in the smelter.

During field studies detailed records of five 8-hour shifts were obtained which included data on all process states crane activities and converter foreman decisions. Being able to copy the latter as closely as possible is an important prerequisite to simulate the actual operating conditions. In fact it became evident that it takes some training to become an efficient model operator.

Before describing the validation runs a brief description of the operating policy of the converter foreman will be given. The smelter is run on an 8-hour shift schedule during which time the converter foreman essentially has control over the running of the smelter.

With four converters in operation at any time the converter foreman charges matte into the converters during distinct matte blowing periods. At the end of each period slag is skimmed which is returned to the reverberatory furnace.

To proceed to a copper blow matte from one converter is transferred to another to give a large enough charge to start on a copper blow. Important in this process is proper timing of transfers to reduce idle times of converters to a minimum while ensuring that matte requirements are uniform. This policy for controlling smelter operations is termed the "matte transfer" policy in contrast to the "no transfer" policy that is discussed in the next chapter.

This scheduling task has to be performed by the model operator during a simulation run to match the actual performance.

To compare operating data with results of simulation runs it is essential to run the model under identical conditions. Initial conditions of processes were calibrated to correspond to that of the field study. However, the decisions of the model operator, if different to those of the converter foreman of a particular shift will not satisfy the requirement of identical conditions.

The detailed data of three shifts obtained in March 1972 was drawn up as an operating sheet and used for the validation runs as a shift plan. In this way the model could be run under identical conditions including the human decisions.

Figure 5.4.1 shows results of one 8-hour simulation run compared to actual operating data based on the matte transfer policy.

Further operating data for operations under the "no matte transfer" policy were obtained in March 1973. While this policy is discussed in Chaptér 6, Figure 5.4.2 is presented here to show the close correspondence of simulation data and operating data as obtained from the different operating policy. On the two Figures material additions and removals and process end points are compared for each converter.

The upper row shows simulation results while the lower row corresponds to the operating data. Material additions mt = matte, tm = transfer matte, fd = flue dust, pg = copper pigs are shown on the lower line while material removals sl = converter slag and cu = blister copper are shown on the upper line. Other material additions thatoccurred (reactor slag, scrap etc.) are not shown on the diagram.

The correspondence of simulated converter operations with data from the real operation was improved to a satisfactory level by adjusting control variables such as blowing rates and service tasks. At this stage further data for matte tapping times was also obtained because of the important influence these distributions have on the availability of matte.

Apart from ensuring correspondence in the time history and material requirements of converters as discussed, data generated by the model was compared with actual data not yet used in calibrating the model. For example it is expected that the slag and matte levels (these are generated by the model) in the reverberatory furnaces change only slightly during an 8-hour shift if an average throughput of matte is obtained. Also the ratio of slag to matte from a converter is generated by the model and was compared with operating records. The amount of matte processed and the quantity of copper produced are, however, the most important measurements of validity since they are the result of all interactions of the system submodels.

While results of the validation runs above were in close agreement with actual data, exact duplication was not expected. The stochastic elements of most models prevent individual events from happening in the same sequence as they were recorded in reality. The matte tapping times for example are obtained from normal distributions in the reverberatory furnace submodels. While in reality a crane could have been free to move the material when it became available, the simulated availability of matte could occur at an earlier or later time when the crane or converter status could have been different. In simulating actual operating conditions, that is following operating decisions in detail this situation could cause an unnecessary delay.

The overall effect of such random variations will, however, not affect the results if long enough operating periods are simulated based on free decisions of the model operator. When simulating several eight-hour shifts the average performance in terms of matte processed and copper produced can be compared. The high variability of both quantities is discussed in detail in the next chapter. Both in reality and in operating the model, the end of an 8-hour shift could come a few minutes before or after a converter is ready to be filled with several ladles of matte which could affect the shift performance by over 10%. However, if the average of several shifts is used this effect is minimised. Two sets of results are compared one for the"transfer policy"in 1972 and one for the"no transfer policy"in 1973. Figure 5.4.3 shows actual matte production for consecutive shifts expressed as a percentage of the average for the month in which operating data was obtained. On the same graph simulated results for matte production during 9 shifts are shown which were based on operating conditions of the same period. It can be seen that while the variation is of the same order the mean of the throughput is slightly lower. Figure 5.4.4 is a similar graph showing the matte throughput when operating under the"no transfer policy". Details of these results are discussed in a different context in Chapter 6. The results are presented here and show that the model is also capable of reproducing operations under the changed policy. The average simulated values here correspond more closely to the actually observed values. The reason for this seems to be the much simpler task of operating the model efficiently under the"no transfer policy". The transfer policy requires more attention and experience to be handled efficiently which is reflected in the lower average shown in Figure 4.3.4.

A vital role was played at this stage by operating personnel. The model was operated according to instructions from operating people who assessed the response of the model. Many comments and queries indicated points of improvement which were implemented where necessary. This process ensured that the model operated to the satisfaction of operating personnel, which ensured their confidence in the model. Furthermore the role of operating people as model operators could be studied to be immitated for further simulation runs.

The fourth test of validity, the ability of the model to correctly predict the response of the system to specified inputs will be discussed in the next section.

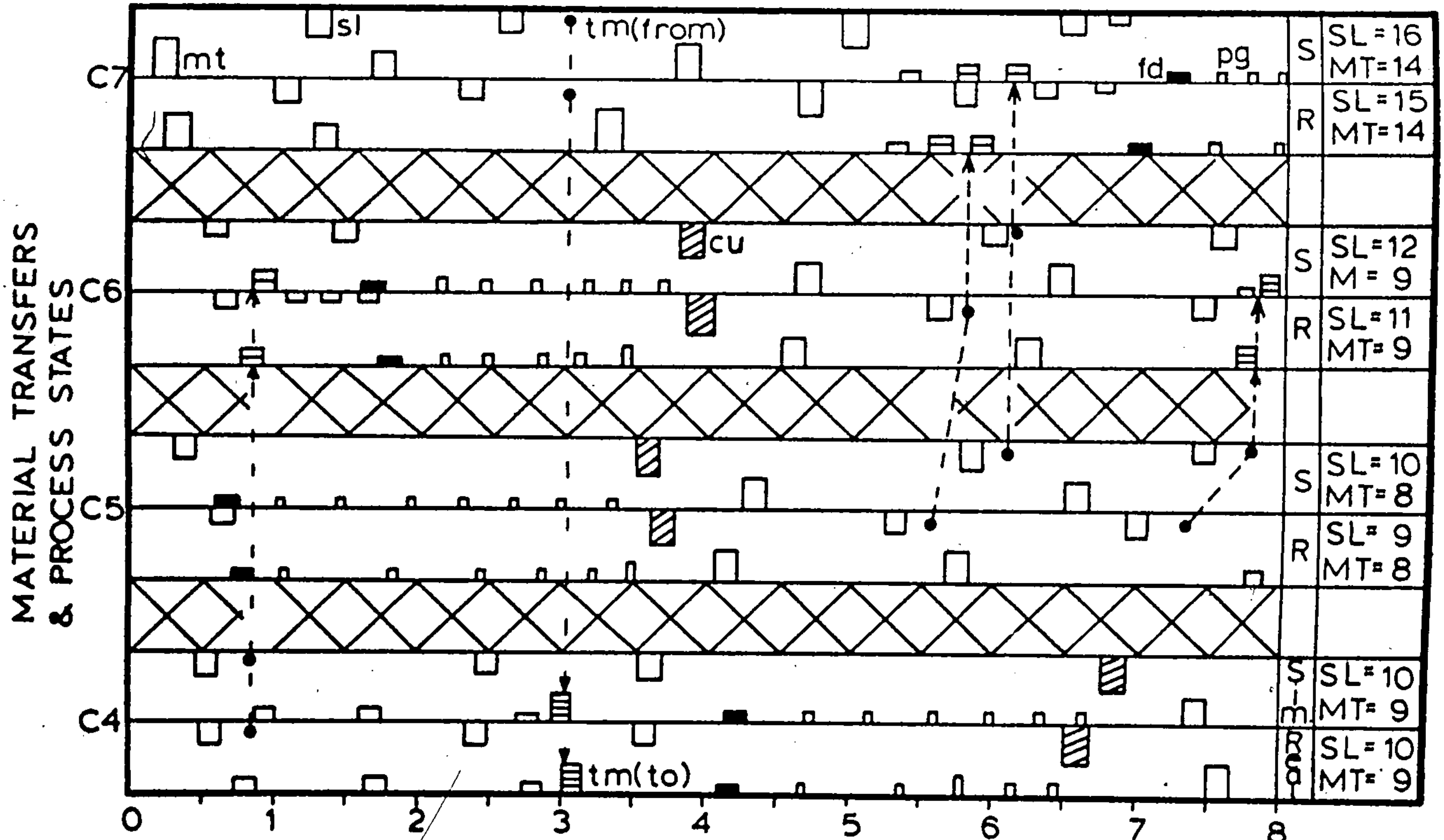


FIGURE 5.4.1 COMPARISON OF REAL AND SIMULATED SHIFT OPERATIONS(TRANSFER)

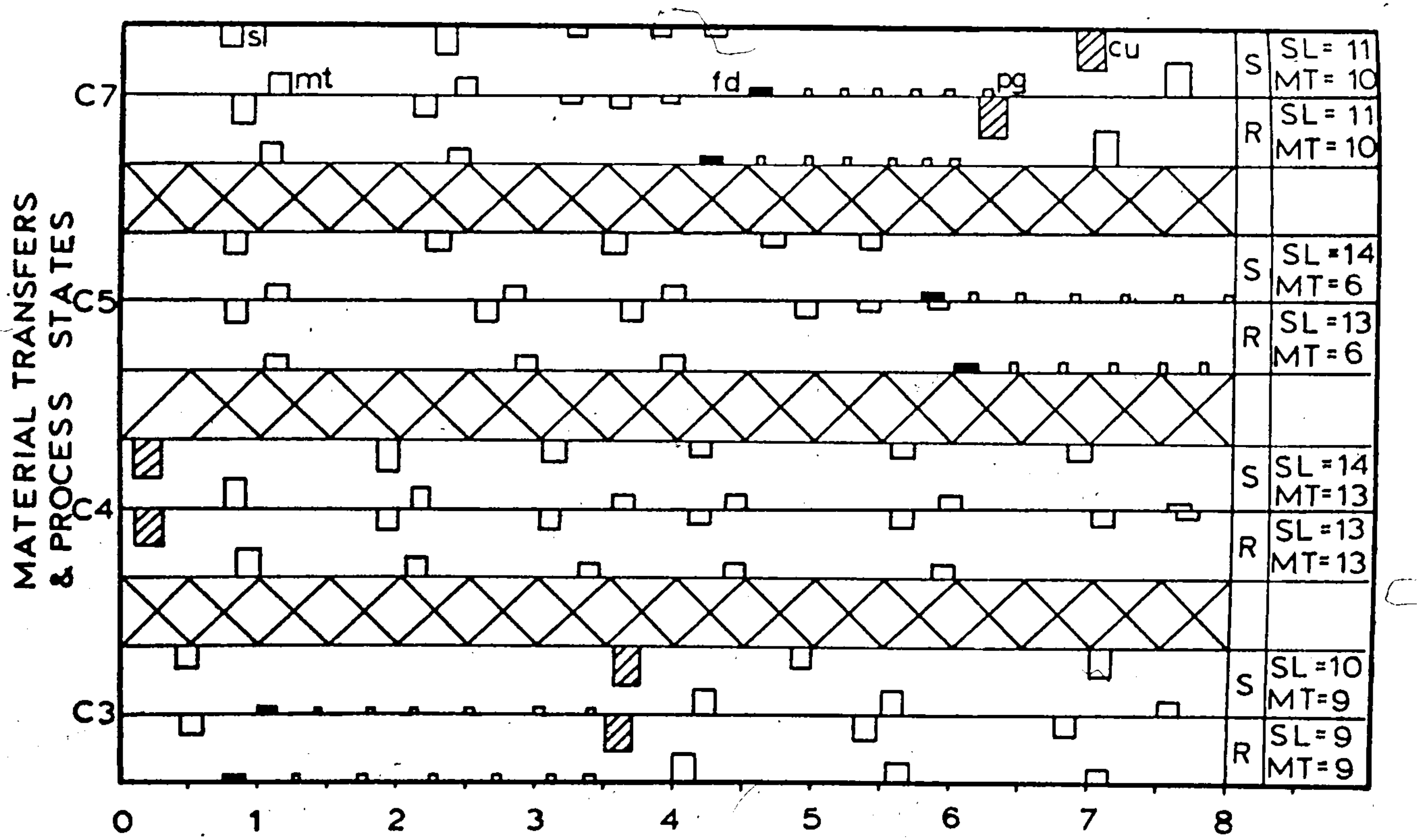


FIGURE 5.4.2 COMPARISON OF REAL AND SIMULATED SHIFT OPERATIONS(NO TRANSFERS)8

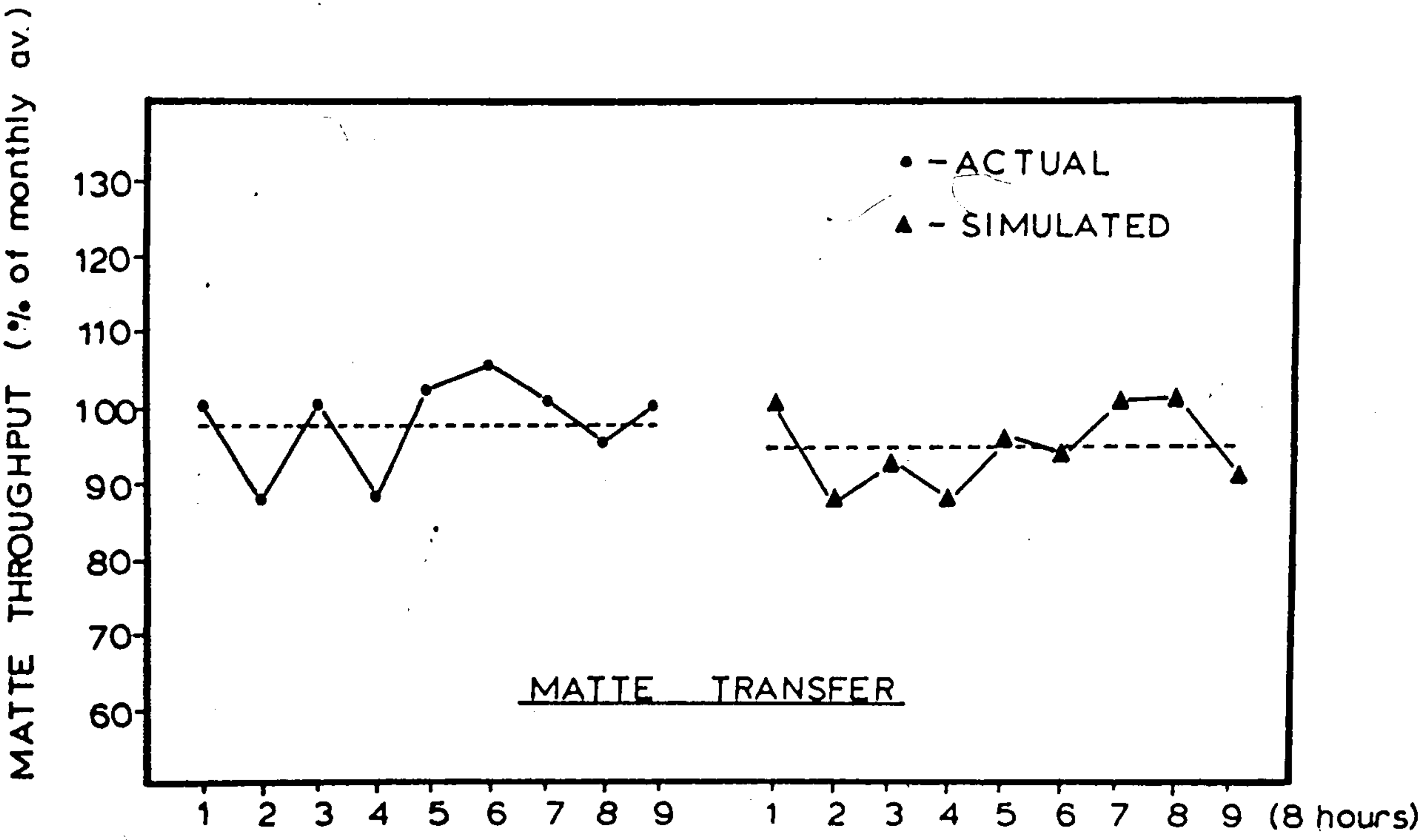


FIGURE 5.4.3 SIMULATED MATTE THROUGHPUT COMPARED TO ACTUAL THROUGHPUT (1972 data)

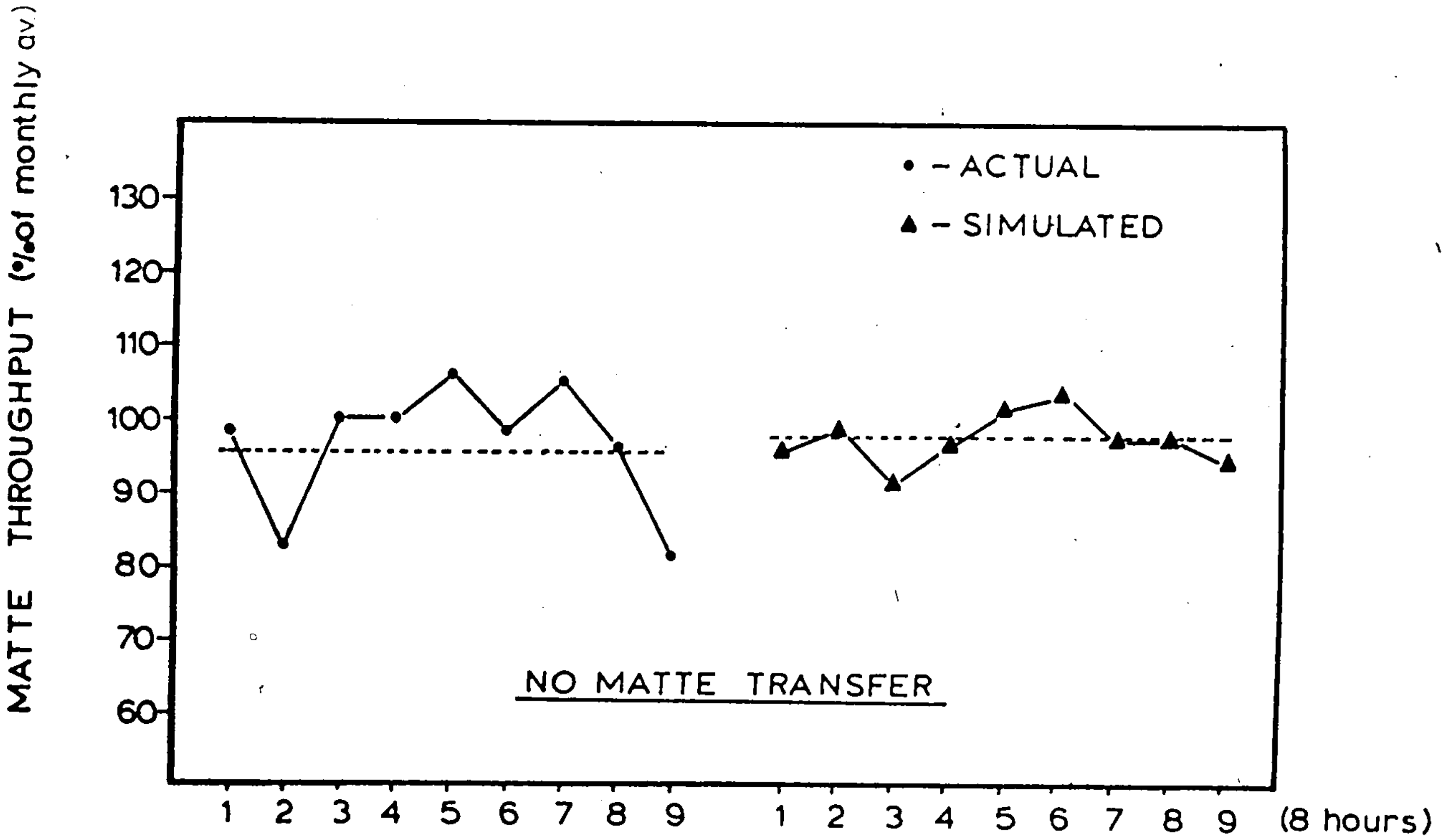


FIGURE 5.4.4 SIMULATED MATTE THROUGHPUT COMPARED TO ACTUAL THROUGHPUT (1973 data)

6.0 MODEL ANALYSIS AND RESULTS BY MANUAL CONTROL

6.1 APPLICATIONS OF THE MODEL

Having established the validity of the model it was analysed to provide information about, and insight into the system it represents.

It was pointed out in Chapter 3 that the mechanism for improving the performance of the system was to be constricted to changes in organising, scheduling and control procedures. In other words the basic processes were fixed. The model could thus be used to analyse the following aspects of the system:

1. Improve converter scheduling
2. Improve crane use
3. Improve plant layout
4. Investigate extension of the plant or capacity increase of some processes.
5. Investigate effect of the startup of a continuous smelting process.
6. Investigate effects of changing the matte grade to increase throughput.
7. Study the use of the model as a training simulator for operating personnel.
8. Study the possibility of developing an on-line smelter control system.

For the purpose of this study the first and the last points were the most interesting to investigate, as they relate directly to the control of the system. Using the policy of "matte transfer" described in Section 5.4 it proved to be difficult to run the model efficiently and it was felt that a "no-matte transfer policy" could provide the same or possibly a better production. This is discussed in Section 6.3.

As it turned out later this policy was implemented for a trial period in the smelter and provided further information on the validity of the model as discussed in the previous chapter.

The investigation of point 8. is discussed in Chapter 7.0. The other points listed were not studied systematically, although some mention will be made of point 2. in this chapter. In Chapter 8 further reference will be made regarding point 7.

6.2 MEASURE OF PERFORMANCE OF THE SYSTEM

To evaluate a simulation run of the model a measure of performance of the system has to be established.

When validating the model it was possible to compare individual system and simulation results directly. However, when trying to improve the performance of the model it is necessary to establish a function that represents the overall profitability of a proposed scheme or policy change.

This expression is generally termed the objective or cost function in optimisation studies. It reflects the profitability of an operation and is expected to increase as the efficiency of a system is increased. The aim in improving the system operation is thus to maximise the objective function. For a stationary linear system it can be represented by:

$$U = \sum_{i=1}^{n} l_i \, x_i \qquad \dots . (1)$$

where x_i = quantity of product or material > 0 for i=1, n

l_i = coefficient of contribution towards the objective

To measure the performance of a dynamic system the objective function can be defined to reflect the performance of the system during a time interval. By recording the performance of the system at equal time intervals it is possible to obtain a measure of performance for the system.

$$\text{i.e.} \quad U(t_j) = \sum_{i=1}^{n} l_i(t_j) \; x_i(t_j) \qquad j = 0,1 \;..\; m \qquad ....(2)$$

where $U(t_j)$ represents the performance of the system in the time interval $(t_j - t_{j-1})$.

In practice such a measure of performance is usually the profit of a system and would include all labour costs, overheads, energy costs, material costs which will affect the profit realised in producing copper at a given price.

The system modelled here does of course not account for all these factors, and an exact cost function for the smelter system with the given boundaries would be difficult to establish.

In the absence of such information the objective function of the system can be represented by:

$$U(t_j) = \sum_{i=1}^{n} l_i(t_j) \, x_i \, (t_j) + C * (t_j - t_j-1) \qquad ....(3)$$

where C is a constant and its effect on the objective function only dependent on the time interval for which $U(t_j)$ is evaluated. If this interval is kept constant then $C \Delta t_j$ includes all fixed costs of the system and other costs which are assumed to be constant over given operating periods.

The quantities contributing to the objective function are all materials entering and leaving the system. All uncontrollable variables are incorporated in $C \Delta t_j$ and are thus assumed to make constant contribution to $U(t)$. Materials that are considered in the model are:

Material leaving:	1.	Anode copper produced
	2.	Reverberatory slag
Materials entering:	3.	Matte
	4.	Flux

5. Heat

6. Air

Materials such as copper scrap, reactor slag and others are assumed to enter the system at a uniform rate and are thus also included in C . Although these materials arrive in discrete batches at irregular times this effect can be approximated by an average uniform input over longer operating periods. Furthermore the quantities involved are small so that no significant error is introduced by this assumption.

The amount of flux, heat and air put into the system and the amount of reverberatory slag removed from it is proportional tp the amount of matte of a given grade processed in a given period of time, thus their effect can be accounted for in the coefficient for matte which will be l_2 while that for copper l_1.
This gives:

$$U(t_i) = l_1(t_i)\,x(t_i) - l_2(t_i)\,x(t_i) - C\,\Delta t_i \qquad \ldots.(4)$$

When considering inputs and outputs to the system only the levels of in-process materials are assumed to remain constant with time. The implications of this assumption are discussed below.. For constant time intervals such as an 8-hour shift the term $C\Delta t$ can be eliminated to give a relative performance of the system only dependent on the matte and copper production.

$$R(t_i) = l_1(t_i)\,x_1(t_i) - l_2(t_i)\,x_2(t_i) \qquad \ldots.(5)$$

This equation will, however, produce a highly variable performance index for mainly four reasons. Copper is produced in one of the three anode furnaces in batches of several hundred tons at intervals of around 12 hours. These discrete batches reflect a copper output that appears to be rather irregular even at daily or longer intervals.

(i) If the copper output is based on the refined anode copper cast, which is the final output from the system, then extremely large fluctuations would appear in the performance.

(ii) Furthermore the storage of copper in the system in form of copper lumps (pigs) causes an apparently lower copper output that will only be removed if long time intervals are considered such as a week or more. (It was assumed above that in-process material levels remain constant).

(iii) In general process cycles require additions and removals of discrete batches of material which lead to fluctuations in the above performance criterion if evaluated at short time intervals.

(iv) There is a considerable time lag between the input and output of material that is not reflected in equation (5)

A simple solution to this problem would be to simulate longer operating periods which was done to obtain reliable figures for average production. Since the simulation of one 8-hour shift takes about one hour on the model, it was still necessary to obtain a reliable measure of performance for that time interval which is a more convenient simulation run than a 24-hour period for a full day.

To obtain a measure of performance for a period of 8 hours the only useful quantity to measure is the amount of matte processed since it does not suffer from the shortcomings discussed above. To eliminate x_1 from equation (4) the relation

$$x_1 = kx_2 + K \qquad \ldots . (6)$$

can be used which merely states that the amount of copper produced is proportional to the amount of matte processed, of a fixed grade, plus a quantity K which consists of Copper Scrap, Reactor copper, etc. This quantity was assumed fixed for a given time period and we thus have:

$$R(t_i) = x_2(t_i) \left[k l_1(t_i) - l_2(t_i) \right] + K \, l_1(t_i) \qquad \ldots .(7)$$

or $$R'(t_i) = x_2(t_i) \; L_2 \qquad \ldots .(8)$$

Equation (8) thus states that the relative performance of the system is given by the amount of matte processed during a given time interval provided the matte grade (% content of copper) remains constant.

This result will seem obvious and does in fact reflect one of the measures of performance in the smelter, namely the amount of matte processed per 8-hour shift or per day. The amount of copper produced is also quoted, but it becomes only meaningful if considered as an average over a longer period of time such as a day or a week.

Over the longer operating periods variations in the matte grade can be more significant making the measure of matte throughput less useful unless it is combined with the corresponding copper production.

Although the matte grade can change considerably even during an 8-hour interval this is rather uncommon and can safely be assumed constant.

The analysis thus shows the assumptions that are made when using the matte production as a performance index. Even with the limitations it provided a good measure for predicting the effect of an improved operating policy in the smelter.

Considering all the assumptions the result of the analysis above might seem of little value. Because of the need for a measure of performance which can be used to compare simulated and real results the above measure was chosen rather than an index that provides a good measure of performance for the model but is not directly applicable to the system.

6.3. MODEL RESPONSE TO "NO-MATTE TRANSFER" POLICY

The effect of operating the smelter without transferring white metal between converters was of particular interest and was the first area to be investigated.

It was pointed out that the transfers were made mainly to be able to schedule converters properly, such that at any time two converters are on a matte stage, the third being on a copper stage while the fourth is getting ready to go on a copper stage. In the "no transfer policy" matte is added to converters during the consecutive blowing periods, while slag is skimmed at the end of each. When a converter contains enough material it proceeds with a copper blow.

After simulating several shifts a good policy for controlling operations was developed and it appeared that the matte throughput could be increased. Figure 6.3.1 shows the matte throughput as obtained by the two different methods. The simulation runs were based on the data of March 1972. An increase of the average matte throughput is indicated in the order of 6-7%. The variation of matte throughput is of the same order as found in reality and it was realised that the significance of the result as given by a student"s t-Test would be low even if more simulation runs were performed. The reasons for this were discussed in the previous section.

The willingness by the smelter management to give this policy a trial run in March 1973 provided an opportunity not only to prove that a predicted increase in production could be achieved, but also served to further validate the model. The ability of the model to accurately respond to different operating conditions had yet to be proven, which represents the fourth test for validity of the model.

The results of the field test are given in the next section.

Based on data obtained in March 1973, several shifts were again simulated according to the two different policies. The results are presented in Figure 6.3.2 and show a similar difference in the throughput of matte which is of the order of 4-5%.

Figure 6.3.3 gives a comparison of process statistics for two 24-hour simulation runs with identical initial conditions based on the two policies. As expected the usage of converters for refining has increased while the non-blowing times decreased slightly. This results in the increased throughput if sufficient material is available and if crane service is not a restricting factor.

Figures 6.3.4 and 6.3.5 show the full simulation reports at the beginning and the end of a no-matte transfer simulation run. The meaning of the variables of each process that appear on these reports is given in Appendices B-E where individual submodels are discussed. Figure 6.3.6 gives the converter blowing states as recorded on stripchart recorders. Again two simulation runs for 24-hours are presented for each of the operating policies.

The stripchart records show the amount of FeS and Cu_2S in the matte and copper blows respectively. In a matte stage several batches of matte are represented by the quantity of FeS which diminishes with time as a proportional amount of slag is produced. The slag skimming periods correspond to a zero FeS level which is followed by another batch of matte addition. (The increments correspond to ladle additions.) The start of refining in the copper stage coincides with the flue dust addition and ends when all of the Cu_2S has been oxidised to copper. After a skull is added copper is skimmed to complete a converter cycle. The spacing of converters is shown. The no matte transfer policy did not produce any irregular effect in fact it was pointed out that a uniform operation was very easy to maintain.

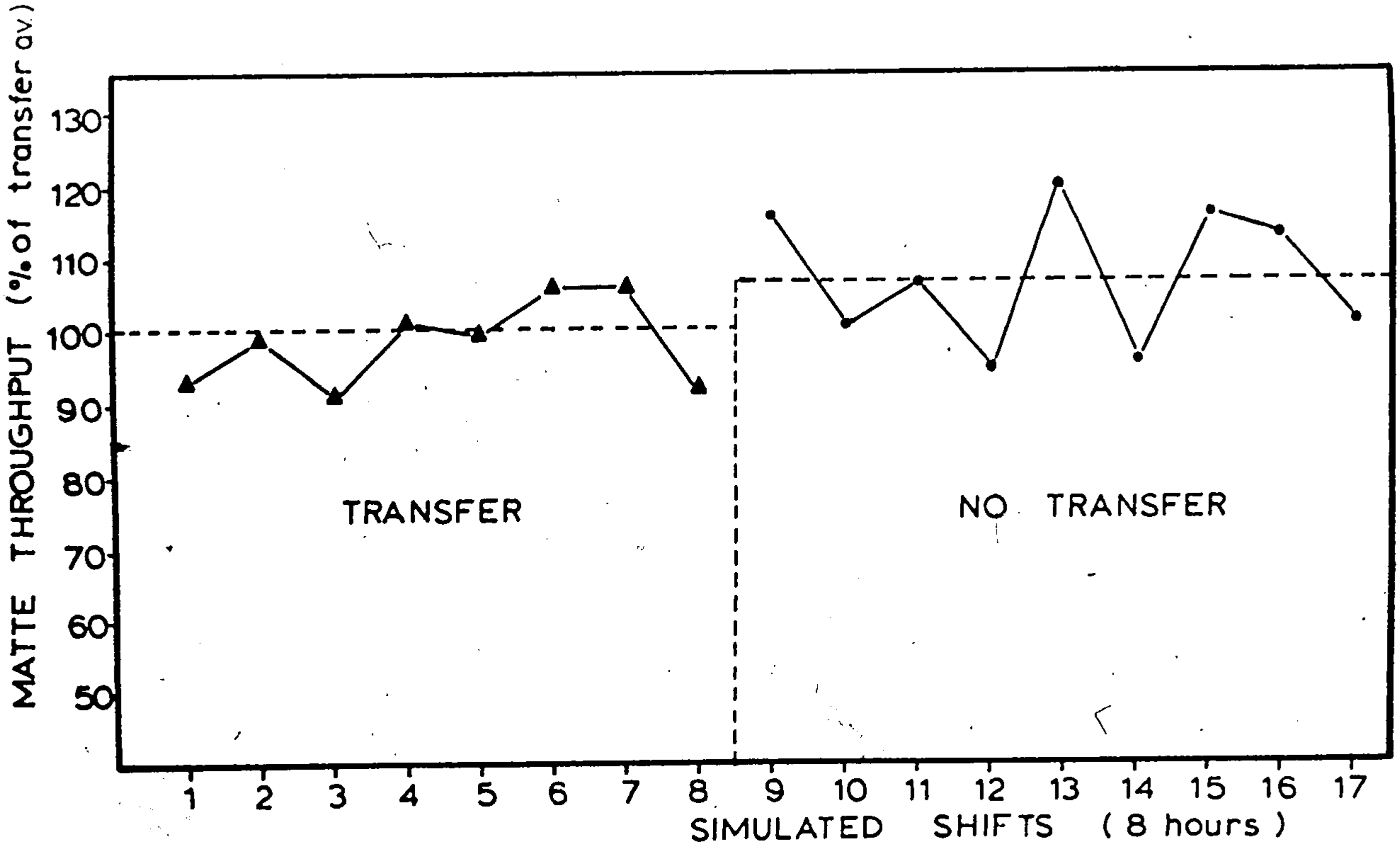


FIGURE. 6.3.1 Matte Throughput for two Policies (1972 data)

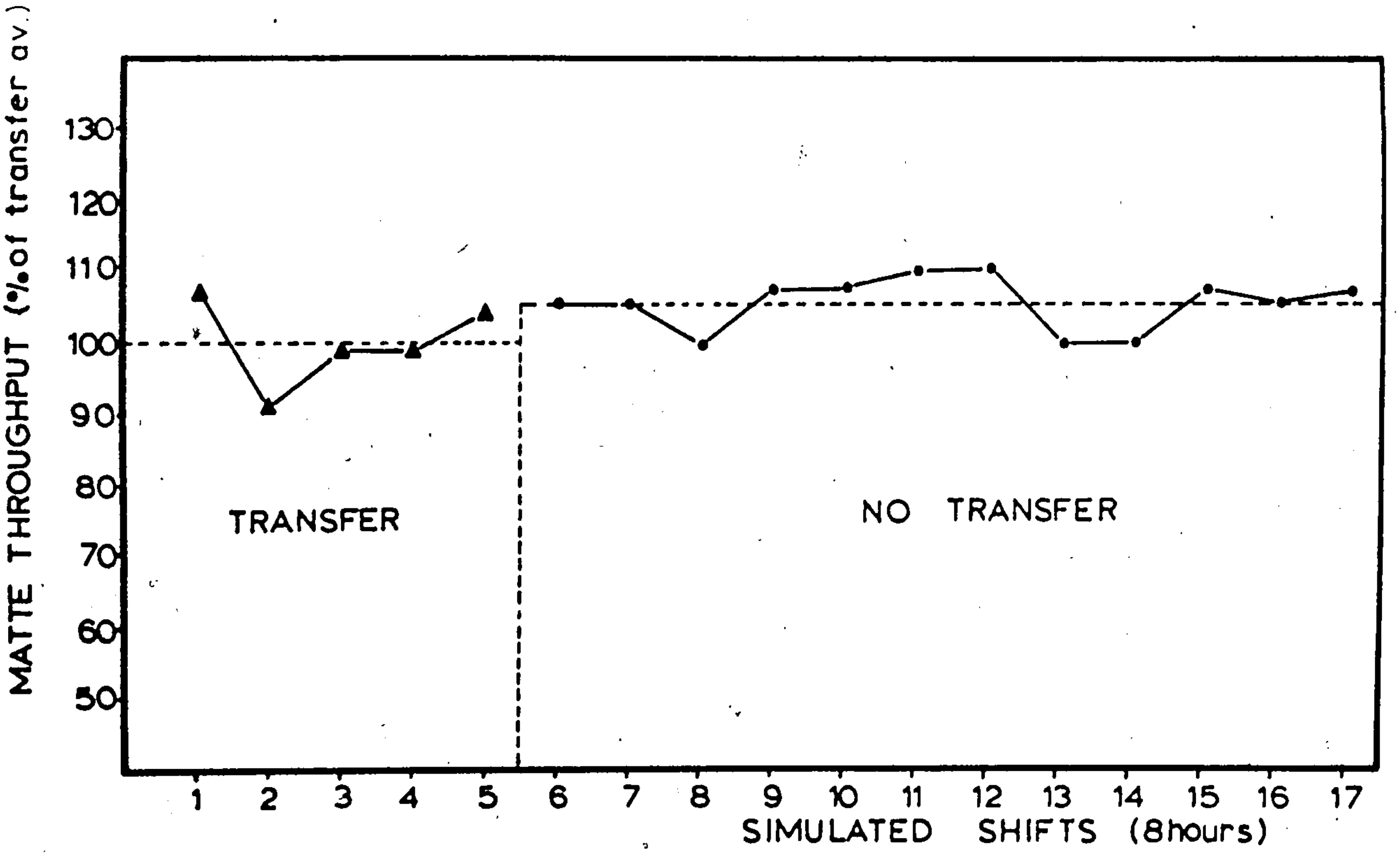
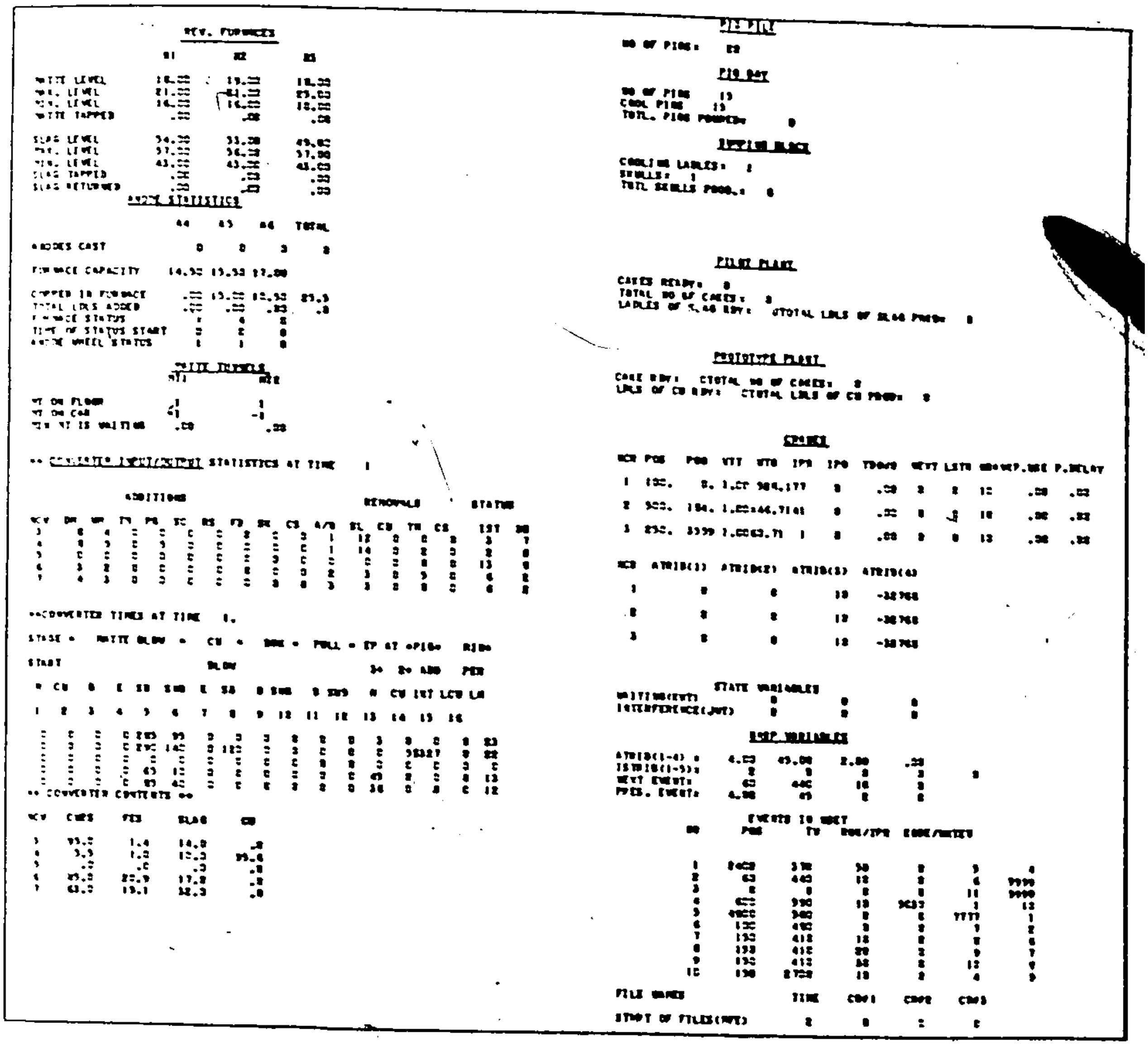


FIGURE 6.3.2 Matte Throughput for two Policies (1973data)

Average statistics on smelter operations	Transfer Policy	No Transfer Policy
Percentage of time blowing in matte stage (converters)	65%	69.5%
Percentage of time not blowing in matte stage (converters)	35%	30.5%
Matte throughput	-	2% higher
Copper throughput	-	2% higher
Percentage of time cranes are busy	61%	63%
Percentage of time cranes are delayed	8.25%	9.15%
Percentage of time a ladle of matte was waiting on a matte car	31.8%	29.3%

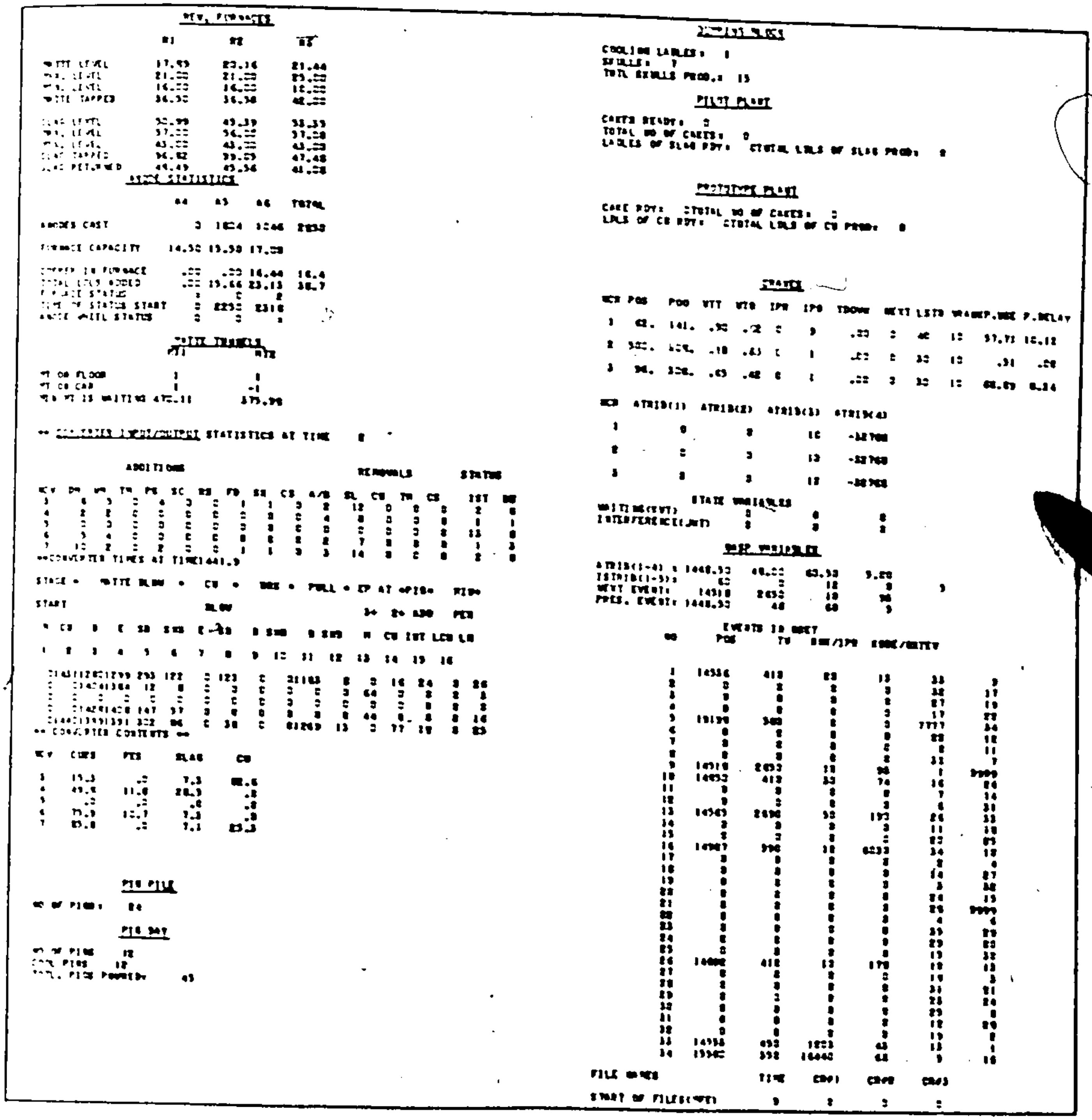
FIGURE 6.3.3 COMPARISON OF PARTICULAR 24-HOUR SIMULATION RUNS FOR MATTE TRANSFER AND NO TRANSFER POLICIES



```
                 REV. FURNACES
                 R1        R2        R3

WHITE LEVEL      16.00     19.00     16.00
MAX. LEVEL       21.00     21.00     25.00
MIN. LEVEL       16.00     16.00     16.00
WHITE TAPPED       .00       .00       .00

SLAG LEVEL       54.00     33.00     49.00
MAX. LEVEL       57.00     56.00     57.00
MIN. LEVEL       43.00     43.00     43.00
SLAG TAPPED        .00       .00       .00
SLAG RETURNED      .00       .00       .00
              ANODE STATISTICS

                   A4    A5    A6   TOTAL

ANODES CAST         0     0     3     8

FURNACE CAPACITY   14.50 15.50 17.00

COPPER IN FURNACE   .00 15.00 12.50  25.5
TOTAL LDLS ADDED    .00   .00   .00    .0
FURNACE STATUS
TIME OF STATUS START
ANODE WHEEL STATUS

              WHITE TUNNELS
              WT1          WT2

WT ON FLOOR
WT ON CAR
TIME WT IS WAITING  .00         .00

** CONVERTER INPUT/OUTPUT STATISTICS AT TIME    1

           ADDITIONS                      REMOVALS     STATUS

** CONVERTER TIMES AT TIME   1.

** CONVERTER CONTENTS **

CNV   CMS   FES   SLAG   CU
```

```
            PIG PILE
NO OF PIGS:    22

            PIG BAY
NO OF PIGS     13
COOL PIGS      13
TOTL. PIGS PRODCD:      0

            SKIMMING BLOCK
COOLING LADLES:   2
SKULLS:   1
TOTL SKULLS PROD.:    0

            PILOT PLANT
CAKES READY:   0
TOTAL NO OF CAKES:   0
LADLES OF SLAG RDY:   (TOTAL LDLS OF SLAG PROD=   0

            PROTOTYPE PLANT
CAKE RDY:   (TOTAL NO OF CAKES:   0
LDLS OF CU RDY:   (TOTAL LDLS OF CU PROD=   0

              CRANES

            STATE VARIABLES
WAITING(CNV)
INTERFERENCE(JWE)

            SMP VARIABLES

             EVENTS IN NSET

FILE NAMES              TIME   CNV1   CNV2   CNV3
START OF FILES(NFE)
```

FIGURE 6.3 4   SIMULATION REPORT AT THE START OF A SIMULATION RUN



```
                    REV. FURNACES

                    R1        R2        R3

WHITE LEVEL         17.99     23.16     21.44
MAX. LEVEL          21.00     21.00     25.00
MIN. LEVEL          16.00     16.00     16.00
WHITE TAPPED        36.50     36.50     42.00

SLAG LEVEL          50.99     49.39     58.33
MAX. LEVEL          57.00     56.00     57.00
MIN. LEVEL          43.00     43.00     43.00
SLAG TAPPED         96.82     99.59     47.48
SLAG RETURNED       49.49     45.56     41.08
               ANODE STATISTICS

                     A4     A5     A6    TOTAL

ANODES CAST           0   1024   1246   2850

FURNACE CAPACITY  14.50  15.50  17.00

COPPER IN FURNACE   .00    .00  16.44   16.4
TOTAL LDLS ADDED    .00  15.66  23.13   38.7
FURNACE STATUS        1      0      2
TIME OF STATUS START  0   8250   8310
ANODE WHEEL STATUS    0      0      1

                WHITE TRANSFS
               RT1          RT2

MT ON FLOOR       1            1
MT ON CAR         1           -1
TIM MT IS WAITING 470.11     375.98

** CONVERTER INPUT/OUTPUT STATISTICS AT TIME     0

            ADDITIONS                       REMOVALS        STATUS

CNV DR WR TR PS SC RS FD SS CS A/S   SL CU TR CS    IST  ND
 3   6  3  0  4  3  0  1  1  3  2    12  0  0  0     2   0
 4   2  2  0  0  0  0  0  0  0  4     8  0  0  0     2   0
 5   0  3  0  0  3  0  0  0  0  0     0  0  0  0     1   1
 6   3  4  0  3  0  0  0  0  0  2     0  0  3  0    13   0
 7  10  2  0  2  0  0  0  0  0  2     7  0  0  0     1   3
                        1  1  3  3    14  0  0  0     2   0
**CONVERTER TIMES AT TIME1441.9

STAGE =   WHITE BLOW  =   CU  =   WRS =  FULL = IP AT APIS=   RID=
START                 BLOW                 3=  2= ADD    PER
 N  CS   B   E  SB  SMB  E-SB   B SMB   B SMB   N  CU IST LCS LB
 1   2   3   4   5    6     7     8       9    10  11  12  13  14  15  16

 3143112821299 293  122   0 123   0   31183   0   0  16  24   3  26
 0   31424138A  12    0   0   3   0   0    0   3  64   3   2   2   3
 0   0    0    0    0    0   0   0   0    0   0   3   0   0   2   2
 0   31429142B 147   37   0   0   0   0    0   0  40   0   0   0  16
 31442139139I 322   96   0  38   0   31269  13   3  77  19   0  25
** CONVERTER CONTENTS **

CNV   CUPS    PTS    SLAG    CU

 3    15.3     .0     7.3   88.6
 4    49.9   11.0    28.3     .0
 5      .0     .0      .0     .0
 6    73.9   10.7     7.3     .0
 7    85.0     .0     7.1   83.3


             PIG FILE

NO OF PIGS:    84

             PIG BAY

NO OF PIGS     12
COOL PIGS     12
TOTL. PIGS POURED:       43
```

```
           SKIMMINGS BLOCKS

COOLING LADLES:   1
SKULLS:   7
TOTL SKULLS PROD.:   15

             PILOT PLANT

CAKES READY:   0
TOTAL NO OF CAKES:   0
LADLES OF SLAG RDY:   CTOTAL LDLS OF SLAG PROD:   0

           PROTOTYPE PLANT

CAKE RDY:   CTOTAL NO OF CAKES:   0
LDLS OF CB RDY:   CTOTAL LDLS OF CB PROD:   0


                CRANES

NCR POS  POD  VTT  VTO  IPR  IPO  TDOWN  NEXT LSTD WAGEP.USE P.DELAY
 1   62.  141.  .50  .72  0   9    .00   0   40   10  57.71 16.12
 2  300.  409.  .10  .43  0   1    .00   0   30   10    .31   .00
 3   94.  338.  .65  .42  0   1    .00   3   30   10  68.09  8.24

NCR  ATRIB(1)  ATRIB(2)  ATRIB(3)  ATRIB(4)
 1       0        8        10     -32768
 2       0        3        13     -32768
 3       8        3        12     -32768

             STATE VARIABLES
WAITING(EVT)          0        0        0
INTERFERENCE(JNT)     8        8        8

            SHIFT VARIABLES

ATRIB(1-4) = 1440.53    48.00    65.50    9.00
ISTRIB(1-5):     60         0       12       0
NEXT EVENT:    14518     2652       10      98       9
PRES. EVENT: 1440.53       48       60       9

              EVENTS IN NSET
        NO     POS      TV     SUC/IPR  LINK/ENTRY

         1   14536     412      83      13     33      9
         2       0       2       2       3     32     17
         3       0       0       0       2     27     19
         4       0       0       0       3     17     22
         5   15190     308       0       3   7777     34
         6       0       2       0       0     22     12
         7       0       2       0       0      2     11
         8       0       0       0       2     31      7
         9   14518    2652      11      98      1   9999
        10   14652     412      33      74     16     26
        11       0       0       0       0      7     14
        12       0       0       0       3      4     31
        13   14565    2696      58     193     26     33
        14       0       0       0       0     11     10
        15       0       0       0       0     22     05
        16   14987     398      18    6233     34     18
        17       0       0       0       0      2      4
        18       0       0       0       0     14     27
        19       0       0       0       0      3     32
        20       0       0       0       0     24     15
        21       0       0       0       0     25   9999
        22       0       0       0       0      4      4
        23       0       0       0       0     35     29
        24       0       0       0       0     29     23
        25       0       0       0       0     13     32
        26   14602     418      13     178     18     13
        27       0       0       0       0     19      3
        28       0       0       0       0     31     21
        29       0       3       0       0     21     24
        30       0       0       0       0     25      0
        31       0       0       0       0     12     20
        32       0       0       0       0     19      0
        33   14958     452    1823      43     13      1
        34   15560     358   16440      43      9     18
FILE NAMES                TIME    CRN1    CRN2    CRN3

START OF FILES(NPT)          9       2       3       0
```

FIGURE 6.3.5   SIMULATION REPORT AFTER SIMULATING THREE 8-HOUR SHIFTS

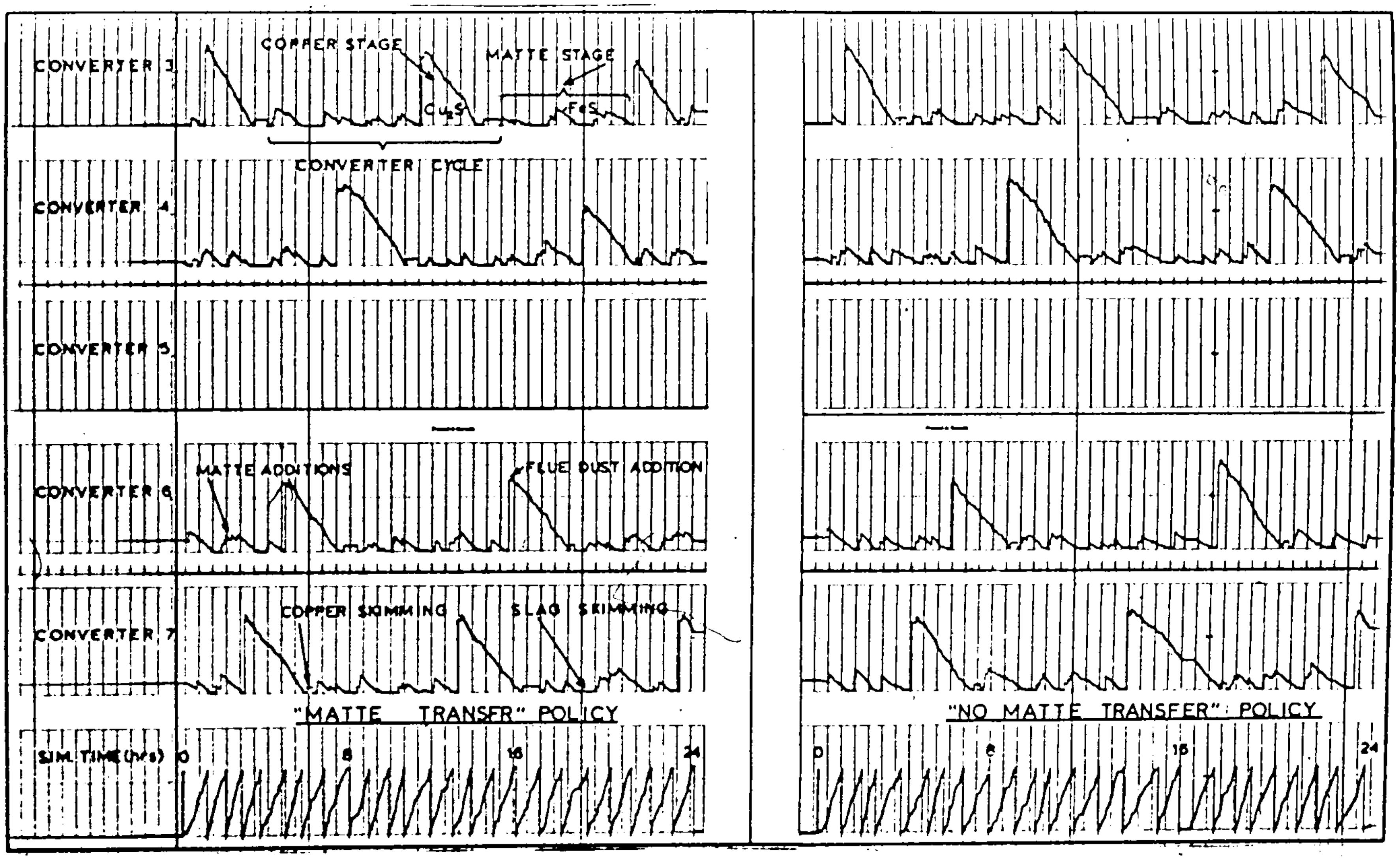


FIGURE 6.3.6 24 HOUR RECORD OF CONVERTER ACTIVITIES FOR TWO DIFFERENT POLICIES

6.4 SYSTEM RESPONSE TO "NO-MATTE TRANSFER" POLICY

In March 1973 a trial run at the smelter was undertaken which was to establish the advantage of operating without matte transfers.

It should be pointed out here that the "no matte transfer" policy is merely a general policy, it does not imply that white metal should never be transferred between converter furnaces. If due to a breakdown or some other delay two furnaces were operating almost in phase it would be advantageous to perform a transfer. This would bring the one converter back to the beginning of a cycle while the other would be advanced to a later matte blowing period. In this way the cycles of the four converters could be spaced if necessary by a transfer. Under normal operating conditions this spacing is done by varying the charge sizes in converter cycles. Good spacing of converter cycles is essential to ensure both a uniform need for matte and to avoid any two converters requiring major service at the same time.

The trial run was successful and matte throughput based on the no matte transfer policy could be increased almost immediately.

Operating personnel found it relatively easy to adjust to the new policy. After 24 hours, when each operating crew had completed one shift based on the new policy, production increased beyond the previous average. Figure 6.4.1 shows the matte throughput 10 days before and after the change-over expressed in terms of the average of the first ten days. An average increase of 6% in matte throughput was achieved over that period. This change is significant since production before the change was considered at a high level. Also, the weekly average of matte throughput reached the highest level ever attained until then during the first week. While the initial 10 days of operation do not reflect any long term effects they show the potential of the policy on a day to day basis. Figure 6.4.2 shows the copper production in the same period. Copper is produced, this was pointed out before, in large batches and shows a considerable variation for that reason. For this reason no conclusive evidence of an increase in copper output is provided by this plot but some increase is indicated. A somewhat lower copper output would also be expected in the first week because of a lower matte grade that was supplied in that period.

The long term effect of the policy change is shown in Figure 6.4.3. The weekly matte throughput for 1973 is plotted as a percentage of the 1972 average. This basis was chosen to eliminate the effect of the low matte throughput in the first weeks of the year. Except for the sixth week after changing to the new policy when the smelter was shut down for 7 1/2 hours a very uniform production was achieved. Using a corrected value of matte throughput for that week the average of the 7 weeks of no transfer policy is 2% higher than the 7 weeks prior to the change. Compared to the 1972 average the increase is 6%.

The trial run was initially planned for 48 hours and was later extended to a week, when it was decided that the new policy should be adopted.

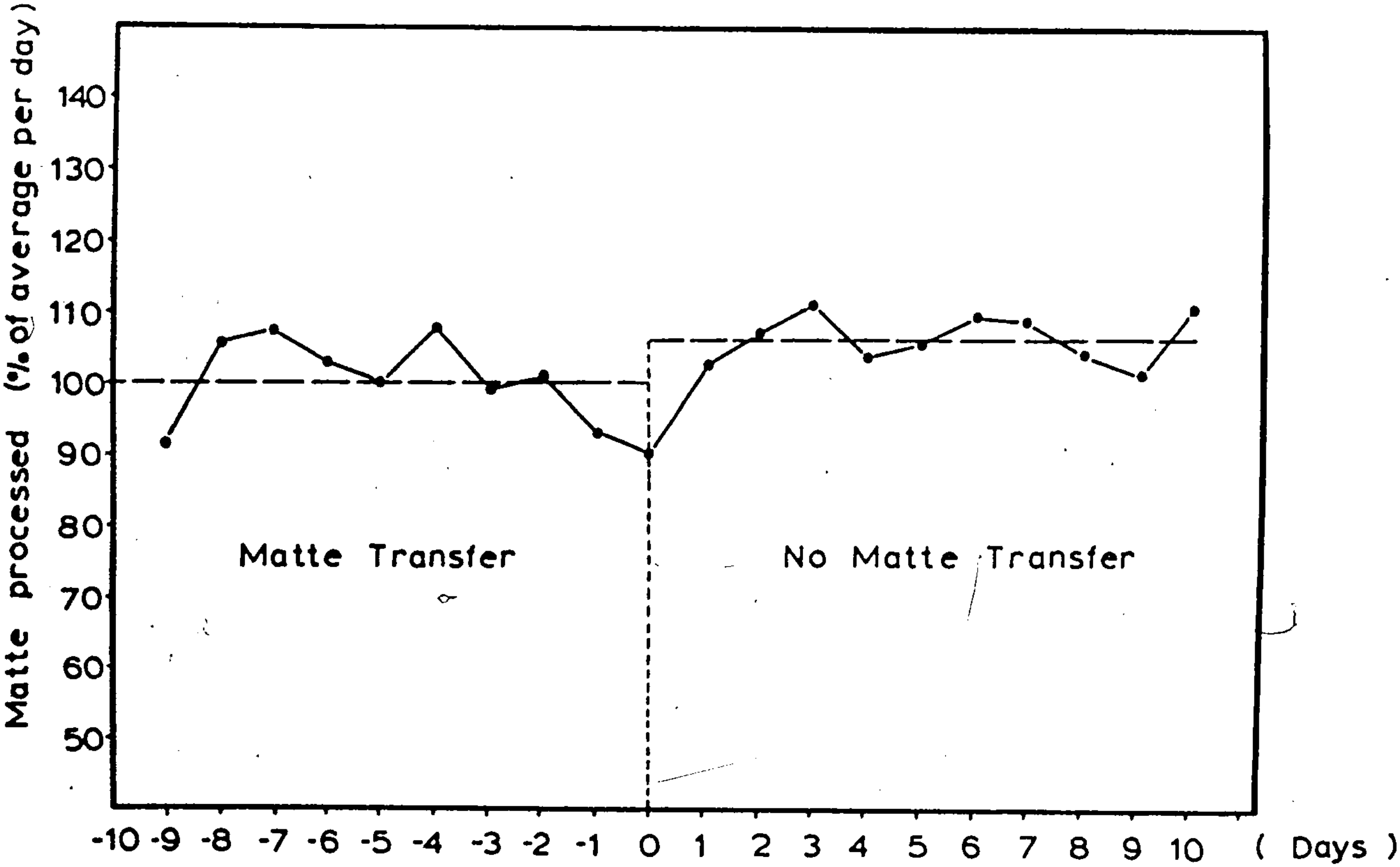


Figure 6.4.1 Matte Throughput before and after policy change based on average of transfer policy

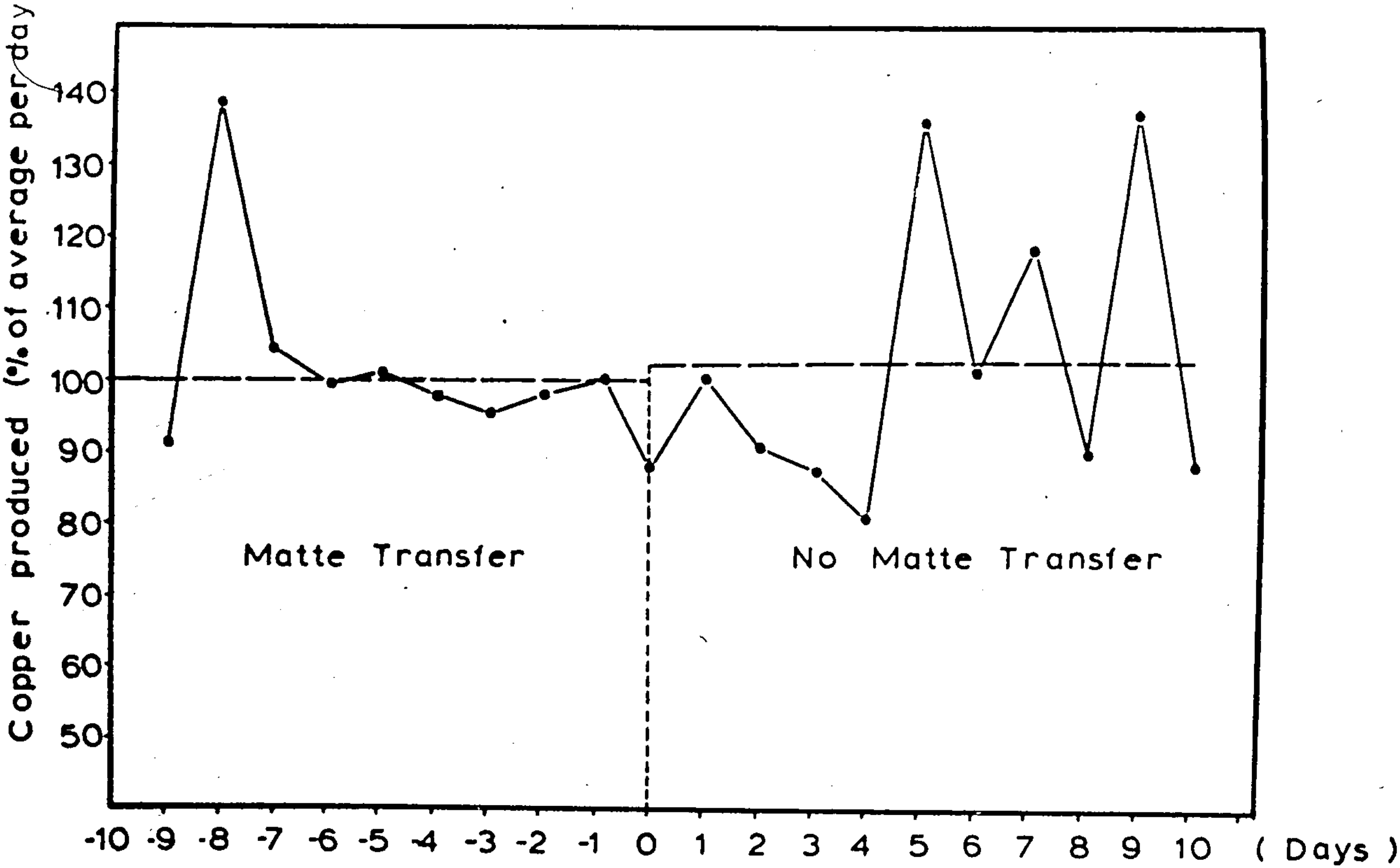


Figure 64.2 Copper Production before and after policy change based on average of transfer policy

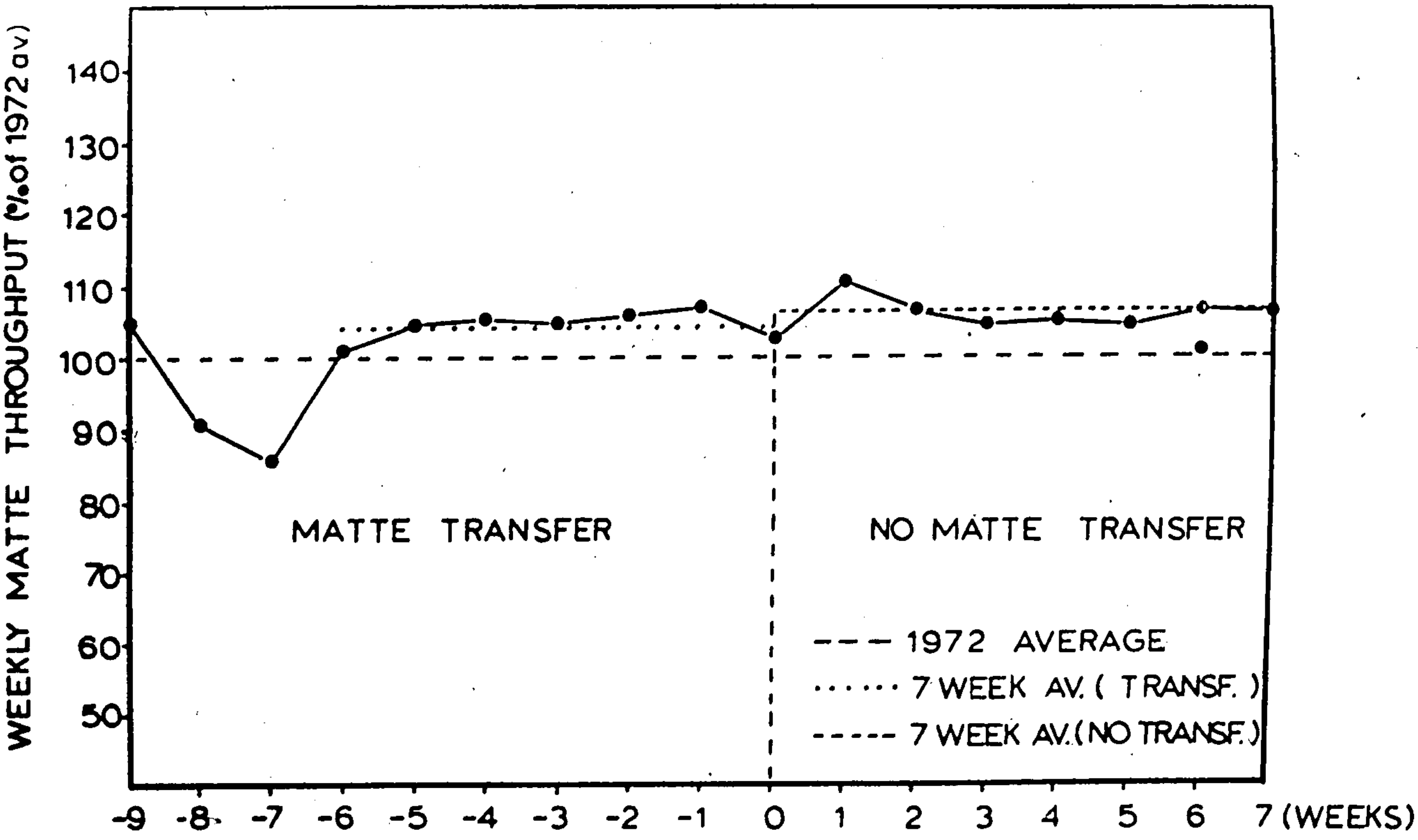


FIGURE 6.4.3 ACTUAL MATTE THROUGHPUT BEFORE AND AFTER POLICY CHANGE

6.5 DISCUSSION OF RESULTS

By changing operating policies or input variables of the model, the model is not changed and thus provides a basis for comparing results of different experimental runs. If, however, a particular change is to be tested in the real system it is important that the basic system is not affected by the proposed change. If the system changes, predictions would have been based on a different model and cannot be expected to correspond with results of plant tests.

It was suggested that the no-matte transfer policy could result in a lower air flow rate in the converters due to larger heads of liquid in the converter. This would adversely affect the reaction rates and reduce the refining capacity of converters. In the converter models a fixed air flow rate is assumed. This aspect of the converter model would have to be a function of the liquid head in the furnaces if such a relationship was observed.

Air flow charts of the trial period were analysed for this effect but no significant lowering was observed. A slight drop in flow rates occurred towards the end of a matte blowing stage but the overall effect was insignificant.

There were two further changes in the system when operating according to the no transfer policy that were not accounted for in the model when simulation experiments were run.

1. During the second week of operation the brick lining in converter #7 showed critical wear and the converter had to operate at reduced capacity. During the third week it was then shut down and replaced by #5 converter.

Apart from a slightly different physical location converter #5 has a lower capacity, and a lower average air flow rate. This change in the system should decrease the overall matte throughput capability by as much as 4%. At the time it was not known that the converter would be shut down and simulation runs were based on the original configuration including converter #7.

2. The start-up of the continuous smelting plant almost coincided with the trial period for the new policy. In the first few weeks the effect was negligible but as the new plant produced more blister copper, which has to be processed in the converters during a copper blow, some of the converter capacity was used for this purpose. To estimate this effect is difficult but it is expected to be in the order of 1/2 - 1%.

Considering both negative effects on the capacity of the system the long term increase of matte throughput of 2% shown in Figure 6.4.3 is in fact in the order of 6% or more. This possible increase was indicated during the first 10 days of operation when both converter #7 was operating and the continuous smelting plant did not operate at full capacity.

The improved operating policy as predicted by the model was thus well substantiated by the data obtained from the plant test. Although there was a considerable variation in the measure of performance, the increase of matte throughput corresponded closely. This correspondence did in fact show that if the effect of any change in the system is represented in the dynamics of the model, the model can successfully predict such effects. It is thus important to ensure that the model still corresponds to the system before any inferences based on model predictions are made.

Of the other areas worth investigating all but one would require similar trial runs to study the effect of different factors until an improved performance was achieved, if indeed this was possible. If several dependent factors were to be investigated a number of experimental simulation runs have to be performed to account for all possible combinations of these factors. Usually the number of experiments is so large that only a selected number of factors are analysed.

Of particular interest was the possibility of producing a submodel that is able to control the smelter operations automatically without the need of any operator attention. This program could if successful control the smelter in a continuous and possibly optimal fashion. This problem and a preliminary solution is presented in the next chapter.

7.0 DEVELOPMENT OF AUTOMATIC MODEL CONTROL

7.1 DEFINITION OF PROBLEM AND OBJECTIVES

In Chapter 6 the successful change to a different operation policy, the "no matte transfer" policy was discussed for which simulation results had predicted a better performance. The policy provided the smelter operator with a different set of operating rules which when adhered to gave an improvement in smelter performance. The decisions that have to be made by the foreman were developed and refined by several simulation runs during which the strategy of no matte transfer was tested. The decision process involved in controlling the overall model can be systematically treated by defining the following four stages that occur continually during a simulation run, or for that matter in the real system:

1. Recognize the need for an operator decision.
2. Establish status of system (resources)
3. Establish projected aims (demand)
4. Establish optimum sequence of tasks, in response to required need.

Every input to the simulation model from the operator has thus been preceded by a sometimes trivial or more frequently by an intricate sequence of thoughts and decisions. An experienced operator will perform some tasks automatically while others will require conscious thought.

In the actual system this process of routine execution of some tasks is very well displayed. If for example the operator has made a decision to skim slag from a particular converter (this might already be done without his consent if there is no immediate other service required), then the crane operations for skimming follow a sequence known to the crane-operator and swamper. In other cases the foreman relates his decisions to a swamper (a man directing cranes to perform certain services) who will see to their execution.

The number of commands issued by a foreman during an 8-hour shift are in the order of 100 or more. The number and complexity of the decision making process increases with the number of variables influencing the overall performance of the system, and only an "Economic Man" as defined by Taylor (T-3) could make optimum decisions in all cases. An Economic Man is presumed to have three properties. He is completely informed, infinitely sensitive and rational. Without going into the literature of decision making behaviour by people such as Cyert (C-2) and Simon (S-5), it can be said that the actual behaviour of human decision maker in a complex system will fall short of the ideal. Even if decisions were perfect he will be subject to fatigue and his performance would decrease in the course of time.

In this Chapter the development of a control procedure is described that allows an automatic operation of the simulation model. Its limitations and possible applications to optimise the real system performance are discussed.

7.2 HEURISTIC APPROACH TO DECISION MAKING PROCESS

The performance of the overall system is to be optimised by generating a decision sequence that results in the best possible throughput of material. In a static, deterministic system (depending on the nature and size of the problem) techniques such as dynamic programming can provide optimal decision strategies.

The smelter system, however, as represented by the simulation model represents a dynamic system changing in an indeterministic fashion with time.

For a problem of this size and structure a heuristic decision model is required which allows for one level of decision, namely that of the converter foreman. Heuristic models or procedures are described by Bowman and Fetter (B-3) as methods for searching a solution space for a solution. When used to represent a decision making process, heuristic procedures generally involve modelling of the human decision making processes. This process provides a great deal of insight into the decision making problem based on which the heuristics can then be refined and extended to produce improved control procedures.

These two goals, the identification of human decision making processes and development of automatic control procedures that could be superior to human operator control, were to be achieved.

The model with the available display unit provided an excellent tool to develop heuristic decision rules. These rules could be tested and improved in an iterative fashion by observing the behaviour of the model until a satisfactory level of control was attained.

While the basic idea is simple the implementation of the heuristic decision involves a fair deal of effort, if the level of sophistication of the human decision making capability is to be achieved. Work in this area has been mainly reported for conventional business models, and McMillan and Gonzales (M-7) report of one model that included aspects of human motivation expressed in terms of an index of stress.

In the smelter environment decisions are governed largely by process states and material availability and the decision model was thus based only on such variables. A control decision rule thus uses a set of state variables and causes the system to change to a new set of values by issuing commands that will initiate this change. Associated with every change is some cost and any decision should be of such nature as to maximise the profit. While the cost function is not used as such decision rules were developed to produce maximum matte throughput which corresponds with the aim of maximising profits. The decision making program developed performs three basic functions:

1. Sensing of information
2. Forecasting of expected activities
3. Making decisions and causing action based on 1 and 2.

By their nature algorithms are not easily generalised. For this reason a description of the rules is given in the next two sections, while the success of the program is discussed in section 7.5.

7.3 AUTOMATIC CONTROL OF SMELTER ACTIVITIES

The automatic scheduling and control of the converter aisle operations are performed in submodules 9-12 of the model. The different submodules are described in Appendix A but for the purpose of this description the four submodules can be considered subprograms of the overall program used for the automatic control. By pressing a sense switch (No.7) the model is placed in "automatic operation", which thus describes the simulation state in which no operator action is required to run the model.

Apart from no commands being issued through the teletypewriter the operation of the model is identical to the manual mode. By resetting the sense switch, the model is returned to the manual mode at any time during operation.

Referring to Figure 4.1.5 on page 45 which shows the interaction of the operator with the model, the basic functions of the operator can be replaced by an automatic control program.

While the display facilities are essential in the manual mode of simulation information transfer from the model to the automatic control programme occurs directly. However, for developing control algorithms and in evaluating them the display facilities were essential. They provide continuous information and show whether or not a decision making algorithm is successful in dealing with all possible conditions that can arise during a simulation run.

Figure 4.2.3 on page 57 shows how the command input function of the operator is taken over by the "Automatic Control Programme" as soon as sense switch 7 is pressed.

The ACP has access to all state variables of each submodel and when called upon to generate a command, the decisions involved will be influenced by the state of related processes.

The task of a model operator of sensing information, making a decision and issuing a command is reproduced in a similar form in the ACP. Corresponding to Table 4.1.1 which lists triggers and actions of operator commands, similar triggers will activate the ACP. Responding to any such input a decision algorithm will generate a suitable command that ensures a continuous model operation.

What follows will be a description of the implementation which will give the best insight into the techniques used in developing the decision rules. Examples will show how particular commands are generated.

Having established the advantage of operating the smelter according to the "no matte transfer policy," the automatic operation was based on this policy. In submodule 9, coded values of the variable ISET are interpreted which are given a value whenever an event occurs which would have required an operator action during simulation. In other words the variable ISET = 1,25 relates the same information to submodule 9 that is also shown on the model display or the typewriter. Instead of the operator reacting to the information, the information is processed and the required response is generated automatically in a way that a satisfactory operation is obtained.

Table 7.3.1 lists all values of ISET with the corresponding event that sets the variable ISET. A brief description of the action that follows is given in the third column on the Table. Essentially every value of ISET is associated with a command that is issued and a corresponding event will be placed in one of the four files, or directly executed. Events are then executed in the usual fashion, that is based on the time, if it is a time event or sequentially, if it was a crane event. In general, events generated in submodule 9 form a series of events for a given value of ISET. To give an example the value

ISET = 3

is set whenever an end point of a matte blow is reached.

The converter number is given by the variable NCV which allows a determination of the quantity of slag actually contained in the converter. This quantity is given by the array CV (NCV,4) and by comparing it with the critical quantity of slag that may be left in the converter at the end of skimming, C MIN (1), it is possible to determine the number of ladles of slag that are to be skimmed. Knowing this quantity, a skimming procedure by either one or two cranes is ordered depending on the availability of the cranes. By pressing a sense switch (SSW(6)), a one-crane operation can be chosen which will override the decision of the program. A sequence of events is then generated which are interpreted and filed into the files in submodule 10 in a similar way as manual operator events are interpreted by submodule 4.

In generating the commands to skim slag, decisions have to be made as to which reverberatory furnace is to receive the slag and which crane is to transport the slag. Having skimmed all the slag, more matte is to be added which requires a decision about the place of supply for the matte. These decisions occur often and every time complete knowledge about the status of the process concerned is required. For this purpose subroutines were developed for each process or submodel which are called to provide the required information. Typically SUBROUTINE MATTE(MTUN) is called if matte is required. The status of both matte tunnels is checked for available matte and either matte tunnel 1 or matte tunnel 2 will supply the required matte. If no matte was available at the instant, a message will be typed out to that effect. Similar routines for other processes exist.

CONV (JZZZ, IT, NBEST, MUL)

REV (NBEST, IRS)

MATTE (MTUN)

ANODE (IT, NBEST)

CRANE (NO, LOCA, LOCB, NBEST)

TWOCR (JC1, JC2, KAPUT, NQ, NCV)

Subroutine SORT (JU, K, MA, MI) is used to establish priorities in case of conflicting requirements of two of the same processes and subroutine TISTAT (IFI, JFI, IT, ISTAT, ICO) is used to establish the status of converters at any time. Subroutine FNDCEV (IVAL, JQ, JATT, KCOL, NSET) establishes whether at any time a particular event has been issued but has not been executed. This is necessary to prevent the same command from being issued twice.

Each of these subroutines represent a certain policy decision and can be replaced at any time by another. Typically, the reverberatory furnaces could accpet converter slag in equal proportions. If each furnace has a specified fraction of converter slag assigned to it, another decision algorithm is used to establish these proportions. Another example is the use of cranes for specific tasks. Subroutine CRANE chooses the best crane to do a particular task based primarily on its availability and the location where service is required. If skimming of slag from a converter is to be performed with two cranes, then subroutine TWOCR performs the selection. Figure 7.3.1 shows a simplified information flow of submodule 9. The program listing is given in Appendix G. For every decision corresponding to a particular value of ISET a number of commands are generated and stored in the arrays:

(1) JSEQ(I)

(2) JCMND(I)

(3) NCRAN(I)

(4) JFROM(I)

(5) JTO(I)

(6) JPR(I)

(7) JQ(I)

Each array contains the coded information that corresponds to the basic command of the type:

```
XXYYNCR, X1, X2, IPR, Q
1    2   3   4   5   6  7
```

The command input in the manual mode is described in Appendix A. If less than the 7 possible variables are sufficient to make up one command, only the significant left hand variables are used. A maximum of twelve commands can be issued at one time which is enough to account for all possible sequences of commands. The arrays are transmitted to submodule 10 via COMMON statements where the commands are interpreted and filed in the crane or time files in the same way as operator commands from the teletypewriter are.

Generation of internal commands is controlled in subroutine EVENTS in the main module. After execution of any event by subroutine EVENTS the program branches to submodule 9 if ISET > 0 which indicates that an internal command should be generated. Upon return to EVENTS the program branches to submodule 10 to file the events if any commands were generated in submodule 9 automatically, i.e., when MODUL = 4 (see Figure 7.3.1). The need for an internal command does not necessarily result in one, i.e. if material is required but not available no command is issued at that time. The number of internal commands is given by NCRAN.

Having described the mechanism of generating commands internally during simulation the logic of the decisions will be described briefly. It should be stressed again that the structure of the program is such that decision algorithms can be modified or replaced at any time. The decisions as presented are based on present operating practice and were modified where necessary after trial runs on the model until a satisfactory automatic operation was achieved. The discussion below will follow the values of ISET as outlined in Table 7.3.1.

ISET = 1: Matte becomes available at Matte Tunnel 1.

This event will cause a search of all converter states which is performed by subroutine TISTAT. If any converter is not blowing and waiting for matte it will receive the highest priority for obtaining matte. Thereafter converters which are blowing but require more matte are assigned priorities. Subroutine CONV and subroutine SORT are used to select the converter with the highest priority which corresponds to the converter with the least amount of matte added in a particular blowing period.

Once the best choice for a converter is made subroutine FNDCEV searches all crane files for the same event which might have been issued before. If such an event is found no further command will be issued. Subroutine CRANE chooses the best crane for the task if a command is issued. If no converter is ready to receive matte the ladle is placed on the floor. No more than three ladles can accumulate on the floor at each matte tunnel. In this case matte tapping is suspended.

ISET = 2: Matte becomes available at matte tunnel 2.

This event is identical to the one above except that the origin of matte is matte tunnel 2.

ISET = 3: End of matte blow in a converter.

If a converter was filled to capacity for a matte blow it requires skimming of slag at the end before more matte can be added. The amount of slag in the converter is given by CV(NCV, 4). In this event slag skimming is scheduled which will leave less than CMIN(1)*TPU Tons in the converter. This operation can be done by a single crane or with two cranes. For a single crane operation sense switch (6) has to be set. The converter slag is returned to one of the reverberatory furnaces. The furnace number is supplied by subroutine REV. One algorithm for distributing slag evenly between the three furnaces and one for a 25%, 25%, 50% distribution, are available. Subroutine REV ensures that no two consecutive ladles of slag are returned to the same furnace to ensure that there is enough time to clean the slag launder between charges. Again the selection of a crane is done by subroutine CRANE if a single crane operation is required. The selection is based on the target and origin of a task, the duration of the present task of a crane and the number of tasks to precede the task. If any crane is not operational it will not accept any commands. For a two-crane operation, subroutine TWOCR chooses a combination of two cranes. If the centre crane is not operational a single crane operation is substituted.

Once all the slag is skimmed, one ladle of matte is charged to the converter. The availability of matte is given by subroutine MATTE. If no matte was available, a message to this effect will appear on the typewriter: "Slag is skimmed only".

ISET = 4: End of a matte stage.

The end of a matte stage differs from the end of a matte blow in two respects. The very last ladle of slag skimmed is not returned to a reverberatory furnace but to a converter in the matte stage. After skimming, flue dust is added to the converter which will proceed then with a copper blow. The maximum quantity of converter slag that may remain is given by CMIN(2).

These operations can also be performed with either one or two cranes. Converters able to receive high copper slag (CS) are determined in subroutine TISTAT and if there are more than one a selection is made in subroutine CONV. If no converter is available, the converter slag is placed on the floor near the particular converter. This situation should never occur in practice but if it did, provision is made for it.

ISET = 5: End of a copper stage in a converter.

At the end of a copper stage, a command is issued to deposit a skull into the converter (Refer to ISET = 6 if no skull is available). Thereafter a number of ladles of copper are removed based on the quantity of copper as given by CV(NCV,5). These tasks can be performed with one or two cranes. Copper is always placed in one of the anode furnaces unless none of them are ready to receive any copper. In that case copper is poured in the pig bay. If there are no moulds available, the bay will be cleared first. (Refer to ISET = 15 where this is prevented from happening).

ISET = 6: (a) A ladle has cooled sufficiently and has been bumped.

(b) Anode furnace becomes ready to accept copper.

If there were no skulls available, the first ladle bumped thereafter will cause the status of all converters to be checked to establish whether a skull is required before skimming copper. If this is the case, the events corresponding to ISET = 5 will be issued, otherwise no action is required. The same applies if an anode furnace became ready again to accept copper, if there were none before.

ISET = 7: Addition of matte to a converter unless it is filled for a particular matte blow.

The availability of matte is established in subroutine MATTE and a command to add a ladle to the converter is issued. If no matte is available or if the same event exists in a crane file (as given by FNDCEV) then no command is issued.

ISET = 8: Converter is empty after removal of all copper.

A time event is issued here for event No. 54 at 15 minutes after the present time when charging of matte to the converter should start. When event 54 is executed as a regular time event, ISET is set to 7 and the action described above follows.

ISET = 11: A copper pig is required by a converter.

In the copper stage, pig requirements are satisifed by a command which causes a crane to pick up a pig at the pig bay and drop it in a converter. If the pig bay was empty or no cool pigs were available, then they will be taken from the pig pile. If there were no pigs at all, a message is typed out to that effect. (Refer to ISET = 15 where this is prevented from happening).

ISET = 12: Copper cakes are available at the pilot plant.

This event causes a crane to move the cakes to the pig pile.

ISET = 13: Reactor slag is available at the pilot plant.

The converter states are checked in TISTAT to find a furnace ready to receive reactor slag. If more than one were available, selection of the best converter is done on the same basis as for ISET = 3 or 4.

ISET = 14: Ladle requires bumping.

Subroutine LADLE records the usage of ladles and requires bumping whenever a ladle has been used a given number of times for transporting material. Since this condition when a ladle requires bumping can arise at the same time that another event is to be scheduled automatically, the variable ISET would lose one of the two coded commands. For this reason the number 5000 is added to ISET in subroutine LADLE whenever bumping is required while the original value of ISET is retained. At the beginning of submodule 9 a numerical value of

$$ISET > 5000$$

will cause both the bumping of a ladle and any other command code, i.e, ISET - 5000 to be executed. In addition to causing a crane to bump a ladle at the bumping block, it is instructed to deposit another ladle at the bumping block for cooling.

ISET = 15 (a) Less than 10 pigs available

(b) No anode furnace ready to accept copper.

If any one of the above conditions are detected when picking up pigs at the pig bay or pig pile, then the pig bay will be emptied immediately. The next charge of copper will then go to replenish the stock of pigs in the pig bay.

ISET = 16: Anode furnace is full.

This event triggers the command to skim anode slag from the anode furnace which is charged into a converter in a matte blow. (At the time of programming this event, it was not known that anode slag should be charged into a converter in a copper blow. This change can be readily made.)

ISET = 17: Anode furnace is ready to cast.

Casting of the anode furnace is triggered if the respective anode wheel is ready. If it is not ready, the command will be issued again when ISET = 19.

ISET = 18: Slag available in the going high blow in a converter.

When enough slag is available in the final matte blow, an event is filed which causes a crane to skim slag.

ISET = 19: Casting is completed at one of the anode wheels.

If one of the two anode furnaces both casting on anode wheel 4 were ready to start casting before the other had finished, then this event would be triggered when the anode wheel is ready again.

ISET = 20: Anode furnace ready to refine.

This event causes an anode furnace to start the refining process.

ISET = 25: Converter requires collar pulled.

This event causes a crane to pick up the collar puller and pull the collar on a converter. It might be noted here that collar pulling is one task that can be interrupted while in progress by another crane if a higher priority task is to be executed.

A special state variable KWT(NCR) had to be introduced to allow a two crane operation, which will be explained briefly by an example. If 3 ladles of slag are to be skimmed from a converter at the end of which matte is to be added, then the commands issued would be such that each crane skims one ladle of slag first, which each pours into a reverberatory furnace. Then one crane skims the third ladle and waits at the converter until the other crane has picked up and dumped a ladle of matte into the converter. Thereafter the one crane returns the empty matte ladle while the other pours slag into a reverberatory furnace.

Under normal operations, crane tasks are executed independently by the crane and synchronization of tasks from two different cranes is not possible. In a two-crane operation, every crane event that is to be delayed contains a pointer, based on the variable KODE, to the other crane of the paired operation. This value is used to set KWT(NCR). Subroutine TKODE interprets this value in the controlling program and interrupts the sequential execution of crane tasks. This means that in the example given the crane that skims the last ladle of slag from the converter will be delayed thereafter until matte has been dumped into the converter. The event following the dumping of matte in turn contains another pointer (given by KODE in the event) which will cause the waiting crane to complete its task. If the crane with matte was to arrive before the last ladle of slag was skimmed, the same synchronizing would take place in the reverse order. This ensures that matte is not added before skimming is completed.

FIGURE 7.3.1 GENERAL FLOWCHART FOR SUBMODULE NINE

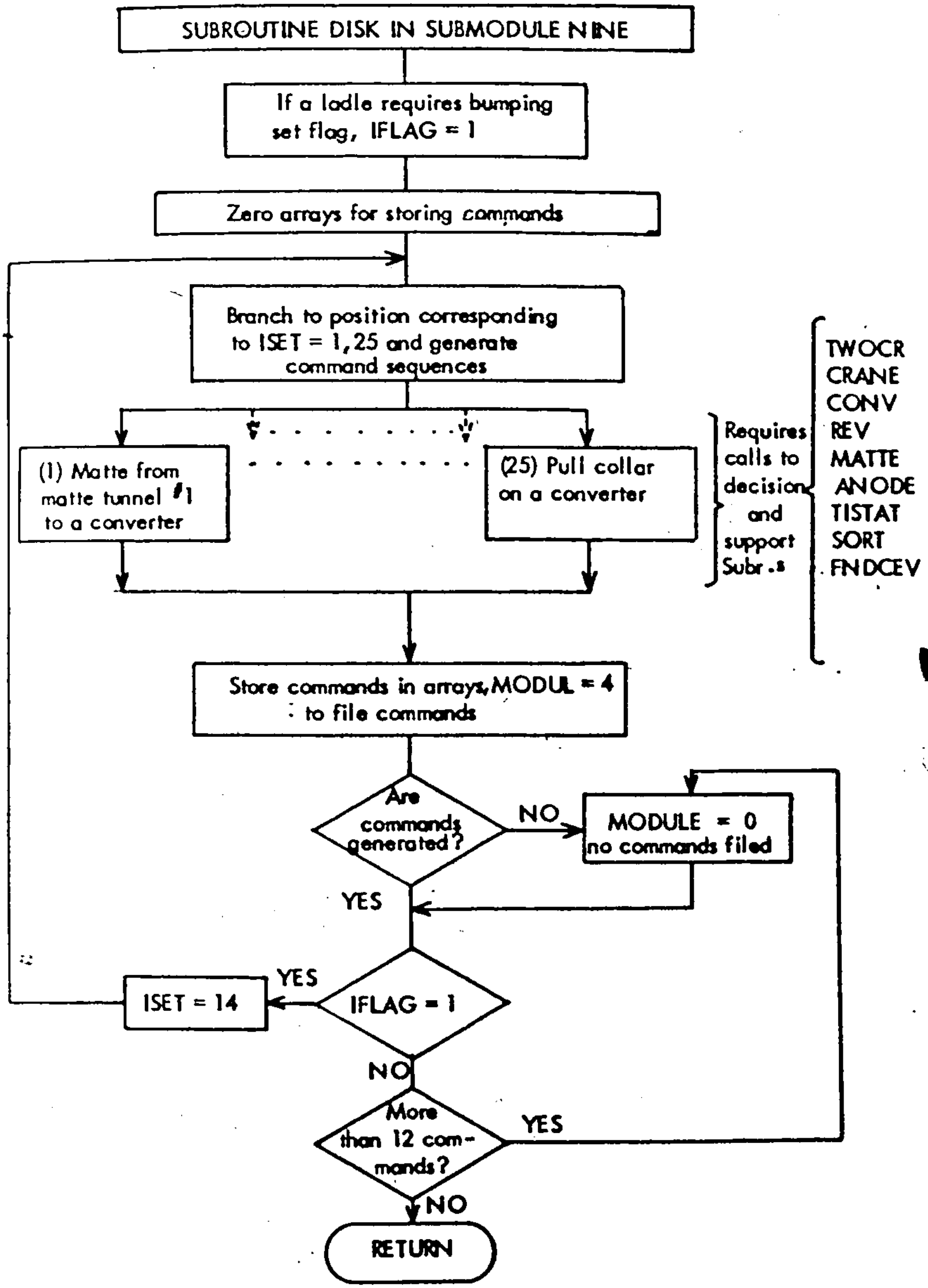

SUBROUTINE DISK IN SUBMODULE NINE
If a ladle requires bumping set flag, IFLAG = 1
Zero arrays for storing commands
Branch to position corresponding to ISET = 1,25 and generate command sequences
(1) Matte from matte tunnel #1 to a converter
(25) Pull collar on a converter
Requires calls to decision and support Subr.s
TWOCR
CRANE
CONV
REV
MATTE
ANODE
TISTAT
SORT
FNDCEV
Store commands in arrays, MODUL = 4 to file commands
Are commands generated?
NO
MODULE = 0 no commands filed
YES
YES
ISET = 14
IFLAG = 1
NO
More than 12 commands?
YES
NO
RETURN

TABLE 7.3.1 - LIST OF CODED VALUES OF VARIABLE ISET

ISET	Event Requiring Response	Command(s) Scheduled Automatically
1	Matte arrives at Matte Tunnel 1	Matte to a converter if required
2	Matte arrives at Matte Tunnel 2	Matte to a converter if required
3	End of matte blow in converter	Skim slag, add more matte at end
4	End of matte stage in converter	Skim high grade slag, add flue dust
5	End of copper stage in converter	Add skull and skim copper from convurter
6	Ladle was bumped for skull	Add skull to converter if required before removing copper
7	Addition of matte to a converter if not filled for particular blow	Add more matte to converter if available at matte tunnels
8	Removal of all copper from converter	Schedule addition of matte for new converter cycle
9	Reserved for Prototype Plant	
10	Reserved for Prototype Plant	
11	Requirement of pigs in copper stage	Addition of pig to converter
12	Pigs available at pilot plant	Move pig to pig bay
13	Reactor slag available at pilot plant	Add reactor slag to converter which is ready to receive it.
14	Ladle at bumping block has cooled sufficiently	Deposit another ladle that requires bumping and bump ladle
15	More pigs required	Empty pig bay
16	Anode furnace ready for skimming before refining	Skim anode slag and dump into a suitable converter
17	Anode furnace finished refining	Trigger casting of anode furnace
18	Slag available in going high blow	Skim slag from converter
19	End of casting of A4 or A5	Trigger casting of other furnace if it was waiting

20	Anode fumace ready to refine	Trigger refining process in anode fumace
25	Converter requires collar pulling	Schedule collar pulling at resp. converter.

7.4 FORECASTING OF EXPECTED PROCESS STATE

Not all control decisions can be based on present and historic data, but require an estimate (schedule) of what might happen in the future. This is also necessary in the smelter model and will be described below by showing the estimates required and how they were obtained. If the model is placed in the automatic mode as described so far, it would not require any operator attention except for two decisions:

1. The decision to proceed with a copper blow before the end of the eighth matte blowing period, to ensure correct spacing of converter cycles.
2. The decision not to fill converters in a matte blow to capacity if this is desirable at any time.

To account for these decisions another subprogram was developed that allows forecasting of the availability of matte at the matte tunnel and the material requirements of the converters. It is on this estimate together with the smelter status that a converter foreman will base his shift schedule.

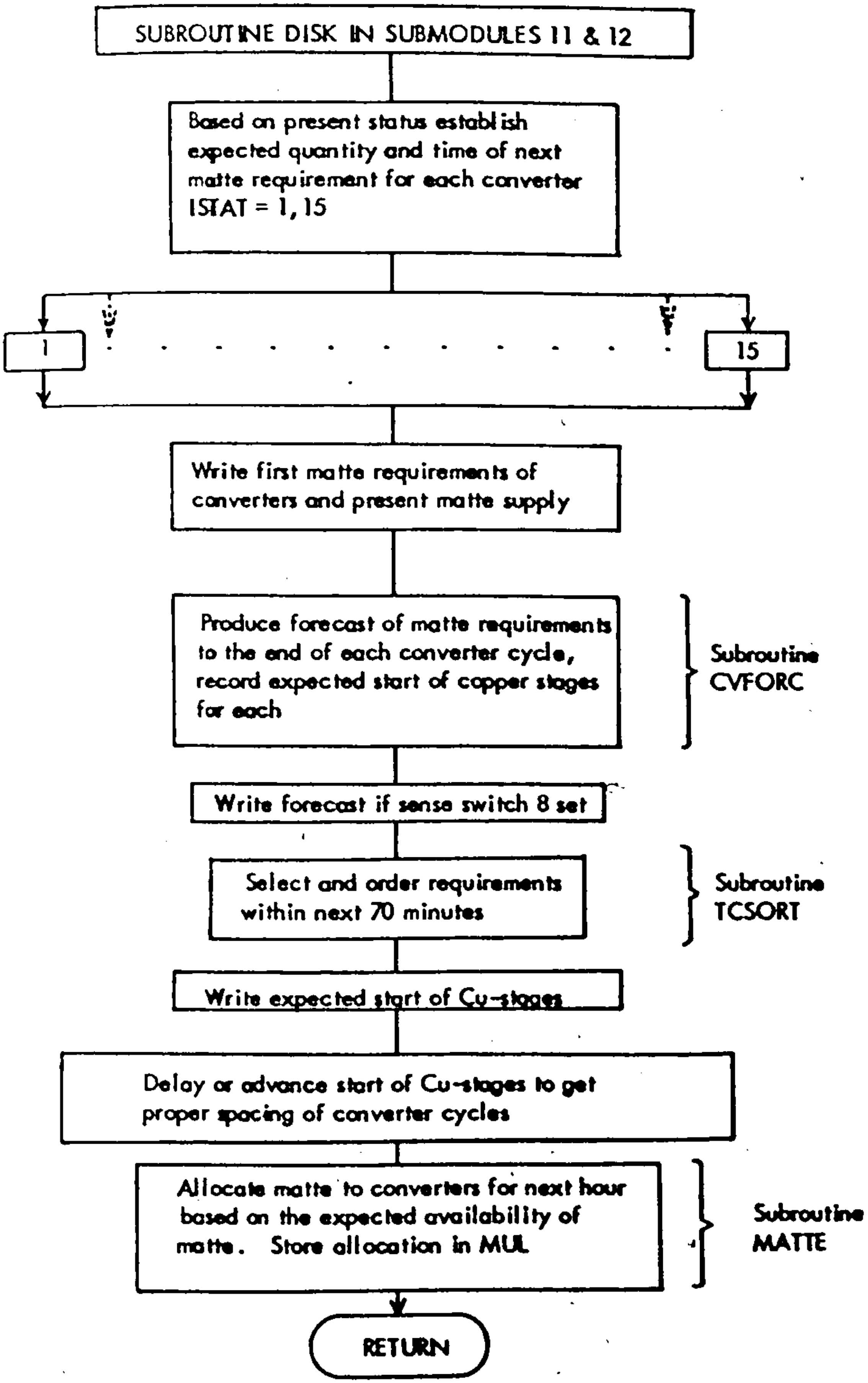
Figure 7.4.1 is a simplified flowchart of this program. Because of its size it is split up into two submodules, submodel 11 and submodule 12, but for the discussion here it can be assumed to be one subprogram. The event #40 triggers this forecasting program which is repeated after a given time interval (60 minutes and 30 minutes were used) by calling the same program to update the previous estimates. Based on data supplied by the converter and matte tunnel models the status of converters and matte tunnels is established at the beginning of the programme. A forecast of quantity and time of matte requirements for each converter cycle is made based on expected blowing times and crane service requirements. This forecast is performed by subroutine CVFORC. (Appendix G). It also provides the time when each converter is expected to proceed with a copper blow. These expected times are used in an iterative algorithm for spacing of converter cycles until a satisfactory separation between copper blows is obtained. Subroutine MTFORC is used in submodule 12 to forecast the availability of matte within the next 70 minutes. This forecast is based on the present matte tunnel status and the expected times between arrivals of ladles of matte. The available matte is then assigned to different converters according to priorities,and additions per blowing period are stored in array MUL (NCV, IBLOW).

If a further refinement of matte charging was desired, this could be very easily implemented. The two-dimensional array MOP(NCV, IBLOW) contains the predicted optimum allocation per blow whereas presently the array MUL(NCV, IBLOW) specifies the maximum allowable charge per blow. The arrays are of the same size and MOP can replace MUL in subroutine CONV in submodule 9 where the actual charge is compared to the allowable limit before any further additions are made.

It was mentioned that a forecast to the end of each converter cycle is made. Of this information, only the expected time to start on the copper blow and the matte requirement within the next 70 minutes are used. Since the forecast is repeated every hour the projections within the next 60 minutes will be the most accurate if used for control decisions.

By pressing Sense Switch (8) the operator can request a printout of the forecast in the automatic mode of operation. The forecast will provide him with fairly reliable information regarding the next 60 minutes and less accurate estimates for later periods.

FIGURE 7.4.1 GENERAL FLOWCHART FOR SUBMODULES ELEVEN & TWELVE


SUBROUTINE DISK IN SUBMODULES 11 & 12
Based on present status establish expected quantity and time of next matte requirement for each converter ISTAT = 1, 15
1
15
Write first matte requirements of converters and present matte supply
Produce forecast of matte requirements to the end of each converter cycle, record expected start of copper stages for each
Subroutine CVFORC
Write forecast if sense switch 8 set
Select and order requirements within next 70 minutes
Subroutine TCSORT
Write expected start of Cu-stages
Delay or advance start of Cu-stages to get proper spacing of converter cycles
Allocate matte to converters for next hour based on the expected availability of matte. Store allocation in MUL
Subroutine MATTE
RETURN

7.5 RESULTS OF THE AUTOMATIC MODEL CONTROL

The smelter model was run in the automatic mode and the response of the controlling algorithms was observed. After removing initial errors the contolling program reached a stage where actions taken were logically correct, but not satisfactory under the given circumstances.

Decision rules were improved to account for this. A good example of this is the selection of a crane for a particular task. In practice this decision is based on the present and future expected availability of a crane and its possible present and future interference with other cranes. The algorithm for selecting the best crane for a particular task was based initially on a simple decision rule and then gradually refined. Shortcomings of the algorithm were visible when cranes selection during simulation was poor. Knowing the exact status of the system at the time, the algorithm could be modified to account for additional variables.

The assignment of cranes to particular tasks is of great importance to achieve efficient smelter operation, and was given much attention. Below the stages in developing crane selection rules are given as the level of sophistication increased.

1. Based on location of crane and task only.
2. Based also on breakdown state of cranes.
3. Based also on the availability of a crane, i.e. if (1) and (2) satisfied but the crane is occupied for some time another choice is made.
4. Based also on the duration of present task. (The duration of some tasks such as collar pulling is so long that any urgent task will not be assigned

to a crane engaged in that task)

If a "two-crane" operation is desired the combination of their states has to be analysed before a selection is made based again on the above rules.

In a similar way other process selection routines were refined to give the most realistic control possible. This process should be continued until a set of decision rules is derived that shows the degree of accuracy required.

It was possible to arrive at a satisfactory level of control by just utilising all the memory space in submodule 9. Because of the need for cross-references the creation of another overlay sub-module was not considered.

While model control in the automatic mode of operation is not yet as sophisticated as in the manual operation the automatic operation does achieve the same level of production. This is possible by utilising slack capacity of the cranes for example.

Table 7.5.1 compares converter and crane statistics for manual and automatic control of the simulation model. The numerical values presented are averages of separate simulation runs. A comparison of single and two-crane operation is also given which shows the superiority of the single crane operation. Although two-crane operations can save valuable non blowing times of converters a very sophisticated crane selection algorithm would be required to achieve this.

The merit of a particular set of decision algorithms is judged by their ability to reproduce and improve upon a decision procedure of a human model operator. Although it was possible to develop algorithms that approach the level of performance of human control, it is expected that the decision rules for the crane selection, converter selection and forecasting algorithms could yet be improved to give an even better performance. While the logic developed for the decision model is not in a form that would allow direct control of the smelter it was possible to show that the decision processes involved in smelter operations can be rationalised. This is a necessary prerequisite before any further work in this area should be pursued.

Converter and crane statistics		Automatic "single-crane" operation	Automatic "two-crane" operation	Manual Operation
Converter blowing in matte stage	(%)	69	70	69.5
Converter not blowing in matte stage	(%)	31	30	30.5
Crane #1 busy	(%)	66.5	69.5	52.0
Crane #3 busy	(%)	72.5	78.5	58.5
Crane #1 delayed by #3	(%)	15.2	19.9	6.1
Crane #3 delayed by #1	(%)	12.3	13.7	4.7

TABLE 7.5.1 RESULTS OF SIMULATED SMELTER OPERATIONS BY AUTOMATIC CONTROL

8.0 CONCLUSIONS

A new technique to provide solutions to an important class of industrial problems is presented. The ability of interactive simulation for modelling systems of discrete and continuous processes using a hybrid computer was demonstrated by producing a model of a copper smelter system. The copper smelter studied represents a typical example for complex control systems encountered in practice.

The development of the model was based on existing knowledge of processes and readily measurable process variables were used to produce semi-empirical mechanistic models of smelter processes. It was necessary to include the model of a human operator which is achieved by allowing an operator to have control over those decisions in the simulation model that are also exercised by a foreman in the smelter. A display unit of smelter process states provides the major feedback on the progress of simulation. This device proved to be of considerable value in developing realistic submodels of all smelter processes. It provides a meaningful picture and the essential information of the simulation model to the model operator. This aspect was important to operating personnel who were consulted on the accuracy and realistic behaviour of the model. Mainly due to the display unit it is possible for smelter operators to gain full understanding of the model operation within a short period of time. The usefulness of this model as a training tool for operators is recognised and it is expected to be used as such in future.

Since the overall model consists of both discrete submodels which are based on the event structure of the GASP simulation language and analog models which operate in parallel it was possible to develop a model of minute by minute smelter operations which correspond closely to reality. The stochastic nature of several processes was included in the model which improves the accuracy of the system representation, but adds to the difficulty of testing and validating the model.

After careful validation the model was used to improve the model performance, which could thereafter be implemented in the real system. Several areas for improvement are indicated of which one was studied in detail. An increased throughput of the system was predicted based on a different control policy. During plant tests this prediction was substantiated which provided additional confidence in the accuracy of the model. The change in policy has since been adopted in the smelter.

The model as presented is an off-line model w.r.t the real system that operates at a speed up factor of 20:1 and thus does not allow any continuous process monitoring.

One aspect of this work was to study the possible use of an on-line computer control system for the entire smelter which would continuously control smelter processes in an optimal way.

As a first step an automatic control program was developed which performs the decision task of the model operator. The level of sophistication obtained was satisfactory to suggest that on-line control seems feasible given that suitable process measuring devices are available. It is expected that the implementation of such a system could provide the smelter operator with such information that leaves him more time for higher level decisions.

Necessary routine decisions for optimal smelter performance could be suggested by the computerised control system. If the wealth of short term and long term information of the smelter can be made available to a control program in a reliable form then such a program could become a valuable aid in copper smelter or similar operations.

REFERENCES

A-1. Ashour S. and Bindingnavle, S.G., An optimal design of a soaking-pit rolling-mill system, Simulation, June 1972, pp. 207-214.

A-2. Ashour, S., and Vaswani, S.D., A GASP Simulation study of job-shop scheduling, Simulation, January 1972, pp. 1-10.

A-3. Ackoff, R.L., and Sasieni, M.W., Fundamentals of Operations Research, John Wiley & Sons, New York, 1968.

A-4. Anderson, J.N., Reverberatory Furnace and Converter Practice at the Noranda and Gaspe Smelters, Extractive Metallurgy, 1960, pp. 133-158.

A-5. Anderson, J.N. Reverberatory Furnace Practice at Noranda, Journal of Metals, June 1954, pp. 745-758.

A-6. Allen, G.J., Beals, G.C., Fish, C.R. Hewlett, R.F., Pana, M.T., Empirical Models of a Copper Reverberatory Furnace - Preliminary Results, Quarterly of the Colorado School of Mines, pp. 385-415.

B-1. Buxton, J.N., Simulation Programming Languages, North-Holland Publishing Company, Amsterdam, 1968.

B-2. Box, G.E.P., Muller, M.E., A note on the generation of normal deviates, Annals of Mathematical Statistics, Vol. 28, 1958.

B-3. Bowman, E., H., and Fetter, B F., Analysis for Production and Operations Management, Richard D. Irwin, Inc., Illinois, 1968.

B-4. Bekey G.A, and Karplus, W.J., Hybrid Computation, John Wiley & Sons, N.Y., 1968.

C-1. Cox, G.E., and Ralston, D.H., The Application of Computers to Shop Control, The Production Engineer, 1966, pp. 10-23.

C-2. Ceyert, R.M., and March, J.G., A Behavioral Theory of the Firm, Engelwood Cliffs, N.Y., Prentice Hall Inc., 1963.

C-3. Chemical Plant Simulation, McMaster University Publication, Hamilton, March 1969.

C-4. Chatfield, C., Statistics for Technology, Penguin Books, Baltimore, U.S.A., 1970.

D-1. Davies, A.J., A Simulation Model of a Copper Smelter, O.R.S.S.A. Conference, Nov. 1970, pp. 1-10.

D-2. Davies & J.S. Thixton, The Use of Simulation in Evaluating Plant Changes in a Copper Smelter, The Metallurgical Society of AIME, Paper No. A72-84, 1972.

D-3. Dudgeon, E., Nenonen, L.K. Balogh, A., Fowler, P.L., Measurements for Control of a Copper Converter, 1966 Operating Metallurgy Conference, Metallurgical Society of the AIME., Dec., 1966, Philadelphia, Pa, 159-185.

E-1. EAI 680 Reference Handbook, Electronic Associates, Inc., 1970.

F-1. Foreman, J.H., Copper Smelter Process Computer Applications, 1965 Operating Metallurgical Conference, Metallurgical Society of the AIME, Dec. 1965, Pittsburgh, Pa, pp. 465-487.

F-2. Foreman, J.H., Mathematical Model for Copper Converter Control, Journal of Metals, June, 1965, pp. 616-621.

F-3. Fabrycky, W.J., and Torgersen, P.E. Operations Economy - Industrial Applications of Operations, Prentice Hall, Inc., N.Y. 1966.

G-1. GPSS, General Purpose Simulation System, IBM, Technical Publications Department.

G-2. Graefe, U.P., Memory Overlay-Program Note S-037, National Research Council of Canada, May, 1972.

G-3. Gibson, N., Smelting Developments at Rhokana Corporation Limited, Technical Proceed. 7th Mining and Metallurgical Congress, 1961.

G-4. Graefe, U.P., Private Communication, 1973.

G-5. Gordon, G., System Simulation, Prentice-Hall, Inc., Englewood Cliffs, N.Y.

H-1. Harris, I.J., Development of a Mathematical Model for Reverberatory Furnace Heat Transfer, Mineral Processing & Extractive Metallurgy, Vol. 81, 1972,104-108.

H-2. Hyndman, D.E., Analog and Hybrid Computing, Pergamon Press, Oxford, 1970.

I-1. IBM Corporation, Random Number Generation and Testing, Form C20-8011, New York, 1959.

I-2. Itakura, K., Nagano, T., and Sasakura, J., Converter Slag Flotation - its Effects on Copper Reverberatory Smelting Process, Journal of Metals, Vo. 21, No. 7, July 1969, pp. 30-34.

K-1. Kiviat, P.J., Villaneuva, R. and .Aarkowitz, H.M., The Simscript II Programming Language, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1968.

K-2. Krause W., Prahl, H. and Eisenaecher, Probleme der Prozessteuerung beim Verblasen von Kupferrohstein im Konverter, Neue Huette, Vol. 16, No. 4, April, 1971, pp. 206-212.

K-3. Kochenburger, R.J., Computer Simulation of Dynamic Systems, Prentice-Hall, Inc., New Jersey, 1972.

L-1. Lippitt, D.L., Simulation of the Basic Oxygen Fumace, Metals Engineering Quarterly, May 1964 , pp. 58-61.

L-2. Love, R.D. and Taha, H.A., A Program Simulating the Operation of an Overhead Crane, Joumal of Industrial Engineering, Vol. 16, No.2, March 1965, pp. 87-93.

M-1. Maisel, H. and Gnugnoli, G. Simulation of Discrete Stochastic Systems, Science Research Associates, Inc. 1972.

M-2. Martin, F.F., Computer Modeling and Simulation, John Wiley & Sons, New York, 1968.

M-3. McKerrow, G.C., The Gaspe Smelter, Joumal of Metals, Vol. 9, Sept. 1957,1114-1117

M-4. Mihram, G.A., Simulation: Statistical Foundations and Methology, Academic Press, N.Y., 1972.

M-5. Mihram, G.A., Some Practical Aspects of the Verification and Validation of Simulation Models, Operational Research Quarterly, Vol. 1, 1972, pp. 17-29.

M-6. Martin, F.F., Computer Modelling & Simulation, John Wiley, N.Y., 1968.

M-7. McMillan, C., Gonzales, R.F., System Analysis, A Computer Approach to Decision Models, R.D. Irwin Inc., Illinois, 1969.

N-1. Nenonen, L.K., and Graefe, U., A project on Smelter Scheduling, Laboratory Technical Report - CS-63, Dec. 1971.

N-2. Nenonen, L.K., and Pagurek, B., Conjugate Gradient Optimisation applied to a Copper Converter Model, Automatica., Vol. 5, Pergamon Press, 1969, 801-810.

N-3. Niemi, A., and Koskinen, T., Simulation of the Copper Converter, Transactions (Mineral Processing and Extractive Metallurgy), Vol.77, Dec. 1968, pp. 201-208.

N-4. Nenonen, L.K. Private Communication, 1973.

N-5. Naylor, T.H., Balintfy, J.L., Burdick, D.S., Kong Chu, Computer Simulation Techniques, John Wiley & Sons, Inc., N.Y., 1966.

O-1. Operations Analysis, Process Design, Process Control in the Canadian Mineral Recovery and Metal Production Industries, May 1972 Conference, N.R.C. Associate Committee on Automatic Control.

P-1. Pritsker, A.B. and Kiviat, J., Simulation with GASP 11, Prentice-Hall, Inc., New Jersey, 1969.

S-1. Schuerger, T.R., and Tiskus, J.R., On-Line Process Control Computers in the Steel Industry, Metals Engineering Quarterly, August 1971, pp. 1-4.

S-2. Shapiro, P., Choosing a Simulation System, Electronics, Sept. 1968, pp. 91-94.

S-3. Schmidt, J.W. and Taylor, R.E., Simulation and Analysis of Industrial Systems, Richard D. Irwin, Inc. 1970.

Leaf 159 omitted in page numbering.

S-4. Svrcek, W.Y., and Sandholm, D.P., An Experience in the Selection of a Computer Simulation Technique, Simulation, Dec. 1972, pp. 245-246.

S-5. Simon, H.A., The New Science of Management Decisions, Harper and Bros. N.Y., 1960.

S-6. Sayre, K.M., and Crosson, F.J., The Modelling of Mind: Computers and Intelligence, Simon and Schuster, New York, 1968.

T-1. Templeton, F.E. and Hankley, W.J., Dynamic Optimal Control of a Process with Discrete and Continuous Decision Variables, 1970, International Symposium, Cdn. Inst. Mining & Metallurgy, Montreal, P.Q., 1971, pp.1-44.

T-2. Tocher, K.D., Review of Simulation Languages, Operations Research Quarterly, Vol. 16, No.2, 1964, pp. 189-217.

T-3. Taylor, D.W., Decision Making and Problem Solving, in Handbook of Organizations, J.G. March (ed), Rand McNally and Co., Chicago, 1965.

V-1. Van Horn, R., Valdiation, Symposium on the Design of Computer Simulation Experiments (Editor T.H. Naylor), Duke University Press 1969.

W-1. Wigan, M.R., The Fitting, Calibration and Validation of Simulation Models, Simulation, May 1972, pp. 188-192.

APPENDIX A

CONTROL AND ARRANGEMENT OF OVERALL SIMULATION MODEL

A.1 MODES OF OPERATION

The mechanism of hybrid simulation as implemented in this study is described here. Together with Appendices F and G it provides a detailed overall picture of the simulation model.

After initialising the model at the start of a simulation run control of the simulation is maintained in subroutine GASP. A general flowchart of subroutine GASP is given in Figure A-1

The two possible modes of operation of the simulation model (OPERATE and HOLD) are clearly seen on this diagram. (The INITIAL CONDITION mode is only used to set up the analog program before the simulation is commenced).

The area enclosed by dotted lines corresponds to the OPERATE mode while all other operations are performed in the HOLD mode. The function of the controlling digital program in the OPERATE mode is that of checking the logical links between the analog and the digital computer by continuously reading the logical states of sense lines and comparators. This is shown on Figure A-2 which shows the detailed flowchart of subroutine GASP. In the OPERATE - mode the analog computer executes all the functions of the analog submodels including those of the timing circuit and the time display.

When the analog computer is placed in the HOLD mode the remaining sections of the program are executed including all other subprograms to which the program might branch. Simulation time will thus only advance in the OPERATE mode, during which time only continuous variables can change. Although the HOLD mode can be described as being one in which all analog states are frozen it does not mean that no state change of the models occur.

In the HOLD mode all discrete changes to analog as well as digital variables occur which gives it an equally important part in the execution of the simulation programme.

In the following sections the two modes of operation will be discussed.

A.2 THE OPERATE MODE

The analog converter and crane submodels are treated separately in the following Appendices and will not be discussed further here. During the OPERATE mode these submodels are executed in parallel and represent the dynamic aspects of the model. There are two further analog circuits, the timing circuit and the time display circuit. Figure A-3 shows the circuit diagram of the timing circuit. Its purpose is that of linking the discrete event simulation to the analog simulation. At the start of an OPERATE period comparator C39 is set to a positive value TREL which corresponds to the time interval to the next discrete event in the time file. (The time file contains all discrete events in chronological order whose execution produces the discrete event simulation). Integrator A35 integrates during the OPERATE period until the condition TREL = TAN is reached when comparator C39 changes its logic state which resets A35 to the INITIAL CONDITION. This condition is sensed by the control programme which places the computer in the HOLD mode while executing the discrete event before removing the next event from the time file and resetting the timing circuit. The potentiometer P35 is set when initialising the programme to give the correct rate of integration TTC. With a speed-up factor BETA = 20 and a maximum time interval TIMEX = 60 min the scaling of the circuit is complete, and the integration is performed according to:

$$\frac{TAN}{TIMEX} = \int_0^t TTC\,dt + 0 \qquad \ldots(1)$$

where

$$TTC = \frac{BETA}{TIMEX*60}.$$

Figure A-4 shows the time display circuit. It consists of a single integrator whose scaling is similar to that of the timing circuit described above. The potentiometer setting is multiplied

TTC = 0.6 TTC

to give a 1/10 volt/min reading for the elapsed simulation time. The integrator output is displayed on a digital voltmeter as minutes. Every hour the integrator is reset to its INITIAL CONDITION in a time event (event 59) by setting control line 6 high. The hour is also updated in that event. The hour reading is scaled and its value is set on D/A - 10 which is displayed on another digital voltmeter. By selecting suitable scales on the voltmeters a simulation time reading correct to the nearest minute is obtained.

An OPERATE period can be terminated by one of four possible conditions.

1. The timing circuit reaches TREL
2. Converter submodel reaches an end point
3. Crane submodel reaches the end of a task
4. Model Operator interrupts simulation. (Usually to issue a command to a particular submodel)

All four conditions will be sensed by the controlling program and the operations that follow in each case are discussed in the next section.

If an OPERATE period is terminated by a condition other than a time event the simulation at that time is not given by TNOW and is computed by FUNCTION routine TIME(TLAST). The intermediate time TAN is read from the timing circuit and converted to give the present simulation time.

i.e. TIME = TLAST + TAN*TIMEX

where TLAST = TNOW only when a time event has just occurred

A.3 HOLD-MODE

Referring to the lower section of Figure A-1 each of the four possible conditions resulting in a HOLD call will cause the programme to branch to a different section for further execution.

The first condition corresponding to a time event will now be discussed. The program branches to statement 99 in Figure A-2 where the event given by JEVNT is executed at time TNOW through a call to subroutine EVENTS. Based on the event number a section of the program is called from disc to execute the particular event. (Execution of an event involves executing a section of a submodel which corresponds to the event number). The arrangement of submodules for the memory overlay technique is given later. After execution of the time event which is described by 4 ATTRIBUTES

ATRIB(1)	=	Event Code 1 to 999
ATRIB(2)	=	Time at which event occurs
ATRIB(3)	=	Other characteristics of the event
ATRIB(4)	=	Other characteristics of the event

the next time event in line is removed from the time file by subroutine RMOVE. Subroutine SETIME is shown in Figure A-5 which is then called to reset the timing circuit shown on Figure A-3. The four attributes of the next event are stored in different variables than ATRIB(1 to 4) since these variables are reset when other events occur before TREL has elapsed.

The GASP-Simulation language provides a choice of scale for variables used in the fixed point filing arrays. That is values of ATRIB are stored with one decimal point if a SCALE = 10 is used, which was the choice in this model. With the minutes as units this gives a timing accuracy in the order of six seconds. A larger time scale such as 100 was not required, which would also have limited the length of simulation to less than six hours. With the present choice it is possible to simulate at least six consecutive 8-hour shifts without exceeding 32767/10 minutes. (32767 being the largest integer number possible on the computer)

This period of 48 hours of simulation might seem short for a simulation run, but since the model reproduces all detailed operations and is run by an operator it would take a minimum of six real-time hours toperform this simulation run. If in future a longer simulation period was required possibly by automatic operation it would be quite simple to extend the simulation period. Another temporary time file could be created in which all events to occur later than 48 hours could be filed - starting again from zero time. When the last event before 48 hours was removed from the time file all events could be transferred back to the time file and the simulation could go on for several 48 hour intervals.

When calculating statistics based on the total simulation time previous 48 hour intervals have to be accounted for.

If no other event had occurred (such as a crane reaching its target) at the same time then the program would branch to statement 72 and place the analog computer back into the OPERATE mode. Excluding other interruptions this process could continue and would represent a real-time discrete event simulation with a 20:1 speed up factor.

If the second condition end point of a converter, became true the corresponding discrete event IARG is executed. This is done by calling subroutine EVNTS in the same way as described above whereafter the OPERATE mode continues. Reference can be made to Appendix D, where details of the converter submodels are given. If the third condition arose, when a crane has completed a task, the program branches to statements 199, 299, 100 depending on the crane number in Figure A-2 on the second page of the flowchart.

The state variable JWT (NCR) $\neq 0$ is an indication that a crane cannot proceed with its task, i.e. it is delayed by another crane. In the case that the crane is not delayed the event to be executed at the end of a crane task is again executed by a call to subroutine EVNTS. Unless there is no other event in the crane file the next crane event is removed from the particular crane file, crane interferences are resolved and any delayed cranes are reset to continue before returning to the OPERATE MODE. The details of these operations are discussed in Appendix B.

In subroutine GASP statistics on crane activities are also recorded by summing the "busy" and "delayed" periods of time in TCRAN(NCR) and SP(NCR).

The fourth condition to place the analog computer in the HOLD mode is given when a push button on the digital computer console is pressed which sets sense switch 1. In this case the control of the program goes to statement 50. A call to subroutine RD is made where any number of operator commands are accepted from the teletypewriter, decoded and translated into events and filed in the crane and/or time files. This section is described below. After the last input command the first crane events are removed if any cranes were idle and the program returns to the operate mode.

If there were no other source of time events other than time events themselves, then whenever the timing circuit was set it would operate until the required time interval between events had elapsed. Since both operator commands and crane activities can cause filing of time events it is necessary to search the time file whenever one of the two conditions occur. This is done by subroutine TSTFL (Figure A-6) which causes the timing circuit to be reset if a time event was to occur before the one previously scheduled. This ensures that all events are executed in the correct time sequence. During a HOLD stage any combination of the four conditions leading to the HOLD call can occur and will be executed consecutively.

The variable MODUL is used when operating the model in the automatic-mode only. If during simulation a command is to be generated internally or if forecasting of material requirements was necessary then this is indicated by MODUL = 4. That is before the controlling program returns to the OPERATE mode the program branches to the section where commands are internally generated or the forecasting is performed. Subroutine TKODE is called whenever a crane event is removed from a file. In the automatic mode TKODE is used to synchronise tasks of two cranes as discussed in Chapter 8.

A-4 PROGRAM DIVISION FOR MEMORY OVERLAY

Table A-1 lists the twelve submodules of the program as used for the memory overlay. A submodule refers to a section of the program that is characterised by a file name and if not in core it is stored on one of two discs available with the EAI 640 digital computer.

The program sections in each file do not necessarily refer to a particular subprogram, but are chosen in such a way that no reference from one submodel to another is necessary.

At any time only the main module is in core with any one of the submodules. In this way it is possible to utilise the 16K memory for a program of much larger size.

The main module consists of a short main program (see Appendix G) which first loads the submodule #1 to initialise the model and then branches to subroutine GASP from where the simulation run is then controlled. If any call to a submodule is made from the main module this is done by calling subroutine EXEC(NAME) with the corresponding submodule number as an argument. Every submodule has the same name subroutine DISC which is called by this routine.

A detailed description of the overlay system is given in Ref. (G-2)

Referring to Table A-1 submodules 2, 3, 6, 7, 8 thus contain digital submodels and digital parts of the digital-analog models.

Submodule 4 is used for interpreting and generating operator commands which are described in the next section.

Submodule 5 provides information on model states by writing variables and statistics when requested. No further description of this submodel is given, the listing in Appendix G is self-explanatory.

Submodules 9, 10, 11, 12 are used only if the model is run in the automatic mode.

Submodule 1 is called into core only once at the beginning of a simulation run to initialise a run. Model variables requiring initialisation are contained in the labled COMMON blocks BLOK1 and BLOK2 (see Appendix F). Their values are read in from data cards in submodule one together with data cards as required for the GASP discrete event simulation. These include initial events to start certain smelter processes. At the end of Appendix G a typical set of data cards is shown.

Subroutine DATAN initialises all GASP variables, and analog models are initialised by setting corresponding analog components. Based on initial conditions the display unit is initialised by setting all light indicators.

A.5 COMMAND INTERPRETATION AND TRANSLATION

The program for command interpretation and translation was developed by Dr. U.P. Graefe and the description is taken from Reference (G-4).

Operator commands can be issued through the teletypewriter by pressing senseswitch 1 until ENTER is typed on the teletype. Thereafter a command can be entered, After the carriage return key is pressed, the teletype will prompt for the next command by typing ENTER etc. If a typing error is made in a command, the command may be cancelled by issuing a CC command immediately after the faulty command. The command // will terminate command inputs. Except for the CC and // commands, the commands consist of two two character command syllables followed by variable lists of various length, depending upon the type of command.

A list of acceptable commands is shown in Table A-2.

Translation of operator commands is performed by subroutine RD and the routines in submodule 4 (See Appendix G). As soon as the operator pushes sense switch 1 on the digital computer console the analog computer is put in the HOLD mode and subroutine RD is called.

Subroutine RD will prompt the operator for input by typing ENTER. Upon entering commands submodule 4 is loaded in order to gain access to the interpreting routines. When entering several commands in sequence a one command delay will be noticed, since a command will only be interpreted after the succeeding command has been entered. This is to allow for correcting a faulty command just after it has been entered by following up the faulty command by the CC command.

A flow chart for subroutine RD is shown in Figure A-7. The command syllables of the command to be translated are sent to submodule 4 via the variables IDIX and ID2X in common block ARG, while the other command variables are transmitted in array IBUFX which is part of common block BLOK2.

With the exception of the FIX, STOP, OIL, INFO, LITE, GW, GOWT, REFN and CAST command, subroutine DISK assigns two codes to each command; a ISEQ code cnd a ICMD code. The ISEQ code is equal to 1 for pick commands, equal to 2 for dump commands and equal to 3 for pick-and-dump commands. The ICMND code is a number presently between 1 and 13, which determines the type of material to be picked up or dumped. Table A-3 shows the abbreviations used for components and materials which are marked by an asterisc in Table A-1.

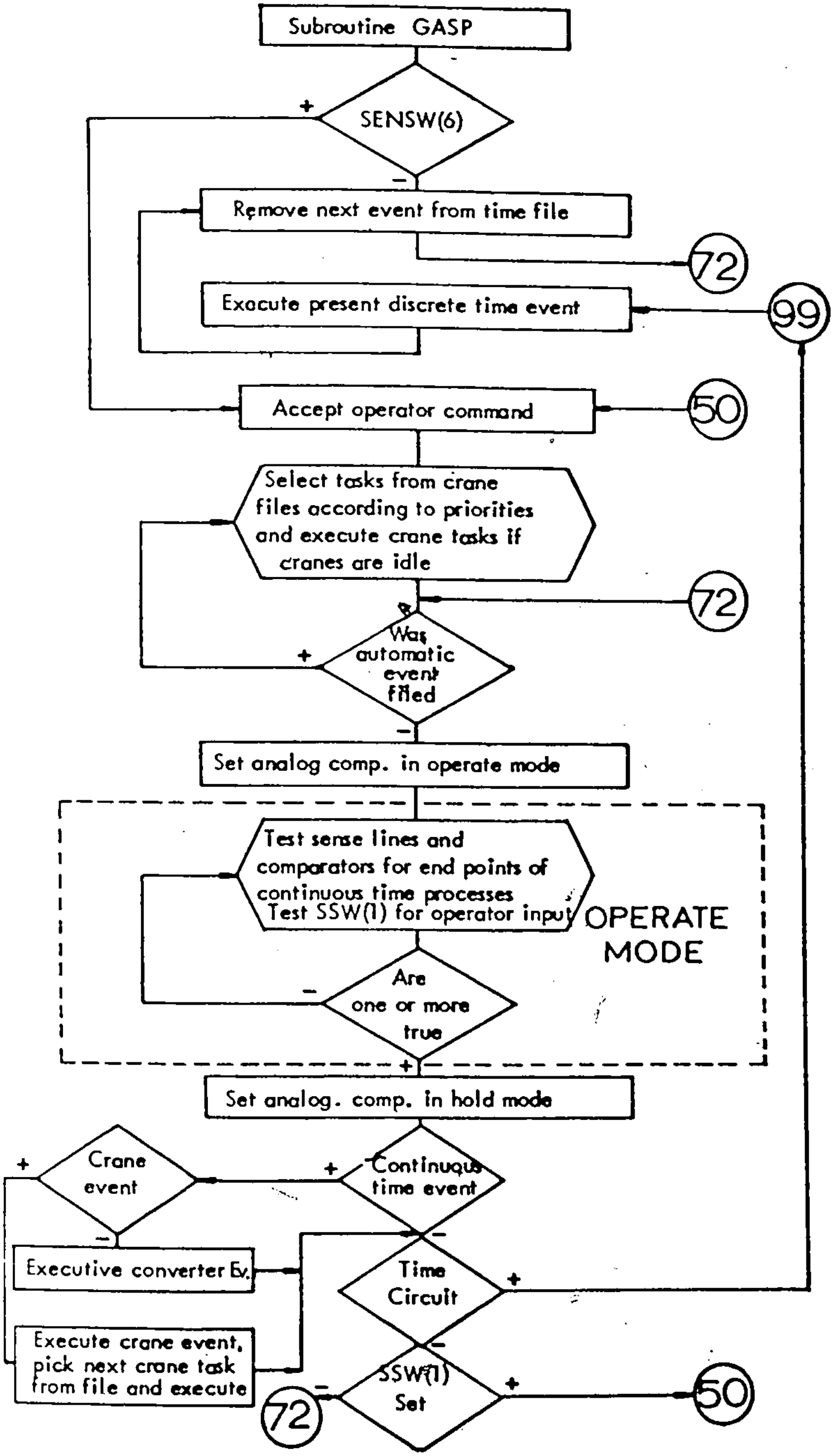
A command is executable if it is a valid command involving the pick-up of material which is available at the specified location, or the dumping of a material into a device that is ready to receive the specified material.

The FIX, STOP, OIL, LITE, GW, GOWT, REFN and CAST commands are executed directly by subroutine DISK. For the INFO command the component code for the device in question is stored in variable IX of the common block ARG, which causes submodule 5 to be loaded and executed as soon as control returns to subroutine RD.

In subroutine CMNDG, the quantity Q of material being picked up is checked. If it was specified as being larger than 2.0, it is set to 2.0. A zero value for Q, TW and IPR - that is quantity of material, waiting time and priority respectively - is meant to signify that default values are to be substituted for them. The default value for Q is 1.0. Default values for TW and IPR depend on the command, and are picked out of the two dimensional arrays IWTNGT and IPRA respectively, using codes ISEQ and ICMND as array indices. The stored default waiting times are given in seconds, and hence require conversion to minutes.

Following this an event code is picked and the event filed in the appropriate crane file using subroutine GWAIT. If a command is issued to a crane which is not operational at the time, the event is not filed, but a message - CRN "CRANE "DOWN TILL "TIME" - is typed out instead.

FIGURE A-1 General Flowchart of GASP


Subroutine GASP
SENSW(6)
+
-
Remove next event from time file
72
Execute present discrete time event
99
Accept operator command
50
Select tasks from crane files according to priorities and execute crane tasks if cranes are idle
72
Was automatic event filed
+
-
Set analog comp. in operate mode
Test sense lines and comparators for end points of continuous time processes
Test SSW(1) for operator input
OPERATE MODE
Are one or more true
-
+
Set analog. comp. in hold mode
Crane event
+
-
Continuous time event
+
-
Executive converter Ev.
Execute crane event, pick next crane task from file and execute
Time Circuit
+
-
SSW(1) Set
-
+
72
50

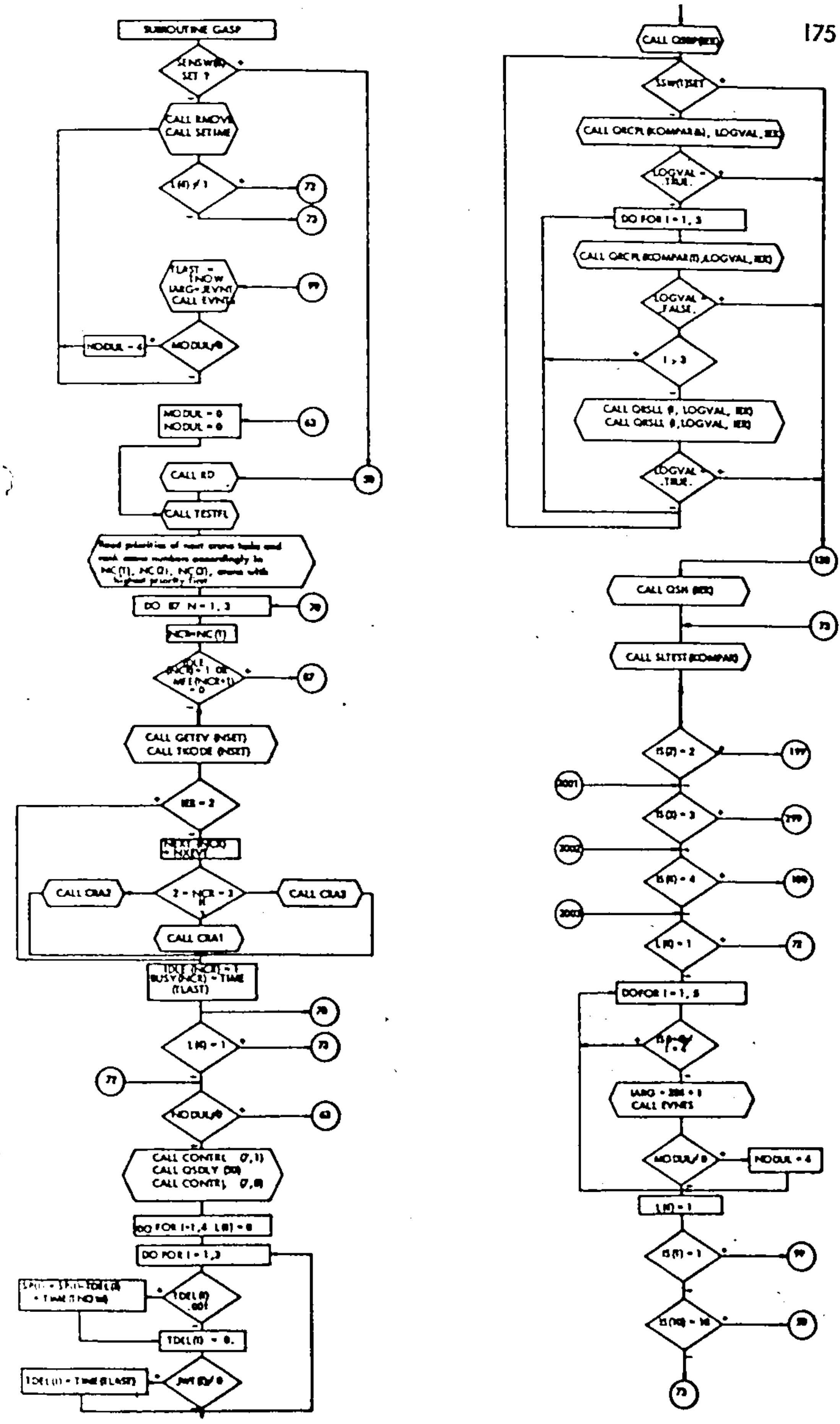
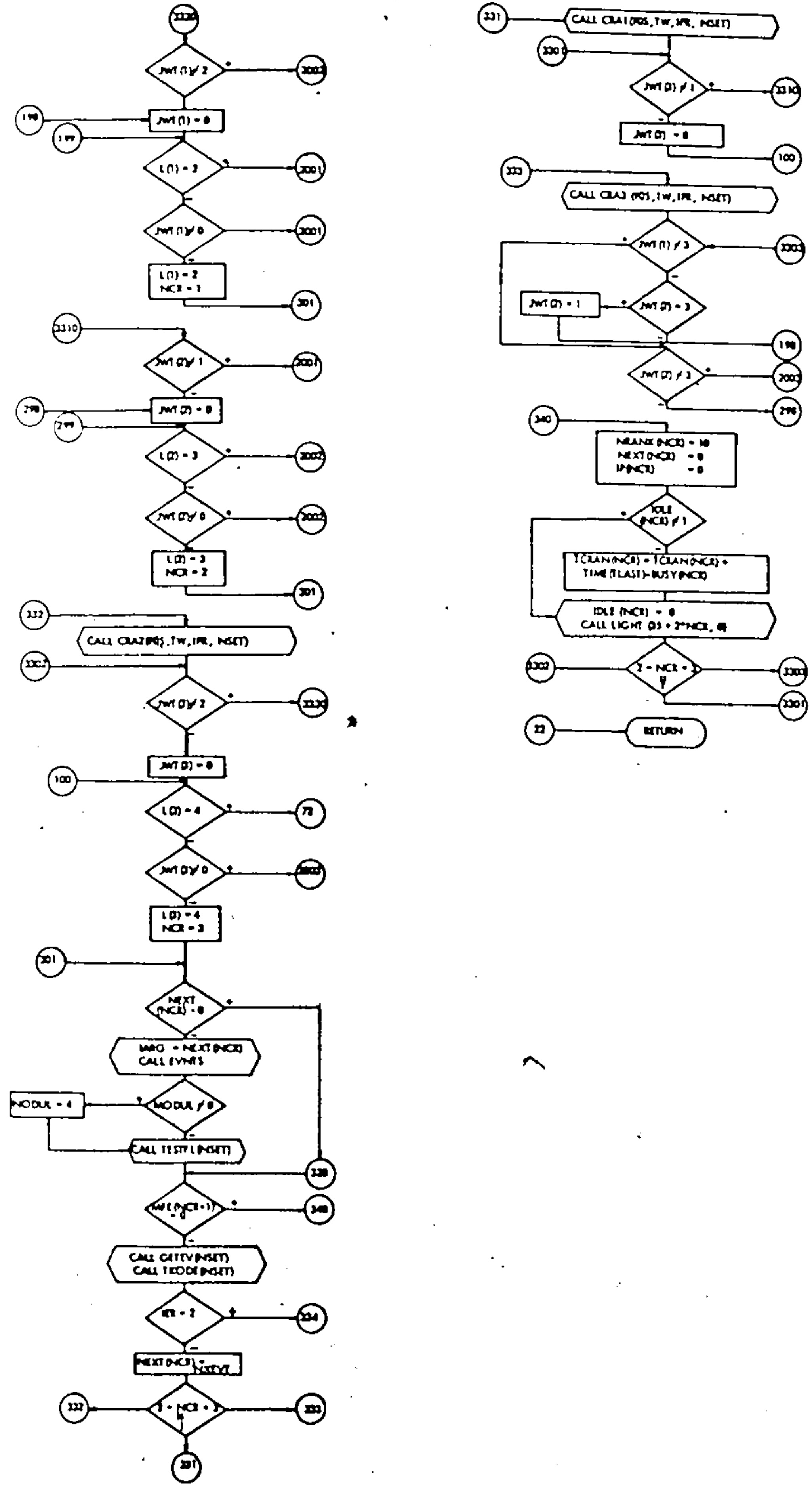


FIGURE A-2 DETAILED FLOWCHART OF SUBR. GASP


3330
JWT (1) ≠ 2
2002
198
JWT (1) = 0
199
L(1) = 2
2001
JWT (1) ≠ 0
2001
L(1) = 2
NCR = 1
301
3310
JWT (2) ≠ 1
2001
298
JWT (2) = 0
299
L(2) = 3
2002
JWT (2) ≠ 0
2002
L(2) = 3
NCR = 2
301
332
CALL CRA2(POS,TW,IPR, NSET)
3302
JWT (3) ≠ 2
3330
JWT (3) = 0
100
L(3) = 4
78
JWT (3) ≠ 0
2003
L(3) = 4
NCR = 3
301
NEXT (NCR) = 0
IARG = NEXT (NCR)
CALL EVNTS
NODUL = 4
MODUL ≠ 0
CALL TESTFL (NSET)
338
IWE (NCR+1) = 0
340
CALL GETEV (NSET)
CALL TKODE (NSET)
IER = 2
334
NEXT (NCR) = NXTEVT
332
2 = NCR = 3
333
331
331
CALL CRA1(POS,TW,IPR, NSET)
3301
JWT (3) ≠ 1
3310
JWT (3) = 0
100
333
CALL CRA3 (POS,TW,IPR, NSET)
JWT (1) ≠ 3
3303
JWT (2) = 1
JWT (2) = 3
198
JWT (2) ≠ 3
2003
298
340
NRANK (NCR) = 10
NEXT (NCR) = 0
IP(NCR) = 0
IDLE (NCR) ≠ 1
TCRAN (NCR) = TCRAN (NCR) +
TIME (TLAST)–BUSY (NCR)
IDLE (NCR) = 0
CALL LIGHT (35 + 2*NCR, 0)
3302
2 = NCR = 3
3303
3301
22
RETURN

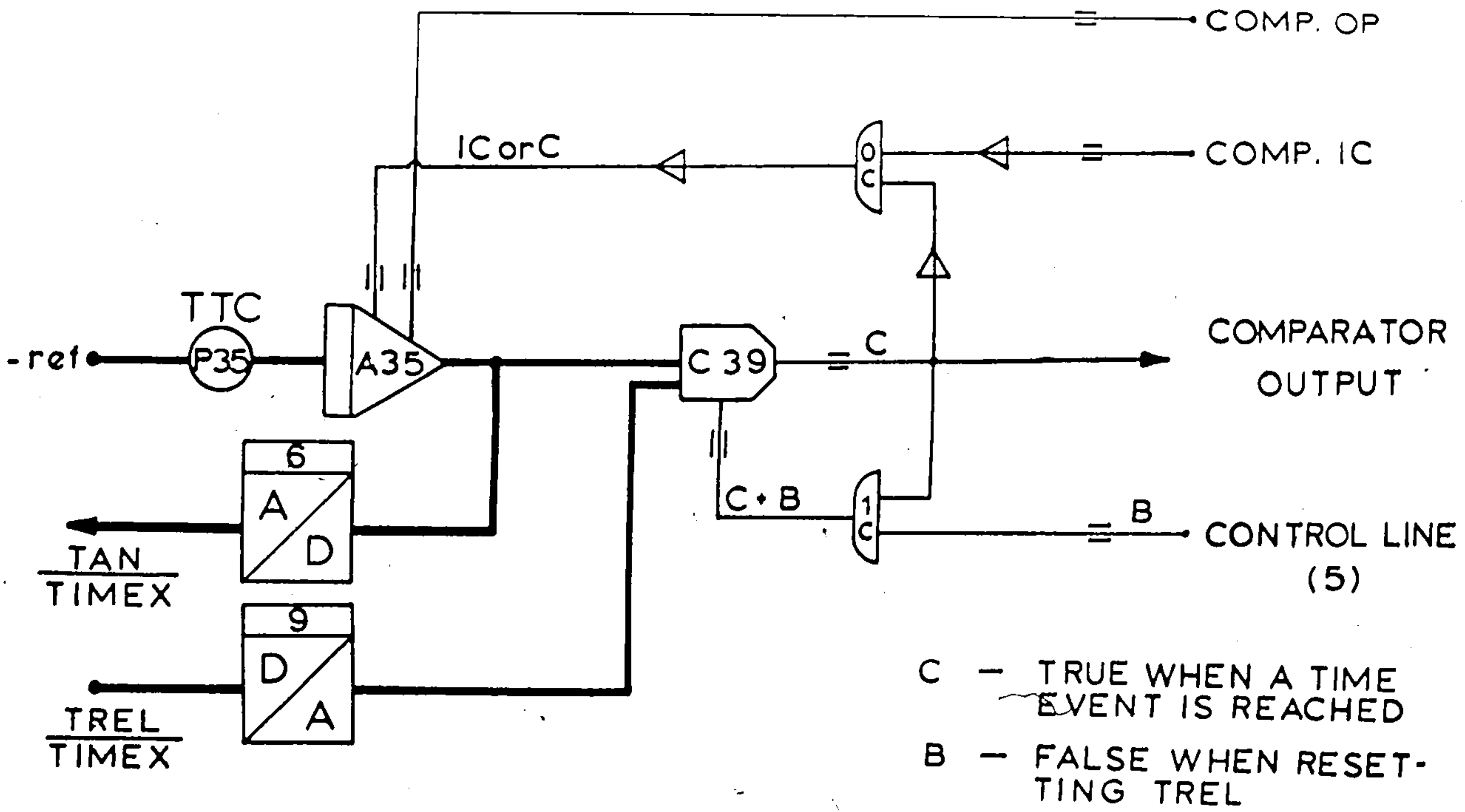


FIGURE A-3 TIME CIRCUIT LINKING DISCRETE DIGITAL TO CONTINUOUS ANALOG SIMULATION

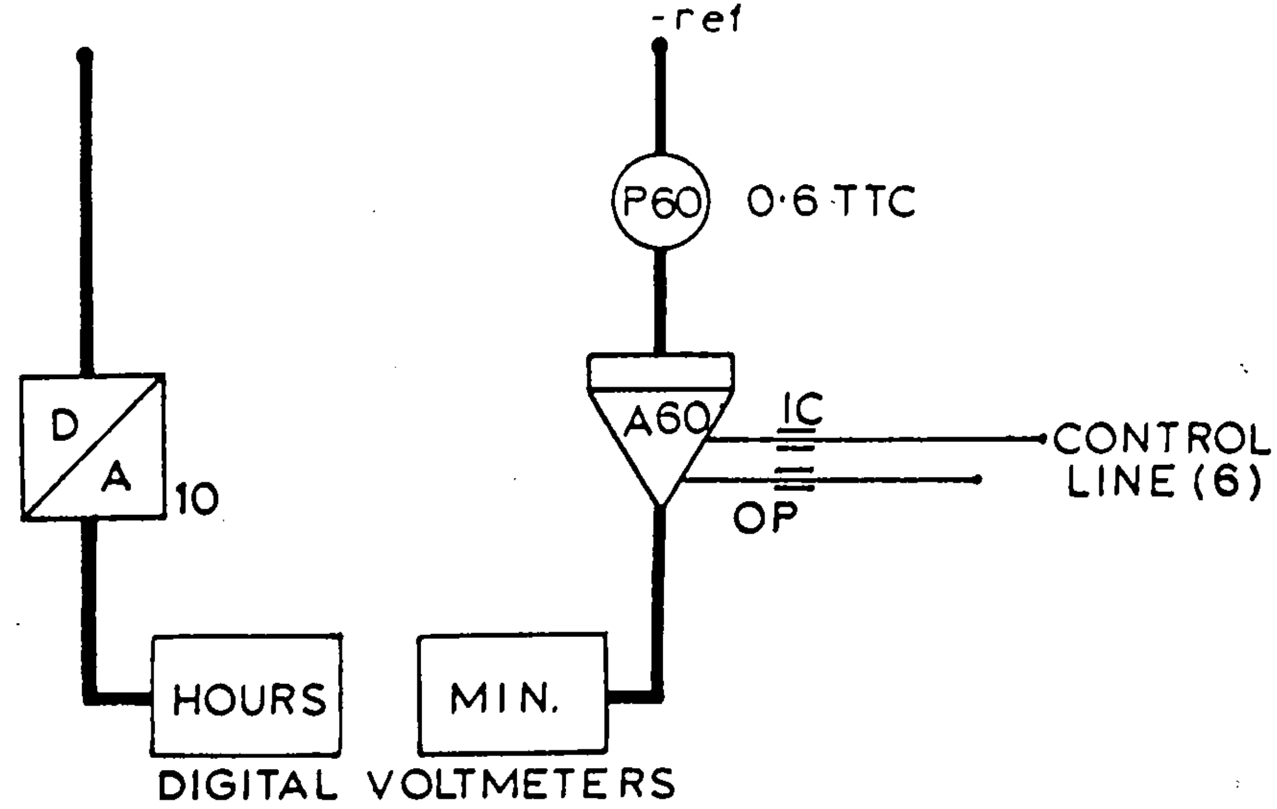


FIGURE A-4 TIME DISPLAY CIRCUIT

FIGURE A-5 FLOWCHART FOR SUBR. SETIME

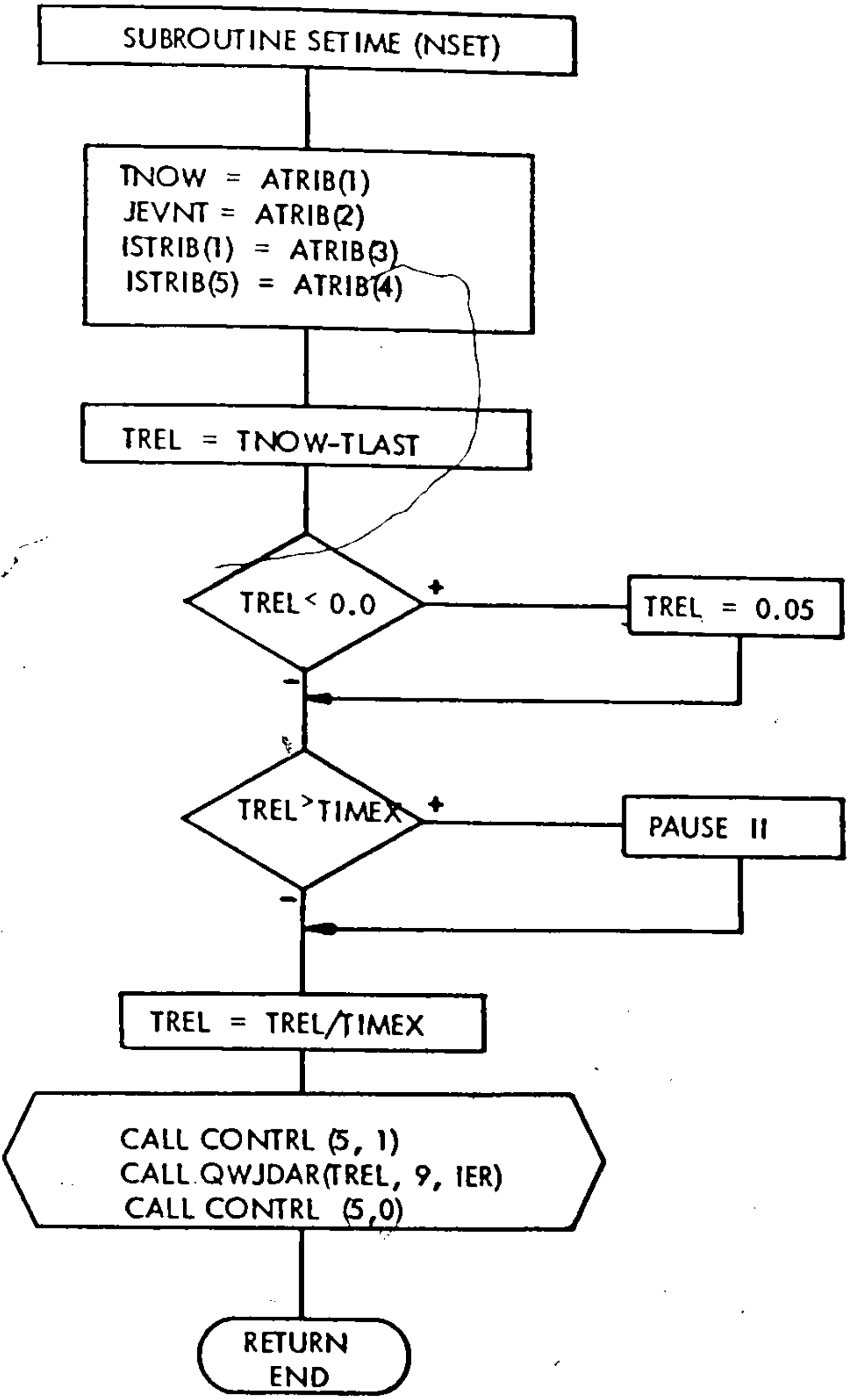
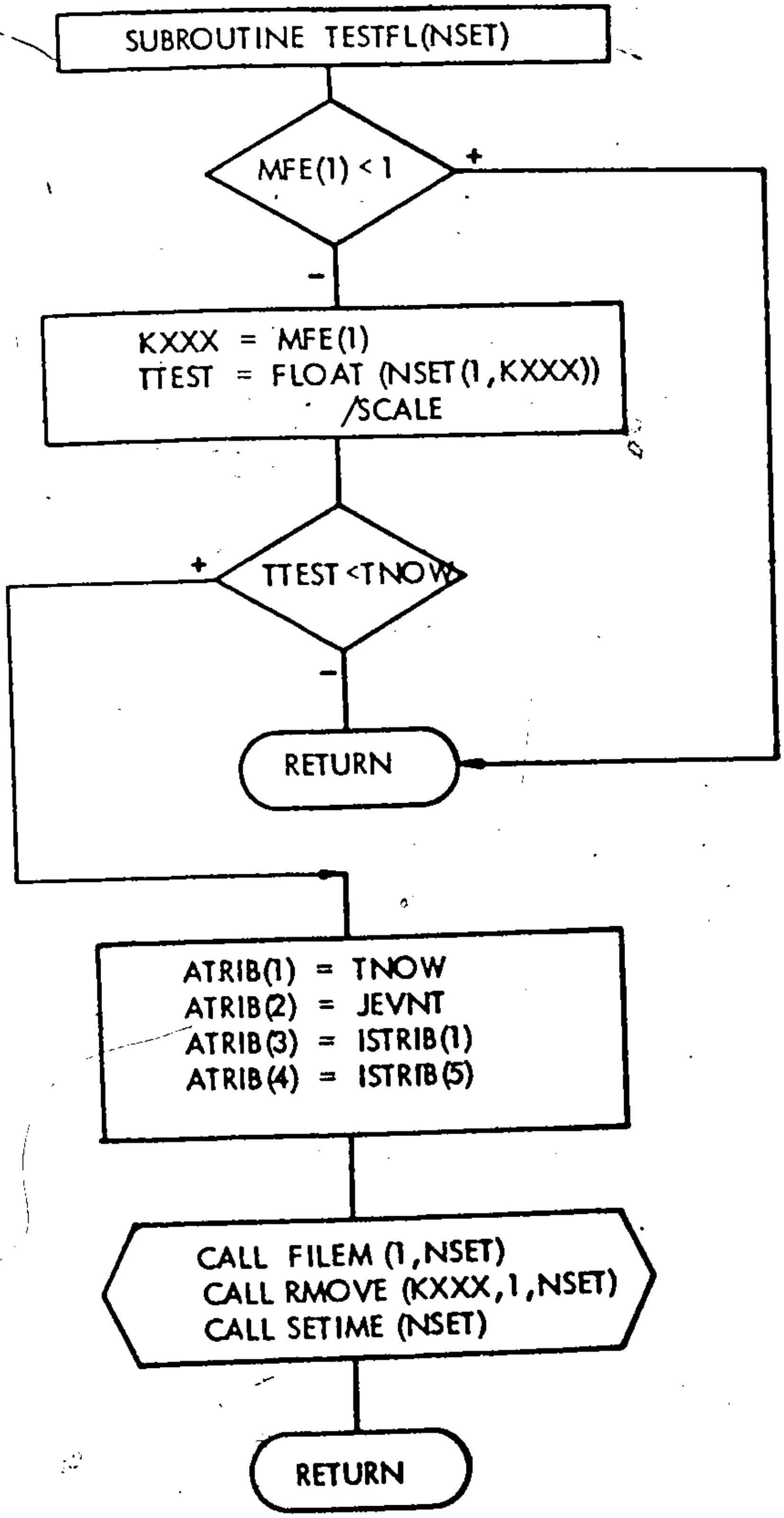


FIGURE A-6 FLOWCHART FOR SUBR. TSTFL



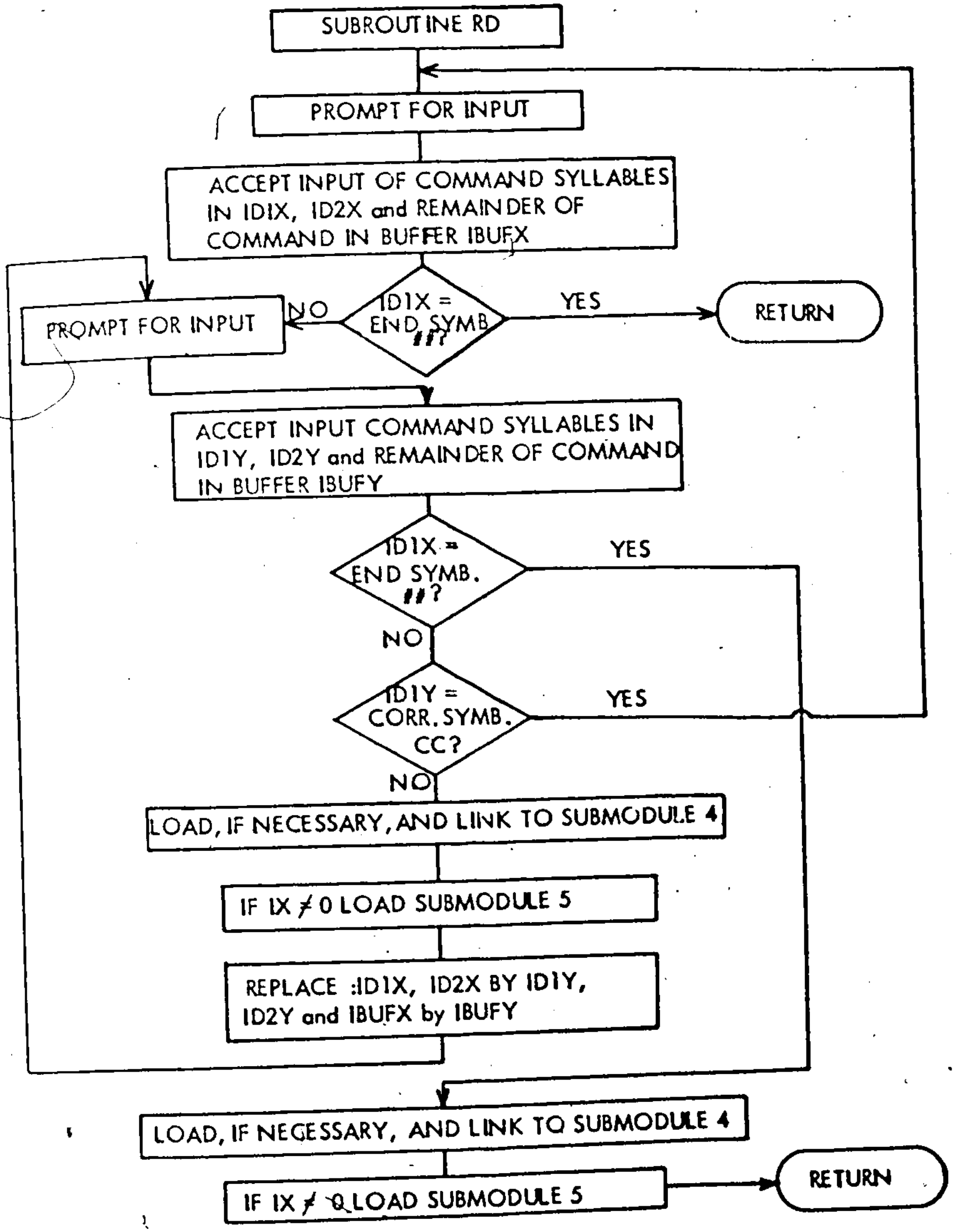
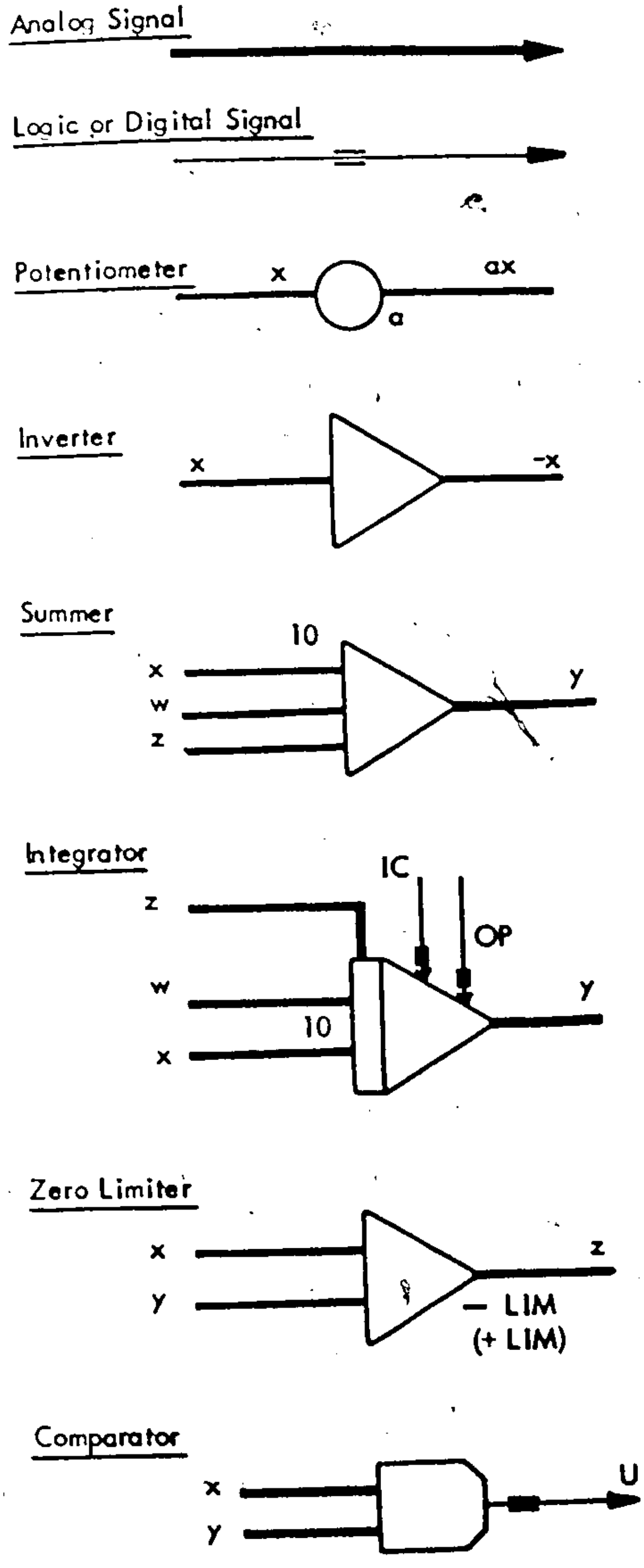

SUBROUTINE RD
PROMPT FOR INPUT
ACCEPT INPUT OF COMMAND SYLLABLES IN ID1X, ID2X and REMAINDER OF COMMAND IN BUFFER IBUFX
ID1X = END SYMB ##?
YES
RETURN
NO
PROMPT FOR INPUT
ACCEPT INPUT COMMAND SYLLABLES IN ID1Y, ID2Y and REMAINDER OF COMMAND IN BUFFER IBUFY
ID1X = END SYMB. ##?
YES
NO
ID1Y = CORR. SYMB. CC?
YES
NO
LOAD, IF NECESSARY, AND LINK TO SUBMODULE 4
IF IX ≠ 0 LOAD SUBMODULE 5
REPLACE :ID1X, ID2X BY ID1Y, ID2Y and IBUFX by IBUFY
LOAD, IF NECESSARY, AND LINK TO SUBMODULE 4
IF IX ≠ 0 LOAD SUBMODULE 5
RETURN

FIGURE A—7 FLOWCHART FOR SUBR. RD

FIGURE A-8 HYBRID COMPUTING COMPONENTS



$a < 1$

$y = (w + z + 10x)$

$y = -z$ (INITIAL CONDITION)

$y = -z - k \int_0 (w+10x)dt$ (OPERATE)

$y = -z - k \int_0^{1} (w+10x)dt$ (HOLD)

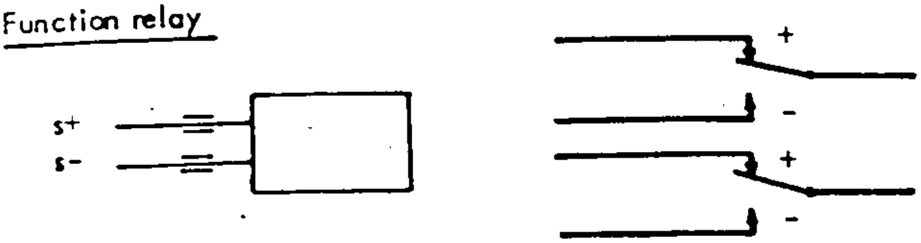
k = Analog Time Scale

$z = -(x+y)$ for $z > 0$ (+LIM)
or for $z < 0$ (-LIM)

$z = 0$ for $z < 0$ (+LIM)
for $z > 0$ (-LIM)

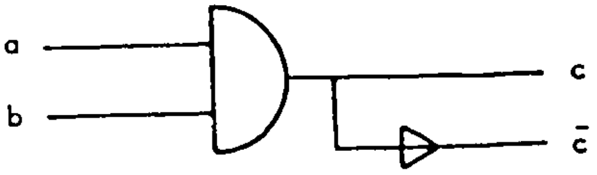
U true for $(x + y) > 0$

Function relay



Relay under logic control with manual override

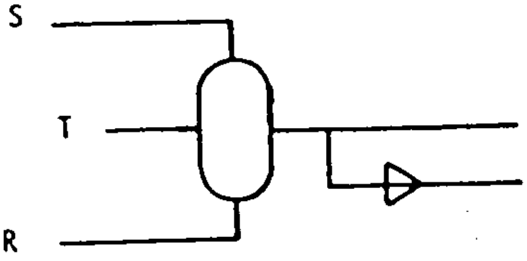
And Gate



c true if a and b true

$\bar{c}$ inverted signal of c

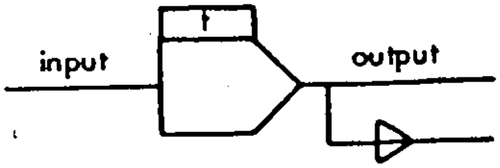
Individual Flip/Flop



S- input to set F/F

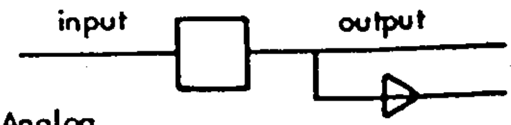
R- input to reset F/F

Monostable Timer



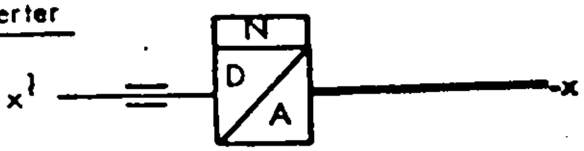
Output high for t seconds after arrival of input

Logic Differentiator



Output high for one clock period after arrival of input

Digital to Analog Converter



x^1 = quantizised x

Analog to Digital Converter

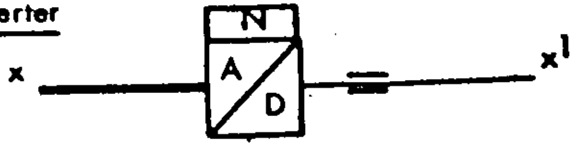


TABLE A-1 - LIST OF FILE NAMES OF THE PROGRAM AS REQUIRED FOR THE MEMORY OVERLAY

Submodule No.	File Name	Contents of files or function
–	UPGCRN	Crane and other subroutines
–	UPGGSP	Required GASP subroutines
–	UPMNC1	Controlling program GASP, files · UPGCRN, UPGGSPand other system routines
1	UPS1C1	Initialising entire simulation model
2	UPS2C1	Events 1-38
3	UPS3C1	Events 60-169
4	UPS4C1	Interpretation of manual commands
5	UPS5C1	Information on model status
6	UPS6C1	Events 170-249
7	UPS7C1	Events 250-300
8	UPS8C1	Events 39-59
9	UPS9C1	Automatic event generation
10	UP10C1	Automatic event interpretation
11	UP11C1	Forecasting and scheduling of material availability and requirements
12	UP12C1	Continuation of UP11C1

TABLE A-2 - INPUT COMMANDS

1st Symb.	2nd Symb.	List of Variables					Action
CC							Correct previous command
//							End of commands
RE	FN	Fumace					Start refining
CA	ST	Fumace					Start casting
GW	XX*	Crane #,	Waiting time,	Priority,			Go to specified device XX and wait
GO	WT	Crane #,	Position,	Waiting time	Priority,		Go to specified position and wait
PU	LL	Crane #,	Device Code,	0,	Waiting time,	Priority,	Pull collar
BU	MP	Crane #,	Waiting time,	Priority			Bump Ladle
PK	YY*	Crane #,	Device code,	Quantity,	Waiting time,	Priority	Pick up specified material YY
DP	YY*	Crane #,	Device code,	0,	Waiting time,	Priority	Dump specified material YY
OI	LZ						Oil crane #Z Z = 1,2,3
IN	FO	Device code, XX*	Blank or 9				Provide information, 9 after device code results in a teletype output
PD	YY*	Crane #,	Source device code,	Destination device code,	Quantity		Pick up and dump specified Material YY
LI	TE	Light #,	0, for off 1, for on				Set Individual light indicator
ST	OP	Conv. code, Crane #	Length of stop,				Converter repair
FI	X						Move crane to pre-established position without going through crane files

TABLE A-3 - DEVICE AND MATERIAL CODES FOR COMMAND INPUT

*XX is the 2-character device code		**YY is the 2-character material code	
R_i	Reverb No. i	SL	Converter Slag
A_i	Anode No. i	MT	Matte
M_i	Matte Tunnel No. i	TM	Transfer Matte
C_i	Converter No. i	CU	Copper
F_i	Flue Dust Ladle No. i	PG	Pigs or Cake
PP	Pig Pile	SC	Scrap
PB	Pig Bay	RS	Reactor (pilot plant) Slag
BB	Bumping Block	FD	Flue Dust
PI	Pilot Plant	SK	Skull
NE	North End	LD	Ladle
SE	South End	CP	Collar Puller
GP	Gasp (for Cmnd, INFO only)	CS	High Grade Converter Slag
AL	All (for Cmnd, INFO only)	AS	Anode Slag
PT	Prototype Plant		

APPENDIX B

CRANE SUBMODELS

B.1 DESCRIPTION OF CRANE TASKS

In modelling the activities of the three overhead gantry cranes in the converter aisle the following terms are used frequently:

command - instruction to a crane to perform a task

task - activity of moving a crane from one point to another in the converter aisle with or without any load where it will remain for a specified period of time, at the end of which an event can be triggered by the crane.

origin - point of departure of crane

target - destination of crane when responding to a command

waiting time - period of time that a crane remains at the target which corresponds to the duration of the activity at that point

crane file - memory locations containing a number of crane tasks that have not been executed.

It is possible to break down all crane activities into four basic phases:

1. Establish feasibility of an operation.
2. Movement from origin to target.
3. Waiting at target for the time it takes to perform an operation.
4. Recording the change of state of a process caused by the crane activity.

The following assumptions are made:

1. A crane moves at its maximum speed, unless it is slowed down by a slower crane.
2. Crane movements along the aisle are always longer than across, i.e. the trolley movement is not considered to contribute to the duration of a crane movement.
3. The duration of any lifting lowering or other operation can be represented by average times for such tasks.
4. The change of state of a process receiving service from a crane occurs instantly and at the end of the servicing period.
5. Every task has a unique priority assigned to it which increases every time the task is interrupted or delayed.

The four phases of a task are executed partly by a digital and partly by an analog crane model. It is important to note that these tasks are performed in parallel and independently by each crane. Once a command is issued to a crane the task is recorded and the model branches to the respective subroutine. Figures B-1, B-2 and B-3 are flowcharts of each of these CRANE subroutines.

PHASE ONE

In the first phase the feasibility of executing the task is established. Apart from the target and waiting time the priority and state of activity is used to test whether an interference with another crane exists. No conflict arises if the target of the crane ready to execute a task is separated more than 20 ft. from the next crane, and the task will be executed instantly. Some examples will show best how the conflicts are resolved if interferences of two cranes occur. The logic involved is shown on the flowcharts of the crane routines.

The arrangement of the cranes as it exists in the smelter is used here where crane No. 1 is on the south end, crane No.2 on the north end, and crane No.3 in the centre.

1. If No. 1 crane is to move beyond the present position of both other cranes which are idle, then these move out of the way. Without delay the task is executed.

2. If in the same situation as in 1. crane No.3 is busy with a task that is to take less than 2 minutes, then crane No. 1 is delayed until that time, when it will take up its task with a priority increased by one unit in a range from 1-9. (See also Table B-1 for a description of all variables of the crane models.)

3. If in the same situation as in 2. crane No.3 was only about to start its task which had a priority lower than that of crane No. 1, then the task of crane No.3 would be delayed until crane No. 1 had completed its task. (The situation would be reversed if the priority of crane No. 1 was the lower of the two.)

4. If an interference between the cranes 1 and 2 occurs with crane No.3 being idle, then the situation is the same as the above with the idle centre crane responding freely to any movements.

5. If however, both cranes No.3 and No.2 obstruct the movement of No. 1 crane, then first the interference with crane No.3 is resolved according to the same rules. If No. 1 crane is successful, then the interference with crane No. 2 is resolved. If crane No.3 is executing a task that can delay No. 1 crane, then crane No.2 will be unaffected. If on the other hand crane No.2 is the cause of a delay then crane No.3 will continue its task unaffected.

In general any task that is delayed will be commenced as soon as the crane causing the delay has completed its task.

The situation where a task of lower priority will be executed first arises when the task has been started already and cannot for practical reasons be interrupted. In effect the priority increases as the end of the task approaches. On the other hand tasks of long durations such as "collar pulling" can be interrupted if necessary to allow passage of another crane.

This concludes the phase of establishing the feasibility of a crane operation. It is the most difficult one and only by refining the decision algorithms and assigning correct priorities to different tasks it was possible to obtain realistic crane models. Both phases two and three of a crane task are performed by the analog model of the crane. Figure B-4 is the circuit diagram for one analog crane model. There is one for each crane.

PHASE TWO

Once interferences are resolved the crane circuit (Fig. B-4) is set according to the task that is to be performed. The variables, PO(I), the target position and WT(I), the waiting time, together with the crane number are specified when calling the subroutine GOWT, which performs the setting up of the circuit. A flowchart of GOWT is shown in Figure B-5.

The crane circuit is composed of analog components that are controlled by logic signals, the functions of each will be described below. The equation describing the movement of the crane along the converter aisle is:

$$X = \int_{T_o}^{T_1} \dot{X} \, dt + X_o$$

where X_o = position of crane at time T_o

X = position of crane after time T_0

$\dot{X}$ = speed of crane

This equation reads as follows after scaling if the corresponding variables are used:

$$\frac{XCR}{XMAX} = \int_{T_o}^{T_1} \frac{XDOT}{XMAX} \, dt + \frac{Xo}{XMAX} \quad \ldots.(2)$$

The assumption that cranes move at the fixed maximum speed was specified earlier which gives:

$$\frac{XCR}{XMAX} = \frac{XDOT}{XMAX} \int_{T_o}^{T_1} dt + \frac{Xo}{XMAX} \quad \ldots.(3)$$

This integration is performed by integrator I-1 while the speed is specified by potentiometer P1. The direction of travel forward or backward is determined from the previous position and is specified by function relay F1 which is switched to a positive or negative reference voltage. Given the initial position (P2) and a speed, the crane position will at all times be available to the digital model by reading the A/D - converter that gives XCR. If the analog model is placed in the OPERATE mode the crane will move until it reaches its target position, where the cranemovement stops.

This is achieved by subtracting the target position (XTAR=P0(I)) from the crane position and finding when its absolute value becomes zero:

$$\frac{XCR}{XMAX} - \frac{XTAR}{XMAX} < 0.001 \qquad \ldots.(4)$$

The subtraction and finding of the absolute value is performed by the two summers (S1, S3) and one limit summer (S2) as shown. If the value reaches .001, the value given by potentiometer P3, comparator C1 changes sign, which in turn affects the logic circuit connected to it. At this point the logic signal A will cause the individual integrator I1 to stop integrating, i.e., the crane will stop once the target is reached.

PHASE THREE

When a task is to be executed by the crane both the target position XTAR and the time to wait there TWAIT (TWAIT = WT(I)) is set. The lower half of the circuit simulates the waiting period according to:

$$\frac{T}{TMAX} = \frac{C}{TMAX}\int_{T_1}^{T_2} dt + \frac{CONST}{TMAX} \qquad \ldots.(5)$$

Where T is the time the crane waits at its target, $\frac{\text{CONST}}{\text{TMAX}}$ will be zero at the beginning of every waiting period. The integration is performed by integrator I2 while the rate C is set on potentiometer P4. The rate is dependent only on the time scale of the simulation which could have any desirable value. Since a speed-up of BETA = 20 was used:

$$C = \frac{\text{BETA}}{\text{TMAX} \times 60} = .0167 \qquad \text{(in volts/sec.)}$$

Scaling of the analog circuits is completed by choosing the maximum travel distance and task durations for which the following values were found suitable:

$$\text{XMAX} = 500 \text{ ft.}$$

$$\text{TMAX} = 20 \text{ min.}$$

Returning to the point where a crane had just reached its target and integrator I1 was placed in the HOLD mode by the logic signal A, the same signal willalso place the waiting time integrator I2 into the OPERATE mode to start the waiting period. Time T is compared against the required time TWAIT by comparator C2. When T = TWAIT logic signal returns integrator I2 to the HOLD mode. This basically concludes the two phases of moving and waiting which will represent a continuous process between time T_o and T_2.

At the end of the waiting time the logic signal C also sets a flip flop FF1 by producing a pulse signal in the differentiator D1. A sense line is connected to the output LS of the flip-flop and is continuously sampled in the controlling program. If the logic value LS changes it indicates the end of a waiting period and the controlling program places the entire simulation in the HOLD mode. This is done by connecting both integrators I1 and I2 to the computer mode signals as shown. Before describing the last phase of the crane task the function of the logic circuitry is summarised by a set of logic equations

where A, B, C, D = logical true signal

$\bar{A}$, $\bar{B}$, $\bar{C}$ = logical false signal

OP = computer operate mode signal

IC = computer initial condition mode signal

LM = $\bar{A}$ and B

LN = A and OP

LO = $\bar{A}$ and $\bar{C}$

LP = $\bar{A}$ and $\bar{C}$ and OP

LQ = A or C

LR = C and B

LS = C or D (C = true when task is done until reset.)
(D = false at start of every OPERATE period)

PHASE FOUR

In the fourth phase the crane task is completed by executing an event caused by the crane activity. The following example will explain this: The action of moving to a converter and adding a ladle of matte to it has to be recorded by the converter model as a material addition. At the end of the time period that the crane waits at the target the material has thus been added but the converter model is unaware of this. An event code associated with the particular task is contained in the variable NEXT (I) for the particular crane while ISAVE(I)/100 contains the quantity of material if it is other than unity. The controlling program will use the value of NEXT(I) to branch to the respective submodel where the state of the submodel will be updated accordingly. For example, the addition of a ladle of matte into converter No.3 will correspond to an event number 80 if it originates from matte tunnel No.1. This code will allow the controlling program to branch to that section of the converter submodel where the addition is recorded.

This completes the execution of one crane task and all other tasks will be executed in a similar way. Commands entered via the teletypewriter are encoded and a typical command

PDMT1, M1, C3

would stand for "Pick up matte at matte tunnel No. 1, pour it into converter No.3 and return the empty ladle to matte tunnel No. 1." This command in fact consists of three basic tasks (3 movements, with a time period for lifting or depositing at the end of each). The target positions, waiting times and priorities for each task are stored internally and need not be specified. A detailed description of command input and interpretation is given in Appendix A.

In the next section the continuous filing and retrieving of a series of crane tasks will be dealt with.

B.2 EXECUTION OF CRANE TASKS

Modelling of the crane tasks has been described in the previous section. In this section the control of a number of tasks waiting to be executed is to be described.

All crane tasks are issued via the teletype input if the model is operated manually. If automatic operation is desired,which is described in Chapter 7, the same type of tasks are used with one additional variable included. All tasks are then stored in the crane files numbered 2 for crane No. 1, 3 for crane No.2, and 4 for crane No.3. This is done in the same way as time events are stored in the time file which is labelled No.1. Tasks are stored in the order in which commands are issued by means of the ranking variable NRANK. The term crane task is thus synonymous with crane event to use the terminology of the GASP simulation language. For this particular study the maximum number of attributes associated with each event was chosen to be four to save computer memory. It is necessary to use at least six attributes to describe a crane event uniquely which was achieved by encoding four integer variables before storing them in two locations with the remaining two floating point variables stored separately. This technique proved to be very useful, but is only possible if the range of values that a variable can have ,is limited. When an event is removed from the file for execution the values are then decoded again. Table B-2 lists the six variables with the constraints placed on their numerical values.

Filing a crane event is performed by subroutine CFILE, the flowchart of which appears in Figure B-6, where the change from 6 to 4 attributes is also performed. The opposite is performed when retrieving an event which is done in subroutine GETEV shown in Figure B-7.

Assuming that all crane files are empty, that is there are no tasks to be performed by the cranes, then the busy light on the crane display will be turned off and the variable

$$\text{IDLE(NCR)} = 0 \qquad (\text{NCR} = \text{crane No.})$$

signifies that crane No. NCR is idle. Following the issuing of a crane command, when the computer is still in the HOLD mode, the controlling program GASP performs a search of all crane files (see flowcharts of GASP in APPENDIX A). If more than one crane file is not empty

$$\text{i.e., MFE (NCR + 1)} \neq 0$$

the first crane events in each file will be executed in the order of their priorities. For crane No. 1 having the highest priority this will take the following typical sequence:

1. Remove event from crane file (subr. GETEV)
2. Call resp. digital crane model to establish feasibility of executing task (subr. CRA1).
3. If feasible, set up analog model (subr. GOWT). If not feasible refile crane event with higher priority (subr. CFILE).
4. Repeat 1 - 3 for other cranes in decreasing order of priority.
5. Start simulation by placing computer in OPERATE mode.

Assuming that no other event occurs until the first crane task is completed, the simulation will run with all busy cranes performing their tasks in parallel. At this time a sense line in the analog crane model will change its logic level. During this period the digital program performs no other task than reading logic signals from the analog computer. The end of the crane task is thus recorded by the controlling program which places the analog computer in the HOLD mode to execute the fourth phase of the crane task as explained in Section B.1.

Thereafter the crane file of the particular crane is searched and the next event is executed in the same way as described above. In this way any crane will be busy as long as there are tasks to be performed. Once a particular crane file becomes empty,

$$MFE\ (NCR + 1) = 0$$

the busy indicatcr which was turned on is reset to indicate that the crane is now idle. The crane remains at the last target position until it has to move out of the way of another crane or when it receives another command. Apart from issuing one of several commands that occur regularly in actual practice such as moving a ladle of matte or copper, the crane models will respond to any arbitrary command where the choice of the target, the waiting time and priority can be chosen freely. These variables are checked and if they exceed the allowable limits they will be set to the maximum or minimum permissible value.

B.3 DIGITAL CRANE MODEL

If the combination of three cranes were to operate as described in Section B.2, they could be considered to be ideal models. In reality scheduled maintenance has to be performed on the cranes twice a week during which time they become inoperative. Furthermore breakdowns occur at irregular intervals when again a crane is operative for an undetermined duration. These aspects were included in the crane model by utilizing the event structure of the GASP discrete event simulation language. The problem of maintenance is thus modelled as follows. Since maintenance is performed at a predetermined time a crane maintenance event in the time file will schedule the maintenance for a given time. At this time a typeout will appear on the teletype "Crane is due for maintenance". The operator will respond to this by issuing the following command:

OIL (NCR = 1, 2, 3)

After interpretation an event is filed that will be executed five minutes from the time the command is given. This event is a crane maintenance event causing a crane to become inoperative for the next 90 minutes, the average time required for maintenance. At the same time information will appear on the teletype as to remind the operator when the crane will be operational again. Also the light indicator for the operational state will be turned off and the crane is now unable to accept any commands. If the command OIL is not issued within 30 minutes of the time the crane became due for maintenance, then a reminder to operator will appear every 30 minutes on the teletypewriter until the maintenance is performed. After the maintenance period has elapsed the crane will be operational again until the next maintenance period comes up.

Irregular breakdowns are dealt with in a similar fashion, except that the time and duration of breakdowns are not deterministic quantities. Separate statistics on both quantities for each crane are used and random samples are drawn to determine breakdowns for each crane. This technique of random sampling from operating statistics is used frequently in the model and needs no further explanation here.

At the beginning of any simulation run crane failure times are generated. It was found that a uniform distribution is the best approximation to represent the breakdown behaviour of the cranes. If this time falls within the simulation period of the particular run the crane will break down and the operator is informed of this by a message on the teletypewriter. At the same time the repair time is sampled from a normally distributed function and the expected duration of the breakdown will also be supplied to the operator. Depending on the length he will then decide to use another crane or just wait until the crane is operational again. As in the case of scheduled maintenance the operational light is turned off and the no command can be issued to the crane until it is operational again.

B.4 CRANE STATISTICS

During any simulation run statistics on the cranes are accumulated and can be obtained at any time by giving the command

INFOK (NCR)

where NCR is the crane number. A call to submodule 5 will provide a tabular listing of all statistics together with all variables describing the crane state. A typical typeout is shown on Table B-3. Shown are the variables of the task presently executed as well as those of the previous task. They are rearranged slightly and the value of KODE is shown elsewhere (the previous event number is not shown as it is always reset to zero after executing the previous event). The three statistical quanties recorded are:

1. busy time (%)
2. delay time (%)
3. down time (min.)

Also shown are the attribures of the next events for each of the three cranes. Where the variables described before give information on the present and previous events, the attributes above contain the same information on the 1st future event. A complete listing of all future events can be obtained for each crane by issuing the command:

INFOGP

This provides the information on all three crane files as well as the time file. Table B-4 shows a typical typeout of the filing array together with relevant variables. The starting point of each file is given and by using column 5 and 6 of the array, it is very easy to obtain a full picture of the number and type of events in each file. Column 5 gives the line number of the succeeding event, while column 6 gives the preceding event. 9999 signifies the start of a file and 7777 indicates the end of a file.

Returning to Table B-3 the last two variables typed out KWT(I) and JWT(I) are state variables of the cranes, indicating whether one crane is waiting for another. Their functions can be described as follows:

JWT(I) = J

I = 1,3 J = 1,3 indicates that crane No. I is delayed by crane No. J and will continue when crane No. J has completed its present task.

KWT(I) = J

I = 1,3 J = 1,3 indicates that crane No. I waits for crane No. J and will only continue when triggered by crane No. J. (This variable allows synchronization of a two-crane operation which is used in the automatic operation.)

i.e. JWT(I) = 0 I = 1,3 KWT(I) = 0 I = 1,3

signifies that no crane is delayed or waiting. Table B-5 lists all values input variables to the crane models. These values are fixed quantities during any simulation run.

B.5 CRANE COLLISIONS

While any collision is avoided by resolving crane interferences in the crane models a collision due to the slower crane speed of No.2 crane could occur if it was followed by the faster No. 3 and No.1 cranes. In practice the faster cranes would slow down in such a situation to the slower speed. Since it is only important in the simulation model, to ensure that the overall travelling time is the same, the faster crane could move at its maximum speed and then stop when it comes too close to the slower crane. After the slower crane has moved a far enough distance the other crane can proceed to its target. This concept is implemented in the analog models of the cranes. In Figure B-8 the circuit diagram is shown. The monostable timer is set to a suitable value and will delay the faster two cranes if they were to follow crane No.2. In practice this does not occur very often but it might be important when the possible use of No.2 crane is further investigated.

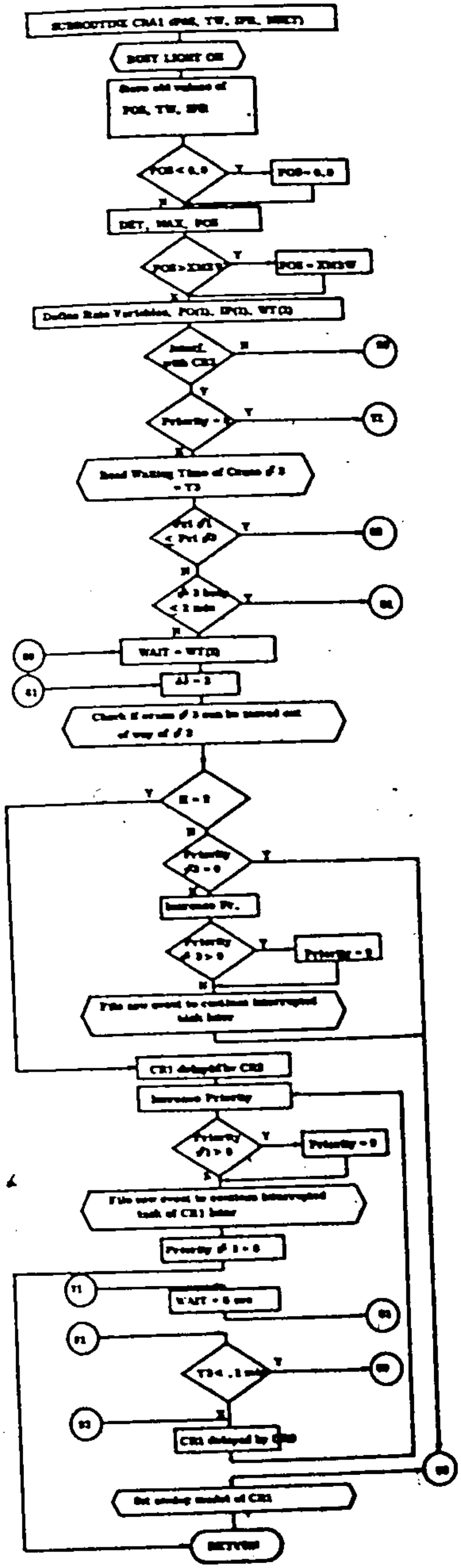


FIGURE B-1 FLOWCHART OF SUBR. CRA1

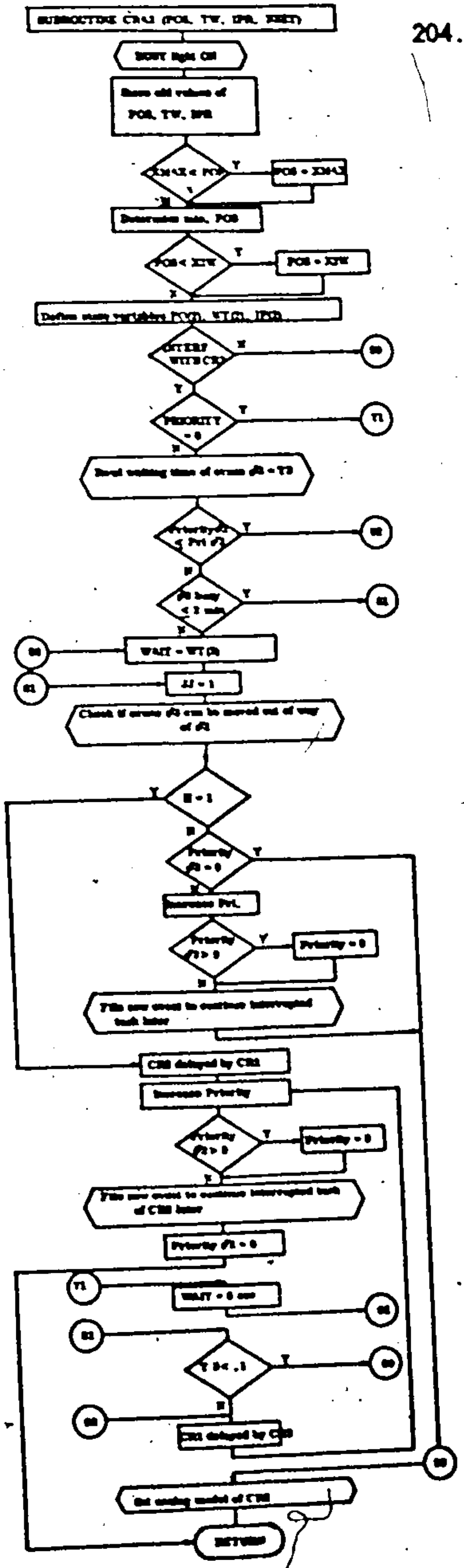


FIGURE B-2 FLOWCHART OF SUBR CRA2

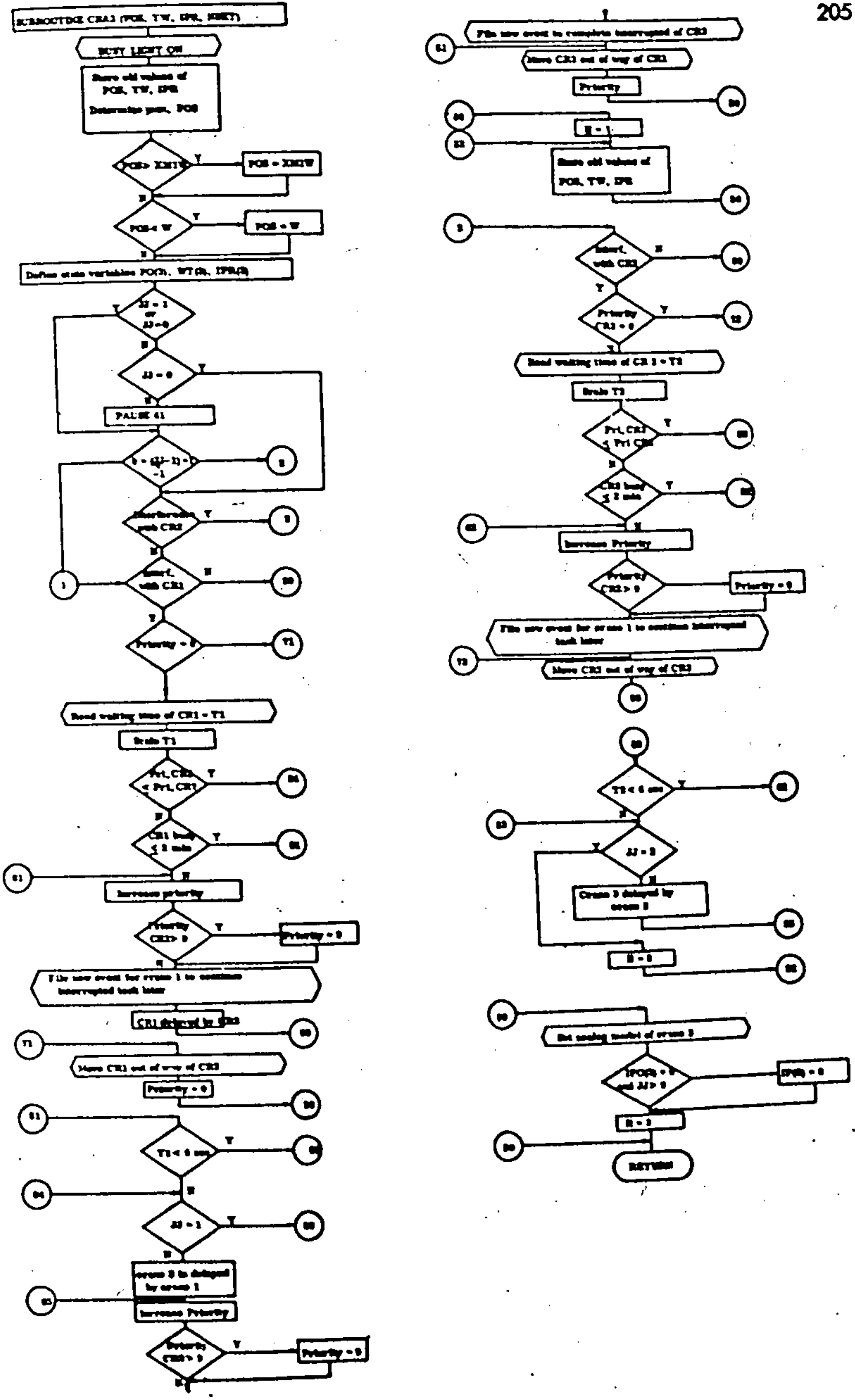

SUBROUTINE CRA3 (POS, TW, IPR, XNEXT)
BUSY LIGHT ON
Store old values of POS, TW, IPR
Determine pos. POS
POS> XNEW
POS = XNEW
POS< W
POS = W
Define state variables PO(3), WT(3), IPR(3)
JJ = 1 or JJ = 0
JJ = 0
PAUSE 41
Interference with CR2
Interf. with CR1
Priority = 0
Read waiting time of CR1 = T1
Scale T1
Pri, CR3 < Pri, CR1
CR1 busy ≤ 2 min
Increase priority
Priority CR3> 9
Priority = 9
File new event for crane 1 to continue interrupted task later
CR1 delayed by CR3
Move CR1 out of way of CR3
Priority = 0
T1 < 0 sec
JJ = 1
crane 3 is delayed by crane 1
Increase Priority
Priority CR3> 9
Priority = 9
File new event to complete interrupted of CR3
Move CR3 out of way of CR1
Priority
Store old values of POS, TW, IPR
Interf. with CR2
Priority CR2 = 0
Read waiting time of CR 2 = T2
Scale T2
Pri, CR3 ≤ Pri CR2
CR2 busy ≤ 2 min
Increase Priority
Priority CR2 > 9
Priority = 9
File new event for crane 1 to continue interrupted task later
Move CR2 out of way of CR3
T2 < 0 sec
JJ = 2
Crane 3 delayed by crane 2
Set analog model of crane 3
IPO(3) = 0 and JJ > 0
IP(3) = 0
RETURN
Y
N
T1
T2
S1
S2

FIGURE B-3 FLOWCHART OF SUBROUTINE CRA3

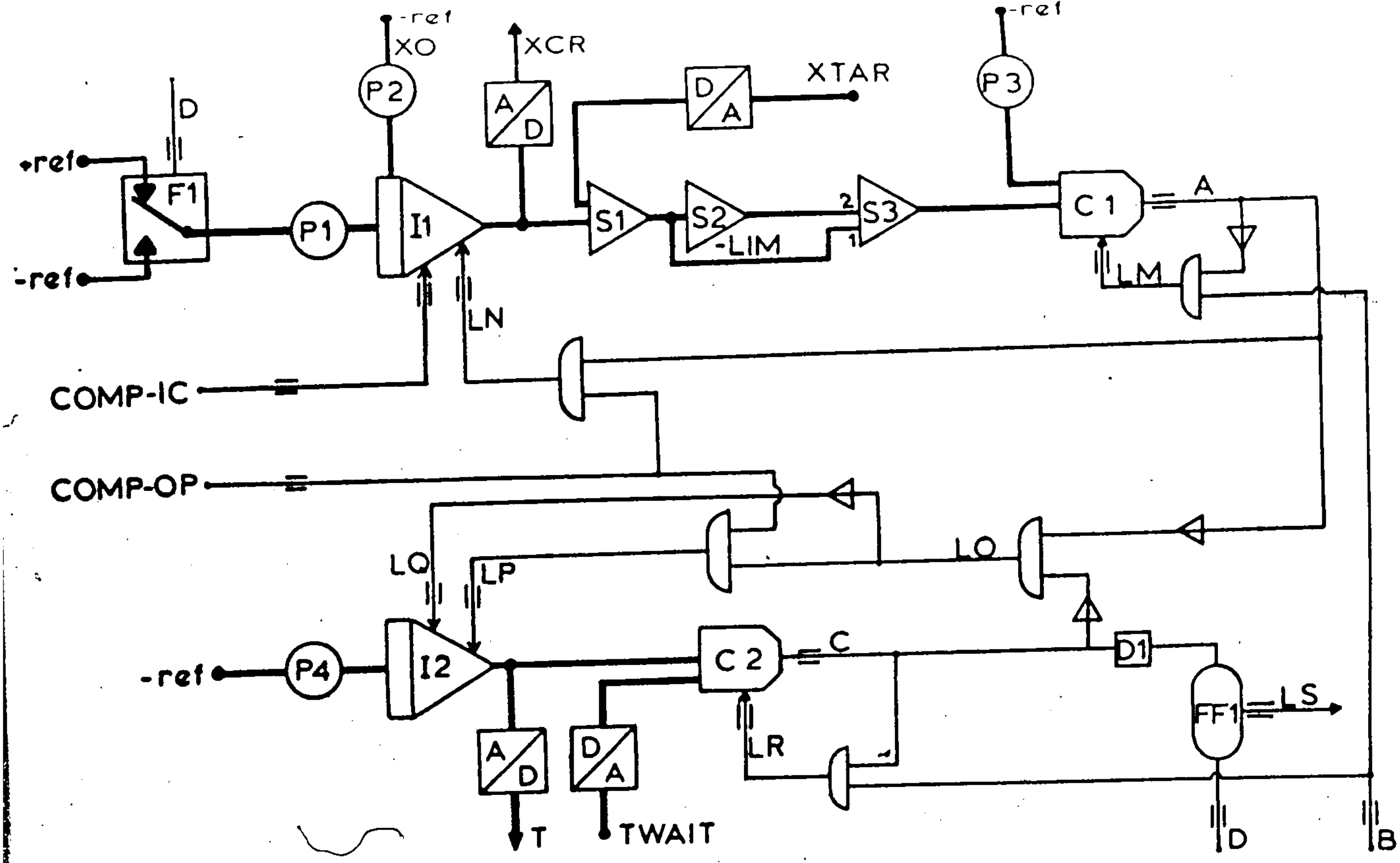


FIGURE B-4 ANALOG CIRCUIT FOR CRANE MODELS

FIGURE B-5 FLOWCHART OF SUBR. GOWT

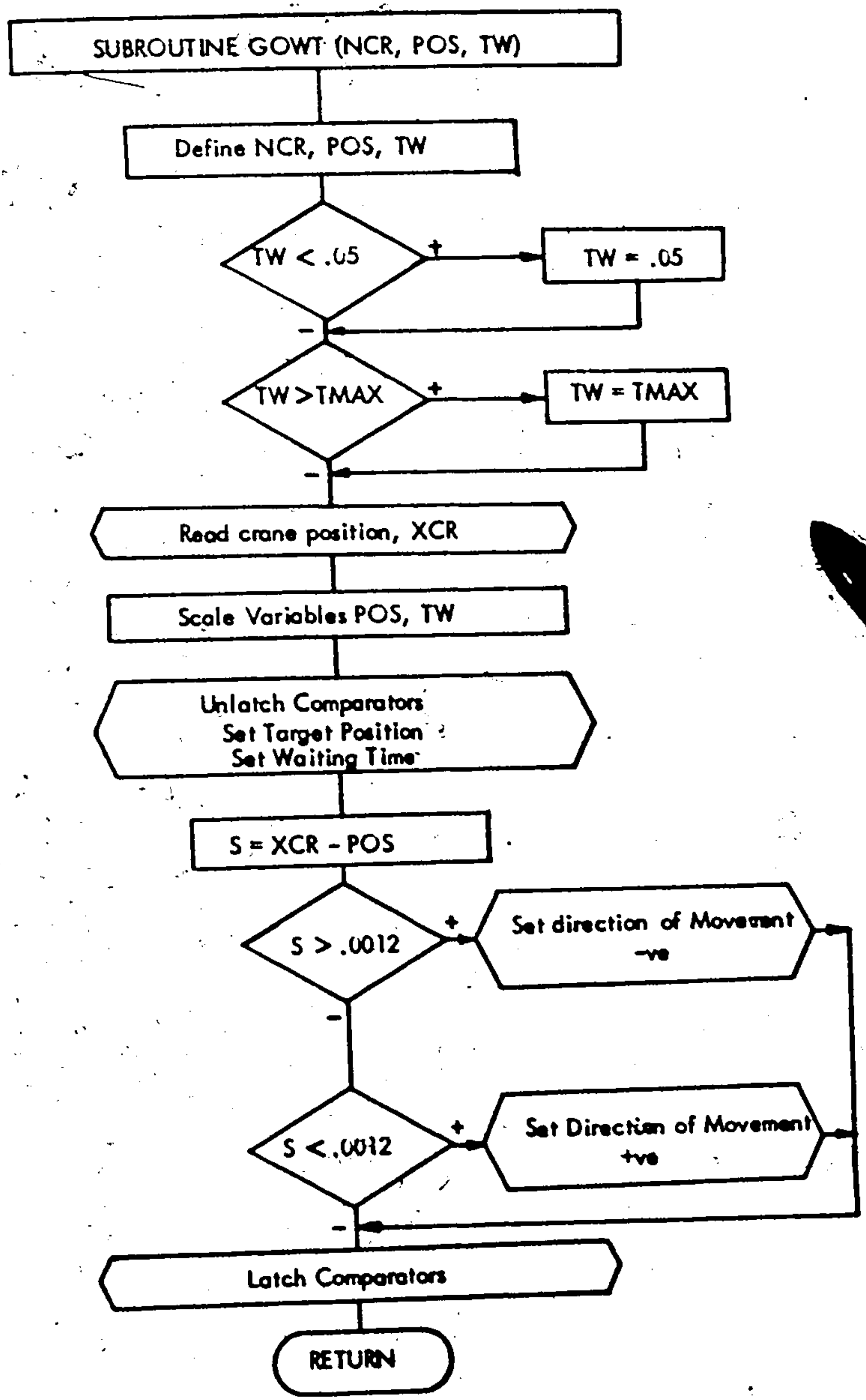


FIGURE B-6 FLOWCHART OF SUBR. CFILE

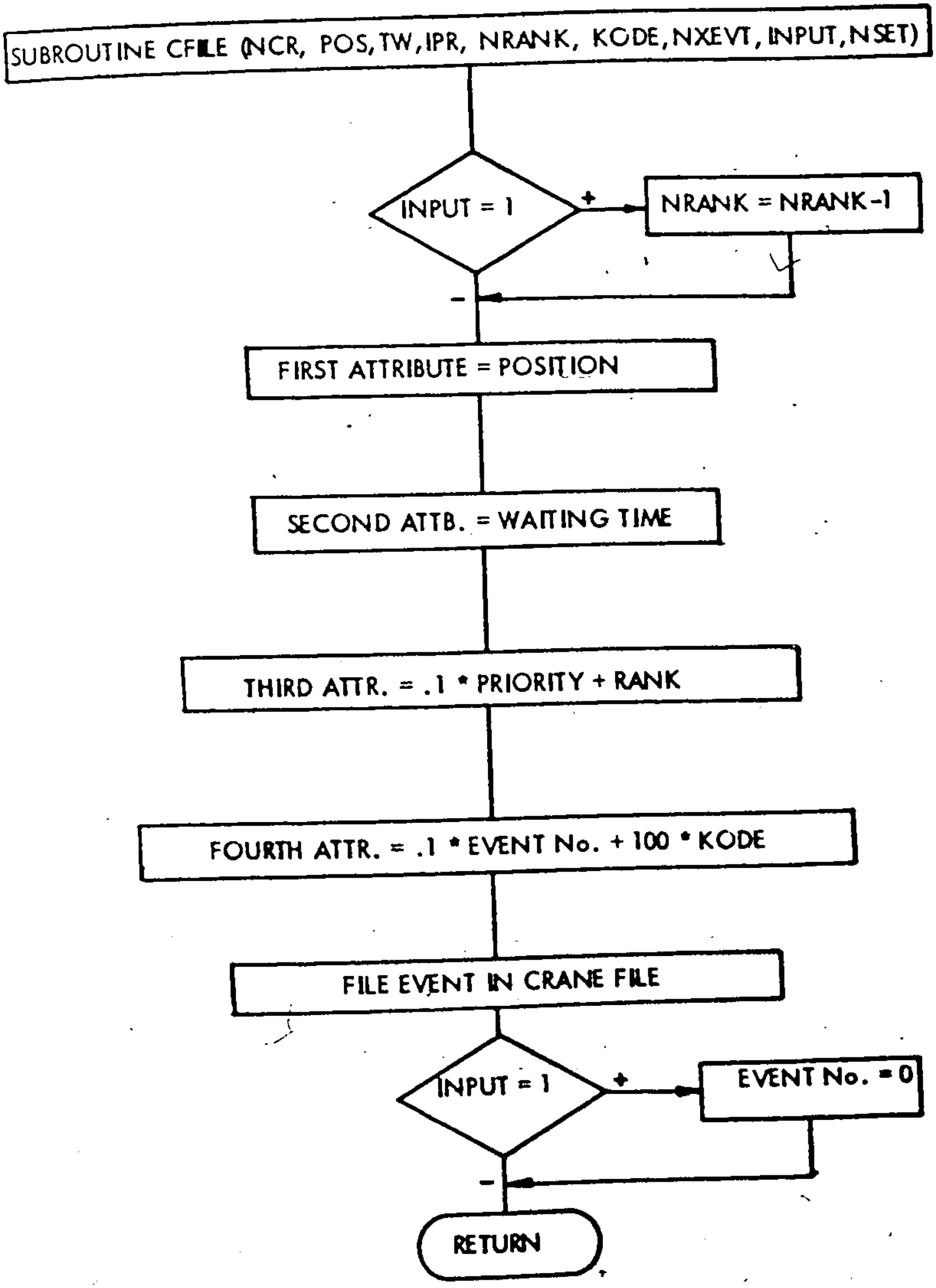
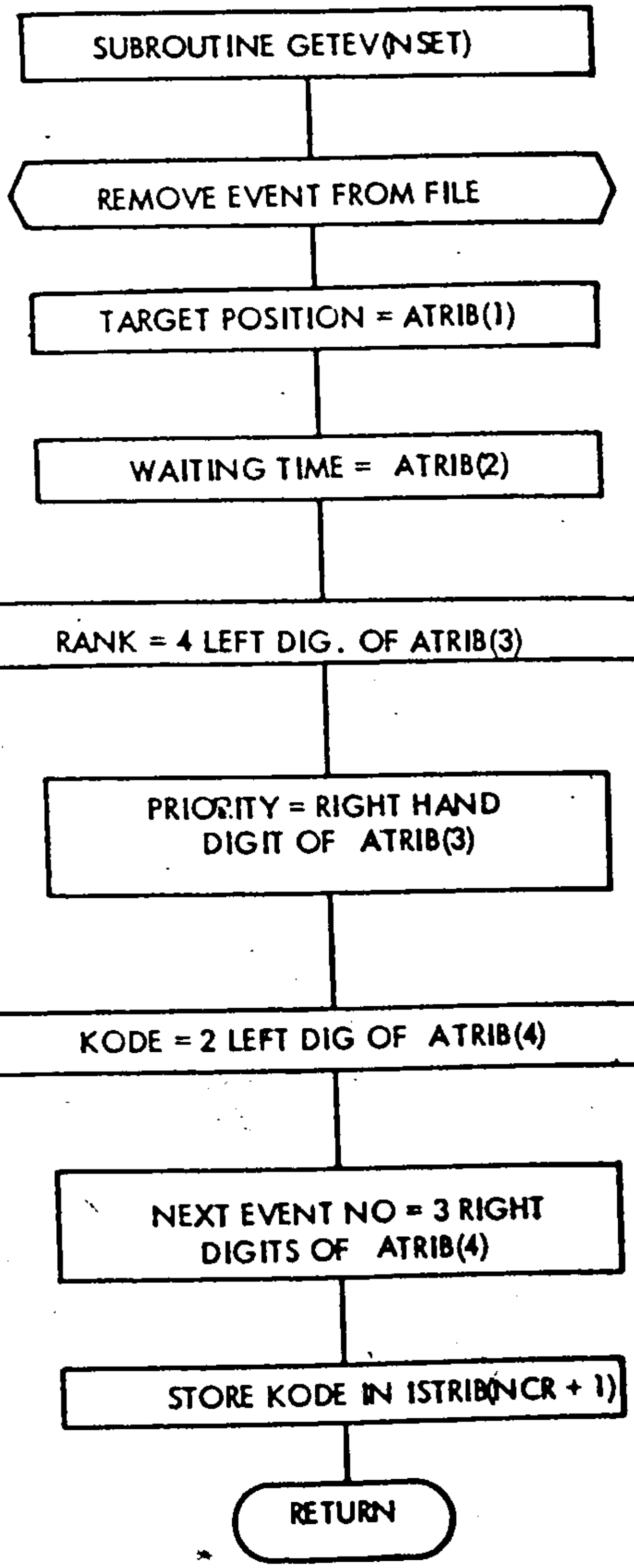

SUBROUTINE CFILE (NCR, POS, TW, IPR, NRANK, KODE, NXEVT, INPUT, NSET)
INPUT = 1
+
NRANK = NRANK-1
-
FIRST ATTRIBUTE = POSITION
SECOND ATTB. = WAITING TIME
THIRD ATTR. = .1 * PRIORITY + RANK
FOURTH ATTR. = .1 * EVENT No. + 100 * KODE
FILE EVENT IN CRANE FILE
INPUT = 1
+
EVENT No. = 0
-
RETURN

FIGURE B-7 FLOWCHART OF SUBR. GETEV


SUBROUTINE GETEV(NSET)
REMOVE EVENT FROM FILE
TARGET POSITION = ATRIB(1)
WAITING TIME = ATRIB(2)
RANK = 4 LEFT DIG. OF ATRIB(3)
PRIORITY = RIGHT HAND DIGIT OF ATRIB(3)
KODE = 2 LEFT DIG OF ATRIB(4)
NEXT EVENT NO = 3 RIGHT DIGITS OF ATRIB(4)
STORE KODE IN ISTRIB(NCR + 1)
RETURN

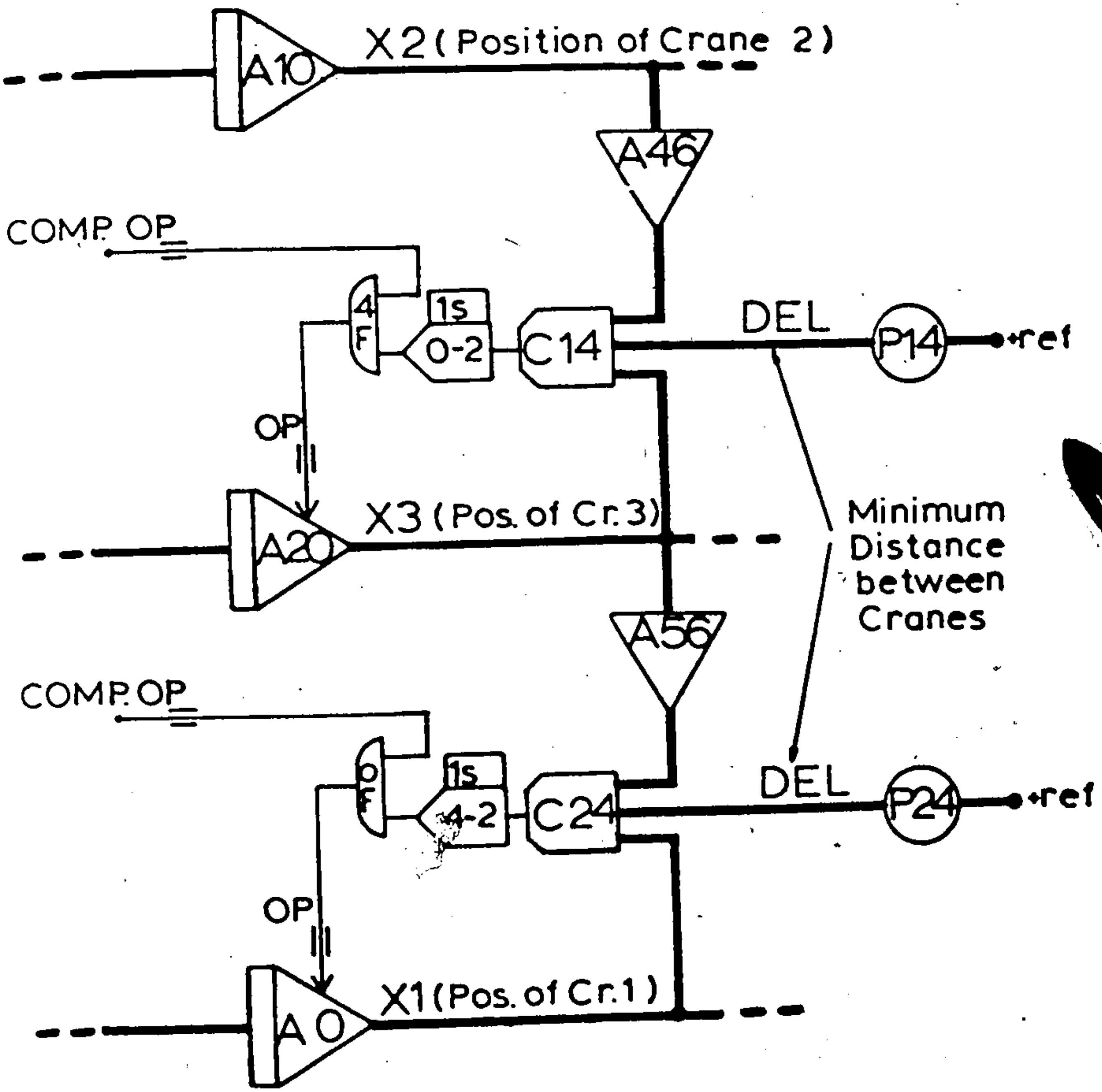


FIGURE B-8 FEATURES OF ANALOG CRANE MODELS TO PREVENTCOLLISIONS

TABLE B-1 - CRANE MODEL VARIABLES

PO (NCR)	present position
POO (NCR)	previous position
WT (NCR)	waiting time of present task
WTO (NCR)	waiting time of previous task
NRANK (NCR)	rank of last entry in crane file
LSTR (NCR)	rank of last event removed from crane file
IP(NCR)	priority of present task
IPO (NCR)	priority of previous task
NEXT (NCR)	event to follow execution of present crane task
IDLE (NCR)	state variable (busy or idle)
KAPUT (NCR)	state variable (operational or down)
NOIL (NCR)	state variable (maintenance required)
KODE	code used to retain different information on crane events.
ISTRIB (NCR+1)	contains the value of code after event is removed from crane file
ISAVE (NCR)	retains 100* quantity of material picked up by a crane
JWT (NCR)	state variable (delayed or not)
KWT (NCR)	state variable (waiting or not)
NCR	Crane Number 1, 2, 3
TDOWN (NCR)	total time crane was broken down
TCRAN (NCR)	total time crane is busy
SP(NCR)	total time crane is delayed
TDL(NCR)	mean of crane repair duration
TDH(NCR)	standard deviation of crane repair duration
CRL(NCR)	lower limit of time between breakdowns
CRH(NCR)	upper limit of time between breakdowns

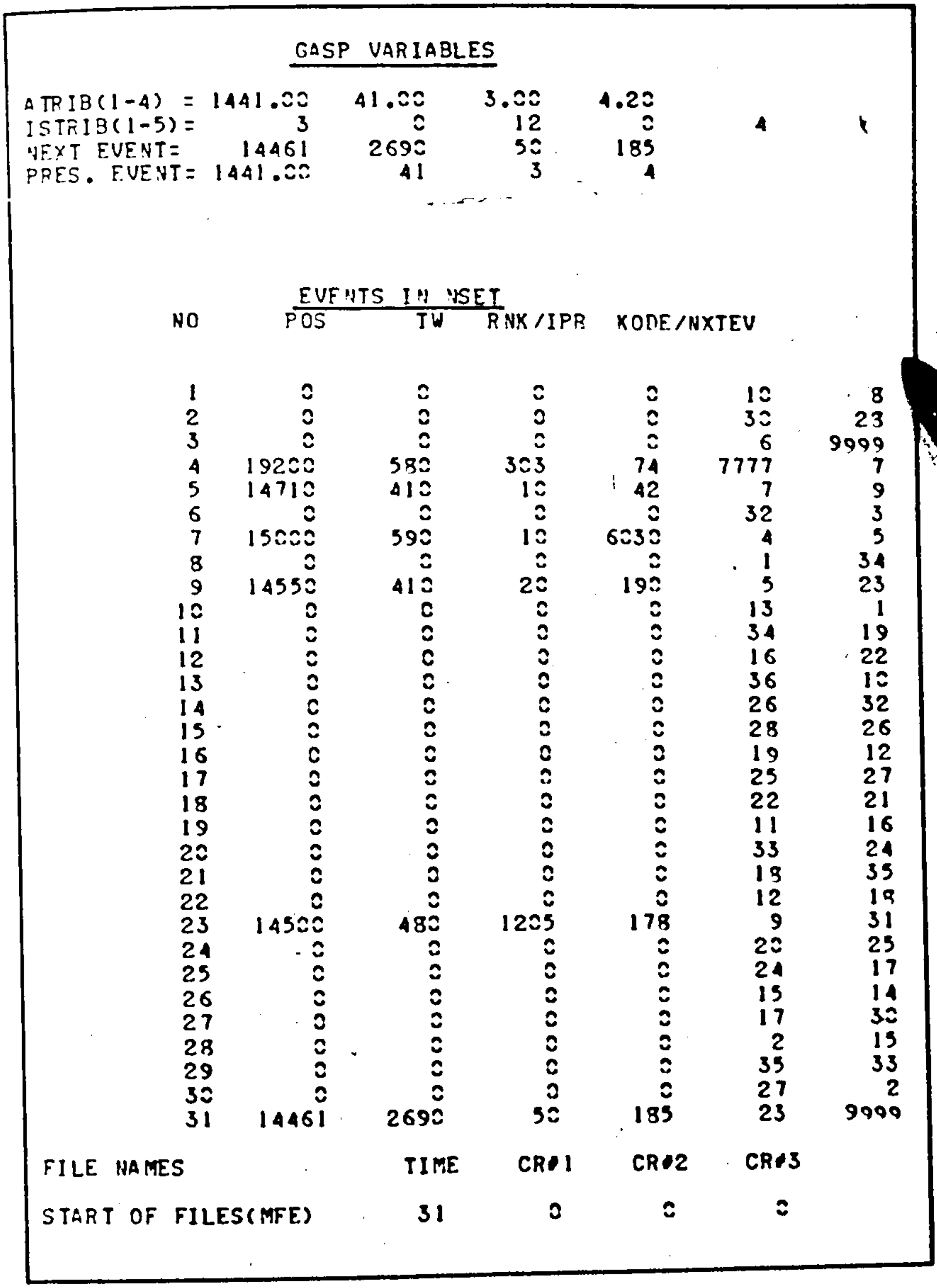
TABLE B-2 - CRANE EVENT VARIABLES

NAME	REPRESENTATION	RANGE OF VARIABLE
POS	- Target position	- any F.P. variable (0-500 ft.)
TW	- Waiting time	- any F.P. variable (0-20 min.)
NRANK	- Rank of task	- 10 up
IPR	- Task priority	- 0 - 9
KODE	- Code for different purposes	- 0 - 32
NXEVT	- Event number to follow crane task	- 0 - 999

```
                    CRANES
NCR POS    POO   WTT   WTO   IPR   IPO   TDOWN   NEXT LSTR NRANKP.USE P.DELAY
 1   62.   264.  .50   .62   0     9       .00   0   40    10   52.52   6.07
 2  469.   308.  .65   .42   0     1       .00   0   30    10     .14    .00
 3  376.   182.  .50   .62   4     5       .00 193   40    40   60.99   4.97

NCR  ATRIB(1)  ATRIB(2)  ATRIB(3)  ATRIB(4)
 1        0         0        10    -32768
 2        0         0        10    -32768
 3        0         0        10    -32768


             STATE  VARIABLES
WAITING(KWT)          0         0         0
INTERFERENCE(JWT)     0         0         0
```

TABLE B-3   TYPICAL SIMULATION REPORT OF CRANE STATUS

TABLE B-4 TYPICAL SIMULATION REPORT ON EVENT FILING ARRAY NSET(6,1), WHERE I=1,60 AND RELATED VARIABLES



```
                       GASP VARIABLES

ATRIB(1-4) = 1441.00     41.00      3.00      4.20
ISTRIB(1-5)=       3         0        12         0         4
NEXT EVENT=    14461      2690        50       185
PRES. EVENT= 1441.00        41         3         4



                       EVENTS IN NSET
         NO         POS          TW    RNK/IPR   KODE/NXTEV

          1           0           0         0         0        10         8
          2           0           0         0         0        30        23
          3           0           0         0         0         6      9999
          4       19200         580       303        74      7777         7
          5       14710         410        10        42         7         9
          6           0           0         0         0        32         3
          7       15000         590        10      6030         4         5
          8           0           0         0         0         1        34
          9       14550         410        20       190         5        23
         10           0           0         0         0        13         1
         11           0           0         0         0        34        19
         12           0           0         0         0        16        22
         13           0           0         0         0        36        10
         14           0           0         0         0        26        32
         15           0           0         0         0        28        26
         16           0           0         0         0        19        12
         17           0           0         0         0        25        27
         18           0           0         0         0        22        21
         19           0           0         0         0        11        16
         20           0           0         0         0        33        24
         21           0           0         0         0        18        35
         22           0           0         0         0        12        19
         23       14500         480      1205       178         9        31
         24           0           0         0         0        20        25
         25           0           0         0         0        24        17
         26           0           0         0         0        15        14
         27           0           0         0         0        17        30
         28           0           0         0         0         2        15
         29           0           0         0         0        35        33
         30           0           0         0         0        27         2
         31       14461        2690        50       185        23      9999

 FILE NAMES                    TIME      CR#1      CR#2      CR#3

 START OF FILES(MFE)             31         0         0         0
```

TABLE B-5 - FIXED CRANE VARIABLES DURING A SIMULATION RUN WITH TYPICAL NUMERICAL VALUES

BETA = 20	speed up factor
XMAX = 500 (ft)	maximum crane movement
TMAX = 20 (min)	maximum length of crane waiting time
SP(1) = SP(3) = 394	ft/min crane speeds
SP(2) = 330	ft/min crane speeds
TOL = 2 (min)	shortest started crane task that is not interrupted
W = 20 (ft)	closest distance between cranes

APPENDIX C

REVERBERATORY AND MATTE TUNNEL SUBMODELS

C-1 EMPIRICAL REVERBERATORY AND MATTE TUNNEL MODELS

While no generally accepted mathematical models for reverberatory furnace operations are available, it is necessary to account for the reverberatory furnace response to material inputs and outputs from the converter aisle. Since the requirements of a reverberatory furnace submodel in the overall simulation of smelter operations are quite different to those of a model to be used for furnace control, the lack of exact theoretical representations of reverberatory furnace practice can be overcome.

Studies of reverberatory furnaces have been based on heat or material balances. The primary concern in developing a reverberatory furnace model, or for that matter any of the simulated submodels, was to use only existing measurable control variables to represent the behaviour of the process. Analysis of reverberatory furnace practice showed that an empirical model based on material balances would produce the most satisfactory representation of reverberatory furnace operation.

Three primary objectives were to be met by the reverberatory model all of which relate to its interaction with the converter aisle operation:

(1) Reproduce the rate of supply of matte of a given grade to the converter aisle.

(2) Represent the availability of both matte and slag in the reverberatory furnace.

(3) Account for the amount of copper lost in the reverberatory slag.

In the empirical model developed it was thus not necessary to account for variables which have no direct effect on above quantities. The usefulness of this might become evident when considering that a regression model of a reverberatory furnace included no less than 29 variables, many of which were difficult to measure.

C-2 ASSUMPTIONS IN THE REVERBERATORY FURNACE MODEL

The following assumptions were made in developing a model of the reverberatory furnace:

(1) The grade of matte from the reverberatory furnace is a fixed input quantity during simulation but can be varied if desired.

(2) The reverberatory furnace operation is a steady state operation, i.e., feeding, firing and consequently the smelting rate is constant unless it is decreased or increased externally.

(3) Matte is tapped from the matte tapping holes as long as there is enough matte in the furnace and as long as there is a demand for it in the converter aisle. (If there is no demand tapping can be temporarily suspended by not removing the last ladle from the matte car.)

(4) The duration of tapping a given quantity of matte or of reverberatory slag is not deterministic but can be represented statistically.

(5) There are definable limits to the levels of both slag and matte in the furnace within which the basic operating practice of slag and matte tapping are fixed.

(6) The assumptions made apply to both dry charged and wet charged furnaces.

Without attempting to give complete justification the following points will support the assumptions on the previous page:

TO: (1) In operating practice it is attempted to keep fluctuations of the matte grade to a minimum. If larger changes occur due to different ore grades, this change can be accounted for in the model by specifying a required higher or lower grade at any time during simulation.

TO: (2) The matte output of reverberatory furnaces is in direct response to the requirement of the converters. This demand is usually uniform. If a drastic decrease or increase occurs the rate of smelting in the reverberatory furnace models can be changed.

TO: (3) The assumption made under (3) corresponds very closely with operating practice.

TO: (4) The process of tapping matte involves several tasks that can hardly be described analytically. (Typically the time it takes to open a matte hole might depend among other variables on the hardness of the clay used to close the tapping hole). This assumption is supported by time studies that were taken of the matte tapping times. These are shown in Figures C-1 and C-2 where the tapping times show a variation very close to a normal distribution.

TO: (5) In practice these limits are not as exactly defined as they have to be in a mathematical representation but such limits exist. The main difference between reality and the model representation being the recognition that such limits have been reached. In practice properties of the material emerging from the tapping holes indicate whether or not such a limit is reached, whereas in the model direct level measurements are recorded. In operating practice level measurements are taken every four or eight hours.

TO: (6) It should be pointed out that the lumped parameter approach in obtaining the smelting rate as a control variable does not need to account for the different material inputs but reflects a given different rate of production for each furnace. In the absence of a relationship between the smelting rate and variables such as rate of fuel consumption, rate of oxygen supply, furnace draught, furnace temperature, etc., this was the most accurate representation possible.

C-3 DESCRIPTION OF REVERBERATORY EVENTS

Although the matte tunnel operation is not part of the reverberatory furnace it is closely related to it and will be described as part of the model.

The model of the reverberatory furnace is composed of a sequence of logically related discrete events. If these events are executed as time advances at the respective times they will reproduce the furnace operation. The events are all stored in the time file with one common time attribute which indicates the time at which the event is to occur. For each of the three reverberatory furnaces the same events are used only with different subscripted variables. The same does not apply for the matte tunnels, where there are separate events for each matte tunnel. This is necessary to account for the difference in the procedure of tapping matte from the furnaces (i.e. both reverberatory furnaces No.1 and No.2 supply matte to matte tunnel No. 1 whereas only reverberatory furnace No.3 supplies matte to matte tunnel No.2).

In Table C-1 a list of events appears with a brief description of the operations performed by them. The abbreviations R1, R2, R3 and M1 and M2 are used to represent the reverberatory furnaces and matte tunnels. Logic and arithmetic operations performed in each event will now be described. Wherever variables are used they are subscripted and are calibrated differently for each furnace. Table C-2 lists and defines these variables.

1. (Event 41) A fixed smelting rate is represented by a given rate of matte level increase in each furnace, RMAT(NRE) in min/ins. This

event is initialised at the beginning of the simulation where the initial level of matte is specified as RMLEV(NRE). Every time this event is executed the matte level is increased by one inch. Based on the rate of matte production a future event is scheduled when the matte level will have increased again by one inch. The increments of one inch were chosen to produce an almost continuous variation of the matte level.

The matte level is compared to the allowable upper and lower limits. If any of these are reached the indicator lights "High Matte Level" or "No Matte Available" will be set to alarm the operator. At the same time slag or matte tapping is suspended, whichever occurs, high or low matte.

If the matte level has reached or exceeded its maximum allowable limit in all three furnaces a typeout will appear on the teletypewriter giving the operator the option of:

(a) Increasing the matte grade, which will reduce the matte production,

(b) Decreasing the fire in a furnace to directly reduce the smelting rate.

Individual changes to each furnace are allowable and the resulting changed rates are shown for verification. In the absence of data relating an increase of matte grade to the decrease in smelting rate, the following estimate is used:

-4 x (% change in matte grade) = % change of smelting rate

This equation expresses the response of the smelting rate as estimated by operating personnel.

2. (Event 42) When a crane returns an empty ladle to matte tunnel No.1, the variable describing the state of the matte car ITROL(ITUN) is checked. The states are given by:

ITROL(ITUN) = 1 - full ladle of matte on car at converteraisle

ITROL(ITUN) = -1 - car near matte tapping hole of rev.

ITROL(ITUN) = 0 - car empty at converter aisle

If ITROL(ITUN) = 1, or -1, no action is taken signifying that the ladle is placed on the floor. If ITROL(ITUN) = 0 it is assumed that the empty ladle is placed on the empty car which thereafter moves the ladle to the matte tapping hole. Matte is tapped if the "matte available" indicator is switched on in either reverberatory furnaces R1 or R2.

i.e., if RMLEV(NRE) > YMMIN(NRE)

Matte is tapped from both furnaces if matte is available in both. If no matte is available in one, matte is tapped in the other. If the matte level has exceeded the upper level in only one, matte is tapped there only. The time that passes before the filled ladle of matte arrives at the converter aisle again is sampled from the distribution shown in Figure C-1. The data shows times for tapping matte from both furnaces and it is assumed that tapping from one furnace takes twice as long, while the time of moving the car to and from the converter aisle remains the same. This time is a fixed variable given byTRAIL (ITUN) and could thus have been lumped into the distribution of tapping times. This was not done to account for breakdowns of the matte car which could delay the matte supply. In the absence of suitable statistics the effect was not considered significant but could be included in the model at any time.

If no matte is available in both furnaces tapping is suspended until the matte level has increased sufficiently. At that time tapping is resumed and the next time a ladle of matte becomes available is determined as described above.

3. (Event 43) The logic follows that of event 42 except that matte is only tapped from reverberatory furnace No.3. The distribution of tapping time is shown in Figure 2.

4. (Event 44) Together with a light indicator "Matte has arrived" a bell ringing on the teletypewriter will inform the model operator of the arrival of a ladle of matte at matte tunnel No. 1. At this time the decrease of matte level due to tapping matte in the reverberatory furnaces is recorded, while light indicators for "low matte" are set if applicable. Furthermore the total number of ladles tapped are recorded as well as the time a full ladle stands on the matte car at the converter aisle.

5. (Event 45) This event for matte tunnel No.2 is similar to event 44 except that it accounts for tapping of matte from reverberatory furnace No.3 only. It might be noted that here the human interaction, i.e., a command from the model operator, is necessary to complete the cycle of tapping

matte from the reverberatory furnace. In the actual operation a crane will be sent to pick up matte from a matte tunnel if there is one available and if a converter is ready to receive it. This same operation is performed in the model if a crane is ordered to pick up matte at a matte tunnel and to return the empty ladle thereafter. Onl y then will the next ladle of matte be filled and returned to the converter aisle to repeat the cycle.

In the automatic mode of operating the simulation model which is described elsewhere, the crane command is issued automatically after the need for matte is established and a converter is chosen to give a smooth converter operation.

6. (Event 46) Once a crane has picked up a ladle of matte at matte tunnel No.1, event 46 is triggered when the state of the matte car is set to "empty". This is also indicated by a green light that is turned on at the matte tunnel. The end of the period of waiting of a full ladle of matte on the car is recorded here.

Even 46 is also used if matte is picked up from the floor. That is, if

$$\text{IFLOR (1)} > 0$$

$$\text{and} \quad \text{ITROL(1)} \neq 1$$

i.e., if there is matte on the floor but no matte on the car, a ladle is picked up from the floor. If there were no ladles of matte left on the floor a light indicating this would be turned off. It is assumed that ladles of matte are always full, the quantity multiplied by 100 being stored by the variable

$$\text{ISAVE(NCR)} = 100 \ .$$

7. (Event 47) Event 47 is the equivalent of event 46 for ~~matte~~ tunnel No.2.

8. (Event 48) The increase of the slag level in the reverberatory furnaces is mainly due to the returned converter slag, however there is also a certain amount of slag produced in the reverberatory furnace. This uniform rate of slag production is represented by the variable

RSLG(NRE) in(ins./min.)

which is used to update the slag level in a way similar to the matte level. If the slag level reaches its lower or upper limit, a light indicator will inform the operator. Event 48 is also used to perform the operation of slag tapping.

The quantity of slag tapped varies due to variations in slag viscosity and due to blocking of tapping holes. Available statistics showed that the variations of tapping rates can be best represented by uniform distributions for each of the three furnaces.

9. (Event 49) If a crane returns converter slag to reverberatory furnace No. 1, the quantity of slag is given by the variable ISAVE(NCR) which is used to record the increase of the slag level in the furnace, SLEV(NRE). If the slag level exceeds the upper limit,

$$SLEV(NRE) > SMAX(NRE)$$

a light indicator will be turned on to inform the operator. On any subsequent occasion no converter slag can be accepted by the reverberatory furnace until the slag level has fallen below the limit. If the command of pouring slag into the reverberatory furnace is given while the "high slag level" indicator is turned on, a message will appear on the teletypewriter to inform the operator of this, (i.e., REVERB #1 CANNOT ACCEPT CV.SLAG.). Each ladle of converter slag that is poured into the reverberatory furnace is counted and stored in TSLAG(NRE).

10. (Event, 50, Event 51). The procedure of returning converter slag to reverberatory furnaces No. 2 and No. 3 is the same as that of No. 1.

It should be noted here that the distribution of converter slag between the reverberatory furnaces is performed by the model operator who can request information on the amount of slag returned to each at any time. If he wishes to distribute slag evenly or in any other fashion the statistics on the reverberatory furnace will allow him to control this as desired.

In the automatic operation a given distribution pattern can be specified according to which the converter slag will then be scheduled to return to the reverberatory furnaces.

11. (Event 52, Event 53) If a ladle of matte is removed from a matte car and placed on the floor near one of the matte tunnels the variable

IFLOR(ITUN)

will be incremented and an indicating light "matte on floor" will be turned on. Again this case is treated in a similar way for both matte tunnels.

C-4 STATISTICS ON REVERBERATORY FURNACES AND MATTE TUNNELS

Table C-3 shows a typical typeout as obtained upon requesting information on the reverberatory furnaces and the matte tunnels. The commands are INFOR(NRE) and INFOM(ITUN) respectively. All relevant quantities are listed as shown and give a full status report on the matte supply.

The matte level relative to its tolerance limits would indicate the amount of available matte while a high slag level could indicate that either or both reverb slag tapping was slow and converter slag was poured into the furnace frequently.

The matte level in a reverberatory furnace at any time t can be expressed as follows:

$$M(t) = f(t, r_m, q_i, l_i) \qquad \ldots.(1)$$

where $M(t)$ = matte level at time t (ins)

r_m = rate of matte smelting (ins/min)

q_i = quantity of matte tapped in a time interval t_i (this value is normally distributed) (ins)

l_i = logical condition during Δt_i (=1 if matte is required and available, = 0 if matte is unavailable or not required)

thus after the period $t_n = \sum_{i=0}^{n} \Delta t_i$ has elapsed

$$M(t_n) = M(t_o) + \sum_{i=0}^{n} r_m \Delta t_i - \sum_{i=0}^{n} q_i l_i \qquad \ldots.(2)$$

Equation (2) reflects the function of both equations (3) and (4) and the logical states as they occur in the course of simulation:

$$\text{RMLEV(NRE)} = \text{RMLEV(NRE)} + 1.0 \times \text{RMAT(NRE)} \qquad \ldots.(3)$$

$$\text{RMLEV(NRE)} = \text{RMLEV(NRE)} - \text{RATI(NRE)} \qquad \ldots.(4)$$

In the reverberatory furnace models these equations are executed whenever necessary and provide the matte level at any time.

In a similar way the slag level is represented by:

$$S(t) = f\,(t, r_i, r_s, \bar{q}_i, \bar{l}_i) \qquad \ldots.(5)$$

where $S(t)$ = slag level at time t (ins)

r_i = quantity of converter slag returned in Δt_i (ins)

r_s = rate of slag production in furnace (ins/min)

$\bar{q}_i$ = quantity of reverberatory slag tapped in time interval Δt_i (this value is uniformly distributed) (ins)

$\bar{l}_i$ = logical condition during Δt_i (= 1 if matte level is below its upper limit, = 0 if above upper limit)

and considering again an elapsed period of time to be made up of n discrete time intervals:

$$t_n = \sum_{i=0}^{n} \Delta t_i$$

then

$$S(t_n) = S(t_o) + \sum_{i=0}^{n} r_i + \sum_{i=0}^{n} r_s \,\Delta t_i - \sum_{i=0}^{n} \bar{q}_i \bar{l}_i \qquad \ldots.(6)$$

Equation (6) is represented in the model by equations (7), (8) and (9) and the state of the logical variable $\bar{l}_i$.

$$\text{SLEV(NRE)} = \text{SLEV(NRE)} + 1.14 \times \text{SLRA(NRE)} \times \text{ISAVE(NCR)} \quad \ldots.(7)$$

$$\text{SLEV(NRC)} = \text{SLEV(NRE)} + \text{RSLG(NRE)} \quad \ldots.(8)$$

$$\text{SLEV(NRE)} = \text{SLEV(NRE)} - \text{POT(NRE)} \times \text{UNFRM (STAPLE(NRE),} \cdot \text{STAPH(NRE)} \quad \ldots.(9)$$

The numerical constant in question (7) accounts for the difference in specific weights of reverberatory slag and converter slag.

In the case of slag tapping the level changes are not directly influenced by decisions of the model operator, i.e., a continuous but non-uniform rate of tapping is assumed as long as the matte level is not too high. If the matte level exceeds its upper limit slag tapping is discontinued to avoid the loss of high grade matte in the reverberatory slag.

The reverberatory furnace model described is based on material balances only. The input of concentrates or calcine to the furnace together with the rate of oil burning determines the smelting rate of matte and accounts for a small part of reverberatory slag production. Both the matte production and the slag production rates were obtained from operating statistics and proved to be a sensitive control variable for the reverberatory furnace.

Matte tapping from the furnaces is a critical operation as it directly affects the overall converter aisle performance. The operating statistics can be represented accurately in the model whereby a realistic rate of matte supply is produced.

Converter slag is returned to the furnaces in response to the operator commands. The operator will attempt to follow regular operating practice in distributing the slag between the furnaces. Reverberatory slag is tapped at a varying rate. The total quantity of matte processed and the Cu loss in the reverberatory furnace slag are the important quantities to consider when estimating the overall efficiency of smelter operation. When considering the advantages of operating at a higher grade matte the benefits would have to be weighed amongst others against the total Cu-losses as supplied by the reverberatory model.

While the level measurements might not reflect the exact volume of molten material in a bath of varying area, they were used to provide the operator with the same measurement that is used in reality.

During the model development and validation phases it was necessary to constantly revise aspects of the model. This was a very useful process since it gave a better understanding of all interactions involved,which when accounted for produced an accurate and meaningful reverberatory furnace operation.

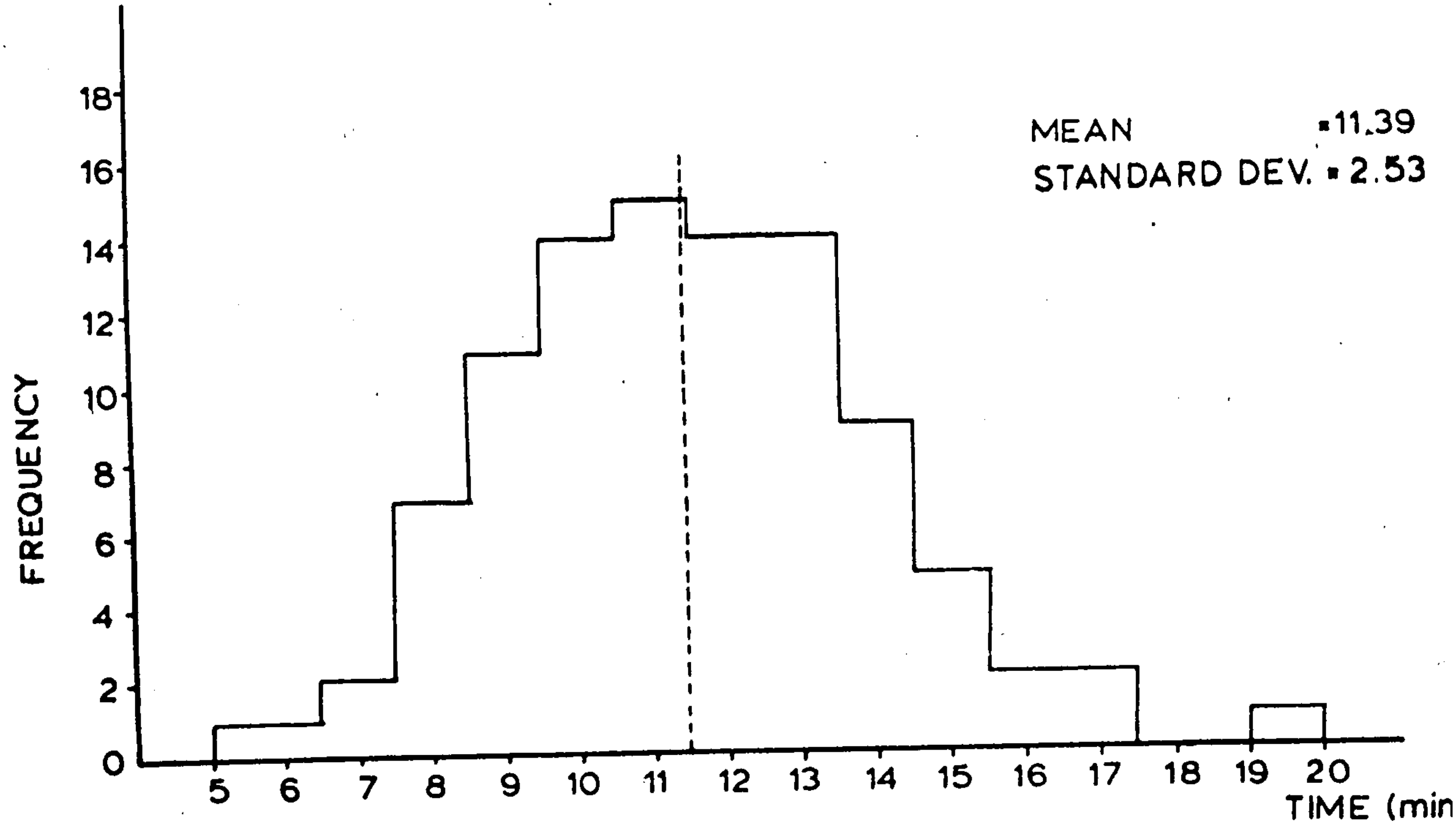


FIGURE C-1 DISTRIBUTION OF MATTE TAPPING DURATION FOR MATTE TUNNEL 1

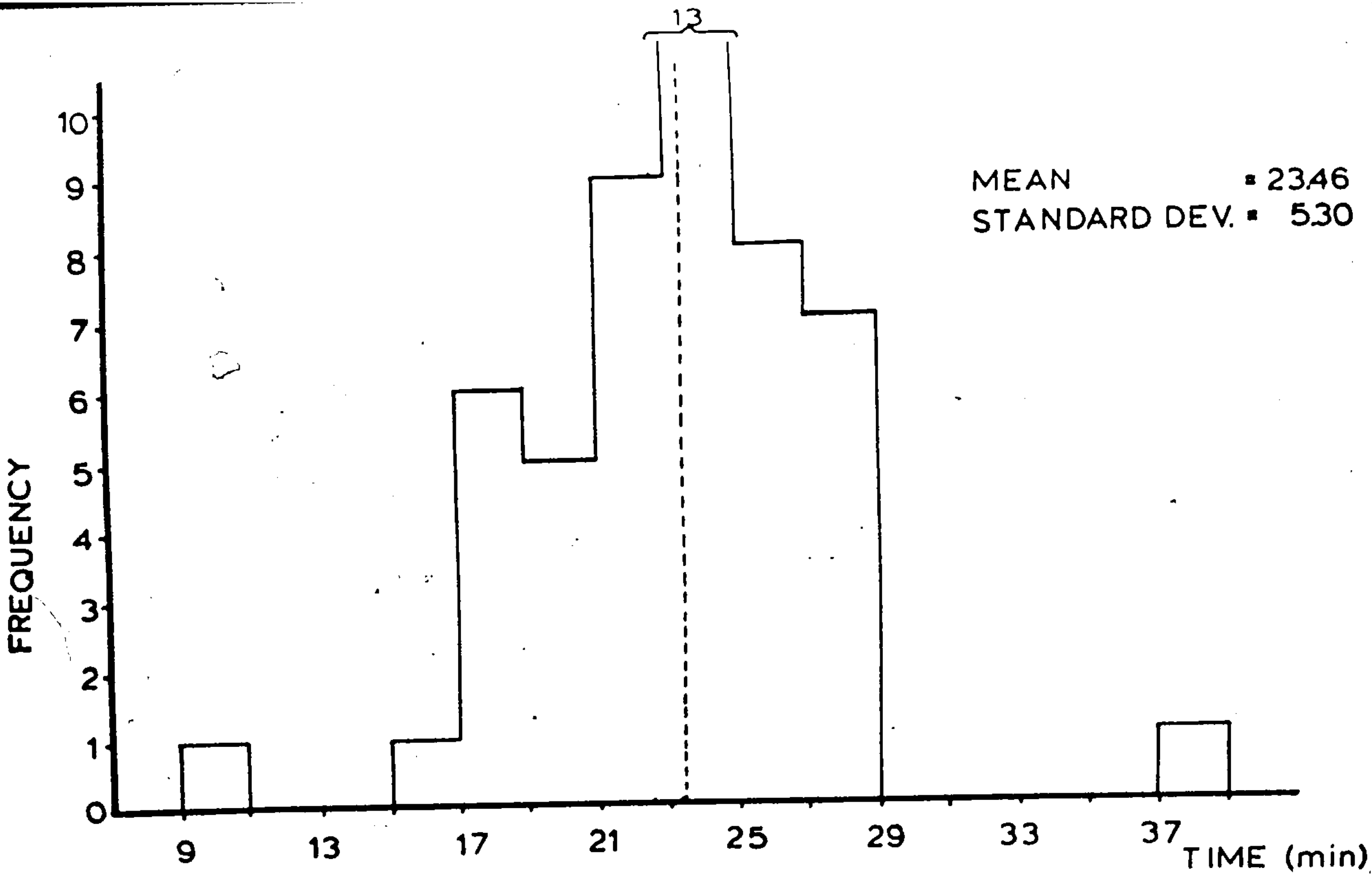


FIGURE C-2 DISTRIBUTION OF MATTE TAPPING DURATION FOR MATTE TUNNEL 2

TABLE C-1 - LIST OF REVERBERATORY FURNACE AND MATTE TUNNEL EVENTS

Event No.	Description of Event
41	Updating of matte level in R1, R2, R3
42	Tapping of matte is initialised at M1
43	Tapping of slag is initialised at M2
44	Full ladle of matte is available at M1
45	Full ladle of matte is available at M2
46	Ladle of matte is removed from car or floor at M1
47	Ladle of matte is removed from car or floor at M2
48	Slag level is updated in R1, R2, R3
	Slag skimming is performed in R1, R2, R3
49	Converter slag is returned to R1
50	Converter slag is returned to R2
51	Converter slag is returned to R3
52	Ladle of matte is deposited on floor near M1
53	Ladle of matte is deposited on floor near M2

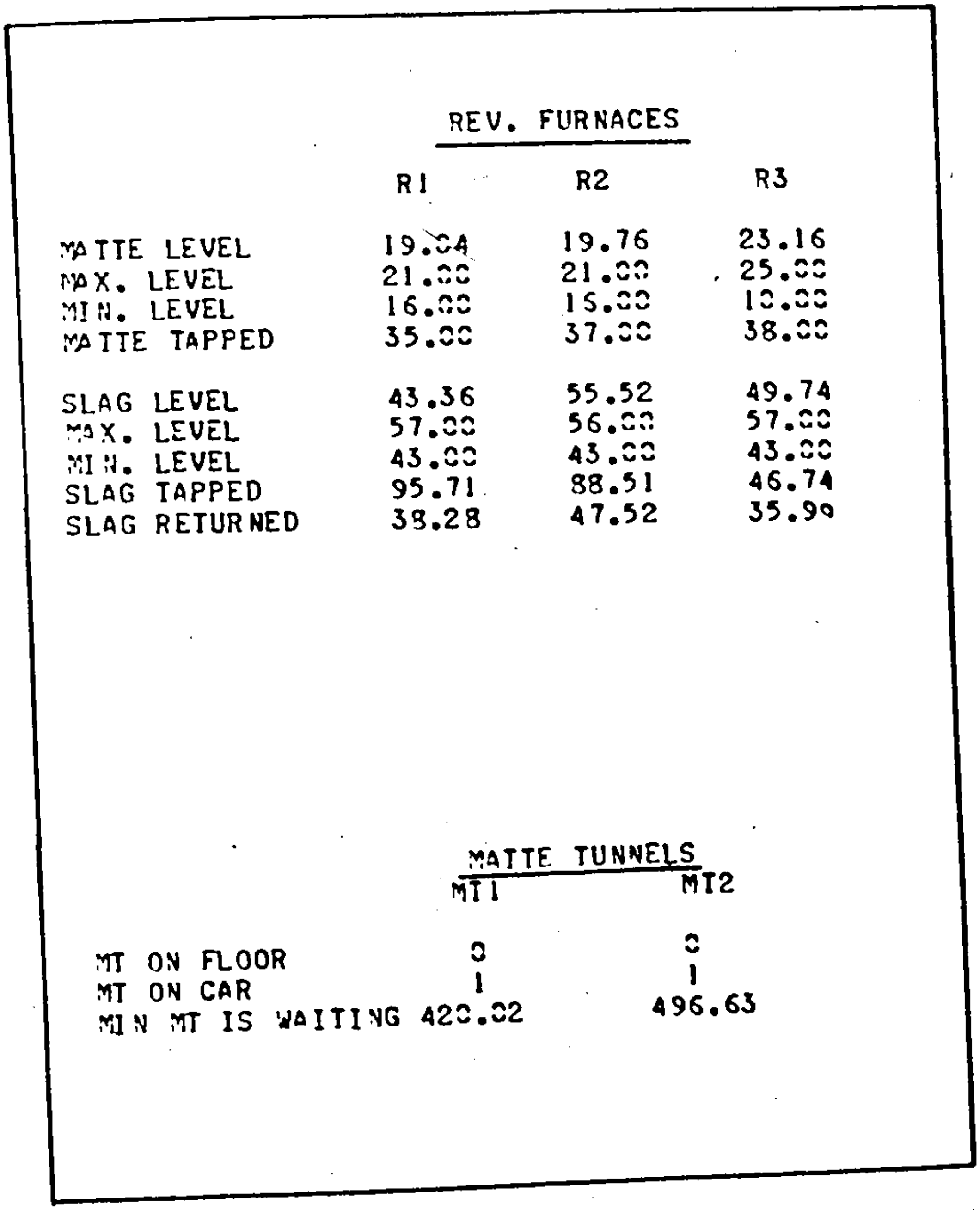
TABLE C-2 - DESCRIPTION OF REVERBERATORY FURNACE AND MATTE TUNNEL CONSTANTS AND VARIABLES

RMLEV(NRE)	-	relative matte level in furnace (ins)
YMMAX(NRE)	-	upper limit of matte level (ins)
YMMIN(NRE)	-	lower limit of matte level (ins)
RMAT(NRE)	-	rate of matte level increase (min/ins)
TR(NRE)	-	interval between matte level increases (min)
TLO(NRE)	-	mean duration of matte tapping (min)
THI(NRE)	-	standard deviation of time duration of tapping (min)
TOLA(NRE)	-	total number of matte ladles tapped
RATI(NRA)	-	conversation factor (ins matte/ladle)
SLEV(NRE)	-	relative slag level in furnace (ins)
SMAX(NRE)	-	upper limit of slag level (ins)
SMIN(NRE)	-	lower limit of slag level (ins)
STAPL(NRE)	-	lower limit of slag tapping rate (pots/min)
STAPH(NRE)	-	upper limit of slag tapping rate (pots/min)
TOTSL(NRE)	-	number of slag pots tapped
RSLG(NRE)	-	rate of slag level increase (in/min)
SLRA(NRE)	-	conversion factor (ins slag/ladle)
POT(NRE)	-	conversion factor (ins slag/pot)
TSLAG(NRE)	-	number of converter slag ladles returned
TAW(ITUN)	-	arrival time of matte car at converter aisle (min).
TOTW(ITUN)	-	total time full ladle stands on matte car (min)

TABLE C-2 (Contd.)

TRAIL(ITUN)	-	time duration of matte car transportation (min)
IFLOR(ITUN)	-	number of matte ladles on floor
ITROL(ITUN)	-	status of matte car
NSLAG(ITUN)	-	matte available at matte tunnels (automatic operation)
NRE = 1,3	-	reverberatory furnace numbers for R1, R2, R3
ITUN = 1,2	-	matte tunnel numbers for M1, M2

TABLE C-3 TYPICAL SIMULATION REPORT ON REVERBERATORY AND MATTE TUNNEL SUBMODELS



```
                              REV. FURNACES

                       R1            R2            R3

MATTE LEVEL          19.04         19.76         23.16
MAX. LEVEL           21.00         21.00         25.00
MIN. LEVEL           16.00         15.00         10.00
MATTE TAPPED         35.00         37.00         38.00

SLAG LEVEL           43.36         55.52         49.74
MAX. LEVEL           57.00         56.00         57.00
MIN. LEVEL           43.00         43.00         43.00
SLAG TAPPED          95.71         88.51         46.74
SLAG RETURNED        38.28         47.52         35.99



                         MATTE TUNNELS
                       MT1              MT2

MT ON FLOOR             0                0
MT ON CAR               1                1
MIN MT IS WAITING 420.02             496.63
```

APPENDIX D

CONVERTER SUBMODEL

D.1 INTRODUCTION

The converter model described here was developed by L.K. Nenonen of the National Research Council and the description that follows was taken from Reference (N-4).

The converter model consists of:

(i) Analog circuitry to represent ongoing chemical activity during blowing period as well as detection of end points.

(ii) A digital computer program module (Submodule 3) which handles material additions to the converter.

(iii) A digital computer program module (Submodule 6) which handles material removals from the converter.

(iv) A digital computer program module (Submodule 7) which responds to end point events, converter breakdowns, collar pulling, etc.

All materials transported by the cranes are specified in terms of their weight and their Fe, Cu, and S contents, the balance being considered slag. When specific material additions are made, these compositions and weights are used in submodule 3 to calculate the corresponding quantities of FeS, Cu_2S and slag if the converter is in its matte stage and Cu_2S, Cu and slag if in the copper stage. Digital-Analog converters are used to transmit the effects of the material addition to the analog computer. Similarly, when material is removed from a converter, using Submodule 6, the digital computer determines the corresponding quantities of FeS, Cu_2S, Cu or slag removed and updates the analog computer using the same Digital-Analog converters.

D.2 PROCESS CHEMISTRY AND CONVERTING RATES

The model of the matte stage behaviour accounts for the reactions:

$$2FeS + 3O_2 \rightarrow 2FeO + 2SO_2 \qquad \ldots.(1)$$

$$2FeO + SiO_2 \rightarrow 2FeO.SiO_2 \qquad \ldots.(2)$$

which give:

$$2FeS + 3O_2 + SiO_2 \rightarrow 2FeO . SiO_2 + 2SO_2 \qquad \ldots.(3)$$

The assumptions made are as follows:

(1) Any free Cu entering with input materials oxidizes instantaneously, according to:

$$4 Cu + 2FeS + O_2 \rightarrow 2Cu_2S + 2FeO$$

the resulting FeO combining with SiO_2 to form fayalite according to (2).

(2) Any free Fe entering with input materials oxides instantaneously according to:

$$2Fe + O_2 \rightarrow 2FeO$$

the resulting FeO combining with SiO_2 according to (2)

(3) Converter temperatures and fluxing conditions are such that magnetite is not produced according to:

$$6FeO + O_2 \rightarrow 2Fe_3O_4$$

(4) Referring to equation (3), equilibrium conditions are assumed to occur at all times such that the rate of reaction is determined entirely by the rate of supply of oxygen.

Additions of free Cu and Fe are accommodated by the model, the necessary adjustments being made in the material addition portion of the computer program.

To calculate the rate of supply of oxygen to the reaction, two input variables are required:

AIR = Measured air flow rate entering the converter, SCFM

EFF = Proportion of the air entering which actually reacts on its way through the converter. This efficiency factor accounts for loss of air through leakage and possible non-equilibrium conditions in the converter which lead to the escape of unreacted oxygen.

Thus, the rate at which 0_2 enters into reaction (3) is:

$$R0 = 0.21 \times \frac{Pa}{RT_s} \times AIR \times EFF \quad \text{pound-moles per minute}$$

$$R02 = .000545 \times AIR \times EFF \quad LBMM \qquad \ldots. (4)$$

The rates of change of all other reactants and products of reaction (3) can be determined based on R02.

$$RFES = -2/3 \times R02 \quad LBMM$$

$$RFES = -.00001597 \times AIR \times EFF \quad TPM \qquad \ldots.(5)$$

$$RS02 = +2/3 \times R02 \quad LBMM$$

$$RS02 = +.00000884 \times AIR \times EFF \quad TPM \qquad \ldots.(6)$$

$$R2FE0.SI02 = +1/3 \times R02 \quad LBMM$$

$$R2FE0.SI02 = +.0000185 \times AIR \times EFF \quad TPM \qquad \ldots. (7)$$

$$RSI02 = -1/3 \times R02 \quad LBMM$$

$$RSI02 = -.00000548 \times AIR \times EFF \quad TPM \qquad \ldots.(8)$$

However, converter flux additions may not be made at the stoichiometric rate given by (8). To permit use of non-stoichiometric rates in the model, a coefficient FRAT is introduced such that the actual rate of silica addition is

$$\mathrm{RSI02} = + .00000548 \times \mathrm{AIR} \times \mathrm{EFF} \times \mathrm{FRAT} \quad \mathrm{TPM} \qquad ....(9)$$

and when FRAT = 1, SiO_2 is added at exactly the rate demanded by reaction (2), when FRAT > 1, excess SiO_2 is added, and when FRAT <1, insufficient SiO_2 is added. Since the flux is actually 68% SiO_2, the total rate of flux added is:

$$\mathrm{RFLUX} = \frac{\mathrm{RSI02}}{0.68} = .00000804 \times \mathrm{AIR} \times \mathrm{EFF} \times \mathrm{FRAT} \quad \mathrm{TPM} \qquad ....(10)$$

of which the balance of 32% is assumed to be inert materials which enter the slag.

In practice, flux is added in an intermittent fashion to achieve converter temperature control. However, flux additions are made continuously in the model at the rate given by (10) whenever the converter is blowing in the matte stage. While the total flux addition per charge will be correct, the uniform distribution of flux addition over the entire matte stage will deviate somewhat from the distribution achieved in practice. The actual rate of change of slag in the converter can be determined from equations (7) and (10) as follows:

$$\mathrm{RSL} = \mathrm{R2FEO.SI02} + (\mathrm{FRAT} - 1) \times \mathrm{RSI02} + \mathrm{RFLUX} \times 0.32$$

where term (1) accounts for fayalite production by (2)

(2) accounts for SiO_2 not involved in reaction (2), i.e., non-stoichiometric SiO_2

(3) accounts for the inert materials entering with the flux.

Substituting

$$\begin{aligned}\mathrm{RSL} &= .0000185 \times \mathrm{AIR} \times \mathrm{EFF} + (\mathrm{FRAT} - 1) \times .00000548 \times \mathrm{AIR} \times \mathrm{EFF} \\ &\quad + .00000804 \times \mathrm{AIR} \times \mathrm{EFF} \times \mathrm{FRAT} \times 0.32 \\ &= .000013 \times \mathrm{AIR} \times \mathrm{EFF}\ (1 + .62\ \mathrm{FRAT}) \qquad .... (11)\end{aligned}$$

Equations (5) and (11) are the key rate equations solved in the model for converter in the matte stage.

During the copper or finish stage, incoming oxygen reacts with Cu_2S as follows:

$$Cu_2S + O_2 \rightarrow 2\ Cu + SO_2 \qquad \ldots.(12)$$

Therefore, the rates are

$$RCU2S = -\ R02\ LBMM$$

$$RCU2S = -.0000434 \times AIR \times EFF\ \ TPM \qquad \ldots.(13)$$

and

$$RCU = +2 \times R02\ \ LBMM$$

$$RCU = .0000346 \times AIR \times EFF\ \ \ TPM \qquad \ldots.(14)$$

Note that the rate of supply of oxygen to the reaction (12) is based on the same efficiency factor EFF as is used in the matte stage. Equations (13) and (14) are solved in the model for converters in the copper stage.

During the matte stage equations (5) and (11) are solved to produce FeS and slag.

Before solving or integrating these rate equations on an analog computer, decisions are required concerning amplitude scaling and time scaling. To transform the analog voltages to corresponding material quantities in tons, the variable TPU (NCV) is used, NCV being the converter number where NCV = 1 to 5 corresponds to converters 3 to 7. Simulation work to date has used BETA = 20, TPU(NCV) = 150 for all converters.

In the model program, the variables AIR and EFF are defined by elements of the two dimensional array CV as follows:

$$CV(NCV,6) = AIR \times 10^{-3}$$

$$CV(NCV,7) = EFF$$

Substituting into the rate equations for RFES, RSL, the equations to be integrated become

$$\text{RFES} = -.01597 \times \text{CV(NCV,6)} \times \text{CV(NCV,7)} \times \text{BETA/TPU(NCV)} \quad \ldots.(15)$$

$$\text{RSL} = +.013(1 + .62\ \text{FRAT}) \times \text{CV(NCV,6)} \times \text{CV(NCV,7)} \times \text{BETA/} \\ \text{TPU(NCV)} \quad \ldots.(16)$$

Similarly, in the copper stage, the following equations must be integrated:

$$\text{RCU2S} = -.0434 \times \text{CV(NCV,6)} \times \text{CV(NCV,7)} \times \text{BETA/TPU(NCV)} \quad \ldots.\ (17)$$

$$\text{RCU} = .0346 \times \text{CV(NCV,6)} \times \text{CV(NCV,7)} \times \text{BETA/TPU(NCV)} \quad \ldots.\ (18)$$

Actual solution or integration of these equations is achieved by the analog computer circuit of Figure D-1, identical for each converter model. Referring to Figure D-1, potentiometers, P_A, P_D are set to the values given by (15) and (16) while P_B and P_E are set to the values given by (17) and (18). P_F and P_G are set to 1/60 to convert units per minute to units per second, the computer time scales being based on units/second. These potentiometers are set at the start of a simulation period when all initial conditions are established.

Thus, once initial conditions have been established for a given blowing period via DA_A and DA_B, control lines A, B are employed to switch the integrators to the operate mode where the outputs of I_A, I_B will change at rates determined by P_A, P_D in the matte stage or P_B and P_E in the copper stage.

D.3 RESPONSE OF BASIC CONVERTER MODEL TO DISCRETE EVENTS

Discrete events can influence these analog computer circuits via the DA or AD converters, comparators, control lines or computer mode control status lines. The types of discrete event which arise are:

(1) Material additions

(2) Material removals

(3) Start and end points of individual blowing periods

(4) Equipment maintenance and breakdowns

Such events can be triggered by operator commands, by commands from the digital computer automatic decision making algorithms, or by changes in state of comparators in the analog computer itself. In particular, when FeS or Cu_2S in a blowing converter reaches 1 ton, this is sensed by the corresponding comparator (C_A in Figure D-1) which switches from a TRUE state to a FALSE state and in so doing, triggers a sequence of other events leading to the switching of the corresponding integrators I_A, I_B to the HOLD mode from the OPERATE mode. In the HOLD mode, voltages on I_A, I_B are held for recording until further action is taken arising from a subsequent material removal or addition event.

The following converter material additions are classified by elements of the IADD (JIO,4):

JIO

(1) Hot Charge Matte from matte tunnel 1

(2) Wet Charge Matte from matte tunnel 2

(3) Transfer matte

(4) Copper pigs

(5) Scrap

(6) Reactor slag

(7) Flue dust

(8) Skulls

(9) High copper slag

Each category of addition is identified by a code JIO and is assigned four attributes: weight, % Cu, % FE, and % S.

Compositions of material removed from the converters are also defined by elements of the IADD array:

<u>JIO</u>

(3) Transfer matte

(9) High copper slag

(10) Regular slag

(11) Blister copper

Other converter model variables are given in Table D-1.

In the model, composition of transfer matte removed from a particular converter is calculated based on the composition of the converter matte, and the corresponding IADD array elements are updated. Thus, when the transfer matte is added to a second converter, these updated compositions are used. However, fixed compositions are used for high copper slag, regular slag, and blister copper.

Fifteen operating states are defined by the variable ISTAT(NCV) where NCV is the converter number. These states are as follows:

(1) Filled for a matte blow and blowing

(2) Blowing on copper

(3) End of a matte blow, slag to be skimmed

(4) End of a matte stage, slag to be skimmed

(5) End of a copper stage, blister copper to be removed

(6) Blowing on matte, able to take more matte

(7) End of matte stage, slag skimmed, waiting for flue dust before starting copper stage.

(8) End of copper stage, skull to be added before tapping blister copper

(9) End of matte blow, slag has been skimmed, waiting for matte before blowing can start

(10) Not used

(11) Blowing on copper, cold copper needed

(12) Collar being pulled

(13) Converter down due to equipment breakdown or rebricking

(14) End of a matte blow, slag can be skimmed or more matte added and current blow continued

(15) Gowing high, at least one ladle of slag can be skimmed

The classification of material addition events is given above and the event numbers for the sine possible categories is given in Appendix F. Corresponding to each event number a material addition event causes a branch to submodule 3 where the following operations are performed:

1. Branch to respective event statement number.

2. Establish legality of command. Depending on the converter status ISTAT (NCV) and the material category JIO the matrix IO(JIO, ISTAT) transfers action to one of the 9 categories of legal material addition. If an illegal command was issued a message is typed out and the command is ignored by branching to the RETURN statement.

3. Based on the material category the respective converter model variables and the analog circuit is updated to account for the addition. Based on the material compositions as given by array IADD the quantities are updated for a matte blowing stage or a copper blow.

4. During a matte stage every addition causes a check whether the accumulated quantity of copper (Cu_2S) has reached a critical level after which a copper blow can be initialised. If this condition is satisfied a message is typed out which gives the operator the choice. If he decides to proceed with a copper blow the present blow will be the last matte blow, the going-high blow before proceeding with a copper stage.

Two further events are treated in submodule 3, the start of a slag blow in a converter analog model and the start of a copper blow. The first occurs when sufficient matte has been added for a particular blow while a copper blow always starts when a ladle of flue dust has been added after all the slag is skimmed in the going-high blow.

If a material removal event occurs a branch to submodule six follows, where the four categories of material removal are treated in a similar fashion as additions are recorded in submodule 3.

The difference being the checks of material quantities remaining after a removal. For each type of material removed one component is more critical than the other, i.e. Cu_2S for transfer matte,and slag is important in the slag skimming operation irrespective of other quantities. However, removal of a particular material category causes a removal of the correct proportions of each of its constituents. The composition is again given by the array IADD.

In addition to the removal events an event recording the presence of an empty ladle at a converter and an event to start a blow after skimming slag during the going-high blow is treated here. Whenever converter slag is skimmed from a converter this is rerouted to a converter instead of being returned to a reverb. furnace.

Submodule seven consists of ten types of converter events, these are also shown in Appendix F. These events respond to beginnings and end points of processes.

The end point of a blowing period is given by the analog model of a converter and depending on the converter status given by ISTAT(NCV) it will indicate the end of a copper or a slag blowing period.

In the copper blow the demand for cold copper pigs is created in submodule seven.by filing an event at time intervals, that are determined at the beginning of a copper blow. In a similar way an event is filed that indicates when enough slag has accumulated in the going high blow to warrant skimming. In both cases light indicators are turned on to which the operator can respond.

Collar pulling results in an addition of material to a converter which is treated by two events, the start and end of collar pulling. Converter breakdowns are treated by digital models utilising breakdown and repair statistics.

One further event to zero converter arrays IAFO, IRFO and MT is executed whenever a new converter cycle starts after removing all the copper or all transfer matte from a converter.

The three arrays together with the material quantities in the converters at any time,given by array CV, provide a complete status report on all converters. This information is automatically provided at the end of every converter cycle or upon request. A typical typeout is given on Table D-2.

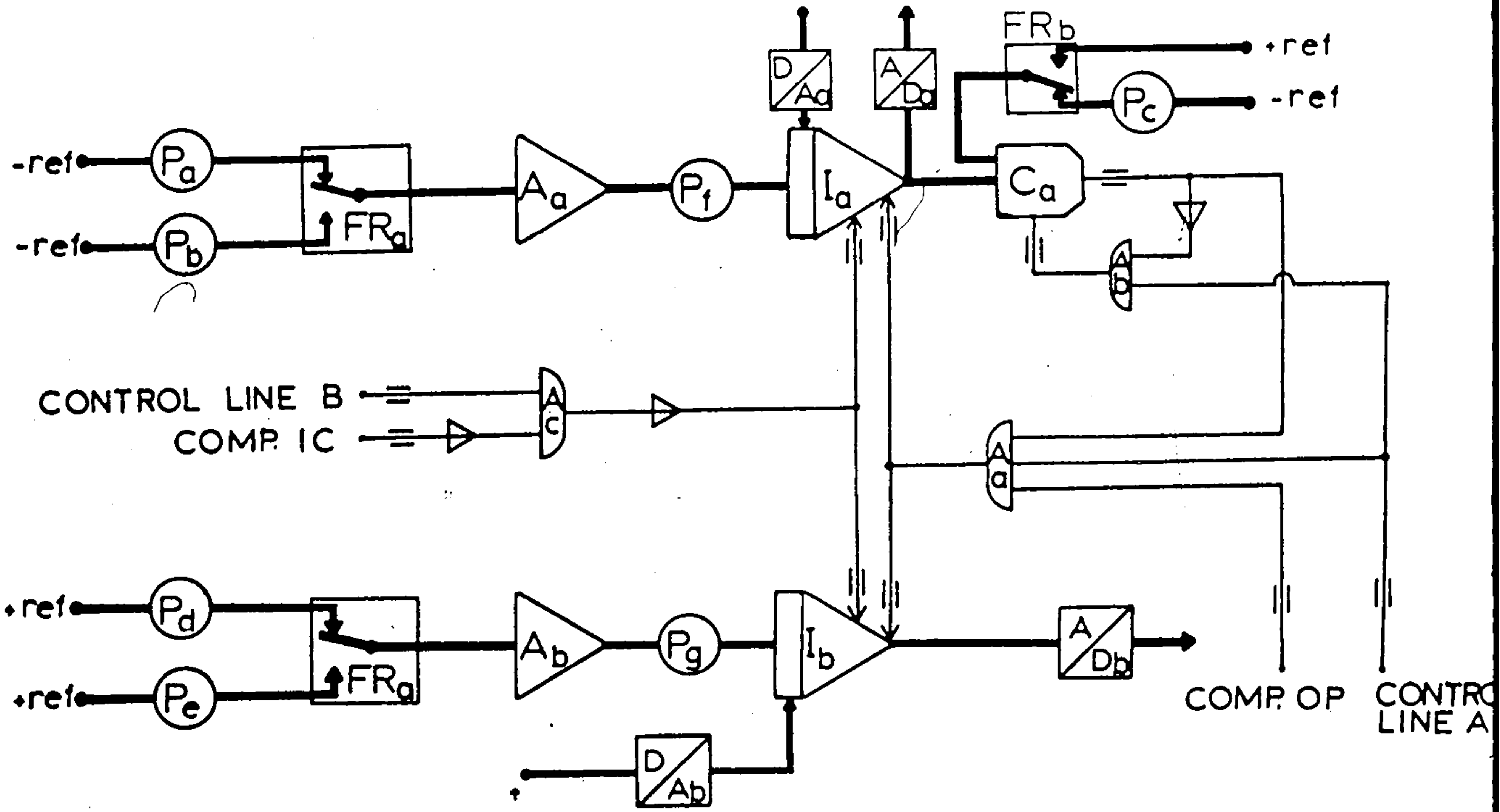
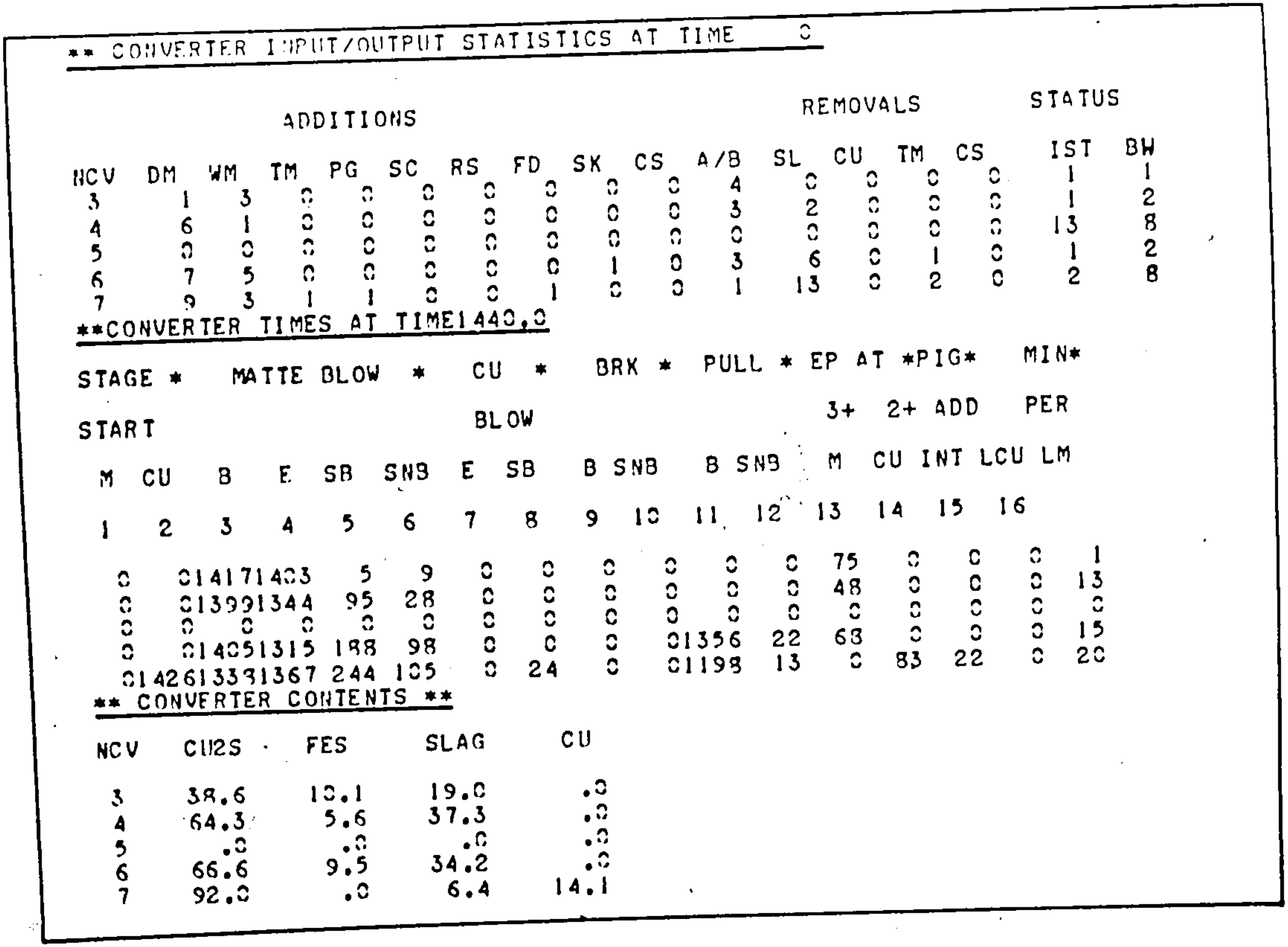


FIGURE D-1 ANALOG CIRCUIT FOR CONVERTER MODELS

ARRAY NAME	INDEX I	INDEX J	DESCRIPTION
CV(I,J)	Converter No.		Converter air flow, efficiency, capacity, current contents
IADD(I,J)	Input material category		Weight, % Cu, Fe, and S
IO(I,J)	Input or output material category	Converter status	Validity check and program directory for converter material additions or removals
IAFO(I,J)	Converter No.	Input material category	Record of material additions during current converter charge
IRFO(I,J)	Converter No.	Output material category	Record of material removals during current charge cycle
MT(I,J)	Converter No.		Record of converter event times during current charge cycle
ICV(I,J)	Converter No.		Computer Hardware assignments
PCV(I,J)	Converter No.		Potentiometer assignments
IREP(I,J)			General Converter Data Storage array

TABLE D-1: DESCRIPTION OF CONVERTER MODEL VARIABLE ARRAYS



```
** CONVERTER INPUT/OUTPUT STATISTICS AT TIME      0

                    ADDITIONS                             REMOVALS        STATUS

NCV  DM  WM  TM  PG  SC  RS  FD  SK  CS  A/B  SL  CU  TM  CS    IST  BW
 3    1   3   0   0   0   0   0   0   0   4    0   0   0   0     1    1
 4    6   1   0   0   0   0   0   0   0   3    2   0   0   0     1    2
 5    0   0   0   0   0   0   0   0   0   0    0   0   0   0    13    8
 6    7   5   0   0   0   0   0   1   0   3    6   0   1   0     1    2
 7    9   3   1   1   0   0   1   0   0   1   13   0   2   0     2    8
**CONVERTER TIMES AT TIME1440.0

STAGE *   MATTE BLOW   *   CU   *   BRK *  PULL * EP AT *PIG*   MIN*

START                     BLOW                    3+  2+ ADD    PER

 M  CU   B   E  SB  SNB  E  SB   B SNB   B SNB   M  CU INT LCU LM

 1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16

  0  014171403   5   9   0   0   0   0     0   0  75   0   0   0   1
  0  013991344  95  28   0   0   0   0     0   0  48   0   0   0  13
  0   0   0   0   0   0   0   0   0   0     0   0   0   0   0   0   0
  0  014051315 188  98   0   0   0  01356  22  68   0   0   0  15
 014261333 1367 244 105   0  24   0  01198  13   0  83  22   0  20
** CONVERTER CONTENTS **

NCV   CU2S    FES    SLAG    CU

 3    38.6   10.1   19.0    .0
 4    64.3    5.6   37.3    .0
 5      .0     .0     .0    .0
 6    66.6    9.5   34.2    .0
 7    92.0     .0    6.4  14.1
```

TABLE D-2   TYPICAL SIMULATION REPORT OF CONVERTER FURNACES

APPENDIX E

ANODE FURNACE AND MISCELLANEOUS MODELS

E-1 ANODE FURNACE SUBMODEL

The submodels described below were developed by Dr. U.P. Graefe of the National Research Council and the description is taken from Reference G-4.

The operation cycle of an anode furnace consists of filling the furnace with copper, a ladle or less at a time until its holding capacity is reached or until the operator decides to start refining. The refining time is assumed to be proportional to the number of ladles of copper in the furnace. Upon completion of refining, the furnace awaits a casting command from the operator. Casting can only take place if an anode wheel is available. Furnace 6 casts onto anode wheel No.3 while furnaces 4 and 5 share anode wheel No.4. The actual casting time depends on the furnace and the quantity of copper in the furnace. After completion of casting a certain amount of time is required to get the furnace ready to accept copper again.

The furnaces can thus be modelled by the following six types of events:

1. Event 30 to 32, Pouring of copper into anode furnace 4, 5 or 6 respectively
2. Event 34, Start the refining process
3. Event 35, Furnace ready for casting
4. Event 36, Start casting
5. Event 37, Finished casting
6. Event 38, Anode furnace ready for copper input.

To prevent issuing the refined and cast commands at the wrong time during the anode furnace cycle, the furnace and anode wheel states are saved in the three element array IAFFL (one element per furnace) and the two element array IAWHL (one element per anode wheel). The states are coded as follows:

IAFFL(I) = 0 when waiting for copper
= 1 when full
= 2 when refining
= 3 when finished refining
= 4 when casting
= 5 when finished casting

IAWHL (J) = 0 when not casting
= 1 when casting

where I = 1,2,3 for furnaces 4, 5 and 6 respectively
and J = 1,2 for anode wheel 3 or 4 respectively

Pouring of copper into a furnace results in updating of the amount of copper held, provided of course, that the furnace is in state 0. The 'more copper required' and 'furnace full' lights are switched appropriately and the furnace state is set to 1 when the present level differs by less than one ladle from full capacity.

A refining command for a particular furnace results in a check to see if the furnace is indeed ready to refine. If so, appropriate status lights and furnace state are changed, and an event No. 35 is scheduled using a furnace dependent uniform distribution and the amount of copper in the furnace to determine the refining time.

After completion of the refining process event 35 is executed which notifies the operator that the particular anode furnace is ready for casting. The operator originated casting command results in checking for the correct furnace state and availability of an anode wheel. If the furnace is indeed ready to cast, and a wheel is available appropriate status indicators are set and an event No. 37 is scheduled to occur at a time derived from a furnace dependent uniform distribution and the amount of copper held in the furnace.

Completion of casting results in setting appropriate status indicators and anode wheel and furnace states. The total number of anodes cast is updated by the number of anodes just cast, and event No. 38 is scheduled, using a furnace dependent uniform distribution for the time of occurrence of this event.

When event No. 38 is executed the furnace status is reset to 0, the copper content is set to zero, the 'furnace ready for copper' light is turned on, and the operator is notified. Table E-1 shows the format of information as obtained from the simulation model in response to the command INFOA (4,5,6). Also shown on Table E-1 is the information on the various other submodels described in the next sections.

E-2 PILOT PLANT SUB-MODEL

Operations of the pilot plant as seen from the converter aisle consists of the production, from time to time, of two copper cakes and pilot plant slag. The copper cakes have to be removed by a crane. A full ladle of pilot plant slag has to be picked up, and after emptying the slag into a converter, the empty slag ladle has to be returned. Thus, the whole pilot plant can be modelledthrough the following five events:

1. Event 1, copper cakes are ready to be picked up.
2. Event 3, copper cakes have been picked up
3. Event 4, slag is ready to be picked up
4. Event 6, slag has been picked up.
5. Event 7, an empty slag ladle has been returned to the pilot plant.

Briefly, when copper cakes are ready the operator is notified, the total number of cakes is updated and the next event No. 1 is filed using uniform distribution for the cake production time.

The pick-up of cakes involves checking to see that cakes are indeed available for pick-up, and notifying the operator when all available cakes have been picked up.

When slag is ready the operator is notified of this fact and the next such event is scheduled at a future time determined from an appropriate uniform distribution.

Pick-up of slag involves some bookkeeping and changing the light indicator status for the pilot plant.

The return of the slag ladle just causes the 'empty ladle required' light to be turned off.

E-3 COPPER PIGS

Copper pigs are produced by pouring blister copper into moulds. There are 30 such moulds in the pig bay and one ladle of copper produces about five pigs. After cooling for approximately three hours the pigs are ready for pick-up. They may be dumped either into a converter or transported to the pig pile. Pigs on the pile may be picked up later for dumping into a converter. Thus the copper pig submodel can be represented by five events.

1. Event 21. pigs have been poured
2. Event 27. Some pigs have cooled sufficiently to be picked up
3. Event 23. Pick.up of pigs from pig bay.
4. Event 24. Dump pigs into the pig pile.
5. Event 26. Pick up of pigs from pig pile .

Before pigs can be poured, a check is made to see if there are enough mculds available. If, after filling all available moulds there is still some copper left, the operator is notified of the amount of copper left. As soon as more than 26 moulds are filled, the 'moulds available" light is turned off. Pouring of copper pigs also results in scheduling an event No. 27 to occur at a fixed time TPCOOL minutes from now.

When the pigs have cooled sufficiently event No. 27 occurs which increments the number of cold pigs available and notifies the operator of the fact that cool pigs are available.

Pick-up of pigs from the pig bay results in a check to see if the required number of pigs is available. If all pigs have been picked up, the 'pigs available' light is turned off, and the 'moulds available' light is turned on as soon as less than 26 moulds are occupied.

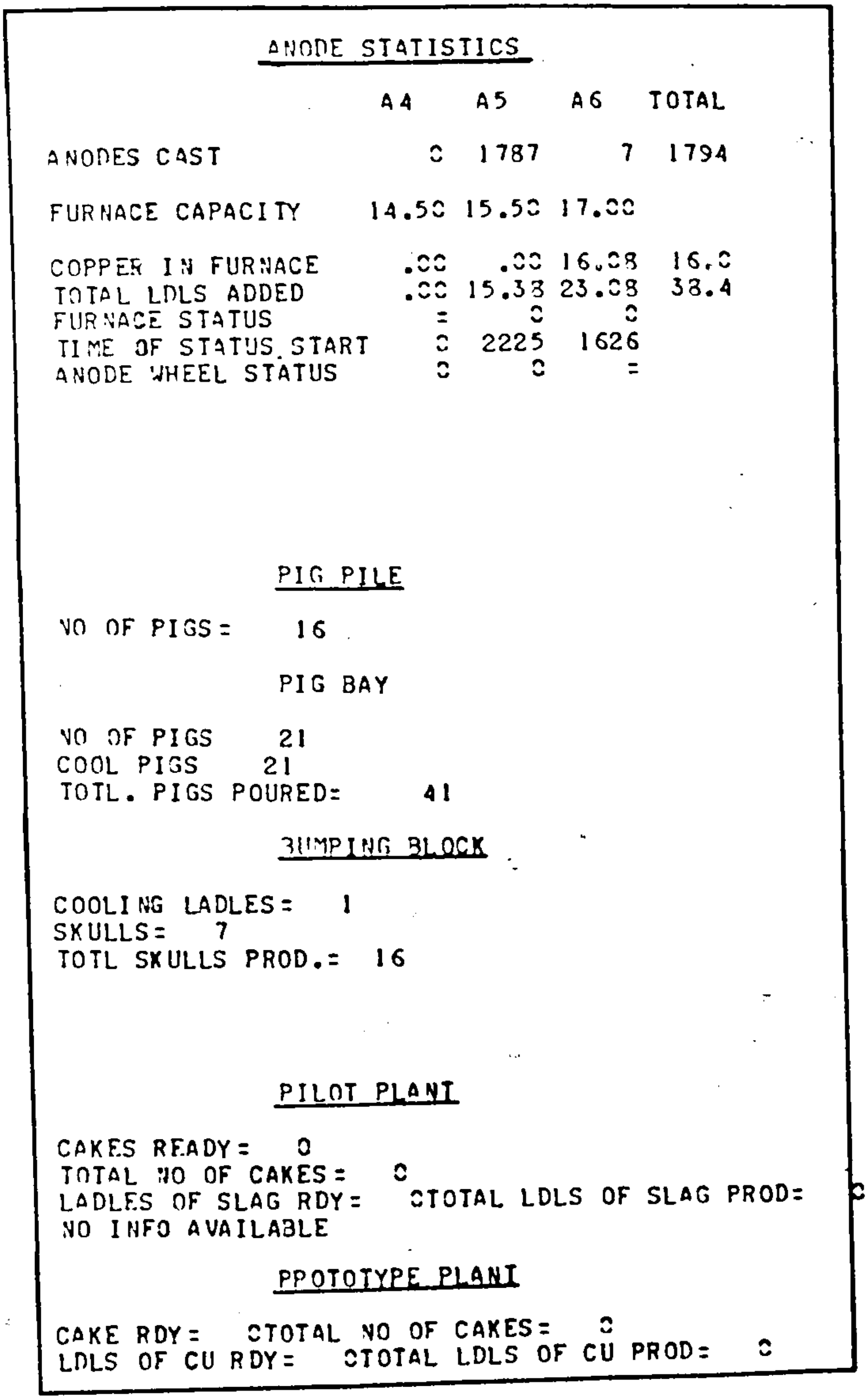
Dumping of pigs or copper cakes onto the pig pile results in updating the number of pigs on the pig pile. Note that no distinction is made here between copper pigs or copper cakes. As long as there is at least one pig on the pig pile, the 'pigs available' light is on. Pick-up of pigs from the pig pile results in a check to see if the required number of pigs is available, and decrementing the number of pigs held on the pig pile.

E-4 LADLES - BUMPING BLOCK - SKULLS

No attempt is being made to keep track of individual ladles. Whenever a ladle is filled with matte, any type of slag, transfer matte, or copper, a variable NLC which counts the number of ladle usages, is incremented by 1. When this variable reaches the value NFL which is an average number of ladle usages between bumping, a light is turned on to inform the operator that a ladle (any ladle) requires bumping. The operator may now deposit a ladle near the bumping block for cooling. This constitutes event No. 13. After having cooled for some time, the operator may command a crane to bump the ladle, so gaining a skull in the progress. This operation is performed in event No. 11.

This submodel keeps track of the number of skulls available at the bumping block. Execution of event No. 11 adds a skull and execution of event No. 17, the 'completion of a skull pick-up' removes a skull from the supply. Light No. 5 which indicates the availability of skulls, is switched off and on appropriately.

TABLE E-1. TYPICAL SIMULATION REPORT ON ANODE FURNACE AND MISCELLANEOUS SUBMODELS.



```
                     ANODE STATISTICS

                              A4     A5     A6   TOTAL

ANODES CAST                    0   1787      7   1794

FURNACE CAPACITY           14.50  15.50  17.00

COPPER IN FURNACE            .00    .00  16.08   16.0
TOTAL LDLS ADDED             .00  15.38  23.08   38.4
FURNACE STATUS                 =      0      0
TIME OF STATUS START           0   2225   1626
ANODE WHEEL STATUS             0      0      =




                     PIG PILE

NO OF PIGS=    16

                     PIG BAY

NO OF PIGS    21
COOL PIGS    21
TOTL. PIGS POURED=     41

                     BUMPING BLOCK

COOLING LADLES=   1
SKULLS=   7
TOTL SKULLS PROD.=  16



                     PILOT PLANT

CAKES READY=   0
TOTAL NO OF CAKES=   0
LADLES OF SLAG RDY=   0TOTAL LDLS OF SLAG PROD=   0
NO INFO AVAILABLE

                     PROTOTYPE PLANT

CAKE RDY=   0TOTAL NO OF CAKES=   0
LDLS OF CU RDY=   0TOTAL LDLS OF CU PROD=   0
```

APPENDIX F

In this Appendix the following reference lists for the overall model are given:

F.1 LIST OF SUBROUTINES

NAME	FUNCTION PERFORMED	SUBMOD.# OF FILE NAME	PAGE
ANODE	Searches for best anode fumace to receive Cu	9	24
ANTIME	Provides duration of anode fumace event	CRANE	29
BELLS	Rings bell of teletypewriter	CRANE	32
CDCK	Checks correctness of next event in crane tasks	CRANE	29
CFILE	Encodes crane task variables and files crane events	CRANE	32
CHECK	Checks priority of crane event and crane no.	CRANE	32
CLOCK	Converts simulation time (TIME) to hours and minutes	CRANE	33
CMNDG	Generates and files commands	4	11
CONCHK	Checks converter status	4	12
CONTRL	Sets and resets control lines	CRANE	32
CONV	Finds best converter to receive material	9	24
COREAD	Enables reading from buffer array core	CRANE	33
CRANE	Finds best crane	9	23
CRA1	Establishes feasibility of move by crane 1	CRANE	30
CRA2	Establishes feasibility of move by crane 2	CRANE	30
CRA3	Establishes feasibility of move by crane 3	CRANE	30
CVFORC	Produces forecast of matte requirements of converters.	11	27

NAME	FUNCTION PERFORMED	SUBMOD # OF FILE NAME	PAGE
DATAN	Initialises GASP simulation, reads GASP data	GASP	34
DRAND	Produces a pseudo random number	CRANE	33
ERROR	Types ERROR if GASP error occurs	GASP	35
EVNTS	Branches to a corresponding event of a submodel	MAIN	3
EXIT	Calls EXIT after GASP error occurs	GASP	35
EXEC	Reads any submodule from disc	CRANE	33
FILEM	Files GASP type event in any file	GASP	34
FIND	Finds event in a file	CRANE	29
FLEFND	Checks if all submodules exist on disc	MIAN	3
FNDCEV	Finds any crane event issued but not executed	9	24
GASP	Performs simulation control	MAIN	1
GETEV	Removes crane event from file and decodes event variables	CRANE	29
GOWT	Sets analog components of crane models	CRANE	30
GWAIT	Files crane events with increasing rank	CRANE 3	31
LADLE	Submodel of ladles - records usage of filled ladles	CRANE	32
LIGHT	Sets or resets light indicators on display unit	CRANE	32
LOCFND	Finds device location from corresponding code	CRANE	29
MATTE	Provides information on present availability of matte	9	24

NAME	FUNCTION PERFORMED	SUBMOD# OF FILE NAME	PAGE
MTFORC	Produces forecast of matte availability	12	28
QRBADR*	Reads analog to digital converter value		
QRSLL*	Reads logical signal of sense line		
QRCPL*	Reads comparator logic state		
QSIC*	Selects analog computer mode Initial Condition		
QSSECN*	Selects analog time scale SEC.NORMAL		
QSDLY*	Causes analog delay period		
QWFRL*	Sets function relay true or false		
QWJDAR*	Sets value on digital to analog converter		
QWPR*	Sets value on potentiometer		
QSH*	Selects analog computer mode	HOLD	
QSOP*	Selects analog computer mode	OPERATE	
RD	Accepts operator commands from typewriter input	MAIN	2
REV	Selects best reverberatory furnace for conv.slag	9	
RMOVE	Removes GASP type event from a file	GASP	34
RNORM	Finds a normal variate of a given distribution	CRANE	32
SET	Updates filing pointers for GASP array NSET	GASP	34
SETEST	Accepts specific operator command for automatic operation (debug)	MAIN	3
SETIME	Sets timing circuit	CRANE	29
SLTEST	Reads all logic links of analog submodels and sense switch (1)	CRANE	29
SORT	Finds best furnace if tie exists	9	24
STATS	Produces typeout of information on any submodel	5	13

NAME	FUNCTION PERFORMED	SUBMOD # OR FILE NAME	PAGE
TESTFL	Finds first time event and resets timing circuit if required	CRANE	29
TCSORT	Sorts material requirements chronologically	11	27
TISTAT	Finds converters in a particular status	9	24
TIME	Computes simulation time from analog timing	CRANE	31
TKODE	Delays crane operation for synchronisation of "two crane" operation	MAIN	2
TWOCR	Selects combination of two cranes for a task	9	23
UNFRM	Finds uniform variate of a given distribution		33
*	System linkage routines used to set and read analog components.		

F.2 LIST OF EVENTS

Event No.	Submodule	Event
1	2	Copper cakes ready
2	2	Copper cakes ready at prototype plant
3	2	Cakes have been picked up
4	2	Slag ready
5	2	Copper cakes picked up at prototype
6	2	Slag has been picked up
7	2	Empty slag ladle has been returned
8	2	CU ready at prototype
9	2	CU picked up at prototype
10	2	Ladle returned to prototype
11	2	Completion of bumping
12	2	Pick up of scrap or flue dust
13	2	Ladle deposited for cooling
14	2	Anode slag picked up at A4
15	2	Anode slag picked up at A5
16	2	Anode slag picked up at A6
17	2	Skull has been picked up
18	2	Pull collar on A4
19	2	Pull collar on A5
20	2	Pull collar on A6

Event No.	Submodule	Event
21	2	Pigs have been poured.
22	2	Pick anode collar at A4
23	2	Picked up pigs at pig bay
24	2	Dumped pigs on pig pile
26	2	Picked up pigs at pig pile
27	2	Some pigs have cooled
28	2	Pick anode collar at A5
29	2	Pick anode collar at A6
30	2	Poured copper into anode F. No.4.
31	2	Poured copper into anode F. No.5.
32	2	Poured copper into anode F. No.6.
33	2	Mould available at pig bay
34	4	Start refining
35	2	Furnace 4, 5 or 6 ready for casting
36	4	Start casting on a wheel No. 4 or 3
37	2	Finished casting on a wheel No. 4 or 3.
38	2	Furnace 4, 5 or 6 ready for copper input
40	8	Scheduling and forecasting for automatic operation
41	8	Matte level rises in reverb.furnace
42	8	Empty ladle arrives at matte Tun. No. 1.
43	8	Empty ladle arrives at matte Tun. No.2.
44	8	Full ladle arrives at Tun. No. 1
45	8	Full ladle arrives at Tun. No.2.
46	8	Crane removes full ladle from No. 1.

Event No.	Submodule	Event
47	8	Crane removes full ladle from No.2
48	8	Slag is skimmed from reverb.
49	8	Converter slag is retumed to reverb.1
50	8	Converter slag is returned to reverb.2
51	8	Converter slag is retumed to reverb.3
52	8	Deposit full ladle of matte near Mt.1
53	8	Deposit full ladle of matte near Mt.2
55	8	Crane failure
56	8	Daily maintenance scheduled
57	8	Execute daily maintenance
58	8	End of shift
59	8	Time display is updated
60-64	3	Add scrap
65-69	3	Add reactor slag
70-74	3	Add pigs
75-79	3	Add high copper slag
80-84	3	Add wet charge reverb matte
90-94	3	Add dry charge reverb matte
95-99	3	Add transfer matte
104	3	Start copper blow
105-109	3	Add flue dust
110-114	3	Add skull
115-119	3	Start slag blow
170-174	6	Remove high copper slag (automatic mode only)
175-179	6	Remove regular slag

Event No.	Submodule	Event
180-184	6	Remove transfer matte
185-189	6	Remove blister copper
190-194	6	Return empty slag ladle to converter
200	6	Start blowing after going-high skim
250-254	7	End of slag stage
255-259	7	Start converter breakdown
260-264	7	End of converter breakdown
265-269	7	Cold CU needed
270-274	7	Indicate that slag may be skimmed on going high.
275-279	7	Start collar pulling
280-284	7	End collar pulling
285-289	7	End of a blowing period
290-294	7	End of finish stage
295-299	7	Zero arrays for next charge

F.3 VARIABLES IN 'NAMED COMMON' BLOCKS

List of Variables in Common Block BLOK1

PO(3)	Present crane position.
WT(3)	Present crane task time.
IP(3)	Present crane task priority.
SP(3)	Crane speed for initialization.
	Crane delay time during simulation.
POO(3)	Previous crane position.
WTO(3)	Previous crane task time.
IPO(3)	Previous crane task priority.
NC(3)	Crane number when selecting crane tasks from files.
NP(3)	Time when next ladle of matte is available.
JWT(3)	State of crane (delayed).
KWT(3)	State of crane (waiting).
TCRAN(3)	Crane busy time during simulation.

List of Variables in Common Block BLOK2

ICAPA(3)	100 * ladle capacity of anode furnaces
ILADA(3)	100 * number of ladles of copper in anode furnaces
IBUFX(12)	X-buffer for reading operator commands.
IBUFY(12)	Y-buffer for reading operator commands.
NRANK(3)	Rank number with which last crane command was filed.
ISTRT(3)	Matte tunnel number from which crane picked matte.
LOCDEV(31)	Co-ordinates of components in converter aisle (feet, from south end)
IWTNGT(17,2)	Default waiting time for tasks in seconds.
IPRA (17,2)	Default priorities for tasks.
NLC	Number of full ladles carried since last bumping a ladle.
NFL	Number of ladle fillings before bumping is required.
NSKL	Number of skulls at bumping block.
NCOOL	Number of cooling ladles at bumping block
NPPB	Number of pigs in the pig bay.
NCOLP	Number of cool pigs in the pig bay
NPPP	Number of pigs in the pig pile.
NCTOT	Number of cakes now ready at the pilot plant.
NCPPL	Number of cakes produced at pilot plant during simulation, including initial value of NCTOT.
NTANC	Total number of anodes cast.
ICNTRW	Control word to set control lines.
IER	Miscellaneous, usually error word in linkage routines.
ISAVE(3)	100 * content of ladles carried by cranes.

NAPL	Total number of skulls produced during simulation.
NOIL(3)	Cranes have been oiled if = 1 .
IFLOR(2)	Ladles of matte on floor.
ITROL(2)	Status of ladle on matte car.
JJ	Dummy for cranes.
II	Dummy for cranes
IDLE(3)	Crane status, O is idle 1 is busy.
KAPUT(3)	Crane status, O is operational 1 is down for repair.
NEXT(3)	Event code for next event to be executed by a crane.
LSTR(3)	Rank of last event removed from crane file.
NSLAG(3)	Number of ladles of slag returned to reverb.
IAFFL(3)	Anode furnace status.
IAWHL(2)	Anode wheel status.
NEVC(5)	Converter blow number.
IADD(11,4)	Weight and composition of converter input material.
ICV(5,9)	Computer hardware assignments for converter models.
ISTAT(5)	Current converter status.
JSTAT(5)	Converter status at time of breakdown.
NSTAT(5)	Converter status, not used presently.
NAPLC(3)	Number of anodes cast per ladle of copper.
IHGS(5)	100 * ladle content of converter slag standing near converter.
IRFO(5,5)	Quantities of materials removed from converter during the current charge.
IAFO(5,15)	Quantities of materials added to the converters during the current charge.

MT(5,20)	Converter blowing/non blowing and starting times during current charge.
NCV	Converter number 1 to 5.
NTLAD(3)	100 * total number of ladles of copper added to anode furnaces.
NANCST(3)	Total number of anodes cast.
IATIME(3)	Time when present anode furnace state was begun.
NTPPB	Total number of pigs poured in pig bay.
NCPT	Total number of cakes produced by prototype.
IRESL	Number of ladles of reactor slag presently available.
IPTCU	= 1 if copper presently available from prototype.
IPK(6,2)	Equivalenced with JQ(12) to store material quantities for automatic operation (AO).
IDP(6,2)	Equivalenced with JPR(12) to store priorities for automatic operation (AO).
NCPTT	Number of copper cakes presently available from prototype.
LCPU	Total ladles of copper produced by prototype.
LRS	Total ladles of slag produced by pilot plant.
IRS	Reverb furnace which received ladle of slag last.
IPRIT(5)	Priorities of converters to receive matte (automatic operation AO)
INHOUR	Initializes subroutine clock at start of simulation.
NACOL(3)	Number of anode collars available from anode furnaces.
ITCOR(15)	Ranking variable used in forecasting of time of matte requirements.(AO)
IBLOW(15)	Blow number for matte requirements (AO).
MOP(5,8)	Matte allocation to converters per blow. (AO).
MTREQ(5)	Quantity of first matte requirement after forecast (AO).

MTAV(2)	Quantity of matte available at time of forecast (AO).
IREP(5,6)	Reserved for converter status report.
MTOR(15)	Matte requirement within forecasting period (AO).
ICNO(15)	Converter number corresponding to MTOR(AO).

List of Variables in Common Block BLOK3

TPIA, TPIB	Lower and upper limits of uniform distribution (LUD) of time between copper cakes.
TIPBLA, TPISLB	Lower and upper LUD of time between slag ready.
TPCOOL	Time in minutes for pigs to cool.
TAF 6A, TAF 6B)) TAF5A, TAF5B)) TAF4A, TAF4B)	Lower and upper LUD of time for refining per ladle in anode furnaces 6, 5 and 4.
TSCT4A, TCST4B)) TCST5A, TCST5B)	Lower and upper LUD of time for casting per ladle for anode furnaces 4 and 5.
TAFR5A, TAFR4B)) TAFR5A, TAFR5B)	Lower and upper LUD of time to get anode furnace 4 or 5 ready after casting.
XMAX	Maximum crane displacement in feet from south end.
BETA	Speed-up factor for simulation.
TMAX	Maximum crane task time in minutes.
TIMEX	Maximum time interval between time events.
TLAST	Time of execution of last time event.
TOL	Length of shortest started crane task that is not interrupted by another crane task.
W	Closest distance between cranes.
RMLEV(3)	Relative matte level in reverb furnace (inches)
YMMAX(3)	Upper limit of matte level in reverb furnace (inches)
YMMIN(3)	Lower limit of matte level in reverb furnace (inches)
RMAT(3)	Rate of matte level increase in (min/inch).
TR(3)	Interval between matte level updating (minutes).

TLO(3), THI(3)	Mean and variance of matte tapping time in minutes.
TOLA(3)	Total number of ladles of matte tapped.
TAW(2)	Arrival time of matte car at converter aisle (minutes).
TOTW(2)	Total time full ladle stands on matte car (minutes)
TRAIL(2)	Time duration of matte car transportation (minutes).
RATI(3)	Conversion factor (inches of matte/ladle).
SLEV(3)	Relative slag level in furnace in inches.
SMAX(3), SMIN(3).	Upper and lower limit of slag level (inches).
STAPL(3), STAPH(3)	Upper and lower limit of slag tapping rate (pots/min.)
TOTSL(3)	Number of pots of slag tapped.
RSLG(3)	Rate of slag level increase (inches/min).
SLRA(3)	Conversion factor (inches of slag/ladle).
POT(3)	Conversion factor (inches of slag/pot).
TDOWN(3)	Total break-down duration of cranes.
TDL(3), TDH(3)	Lower and upper LUD of crane down time (minutes)
CRL(3), CRH(3)	Lower and upper LUD of operational time of cranes (minutes)
TCST6A, TCST6B	Upper and lower LUD for casting time on anode furnace 6.
TAFR6A, TAFR6B	Lower and upper LUD for getting anode furnace 6 ready after casting.
HOURS	Clock display time (hours).
TSLAG(3)	Ladles of converter slag returned to reverbs.
CV(5, 9)	Converter contents, blowing rates and capacities.
TPU(5)	Scale factor for converter model voltages.
TREQ(5)	Time of first matte requirement after forecast.
TAV(2)	Time of next matte arrival at converter aisle.

TCEND(5)	Forecasted time for start of copper blow in converters (AO).
TCOR(15)	Times of matte requirements within forecasting period (AO).
TCONT(5)	Reserved for automatic operation (AO).

List of Variables in Common Block BLOK4

NCR	Crane number
POS	Target position of crane
IPR	Priority of crane event.
TW	Waiting time of crane event.
KODE	Various uses.
NXEVT	Event to be executed at end of crane task.
IS(10)	Buffer to store sense line and comparator states when switching to frozen time mode.
LOGVAL	Logical variable, various uses.
NFRNC	Anode furnace number.
NRE	Reverb furnace number

List of Variables in Common Block SAVE

NSET (6,60)	GASP filing array
MODSCT(13)	Track and sector number of submodules on disk.

List of Variables in Common Block ARG

IX	Event number to be executed.
ID1X	First command syllable
ID2X	Second command syllable
ISET	Type of input command (PK, DP, PD)
JSEQ(12)	Type of input command (material)

JCMND(12))
)
NCRAN(12))
	Used to communicate command and crane information between submodules 9 and 10.

ICO(5)	Dummy, used in submodule 9 for converter number storage.
MODUL	Flag to load submodules 9 to 12.

F.4 LIST OF MODEL DISPLAY LIGHTS

LIGHT NUMBER	STATE OF PROCESS WHEN LIGHT IS ON
0	Reset all lights
1	Copper cakes available at PI
2	Slag ready at PI
3	Ladle requires bumping
4	Ladle cooling prior to bumping
5	Skull available at bumping block
6	Pigs available at pig bay
7	Pigs available at pig pile
8	A4 more copper required
9	A5 more copper required
10	A6 more copper required
11	A4 full and waiting
12	A6 full and waiting
14	A4 ready for casting
15	A5 ready for casting
16	A6 ready for casting
17	Empty moulds in pig bay
18	Empty slag ladle required at PI
19	Full ladle of matte near MTI

LIGHT NUMBER	STATE OF PROCESS WHEN LIGHT IS ON
20	Full ladle of matte near MT2
21	Empty ladle required at matte tunnel 1
22	Full ladle waiting at matte tunnel 1
23	Same as 21 for tunnel 2
24	Same as 22 for tunnel 2
25	Matte available in No.1
26	Matte level too high in No.1 (reaches slag hole)
27	Slag level is high
28	Same as 25 for No.2
29	Same as 26 for No.2
30	Same as 27 for No.2
31	Same as 25 for No.3
32	Same as 26 for No.3
33	Same as 27 for No.3
34	Saved for possible 3rd tunnel
35	-"-
36	Crane 1 operational
37	Crane 1 busy
38	Crane 2 operational
39	Crane 2 busy
40	Crane 3 operational
41	Crane 3 busy
42-46	Matte blow, converter
47-51	Copper blow, converter

LIGHT NUMBER	STATE OF PROCESS WHEN LIGHT IS ON
52-56	End of blow, converter
57-61	Additions required, converter
62-66	Empty ladle to be retumed to converter
67-71	Skimming required, converter
72	Anode wheel No.4 casting
73	Anode wheel No. 3 casting
74	Cakes ready at P.T.
75	Copper ready at P.T.
76	Empty ladle to be retumed to P.T.
77-79	Spare
80-89	Not used
90-99	Used as control lines

F.5 LIST OF PAUSES

PAUSE NUMBER	CAUSE	ACTION TAKEN WHEN CLEARED
1	Invalid command entered.	Command ignored
2	Invalid anode furnace no.	Command ignored
3	Invalid crane number	Command ignored
4	Event code outside range 1-295	No event is executed
5	No such event	No event is executed
6	Set monostable times 4-0, 4-1 switch serves on display unit ON.	Execution continues
11	Next time target further than TIMEX minutes away (subr. SETIME).	Set next time target TIMEX minutes from now
17	ISET greater than allowable value in DISK 9	Call ignored
21	Subr. CONV	Call ignored
41	Incorrect branch to CRA3 routine	Branches to CRA1 routine
55	In event 55 an incorrect crane failure state	Crane failed
74	Error in FIND, column No. less than 1. (GASP error 89)	Not recoverable
75	Error in SET (GASP error 88) Error in FNDCEV,column No. less than 1	Not recoverable
76	Error in RMOVE, column No. less than 1 (GASP error 97)	Not recoverable
77	Overlap in filing array, Subr. FILEM (GASP error 87)	Not recoverable

PAUSE NUMBER	CAUSE	ACTION TAKEN WHEN CLEARED
150	Pause allows setting analog components to 'fix' cranes	Execution continues
222	Converter status is larger than 15	ISTAT (NCV) = 1

F.6 LIST OF SENSE SWITCH FUNCTIONS

SENSE SWITCH NO.	FUNCTION
1	Push to enter a command
2	Push to suppress typeout from module 2, 8, 'crane idle' messages and light messages
3	Push to suppress typeout of ATRIB from modules 3,6,7.
4	Push to suppress typeout from modules 3,6,7.
5	Reset to suppress typeout in subroutine TCODE.
6	Push before going to start program during error recovery, and release after ENTER has been typed. In automatic mode, push if single crane operation is desired, release if two crane tasks are desired.
7	Push for automatic mode of operation.
8.	Suppress typeout from modules 9, 10, 11 and 12.

APPENDIX G

PROGRAMME LISTING

MAIN MODULE

```
 DECK FOR MAIN MODULE
 LOGICAL LOGVAL
 DIMENSION IDUM(44)
 DIMENSION MCONME(13),AMFND(3)
 COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,NX,NXC,NOQ,SCALE,ISEED,TNOW,
1TBEG,TFIN,NXX,NPRNT,NCROR,OUT
 COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
*,ISTR(4)(5)
 COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTU(3),IPO(3),NC(3),NP
1(3), JWT(3), KWT(3), TCRAN(3)
 COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
1 NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNGT(17,2),  IPRA(17,2),
2 NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
3 NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
*  ITROL(2),   JJ,   II,
4IDLE(3),KAPLT(3),NEXT(3),LSTR(3),NSLAG(3)
5,IAFFL(3),IARML(2),NEVC(5),IADD(1,4),ICV(5,9)
6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
7,INUS(5),IAFO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
*,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
*NOP(5,6),NTREQ(5),NTAV(2),IREP(5,6)
 COMMON /BLOK2/ ISLAK(44)
 COMMON /BLCK3/  TP1A,  TP1B,  TPISLA,  TPISLB,  TPCOOL,
1 TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST6B,
2 TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATE(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
6(3),TDOWN(3),TDL(3),TDH(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
 COMMON /BLOK4/ NCR,  PCS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
* LOGVAL,NFRNC,NRE
 COMMON /SAVE/ NSET(6,60),MODSCT(13)
 COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
1 NCRAM(12),  ICO(5),  MODUL
 EQUIVALENCE (IDUM(1),ICMPA(1))
 DATA NMFND(1),NMFND(3)/  2HUP,  2HC/
 DATA MODNME(1),  MODNME(2),  MODNME(3),  MODNME(4),  MODNME(5),
1 MODNME(6),  MODNME(7),  MODNME(8),  MODNME(9),  MODNME(10)
2 ,MODNME(11),MODNME(12),MODNME(13)
3 /2HS1,  2HS2,  2HS3,  2HS4,  2HS5,  2HS6,  2HS7,  2HS8,  2HS9,
*  2H10,  2H11,   2H12,  2H13 /
 DATA I/-32768/

C****PREVENT  LOADING  I/O ROUTINES THAT ARE NOT REQUIRED
 CALL EXTNT

C  FIND STARTING SECTOR OF CI FILES
C****SKIP IF THIS HAS ALREADY BEEN DONE
 IF(I .NE. -32768) GO TO 2
 DO 1 I=1,12
 NMFND(2)=MCONME(I)
1 CALL FLFFND(NMFND,MODSCT(I),IDUM)

C****CALL MODULE 1
 CALL EXEC(1)

C****DUMMY CALL TO GET CLOCK INTO MAIN MODULE
 CALL CLOCK(I)
 CALL GASP
 END



 SUBROUTINE GASP
 DIMENSION KOMPAR(6)
 DIMENSION TDEL(3)
 DIMENSION L(4)
 DIMENSION BUSYT(3)
 LOGICAL SENSW,LOGVAL
 COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,NX,NXC,NOQ,SCALE,ISEED,TNOW,
1TBEG,TFIN,NXX,NPRNT,NCROR,OUT
 COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
*,ISTR(8)(5)
 COMMON /SAVE/ NSET(6,60),MODSCT(13)
 COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
1 NCRAM(12),  ICO(5),  MODUL
 COMMON/BLOK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
1(3),  JWT(3),  KWT(3),  TCRAN(3)
 COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
1 NRANK(3),  ISTRT(3),  LOCDEV(31),  INTNGT(17,2),  IPRA(17,2),
2 NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
3 NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
*  ITROL(2),  JJ,  II,
4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
5,IAFF(3),IARML(2),NEVC(5),IADD(1,4),ICV(5,9)
6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
7,INUS(5),IAFO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
*,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
*NOP(5,6),NTREQ(5),NTAV(2),TREP(5,6)
 COMMON /BLOK3/  TP1A,  TP1B,  TPISLA,  TPISLB,  TPCOOL,
1 TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST6B,
```

```
*    2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATE(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TDL(3),TDH(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
      COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,NRE
      EQUIVALENCE (IX,IARG)
      DATA KOMPAR(1),  KOMPAR(2),  KOMPAR(3),  KOMPAR(4),  KOMPAR(5),
     *   KOMPAR(6)  /  99,96,69,106,109,39  /
      DATA L(1),L(2),L(3),L(4) /0,0,0,0 /
      DATA BUSYT(1),BUSYT(2),BUSYT(3) /0.,0.,0./
      DATA TDEL(1),TDEL(2),TDEL(3)/0.,0.,0./
      MODUL=0

C******  GO TO 50 IF ON ERROR RECOVERY
      IF(SENSW(6)) GO TO 50

C******   REMOVE NEXT TIME EVENT AND SET TIME TARGET
   10 CALL RMOVE(MFE(1),1,NSET)
      CALL SETIME(NSET)

C******   GO TO 72 ON INITIAL ENTRY INTO GASP
      IF(L(4) .NE. 1) GO TO 72
      J   .73

C******   UPDATE SIMULATION TIME AND EXECUTE TIME EVENT
   99 TLAST=TNOW
      IARG=JEVNT
      CALL EVNTS

C******   SET MODUL TO 4 IF AN AUTOMATIC EVENT WAS FILED
      IF(MODUL .NE. 0) MODUL=4
      J   .10
   63 MODUL=0
      NODUL=0
      J   .64

C******   ACCEPT OPERATOR COMMAND

   50 CALL RD
C******   SCAN FILES FOR NEW ENTRIES
   64 CALL TESTFL(NSET)
   70 IPMI=9
      IPMA=0

C******   SORT CRANES ACCORDING TO PRIORITIES OF PRESENT TASKS
      DO 69 KP=1,3
      KPT=MFE(KP+1)
      ISCALE=SCALE*.1
C      IER IS USED TO TEMPORARLY STORE THE PRIORITY OF CRANE FILE ENTRY
      IER=NSET(3,KPT)-(NSET(3,KPT)/ISCALE)*ISCALE
      IF(KPT .EQ. 0) IER=0
      IF(IER .LT. 0) IER=0
      IF(9 .LT. IER) IER=9
      IF(IER .LT. IPMA) GO TC 67
   66 IPMA=IER
      NC(1)=KP
   67 IF(IER-IPMI) 68,69,69
   68 IPMI=IER
      NC(3)=KP
   69 CONTINUE
      NC(2)=6-NC(1)-NC(3)

C******   IN ORDER OF PRIORITY ,   FOR EACH CRANE SEE IF CRANE HAS
C         MORE TASKS TO DO.   IF SO,   GET NEXT CRANE EVENT AND
C         SET POSITION AND TIME TARGETS.   RESOLVE INTERFERENCE
      DO 87 N=1,3
      NCR=NC(N)
      IF(IDLE(NCR).NE.0) GO TO 87
      IF(MFE(NCR+1).EQ.0) GO TO 87
      CALL GETEV(NSET)
      CALL TKODE(NSET)
      GO TO (77,82) ,IER
   77 NEXT(NCR)=NXEVT
   82 GO TO (83,84,85) ,NCR
   83 CALL CRA1(POS,TW,IPR,NSET)
      J   .86
   84 CALL CRA2(PCS,TW,IPR,NSET)
      J   .86
   85 CALL CRA3(PCS,TW,IPR,NSET)

C******   SET CRANE BUSY
   86 IDLE(NCR)=1
      BUSYT(NCR)=TIME(TLAST)
   87 CONTINUE

C******   RESCAN SENSE LINES AND COMPARATORS TO SEE IF ANY HAVE
C         CHANGED AND NEED SERVICING BEFORE GOING INTO OPERATE
      IF(L(4) .EQ. 1) GO TO 73
   72 IF( MODUL .NE. 0) GO TO 63

C******   RESET FLIP FLOPS FOR CRANES
      CALL CONTRL(7,1)
      CALL QSBLY(50)
      CALL CONTRL(7,0)

C******   RESET L.   RECORD CRANE INTERFERENCE DELAY TIMES
      DO 2006 I=1,3
      IF(.001 .LT. TDEL(I)) SP(I)=SP(I)+TIME(TLAST)-TDEL(I)
      TDEL(I)=0.
      IF(JWT(I) .NE. 0) TDEL(I)=TIME(TLAST)
 2006 L(I)=0
      L(4)=0

C******   PUT ANALOG COMPUTER INTO OPERATE
```

```
      CALL QSOP(IER)

C******   TEST FOR INTERRUPTION DUE TO OPERATOR COMMAND
 1111 IF(SENSW(1)) GO TO 130

C******   TIME TARGET REACHED
      CALL ORCPL(KOMPAR(6),LOGVAL,IER)
      IF(LOGVAL) GO TO 130
      DO 75 I=1,5

C******   CONVERTER END POINT REACHED
      CALL ORCPL(KCMPAR(I),LOGVAL,IER)
      IF(.NOT. LOGVAL) GO TO 130
      IF( I .LT. 4) GO TO 75

C******   CRANE TARGET REACHED
      CALL ORSLL(I,LCGVAL,IFR)
      CALL ORSLL(I,LOGVAL,IER)
      IF(LOGVAL) GO TO 130
   75 CONTINUE
      J    .1111
C******REMOVE NEXT EVENT FROM CRANE FILE  - IF EMPTY SET CRANE IDLE
 3320 IF(JWT(1) .NE. 2) GO TO 2002
  198 JWT(1)=0

C******   IF CRANE 1 EVENT HAS ALREADY BEEN SERVICED,   GO TO 2001
  199 IF(L(1) .EQ. 2) GO TO 2001
          IF( JWT(1) .NE. 0) GO TO 2001
      L(1)=2
      NCR=1
      J    .301
 3310 IF(JWT(2) .NE. 1) GO TO 2001
  298 JWT(2)=0

C******   IF CRANE 2 EVENT HAS ALREADY BEEN SERVICED,   GO TO 2002
  299 IF(L(2) .EQ. 3) GO TO 2002
          IF( JWT(2) .NE. 0) GO TO 2002
      L(2)=3
      NCR=2
      J    .301
  332 CALL CRA2(PCS,TW,IPR,NSET)
 3302 IF(JWT(3) .NE. 2) GO TO 3320
      JWT(3)=0
  100 IF(L(3) .EQ. 4) GO TO 72

C******   IF CRANE 3 EVENT HAS ALREADY BEEN SERVICED,   GO TO 72
          IF( JWT(3) .NE. 0) GO TO 2003
      L(3)=4
      NCR=3

C******   IF CRANE IS IDLE OR HAS NO EVENT TO EXECUTE,   GO TO 320
  301 IF(NEXT(NCR).LE.0) GO TO 320
      IARG=NEXT(NCR)
C******   EXECUTE CRANE EVENT
      CALL EVNTS

C******   SET NODUL TO 4 IF AN AUTOMATIC EVENT WAS FILED
      IF(NODUL .NE. 0) NODUL=4

C******   SCAN FILES FOR NEW ENTRIES
   13 CALL TESTPL(NSET)
C     KAPUT TEST

C******   IF CRANE FILE IS EMPTY,   GO TO 340
  320 IF(NFE(NCR+1).EQ.0) GO TO 340

C******   REMOVE NEXT CRANE EVENT
      CALL GETEV(NSET)
      CALL TKODE(NSET)
      GO TO (76,334),IER

C******   SET NEXT CRANE EVENT
   76 NEXT(NCR)=NEEVT
      GO TO(331,332,333) , NCR

C******   ASSIGN NEW CRANE TARGETS, CONSIDERING INTERFERENCE, AND
C           ONE CRANE WAITING FOR ANOTHER
  331 CALL CRA1(PCS,TW,IPR,NSET)
 3301 IF(JWT(3) .NE. 1) GO TO 3310
      JWT(3)=0
      J    .100
  333 CALL CRA3(PCS,TW,IPR,NSET)
 3303 IF(JWT(1) .NE. 3) GO TO 3330
      IF(JWT(2) .EQ. 3) JWT(2)=1
      J    .198
 3330 IF(JWT(2) .NE. 3) GO TO 2003
      J    .298

C******   CRANE FILE IS EMPTY,   RESET RANK,   NEXT EVENT AND PRIORITY
  340 NRANK(NCR)=10
      NEXT(NCR)=0
      IP(NCR)=0

C******   IF CRANE WAS BUSY,   RECORD BUSY TIME
      IF(IDLE(NCR) .NE. 1) GO TO 349
      TCRAN(NCR)=TCRAN(NCR)+TIME(TLAST)-BUSYT(NCR)

C******   SET CRANE IDLE,   SWITCH OFF BUSY LIGHT
  349 IDLE(NCR)=0
      CALL LIGHT(39+20*NCR,0)
  334 GO TO (13CE,3302,3303) , NCR

C******   PUT ANALOG COMPUTER INTO HOLD
  130 CALL QSH(IER)

C******   TEST SENSE LINES AND COMPARATORS TO SEE WHICH SUBMODEL
C           REQUIRES ATTENTION
```

```
   72 CALL SLTEST(KCMPAR)

C******   CRANE 1
      IF(IS(2) .EQ. 2) GO TO 199

C******   CRANE 2
 2001 IF(IS(3) .EQ. 3) GO TO 299

C******   CRANE 3
 2002 IF(IS(4) .EQ. 4) GO TO 100

C******   AFTER SECOND SCAN GO TO 72
 2003 IF(L(4) .EQ. 1) GO TO 72
      DO 2005 I=1,5

C******   CONVERTERS
      IF( IS(I+4) .NE. I+4) GO TO 2005
      IARG=204+I

C******   EXECUTE CONVERTER EVENTS
      CALL EVNTS
      IF(NODUL .NE. 0) NODUL=4
 2005 CONTINUE

C******   MARK THE FACT THAT CONVERTER EVENTS HAVE BEEN SCANNED ONCE
   71 L(4)=1

C******   TIME TARGET REACHED
      IF( IS(1) .EQ. 1) GO TO 99

C******   OPERATOR COMMAND
 2004 IF(IS(10) .EQ. 10) GO TO 50
      J    .73
   22 RETURN
      END
```

```
      SUBROUTINE TKODE(NSET)
      LOGICAL SENSW
      COMMON TO,IM,INIT,JEVNT,NFA,NSTOP,NT,NNC,NOQ,SCALE,ISEF3,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),ENQ(4),NRANK(4),NFE(4),NLC(4),N1(4),NC(4)
     *,ISTRIB(5)
      COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WT3(3),IP3(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLOK2/ ICAPA(3),   ILADA(3),   IBUFX(12),   IBUFY(12),
     1  NRANK(3),   ISTAT(3),   LOCDEV(3),   IWTNGT(17,2),   IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCCOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOW(2),
     *    ITROL(2),    JJ,    II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPFL(3),IANDL(2),NEVC(5),IADD(11,4),ICVE(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANC(7(1)),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPR(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),TWNDUM,NACOL(3),ITCOR(15),IRCOL(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,8)
      COMMON /BLOK4/ ACR,  POS,  IPR,  TW,  KODE,  NEEVT,  IS(10),
     * LOGVAL,NPRAC,NNE
      IF(KODE.EQ.C.AND.KWT(NCR).NE.0)  GO TO 4
      IF(9.LT.KODE.OR.KODE.LT.1) GO TO 76
      IF(SENSW(5)) WRITE(6,1000) KWT,JWT,NCR,KODE
      IF(KODE.LT.4) GO TO 1
      IF(KODE.LT.7) GO TO 2
      J    .70
    3 KWT(NCR)=KODE
    4 CONTINUE
      CALL CFILE(NCR,PCS,TW,IPR,LSTR(NCR),0,NEEVT,1,NSET)
      KODE=0
      ISTRIB(NCR+1)=0
      IP(NCR)=0
  334 IER=2
      J    .99
    2 KODE=KODE-3
    1 IF(KODE.EC.NCR) GO TO 70
      IF(KWT(KODE).NE.NCR) GO TO 3
      KWT(KODE)=0
      JWT(KODE)=NCR
   70 KODE=0
      ISTRIB(NCR+1)=0
   76 IER=1
   99 CONTINUE
      IF(SENSW(5)) WRITE(6,1000) KWT,JWT,NCR,KODE
 1000 FORMAT(1X,3HKWT,3I4,/1X,3HJWT,3I4,/1X,3HNCR,I4,2X,4HKODE,I4/)
      RETURN
      END
```

```
      SUBROUTINE MD
      COMMON /BLOK2/ ICAPA(3),   ILADA(3),   IBUFX(12),   IBUFY(12),
     1  NRANK(3),   ISTAT(3),   LOCDEV(3),   IWTNGT(17,2),   IPRAT(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOW(2),
     *    ITROL(2),    JJ,    II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPFL(3),IANDL(2),NEVC(5),IADD(11,4),ICVE(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANC,7(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPR(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),TWNDUM,NACOL(3),ITCOR(15),IRCOL(15),
     *NOP(5,8),NEVREQ(5),NTAV(2),IREP(5,8)
      COMMON /ARG/ IS,  IBIN,  IBZN,  IWF,  JSBJ(12),  J[illegible](12),
     1  NCRAN(12),  ICO(5),  MODUL
      DATA IEND,ICORCT /2H  ,2HCC/
      DATA IS/2HIS/
C************
   60 TYPE 1102
C***** ACCEPT A COMMAND FROM THE OPERATOR
      ACCEPT 1101,ID1X,ID2X,(IBUFX(I),I=1,12)

C******   END OF COMMAND
      IF(ID1X .EQ. IEND) GO TO 70

C******   AUTOMATIC MODULE DEBUG COMMAND
      IF(ID1X .EQ. IS) GO TO 7
  502 TYPE 1102
      ACCEPT 1101,ID1Y,ID2Y,(IBUFY(I),I=1,12)

C******   END OF COMMAND
      IF(ID1Y .EQ. IEND) GO TO 3

C******   CORRECTION
      IF(ID1Y .EQ. ICORCT) GO TO 60

C******   AUTOMATIC MODULE DEBUG COMMAND
      IF(ID1Y .EQ. IS) GO TO 6
C******   CALL MODULE 4
      CALL EXEC(4)

C******   CALL MODULE 5 IF INFORMATION IS REQUIRED
    4 IF(IX .NE. 0) CALL EXEC(5)

C******   COPY Y INTO X
      ID1X=ID1Y
      ID2X=ID2Y
      DO 2 I=1,12
    2 IBUFX(I)=IBUFY(I)
      J     .502
C       END OF INPUT COMMANDS
C       IF ANY CRANE WAS IDLE PICK FIRST EVENT FROM RESP. CRANE FILE

C******   CALL MODULE 4
    3 CALL EXEC(4)

C******   CALL MODULE 5 IF INFORMATION IS REQUIRED
      IF(IX .NE. 0) CALL EXEC(5)
   70 RETURN


C******   DEBUG FEATURE . ******   FILE AUTOMATIC COMMANDS
    7 CALL SETTST(IBUFX)
      J     .60
    6 CALL SETTST(IBUFY)
      J     .502
 1101 FORMAT(14A2)
 1102 FORMAT(1X,5HENTER/)
      END


      SUBROUTINE EVNTS
      COMMON /ARG/ IX, ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
     1  NCRAN(12),  ICO(5),  MODUL
      EQUIVALENCE (IX,IARG)
C     IF(IARG .GT. 30) GO TO 12
      IF(  30 .LT. IARG) GO TO 12
      CALL EXEC(2)
      J     .15
C  12 IF(IARG .GT. 99) GO TO 11
   12 IF(  99 .LT. IARG) GO TO 11
      CALL EXEC(8)
      J     .15
C  11 IF(IARG .GT. 169) GO TO 14
   11 IF( 169 .LT. IARG) GO TO 14
      CALL EXEC(3)
      J     .15
C  14 IF(IARG .GT. 249) GO TO 23
   14 IF( 249 .LT. IARG) GO TO 23
      CALL EXEC(6)
   15 IF(IARG .EQ. 0) GO TO 10
      CALL EXEC(5)
   10 IF(ISET .EQ. 0) GO TO 1
      CALL EXEC(9)
      IF(MODUL .EQ. 4) CALL EXEC(10)
    1 IF(MODUL.EQ.11) CALL EXEC(11)
      IF(MODUL.EQ.12) CALL EXEC(12)
      RETURN
   23 CALL EXEC(7)
      J     .15
      END


      SUBROUTINE FLEFND(NAME,ISECT,IA)
      DIMENSION NAME(3),IA(44)
C      READ ONE SECTOR ON TRACK 0
      ISSTRT=0
    5 CALL DSKRI(IA,44,ISSTRT,ISNXT,10)
C      CHECK ALL 11 FILES ON THIS SECTOR
      DO 1 I=1,11
      K=4*(I-1)
C      CHECK EACH FILE FOR DESIRED NAME
      DO    J=1,3
      L=K+J
      IF(IA(L) .NE. NAME(J)) GO TO 1
```

```
    2 CONTINUE
C      FILE HAS BEEN FOUND
      ISECT=IA(K+4)
      J     .6
    1 CONTINUE
C      FILE HAS NOT BEEN FOUND
      ISSTRT=ISNXT
      IF(ISNXT .LE. 127) GO TO 5
C      FILE IS NOT IN DIRECTORY
      TYPE 100,(NAME(I),I=1,3)
  100 FORMAT(3A2,9HNOT FOUND/)
      PAUSE 555
    6 RETURN
      END


      SUBROUTINE SETTST(IBUF)
      DIMENSION IBUF(12)
      COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
     &  NCRAN(12),  ICO(5),  MODUL
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NERNK(3),  ISTRT(5),   LOCDEV(3,3),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSRL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
     *    ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IANML(2),NEVC(5),IADU(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRFO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 LATIME(3),NTPPR,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2)
     *,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),INLCU(15),
     *NOP(5,8),PTREB(5),NTAV(2),IREP(5,6)
      COMMON /BLOK4/ NCR,  POS,  IPR,  TH,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,NRE
      CALL COREAD(IBUF,12)
      READ(6,100) ISET,NRE
      CALL CARD
      NFRNC=NRE-1
      NCV=NRE-2
      CALL EXEC(9)
      IF(MODUL .EQ. 4) CALL EXEC(10)
      RETURN
  100 FORMAT(I7,I1)
      END
```

FIRST SUBMODULE

```
C     DECK FOR FIRST SUBMODULE
      SUBROUTINE DISK
C     THIS IS THE FIRST SUBMODULE.   ITS FILE NAME IS UPSICF
C     IT READS IN DATA INTO COMMON BLOK 1 AND 2.   SETS POTS
C     AND READS GASP VARIABLES
      DIMENSION RFX(8),RCUB(5)
      DIMENSION IBUFX(80)
      DIMENSION IDUM(655),DUM(191)
      DIMENSION PCV(5,7)
      LOGICAL LOGVAL,SENSW
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCROR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     +,ISTRIB(5)
      COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUF(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(3)),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NAL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
     *  ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IAWHL(2),NEVC(5),TADO(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRFO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
     +NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3TMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),VMMAX(3),VMMIN(3),WMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TOL(3),TON(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
      COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,NRE
      COMMON /SAVE/ NSET(6,60),MOOSCT(13)
      COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
     1  NCRAN(12),  ICO(5),  MODUL
      EQUIVALENCE (IDUM(1),ICAPA(1)),(DUM(1),TPIA)
      DATA PCV(1,1),PCV(2,1),PCV (3,1),PCV(4,1),PCV(5,1)/4HP091,4HP065,
     14HP051,4HP021,4HP068/
      DATA PCV(1,2),PCV(2,2),PCV(3,2),PCV(4,2),PCV(5,2)/4HP093,4HP064,
     14HP052,4HP023,4HP070/
      DATA PCV(1,3),PCV(2,3),PCV(3,3),PCV(4,3),PCV(5,3)/4HP115,4HP095,
     14HP073,4HP102,4HP110/
      DATA PCV(1,4),PCV(2,4),PCV(3,4),PCV(4,4),PCV(5,4)/4HP097,4HP067,
     14HP053,4HP016,4HP071/
      DATA PCV(1,5),PCV(2,5),PCV(3,5),PCV(4,5),PCV(5,5)/4HP098,4HP094,
     14HP046,4HP018,4HP072/
      DATA PCV(1,6),PCV(2,6),PCV(3,6),PCV(4,6),PCV(5,6)/4HP086,4HP066,
     14HP076,4HP106,4HP113/
      DATA PCV(1,7),PCV(2,7),PCV(3,7),PCV(4,7),PCV(5,7)/4HP116,4HP096,
     14HP077,4HP107,4HP100/
      DATA RFX(1),RFX(2),RFX(3),RFX(4),RFX(5),RFX(6),RFX(7),RFX(8)/.45,
     1.45,.45,.45,.45,.45,.45,1./
      DATA RCUB(1),RCUB(2),RCUB(3),RCUB(4),RCUB(5)/.70,.57,.71,.67,.95/
      CALL QSHVIN(IER,680)
      ISET=0
      MODUL=0
      TYPE 103
      ACCEPT 101, IBR
      GO TO (17,156,2) , IBR
C      INITIALIZE LIGHTS
   17 CALL LIGHT(0,2)
      KODE=0
      DO 5 IER=1,5
    5 ISTRIB(IER)=0
      READ(4,101) IDUM
      READ(4,102) DUM
      CALL DATAN(NSET)
      IF(SENSW(2)) GO TO 99
      WRITE(6,197) IDUM
      WRITE(6,107) DUM
   99 CONTINUE
      DO 300 JJCV=1,5
      RATE=CV(JJCV,6)*CV(JJCV,7)/TPU(JJCV)
      NSTA=ISTAT(JJCV)
      GO TO(210,220,210,210,220,210,210,220,210,210,220,250,262),NSTA
C ICS FOR SLAG STAGE
  210 NNEVC=NEVC(JJCV)
      QIA=RFX(NNEVC)*CV(JJCV,3)/TPU(JJCV)
      RFES=.016*RATE
      NT(JJCV,13)=QIA/RFES
      CV(JJCV,3)=CV(JJCV,3)*(1.-RFX(NNEVC))
      QIB= CV(JJCV,4)/TPU(JJCV)
      J   .240
C ICS FOR FINISH STAGE
  220 QIA= CV(JJCV,2)/TPU(JJCV)
      QIB= CV(JJCV,9)/TPU(JJCV)
      RCUS=.043*RATE
      NT(JJCV,14)=QIA/RCUS
      IAFO(JJCV,15)=CV(JJCV,2)/20.
      NT(JJCV,15)=NT(JJCV,14)/IAFO(JJCV,15)
  240 CALL QWJDAR(QIA,ICV(JJCV,1),IER)
      CALL QWJCPR(QIB,ICV(JJCV,2),IER)
      J   .300
  250 JNCV=JJCV+2
      WRITE(6,263) JNCV
  263 FORMAT(1X,44HSTART WITH COLLAR PULL NOT ALLOWED,CONVERTER,I2,13HWI
     1LL BE HELD IN ITS INITIAL STATE/)
      J   .300
  262 JNCV=JJCV+2
```

```
      WRITE(6,264) JNCV
  264 FORMAT(1X,9HCONVERTER,I2,20H DOWN FOR REBRICKING/)
 300  CONTINUE
      TC=BETA/(TMAX*60.)
      TTC=BETA/(TIMEX*60.)
      SP(1)=232.0*BETA/60.0
      SP(2)=194.0*BETA/60.0
      CALL QSSTOP(IER)
C*******  INCREASE CRANE SPEED BY  70
      C=SP(1)/XMAX*1.7
      CALL QWPR(4HP002,C,IER)
      CALL QWPR(4HP022,C,IER)
      C=SP(2)/XMAX*1.7
      CALL QWPR(4HP012,C,IER)
      CALL QSRUN(IER)
C     INITIALIZE CONVERTER RATE POTENTIOMETERS
C     USE ISTAT(NCV) FROM BLOK2. CV FROM BLCK3
      DO 990 JJCV=1,5
      NSTA=ISTAT(JJCV)
      FRAT=1.1
C SET CONVERTER RATE PCTS
      RATE=CV(JJCV,6)*CV(JJCV,7)*BETA/TPU(JJCV)
      RATC=RATE*RCUB(JJCV)/CV(JJCV,7)
      RFES=.016*RATE
      RSL=.0115*(1.+.62C*FRAT)*RATE
      RCU=.034*RATC
      RCUS=.043*RATC
      PC=1./TPU(JJCV)
      TNS=1./60.
      CALL QSSTOP(IER)
      CALL QWPR(PCV(JJCV,1),RFES ,IER)
      CALL QWPR(PCV(JJCV,2),RCUS ,IER)
      CALL QWPR(PCV(JJCV,3),PC   ,IER)
      CALL QWPR(PCV(JJCV,4),RSL  ,IER)
      CALL QWPR(PCV(JJCV,5),RCU  ,IER)
      CALL QWPR(PCV(JJCV,6),TNS,IER)
      CALL QWPR(PCV(JJCV,7),TNS,IER)
      CALL QSRUN(IER)
C SET FUNCTION RELAYS
      JCA=ICV(JJCV,6)
      JCB=ICV(JJCV,7)
      JFA=ICV(JJCV,5)
      JFB=ICV(JJCV,9)
      GO TO(991,992,993,994,994,991,996,996,995,995,992,998,998),NSTA
C STATE 1,6  STRING 1
  991 CALL QWFRL(JFB,.FALSE.,IER)
      CALL QWFRL(JFA,.TRUE.,IER)
      CALL LIGHT(JCA,1)
      IBUFY(JJCV)=1
      CALL LIGHT(JCB,1)
      CALL QWFRL(JFB,.TRUE.,IER)
      IST=1
      J   .999
C STATE 2,11 STRING 2
  992 CALL QWFRL(JFB,.FALSE.,IER)
      CALL QWFRL(JFA,.FALSE.,IER)
      CALL LIGHT(JCB,1)
      CALL LIGHT(JCA,1)
      IBUFY(JJCV)=1
      CALL QWFRL(JFB,.TRUE.,IER)
      IST=2
      J   .999
C     STATES 3,9, STRING 3
  993 CALL QWFRL(JFB,.FALSE.,IER)
      CALL QWFRL(JFA,.TRUE.,IER)
      CALL LIGHT(JCA,0)
      IBUFY(JJCV)=0
      CALL LIGHT(JCB,1)
      J   .999
C     STATES 4,5, STRING 4
  994 CALL QWFRL(JFB,.FALSE.,IER)
      CALL QWFRL(JFA,.FALSE.,IER)
      CALL LIGHT(JCA,0)
      IBUFY(JJCV)=0
      CALL LIGHT(JCB,1)
      J   .999
C STATE 9,10 STRING 5
  995 CALL QWFRL(JFB,.FALSE.,IER)
      CALL QWFRL(JFA,.TRUE.,IER)
      CALL LIGHT(JCA,0)
      IBUFY(JJCV)=0
      CALL LIGHT(JCB,0)
      J   .999
C STATES 7,8 STRING 6
  996 CALL QWFRL(JFB,.FALSE.,IER)
      CALL QWFRL(JFA,.FALSE.,IER)
      CALL LIGHT(JCA,0)
      IBUFY(JJCV)=0
      CALL LIGHT(JCB,0)
      J   .999
C     STATE 12,ILLEGAL STARTING STATE,HOLD CONVERTER IN INITIAL STATE
C     STATE 13, CONVERTER BEING REBRICKED
  998 CALL LIGHT(JCA,0)
      IBUFY(JJCV)=0
  999  CONTINUE
  990 CONTINUE
      CALL QSSTOP(IER)
      CALL QWPR(4HP008,0.0,IER)
      CALL QWPR(4HP010,0.0067,IER)
      CALL QWPR(4HP020,0.0334,IER)
      CALL QWPR(4HP009,0.0005,IER)
      CALL QWPR(4HP019,0.0005,IER)
      CALL QWPR(4HP029,0.0005,IER)
      CALL QWPR(4HP030,TC,IER)
      CALL QWPR(4HP040,TC,IER)
      CALL QWPR(4HP050,TC,IER)
      CALL QWPR(4HP035,TTC,IER)
      CALL QWPR(4HP014,.015,IER)
      CALL QWPR(4HPO24,.015,IER)
      TTC=.6*TTC
      CALL QWPR(4HPO60,TTC,IER)
      CALL QSRUN(IER)
C***********
      CALL QSSECN(IER)
   15 CALL QSIC(IER)
      DO 150 JJCV=1,5
      ILITE=ICV(JJCV,6)
      IF(IBUFY(JJCV).EQ.0) GO TO 150
      CALL LIGHT(ILITE,0)
      CALL QSDLY(1000)
      CALL LIGHT(ILITE,1)
  150 CONTINUE
      TYPE 551
 156  CONTINUE
C**********
C      INITIALIZE LIGHT SETTINGS
C
      DO 157 I=1,80
  157 IBUFX(I)=C
C     SPECIFY CONVERTER LIGHTS TO BE TURNED ON
C     DEPENDING ON INITIAL CONVERTER STATES
      DO 810 N=1,5
      I=ISTAT(N)
      GO TO(801,802,803,803,804,808,805,806,805,805,807,810,810), I
  801 IBUFX(N+41)=1
      J    .810
  802 IBUFX(N+46)=1
      J    .810
  803 IBUFX(N+41)=1
      IBUFX(N+51)=1
      IBUFX(N+66)=1
      J    .810
  804 IBUFX(N+46)=1
      IBUFX(N+51)=1
      IBUFX(N+66)=1
      J    .810
  805 IBUFX(N+41)=1
      IBUFX(N+51)=1
      IBUFX(N+56)=1
      J    .810
  806 IBUFX(N+46)=1
      IBUFX(N+51)=1
      IBUFX(N+56)=1
      J    .810
  807 IBUFX(N+46)=1
      IBUFX(N+56)=1
      J    .810
  808 IBUFX(N+41)=1
      IBUFX(N+56)=1
  810 CONTINUE
      DO 10 I=1,3
      SP(I)=0.
      JMT(I)=0
      KMT(I)=0
      TCRAN(I)=0.
   10 IPO(I)=0
      I2=2*I
      I3=3*I
      IF(IAFFL(I) .EQ. 0) IBUFX(I+7)=1
      IF(IAFFL(I) .EQ. 1) IBUFX(I+10)=1
      IF(IAFFL(I) .EQ. 1) IBUFX(I+7)=1
      IF(IAFFL(I) .EQ. 6) IBUFX(I+10)=1
      IF(IAFFL(I) .EQ. 3) IBUFX(I+13)=1
      IF(RMLEV(I) .GE. YMMIN(I)) IBUFX(I3+22)=1
      IF(RMLEV(I) .GE. YMMAX(I)) IBUFX(I3+23)=1
      IF( SMAX(I) .LT. SLEV(I)) IBUFX(I3+24)=1
      IF(KAPUT(I) .EQ. 0) IBUFX(I2+34)=1
   10 IBUFX(I2+35)=IOLE(I)
      DO 11 I=1,2
      I2=2*I
      IF(0 .LT. IFLOR(I)) IBUFX(I+18)=1
      IF(ITROL(I) .EQ. 0) IBUFX(I2+19)=1
      IF(ITROL(I) .EQ. 1) IBUFX(I2+20)=1
      IER=74-I
      IF(IAWNL(I) .EQ. 1) IBUFX(IER)=1
   11 CONTINUE
      IF(0 .LT. NCTOT) IBUFX(1)=1
      IF(0 .LT. INESL) IBUFX(2)=1
      IF(NLC .GE. NFL) IBUFX(3)=1
      IF(0 .LT. NCOOL) IBUFX(4)=1
      IF(0 .LT. NSKL) IBUFX(5)=1
      IF(0 .LT. NCDLP) IBUFX(6)=1
      IF(0 .LT. NPPP) IBUFX(7)=1
      IF(NPPB .LT. 30) IBUFX(17)=1
      IF(0 .LT. NCPTT) IBUFX(74)=1
      IF(0 .LT. IPTCU) IBUFX(75)=1
C      TURN ON  LIGHTS
      DO 1 I=1,80
      IF(IBUFX(I) .EQ. 1) CALL LIGHT(I,1)
    1 CONTINUE
C**********
      PAUSE 6
      CALL CONTRL(1,0)
      CALL CONTRL(2,0)
      CALL CONTRL(4,0)
      CALL CONTRL(2,1)
      CALL CONTRL(3,1)
      CALL CONTRL(4,1)
      CALL GONT(1,100.,1.)
      CALL GONT(2,90C.,1.)
      CALL GONT(3,250.,1.)
      DO 4 I=1,10
    4 IS(I)=0
  101 FORMAT(20I4)
  102 FORMAT(10F8.0)
  103 FORMAT(12HENTER BRANCH/)
  107 FORMAT(1X,10F7.2)
  197 FORMAT(1X,10I6)
  551 FORMAT(1X,16HTIMERS,SERVOS ON/)
    2 RETURN
      END
```

SECOND SUBMODULE

```
 DECK FOR SECOND SUBMODULE
 SUBROUTINE DISK
  THIS IS THE SECOND SUBMODULE, ITS FILE NAME IS UPS2CI
  IT CONSISTS OF THE PREVIOUS EVNTS ROUTINE FOR EVENTS 1 TO 38
 LOGICAL LOGVAL,SENSW
 COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
1THEG,IFIN,MXX,NPRNT,NCROR,OUT
 COMMON ATRIB(6),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NC(4)
*,ISTRIB(5)
 COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
1(3),  JWT(3),  KWT(3),  TCHAN(3)
 COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
1  NRANK(3),  ISTRT(3),   LOCOEV(31),  INTNGT(17,2),  IPRA(17,2),
2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
*   IFROL(2),   JJ,   II,
4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
5,IAFFL(3),IAWHL(2),NEVC(5),IADO(11,4),ICV(5,9)
6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
7,INGS(5),IRPO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(5),
8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2)
9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INNOUR,NACOL(3),ITCOR(15),IBLOW(15),
*MOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
 COMMON /BLOK3/  TP1A,  TP1B,  TP1SLA,  TP1SLB,  TPCOOL,
1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3TMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
4),TH(3),TLO(3),TMI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RAT(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSE(3),RSLG(3),SLRA(3),POT
6(3),TDOWN(3),TDL(3),TDM(3),CRL(3),CRM(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,ISLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCCNT(5)
 COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
* LOGVAL,NFRNC,NRE
 COMMON /SAVE/ NSET(6,60),MODSCT(13)
 COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
1  NCRAN(12),  ICO(5),  MODUL
*****************************************************************

LIGHT MODEL STATE WHEN  LIGHT IS ON                  SW ON IN  SW OFF IN
  1  PI    COPPER CAKES AVAILABLE                    1         3
  2  PI    SLAG READY                                4         6
  3  LADLE REQUIRES BUMPING                          LADLE     11
  4  LADLE COOLING PRIOR TO BUMPING                  13        11
  5  SKULL AVAILABLE AT BUMPING BLOCK                11        17
  6  PIG R PIGS AVAILABLE                            27        23
  7  PIG P PIGS AVAILABLE                            24        26
  8  AN F   MORE COPPER REQUIRED IN FURNACE   4 38,37       14, 34
  9  AN F   MORE COPPER REQUIRED IN FURNACE   5 38,37       14, 34
 10  AN F   MORE COPPER REQUIRED IN FURNACE   6 38,37       14, 34
 11  AN F   FURNACE  4 FULL AND WAITING         30,37,14  34,18
 12  AN F   FURNACE  5 FULL AND WAITING         30,37,14  34,18
 13  AN F   FURNACE  6 FULL AND WAITING         30,37,14  34,18
 14  AN F   FURNACE  4 READY FOR CASTING        35,37     36,18
 15  AN F   FURNACE  5 READY FOR CASTING        35,37     36,18
 16  AN F   FURNACE  6 READY FOR CASTING        35,37     36,18
 17  PIG R EMPTY MOULDS IN PIG BAY               33        21
 18  EMPTY SLAG LADLE RETURNED TO PI             6         7
 72  CASTING ON WHEEL   4                        SUBR. 4   37
 73  CASTING ON WHEEL   3                        SUBR. 4   37
 74  PT  COPPER CAKES ARE READY                  2         5
 75  PT    COPPER IS READY                       8         9
 76  PT    EMPTY LADLE NEEDS RETURNING           9         10

*****************************************************************

   L I S T   O F   E V E N T S       ****************************

***************************   PILOT PLANT   ********************

      1       COPPER CAKES READY
      3       CAKES HAVE BEEN PICKED UP
      4       SLAG READY
      6       SLAG HAS BEEN PICKED UP
      7       EMPTY SLAG LADLE RETURNED

***************************   PROTOTYPE PLANT   ****************

      2       COPPER CAKES READY
      5       COPPER CAKES PICKED UP
      8       COPPER READY
      9       COPPER IS BEING PICKED UP
     10       EMPTY SLAG LADLE RETURNED

***************************   LADLES   *************************

     11       COMPLETION OF BUMPING
     12       PICK UP OF SCRAP OR FLUE DUST
     13       LADLE DEPOSITED FOR COOLING

***************************   SKULLS   *************************

     17       SKULL HAS BEEN PICKED UP
```

```
C*****************************   COPPER PIGS   ********************
C
C           21        PIGS HAVE BEEN POURED
C           23        PICKED PIGS AT PIG BAY
C           24        DUMPED PIGS ON PIG PILE
C           26        PICKED PIGS AT PIG PILE
C           27        SOME PIGS HAVE COOLED SUFFICIENTLY FOR PICK-UP
C           33        MOULDS AVAILABLE AT PIG BAY
C
C*****************************   ANODE FURNACES   *****************
C
C           14        ANODE SLAG PICKED UP AT A4
C           15        ANODE SLAG PICKED UP AT A5
C           16        ANODE SLAG PICKED UP AT A6
C           18        PULL COLLAR ON A4
C           19        PULL COLLAR ON A5
C           20        PULL COLLAR ON A6
C           22        PICK UP COLLAR AT A4
C           28        PICK UP COLLAR AT A5
C           29        PICK UP COLLAR AT A6
C           30        POURED COPPER INTO A4
C           31        POURED COPPER INTO A5
C           32        POURED COPPER INTO A6
C           34        START REFINING
C           35        FURNACE 4, 5, OR 6 IS READY FOR CASTING
C           36        START CASTING ON ANODE WHEEL 3 OR 4
C           37        FINISHED CASTING ON ANODE WHEEL 3 OR 4
C           38        FURNACE 4, 5, OR 6 IS AGAIN READY FOR COPPER INPUT
C
C******************************************************************
      KODE=ISTRIB(NCR+1)
      TTNOW=TIME(TLAST).
      CALL CLOCK(IER)
      IF(.NOT. SENSW(2)) WRITE(6,2000) IX,IER
      IF(IX .LE. 0) GO TO 100
      GO TO (1,  2,  3,  4,  5,  6,  7,  8,  9,  10,
     2  11,  12,  13,  14,  14,  14,  17,  18,  18,  18,
     3 21,   22,   23,   24,   101,   26,   27,   28,   29,   13,
     4  30,  30,  33,  101,  35,  101,  17,  38  1 , IX
  100 PAUSE 4
      J   .500
  101 PAUSE 5
      J   .500
C      EVENT 1, COPPER CAKES READY, TRIGGERED BY EVENT FILE 1
    1 CALL LIGHT(1,1)
      CALL BELLS(4,1)
      NCTOT=NCTOT+1
      NCPPL=NCPPL+1
      ATRIB(1)=TTNOW+UNFRM(TP1A,TP1B)
      ATRIB(2)=1.
      ISET=12
 1000 CALL FILEM(1,NSET)
  500 IF (.NOT. SENSW(7)) ISET=0
      IX=0
  501 IF(SENSW(2)) GO TO 1400
C *** WRITE STATEMENTS
      WRITE(6,3000)NCTOT,NCPPL,NCPU,NTANC,NCOLP,NPPB,NPIGS,NPRP,
     *NCOOL,NLC,NSKL,NPPP,NPRPP
      WRITE(6,3001) ATRIB
 1400 CONTINUE
 1401 RETURN
C       COPPER CAKES AT PROTOTYPE PLANT
    2 CALL LIGHT(74,1)
      CALL BELLS(4,2)
      NCPT=NCPT+4
      NCPTT=NCPTT+4
      ATRIB(2)=2.
  211 ATRIB(1)=TTNOW+160.
      J   .1000
C       CODE 3,  COMPLETION OF COPPER CAKE PICK-UP
    3 NCPU=KODE-10
      IF(NCTOT .LE. 0) TYPE 1402
      IF(NCPU .GT. NCTOT) NCPU=NCTOT
      NCTOT=NCTCT-NCPU
      IF(NCTOT .LT. 1) CALL LIGHT(1,0)
C          OF CAKES PICKED UP IS STORED IN JSAVE
  310 ISAVE(NCR)=NCPU+100
      J   .900
C       CODE 4 TO INDICATE THAT SLAG IS READY AT THE PILOT
C       PLANT AND SCHEDULE THE NEXT TYPE 4 EVENT
    4 CALL LIGHT(2,1)
      LRS=LRS+1
      CALL BELLS(4,4)
      IRESL=IRESL+1
      ATRIB(1)=TTNOW+UNFRM(TP1SLA,TP1SLB)
      ATRIB(2)=4.
      ISET=13
      J   .1000
C       COMPLETION OF COPPER PICK-UP AT PROTOTYPE PLANT
    5 NCPU=KODE-10
      IF(NCPTT .LE. 0) TYPE 1402
      IF(NCPU .GT. NCPTT) NCPU=NCPTT
      NCPTT=NCPTT-NCPU
      IF(NCPTT .LT. 1) CALL LIGHT(74,0)
      J   .310
C       SLAG HAS BEEN PICKED UP AT THE PILOT PLANT
    6 IF(IRESL .LT. 1) GO TO 611
      IRESL=IRESL-1
      CALL LIGHT(2,0)
      CALL LIGHT(18,1)
  610 CALL LADLE
      ISAVE(NCR)=100
      J   .900
  611 TYPE 612
      J   .900
C       RETURN EMPTY SLAG LADLE TO PILOT PLANT
    7 CALL LIGHT(18,0)
      J   .900
```

```
C       OPPER READY AT PROTOTYPE PLANT
   8 CALL LIGHT(75,1)
     LCPU=LCPU+1
     CALL BELLS(4,8)
     IPTCU=IPTCU+1
     ATRIB(2)=8.
     J    .211
C       PICK UP LADLE OF COPPER AT PROTOTYPE PLANT
   9 IF(IPTCU .LT. 2) CALL LIGHT(75,0)
     IPTCU=IPTCU-1
     CALL LIGHT(76,1)
     J    .610
C       RETURN EMPTY COPPER LADLE TO PROTOTYPE
  10 CALL LIGHT(76,0)
     J    .500
C       CODE 11 COMPLETION OF DUMPING
  11 IF(NCOOL .LE. 0) GO TO 110
     NLC=NLC-NFL
     IF(NSKL .EQ. 0) ISET=6
     NCOOL=NCOOL-1
     IF(NCOOL .LT. 1) CALL LIGHT(4,0)
     IF(NLC .LT. NFL) CALL LIGHT(3,0)
     CALL LIGHT(5,1)
     NSKL=NSKL+1
     NAPL=NAPL+1
     J    .500
C       SCRAP OR FLUE DUST PICK-UP
  12 ISAVE(NCR)=100
     J    .500
 110 TYPE 1900
     J    .502
C       CODE 13 DEPOSIT A LADLE FOR COOLING
  13 CALL LIGHT(4,1)
     NCOOL=NCOOL+1
     J    .500
C       ANODE SLAG  IS BEING SKIMMED
  14 NFRNC=IX-13
     IF(IAPPL(NFRNC) .EQ. 0) GO TO 1414
     IF(IAPPL(NFRNC) .NE. 1) GO TO 1444
1414 CALL LIGHT(NFRNC+7,0)
     CALL LIGHT(NFRNC+10,1)
     CALL CLOCK(IATIME(NFRNC))
     IAPPL(NFRNC)=6
     ISAVE(NCR)=50
     ILADA(NFRNC)=ILADA(NFRNC)-50
     ISET=20
     J    .500
1444 IX=NFRNC+3
     TYPE 1450,IX
     J    .500
C       CODE 17 AT COMPLETION OF SKULL PICK-UP
  17 IF(NSKL .LE. 0) GO TO 1406
     NSKL=NSKL-1
     ISAVE(NCR)=100
     IF(NSKL .LT. 1) CALL LIGHT(5,0)
     J    .500
1406 TYPE 1403
     ISAVE(NCR)=0
 502 IX=21
     J    .501
C       CODE 21 COMPLETION OF PIG POURING
  21 NPIGS=.04*FLOAT(ISAVE(NCR))+.5
     ISAVE(NCR)=0
     IF(NPIGS .LE. 0) GO TO 500
     NPPB=NPPB+NPIGS
     IF(NPPB .GT. 30) GO TO 210
C       TURN OFF EMPTY MOULD LIGHT WHEN 27 OR MORE PIGS IN PIG BAY
     IF(NPPB .GE. 27) CALL LIGHT(17,0)
C       FILE EVENT 27 IN TIME FILE TO INDICATE WHEN PIGS ARE COOL
 212 ATRIB(1)=TTNOW+TPCOOL
     ATRIB(2)=27.
     ATRIB(3)=NPIGS
     NTPPB=NTPPB+NPIGS
     J    .1000
 210 IER=NPPB-30
     NPIGS=NPIGS-IER
     ISAVE(NCR)=25*IER
     F=FLOAT(ISAVE(NCR))*.01
     TYPE 2110,F,NCR
     NPPB=30
     IF(NPIGS .LE. 0) GO TO 500
     J    .212
C       PICK UP ANODE COLLAR
  22 NFRNC=1
 221 NF=NFRNC+3
     IF(NACOL(NFRNC) .LT. 1) GO TO 220
     NACOL(NFRNC)=NACOL(NFRNC)-1
     ISAVE(NCR)=100
     J    .500
  23 NFRNC=2
     J    .221
  24 NFRNC=3
     J    .221
 220 TYPE 222,NF
     J    .500
C       CODE 25 COMPLETION OF PIG PICK-UP AT PIG BAY.
  25 NPPPB=KODE-10
     IF(NPPPB .GT. NCOLP) NPPPB=NCOLP
     IF(NCOLP .LE. 0) TYPE 1402
     ISAVE(NCR)=NPPPB*100
     NCOLP=NCOLP-NPPPB
     NPPB=NPPB-NPPPB
     IF(NCOLP .LT. 1) CALL LIGHT(6,0)
     IF((NPPP+NPPB).LT.10) ISET=15
     IF(NCOLP.LT.0) ISET=0
     K=0
     DO 230 I=1,3
     IF(IAPPL(I).EQ.0)K=K+1
 230 CONTINUE
     IF(K .EQ. 0) ISET=15
     IF(NPPB .LT. 26) GO TO 231
     J    .500
 231 ATRIB(1)=TTNOW+IP.
     ATRIB(2)=33.
     J    .1000
C       CODE 24 DUMPING OF PIGS OR CAKES ON PIG PILE
  24 NPPP=NPPP+ISAVE(NCR)/100
     ISAVE(NCR)=0
     CALL LIGHT(7,1)
     J    .500
C       EVENT 26. COMPLETION OF PIG PICK-UP AT PIG PILE
  26 NPRPP=KODE-10
     IF(NPRPP .GT. NPPP) NPRPP=NPPP
     IF(NPPP .LE. 0) TYPE 1402
     ISAVE(NCR)=NPRPP*100
     NPPP=NPPP-NPRPP
     IF(NPPP .LT. 1) CALL LIGHT(7,0)
     IF((NPPP+NPPB) .LT. 10) ISET=15
     IF(NCOLP.LE.0) ISET=0
     J    .500
C       EVENT 27 A NUMBER OF PIGS HAVE COOLED SUFFICIENTLY
C       TO BE PICKED UP
  27 NCOLP=NCOLP+ISTRIB(1)
     IF( NCOLP .GT. 0) CALL LIGHT(6,1)
C       EMPTY PIG BAY ONLY AFTER AT LEAST 10 PIGS HAVE ACCUMULATED
     IF(NPPB .GT. 10) ISET=15
     J    .500
C       EVENT 30, 31, OR 32 POURING A LADLE OF COPPER INTO AN
C       ANODE FURNACE
  30 NFRNC=IX-29
     IF(IAPPL(NFRNC) .NE. 0) GO TO 300
     ILADA(NFRNC)=ILADA(NFRNC)+ISAVE(NCR)
     NTLAD(NFRNC)=NTLAD(NFRNC)+ISAVE(NCR)
     ISAVE(NCR)=0
     IF((ICAPA(NFRNC)-ILADA(NFRNC) .GE. 50) GO TO 500
     CALL LIGHT(NFRNC+10,1)
     CALL BELLS(4,IX)
     IAPPL(NFRNC)=1
     ISET=16
     J    .350
 300 NFRNC=NFRNC+3
     TYPE 301, NFRNC
     J    .500
C       MOULDS AVAILABLE AT PIG BAY
  33 CALL LIGHT(17,1)
     J    .500
C       EVENT 35 ANODE FURNACE 4,5 OR 6 IS READY FOR CASTING
  35 NFRNC=ISTRIB(3)-3
     IAPPL(NFRNC)=3
     CALL LIGHT(NFRNC+13,1)
          CALL BELLS(4,35)
     ISET=17
 350 CALL CLOCK(IATIME(NFRNC))
     J    .500
C       EVENT 37, COMPLETION  OF CASTING ON ANODE WHEEL  4 OR  5
  37 L=72
     NFRNC=ISTRIB(1)-3
     IF(NFRNC .EQ. 3) L=73
     IAPPL(NFRNC)=5
     CALL CLOCK(IATIME(NFRNC))
     LL=76-L
     IAWHL(LL)=0
     CALL LIGHT(L,0)
     CALL LIGHT(NFRNC+7,1)
     CALL LIGHT(NFRNC+10,1)
     CALL LIGHT(NFRNC+13,1)
     TYPE 371,ISTRIB(1)
     CALL BELLS(4,37)
     LL=.0001*FLOAT(ILADA(NFRNC))*FLOAT(NAPLC(NFRNC))+.1
     NTANC=NTANC+LL
     NANCST(NFRNC)=NANCST(NFRNC)+LL
     ILADA(NFRNC)=0
     IX=6
     J    .501
C       PULL COLLAR ON ANODE FURNACE
  18 NFRNC=IX-17
     IF(IAPPL(NFRNC) .EQ. 7) GO TO 180
     IF(IAPPL(NFRNC) .NE. 5) GO TO 500
     IAPPL(NFRNC)=7
     CALL CLOCK(IATIME(NFRNC))
     CALL LIGHT(NFRNC+7,0)
     J    .500
 180 IAPPL(NFRNC)=8
     CALL CLOCK(IATIME(NFRNC))
     CALL LIGHT(NFRNC+10,0)
     CALL LIGHT(NFRNC+13,0)
     NACOL(NFRNC)=NACOL(NFRNC)+1
     ISET=19
     ATRIB(2)=38.
     ATRIB(3)=NFRNC+3
     IF(NFRNC .EQ. 1) GO TO 1004
     IF( NFRNC .EQ. 3) GO TO 370
     ATRIB(1)=TTNOW+UNFRM(TAP05A,TAP05B)
     J    .1000
1004 ATRIB(1)=TTNOW+UNFRM(TAP04A,TAP04B)
     J    .1000
 370 ATRIB(1)=TTNOW+UNFRM(TAP06A,TAP06B)
     J    .1000
C       EVENT 38 ANODE FURNACE 4,5 OR 6 IS AGAIN READY TO ACCEPT COPPE-
  38 NFRNC=ISTRIB(1)-3
     IAPPL(NFRNC)=0
     CALL LIGHT(NFRNC+7,1)
     CALL BELLS(4,38)
     ISET=6
     J    .350
 222 FORMAT(1X,10HNO COLLAR AT A,I1)
 301 FORMAT(9HANODE .,I2,9HNOT READY/)
 371 FORMAT(1X,10HPULL COLLAR ON A,I1)
 612 FORMAT(5HNO BS/)
1402 FORMAT(8HNO CAKES/)
1403 FORMAT(6HNO SKL/)
1450 FORMAT(1X,12HNO SLAG AT A,I1/)
1900 FORMAT(6HNO LDL/)
2000 FORMAT(1X,5HEVENT,I4,7HAT TIME,I5)
2110 FORMAT(F4.2,24H LADLES OF CU LEFT ON CR,I2/)
3000 FORMAT(1X,7HNCTCT,NCPPL,NCPU,NTANC,NCOLP,NPPB,NPIGS,NPRPB,NCOOL,N
    *LC,NSKL,NPPP,NPRPP,/1X,2I6,I5,2I6,I5,3I6,I4,2I5,I6)
3001 FORMAT(1X,5HATRIB/,1X,4F8.2)
     END
```

THIRD SUBMODULE

```
   ECK FOR THIRD SUBMODULE
   SUBROUTINE DISK
C    THIS IS THE THIRD SUBMODULE, ITS FILE NAME IS UPS3CE
C IT CONSISTS OF EVENTS 60 TO 169,ADDITION AND START EVENTS
   LOGICAL LOGVAL,SENSW
   DIMENSION RFX(8)
   DIMENSION  IO(9,15),CVI(2),MUL(5,8)
   COMMON ID,IR,INIT,JEVNT,MFA,MSTOP,NX,NXC,NOQ,SCALE,ISEED,TNOW,
  1TBEG,TFIN,NXX,NPRNT,NCROR,OUT
   COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
  *,ISTRIB(5)
   COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
  1(3),  JWT(3),  KWT(3),  TCRAN(3)
   COMMON /BLOK2/ ICAPA(3),  ILADR(3),  IBUFX(12),  IBUFY(12),
  1  NRANK(3),  ISTRT(3),  LOCDEV(3),  INTNGT(17,2),  IPRA(17,2),
  2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCDLP,  NPPP,  NCTOT,  NCPPL,
  3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
  *   ITROL(2),  JJ,  II,
  4IOLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
  5,IAFFL(3),IAMHL(2),NEVC(5),IADD(11,4),ICV(5,9)
  6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
  7,INGS(5),IRFO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
  8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2)
  9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
  *MJP(5,8),MTREQ(5),MTAV(2),IREP(5,6)
   COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOCL,
  1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
  2  TCST5A,  TCST55,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
  3XNAE,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),VMMAX(3),VMMIN(3),RRAT(3
  4),TH(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTW(2),TRAIL(2),RATI(3),SLEV
  5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
  6(3),TDOWN(3),TCL(3),TOM(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
  7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
  8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
   COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10)
  * LOGVAL,NFRNC,NRE
   COMMON /SAVE/ NSET(6,60),MODSCT(13)
   COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCPND(12),
  1  NCRAM(12),  ICO(5),  MODUL
   DATA IO(1,1),IO(1,2),IO(1,3),IO(1,4),IO(1,5),IO(1,6),IO(1,7),IO(1,
  18),IO(1,9),IO(1,10),IO(1,11),IO(1,12),IO(1,13),IO(1,14),IO(1,15)/
  21,1,1,1,1,2,1,1,2,1,1,1,1,2,1/
   DATA IO(2,1),IO(2,2),IO(2,3),IO(2,4),IO(2,5),IO(2,6),IO(2,7),IO(2,
  18),IO(2,9),IO(2,10),IO(2,11),IO(2,12),IO(2,13),IO(2,14),IO(2,15)/
  21,1,1,1,1,2,1,1,2,1,1,1,1,2,1/
   DATA IO(3,1),IO(3,2),IO(3,3),IO(3,4),IO(3,5),IO(3,6),IO(3,7),IO(3,
  18),IO(3,9),IO(3,10),IO(3,11),IO(3,12),IO(3,13),IO(3,14),IO(3,15)/
  21,1,1,1,1,2,1,1,2,1,1,1,1,2,1/
   DATA IO(4,1),IO(4,2),IO(4,3),IO(4,4),IO(4,5),IO(4,6),IO(4,7),IO(4,
  18),IO(4,9),IO(4,10),IO(4,11),IO(4,12),IO(4,13),IO(4,14),IO(4,15)/
  21,5,1,1,1,1,1,1,1,1,3,1,1,1,1/
   DATA IO(5,1),IO(5,2),IO(5,3),IO(5,4),IO(5,5),IO(5,6),IO(5,7),IO(5,
  18),IO(5,9),IO(5,10),IO(5,11),IO(5,12),IO(5,13),IO(5,14),IO(5,15)/
  26,1,1,1,1,1,1,1,1,1,1,1,1,1,1/
   DATA IO(6,1),IO(6,2),IO(6,3),IO(6,4),IO(6,5),IO(6,6),IO(6,7),IO(6,
  18),IO(6,9),IO(6,10),IO(6,11),IO(6,12),IO(6,13),IO(6,14),IO(6,15)/
  26,1,1,1,1,6,1,1,1,1,1,1,1,1,1/
   DATA IO(7,1),IO(7,2),IO(7,3),IO(7,4),IO(7,5),IO(7,6),IO(7,7),IO(7,
  18),IO(7,9),IO(7,10),IO(7,11),IO(7,12),IO(7,13),IO(7,14),IO(7,15)/
  21,1,1,1,1,1,3,1,1,1,1,1,1,1,1/
   DATA IO(8,1),IO(8,2),IO(8,3),IO(8,4),IO(8,5),IO(8,6),IO(8,7),IO(8,
  18),IO(8,9),IO(8,10),IO(8,11),IO(8,12),IO(8,13),IO(8,14),IO(8,15)/
  26,1,1,1,1,6,1,4,1,1,1,1,1,1,1/
   DATA IO(9,1),IO(9,2),IO(9,3),IO(9,4),IO(9,5),IO(9,6),IO(9,7),IO(9,
  18),IO(9,9),IO(9,10),IO(9,11),IO(9,12),IO(9,13),IO(9,14),IO(9,15)/
  26,1,1,1,1,2,1,1,1,1,1,1,1,1,1/
   DATA MUL(1,1),MUL(1,2),MUL(1,3),MUL(1,4),MUL(1,5),MUL(1,6),
  1MUL(1,7),MUL(1,8)/4,3,2,2,2,2,1,1/
   DATA MUL(2,1),MUL(2,2),MUL(2,3),MUL(2,4),MUL(2,5),MUL(2,6),
  1MUL(2,7),MUL(2,8)/4,3,2,2,2,2,1,1/
   DATA MUL(3,1),MUL(3,2),MUL(3,3),MUL(3,4),MUL(3,5),MUL(3,6),
  1MUL(3,7),MUL(3,8)/4,3,2,2,2,2,1,1/
   DATA MUL(4,1),MUL(4,2),MUL(4,3),MUL(4,4),MUL(4,5),MUL(4,6),
  1MUL(4,7),MUL(4,8)/4,3,2,2,2,2,1,1/
   DATA MUL(5,1),MUL(5,2),MUL(5,3),MUL(5,4),MUL(5,5),MUL(5,6),
  1MUL(5,7),MUL(5,8)/5,4,3,2,2,2,1,1/
   DATA RFX(1),RFX(2),RFX(3),RFX(4),RFX(5),RFX(6),RFX(7),RFX(8)/.45,
  1.45,.45,.45,.45,.45,.45,1./
   DATA NY/2HYE/
   IDIV=1
   III=0
   IDI=0
   RRAT=1.1
   RMTTE=0.
   TTNOW=TIME(TLAST)
   IF(.NOT. SENSW(4)) WRITE(6,2000) IX,TTNOW
   I=IX-59
   GO TO (60,
  160,  60,  60,  60,  65,  65,  65,  65,  65,  70,
  270,  70,  70,  70,  75,  75,  75,  75,  75,  80,
  380,   80,   80,   80,   85,   85,   85,   85,   85,    90,
  490,  90,  90,  90,  95,  95,  95,  95,  95,  101,
  5101,  101,  101,  101,  105,  105,  105,  105,  105,  110,
  6110,  110,  110,  110,  120,  120,  120,  120,  120,  120,
  7120,  120,  120,  120,  125,  125,  125,  125,  125,  130,
  8130,  130,  130,  130,  135,  135,  135,  135,  135,  140,
  1140,140,140,140,145,145,145,145,145,150,150,150,150,150,155,155,
  2155,155,155,160,160,160,160,160,165,165,165,165,165),I
91 PAUSE 9
   J   .900
1000 CALL FILEM(1,NSET)
 500 CONTINUE
C  UPDATE CONVERTER ADDITION INFO
     IAFO(NCV,JIO)=IAFO(NCV,JIO)+ISAVE(NCR)
C CHECK IF COLLAR PULLING IS NEEDED
     GO TO(550,550,550,501,550,501,501,550,501),JIO
550  MT(NCV,20)=MT(NCV,20)+ISAVE(NCR)
     ITL=MT(NCV,20)/100
     IF(ITL.NE.IC) GC TO 501
     ISET=25
     NNCV=NCV+2
     WRITE(IDIV,3009) NNCV
     CALL BELLS(4,IX)
3009 FORMAT(5X,16HPULL COLLAR ON C,I2//)
501  IX=0
1400 IF(SENSW(3)) GO TO 1401
     WRITE(6,3001) ATRIB
1402 CONTINUE
     III=ISTAT(NCV)
     IOI=IO(JIO,III)
     WRITE(6,3002)NCV,III,IOI,JIO,NN,ISAVE(NCR),NCR,NSN,NFN
3002 FORMAT(1X,70HNCV,III,IOI,JIO,NN,ISAVE,NCR,NSN,NFN'/1X,9I4)
     WRITE(6,3C03)(CV(NCV,J),J=1,9)
     WRITE(6,3003) F,SLG,FRS,CUS,FFF,FCU,STARTA,STARTB,TPU(NC.)
     WRITE(6,3C03) CVI(1),CVI(2),RMTTF
3003 FORMAT(9F8.3)
1401 IF(.NOT.SENSW(7)) ISET=0
     RETURN
C    CONVERTER EVENTS
C    DETERMINE CONVERTER NO. NCV FROM EVENTS CODE IX
C    SPECIFY WEIGHT, COMPOSITION OF CONVERTER INPUTS BY SETTI.G
C    JIO IN IACD(JIO,J) TO THE APPROPRIATE VALUE
C    ADD SCRAP
60   NCV=IX-59
     JIO=5
     J   .4000
C    ADD REACTOR SLAG
65   NCV=IX-64
     JIO=6
     J   .4000
C    ADD PIGS
70   NCV=IX-69
     MT(NCV,8)=TTNOW+FLOAT(MT(NCV,8)-MT(NCV,2))
     MT(NCV,14)=FLOAT(MT(NCV,14)+MT(NCV,2))-TTNOW
     IAFO(NCV,15)=IAFO(NCV,15)-(ISAVE(NCR)+10)/100
     JIO=4
     IF(0.LT.IAFO(NCV,15)) GO TC 4000
     CALL LIGHT(56+NCV,0)
     J   .4000
C    ADD HIGH CU SLAG
75   NCV=IX-74
     JIO=9
C SET CS ON FLOOR IF CONVERTER WANTS FD.STATE 7
C MEANS THAT NO OTHER CONVERTER COULD TAKE CS
     IF(ISTAT(NCV).EQ.7) GO TO 750
     IF(ISTAT(NCV).NE.2) GO TO 4000
750  INGS(NCV)=ISAVE(NCR)
     ISAVE(NCR)=0
     J   .900
C  ADD WET CHARGE MATTE
80   JIO=2
     J   .4000
85   CONTINUE
C    ADD REVERB MATTE
90   NCV=IX-89
     IAFO(NCV,10)=IAFO(NCV,10)+ISAVE(NCR)
     J=ISTRT(NCR)
     CALL LIGHT(66+NCV,0)
900  GO TO(91,801),J
C  ADD DRY CHARGE MATTE
 91 JIO=1
     J   .4000
C    ADD TRANSFER MATTE
95   NCV=IX-94
     IAFO(NCV,10)=IAFO(NCV,10)+ISAVE(NCR)/4
     JIO=3
     J   .4000
C START OF COPPER BLOW
 104  CALL LIGHT(51+NCV,0)
C RESET REFERENCE TO CA
     CALL ONFRL(ICV(NCV,9),.TRUE.,IER)
C UNLATCH CA, PUT IA,IB TO OPERATE
     IL=ICV(NCV,8)
     CALL LIGHT(IL,1)
     ISTAT(NCV)=2
     MT(NCV,2)=TTNOW
     CALL LIGHT(96+NCV,0)
     J   .900
C    ADD FLUE DUST
109  NCV=IX-104
C SCHEDULE PIG ADDITION
     ATRIB(1)=TTNOW+FLOAT(MT(NCV,15))
     ATRIB(2)=NCV+264
```

```
      ATRIB(3)=NCV
      CALL FILEM(1,NSET)
      JIO=7
      MT(NCV,2)=TTNOW
      J    .4000
  ADD SKULLS
 110 NCV=IX-109
      ITAT=ISTAT(NCV)
      IF(ITAT.NE.8) GO TO 1100
      CALL LIGHT(56+NCV,0)
      CALL LIGHT(66+NCV,1)
 1100 JIO=8
      J    .4000
  START SLAG BLOW
  RESET REFERENCE TO CA TO OPERATIONAL CONDITION
 115 CALL QMPRL(ICV(NCV,9),.TRUE.,IER)
      IF(ISTAT(NCV).NE.9) GO TO 1151
      MT(NCV,3)=TTNOW
      MT(NCV,6)=MT(NCV,6)+MT(NCV,3)-MT(NCV,4)
      J    .1154
 1151 MT(NCV,9)=TTNOW+FLOAT(MT(NCV,5)-MT(NCV,3))
      MT(NCV,3)=TTNOW
  PUT IA,IB TO OPERATE
 1154 IL=ICV(NCV,6)
      CALL LIGHT(IL,1)
  TURN OFF END POINT LIGHT,SET ISTAT TO 1 FOR SLAG STAGE IF BLOW LIMIT REACHED
      CALL LIGHT(51+NCV,0)
      NBLW=NEVC(NCV)
      LAD=IAFO(NCV,10)/100
      IMUL=MUL(NCV,NBLW)
      IF(LAD.GE.IMUL) GC TO 1155
  MORE MATTE ALLOWED
      ISET=7
      ISTAT(NCV)=6
      J    .500
 1155 NNCV=NCV+2
      WRITE(IDIV,1150) LAD,NNCV,NBLW
 1150 FORMAT(1X,I2,1X,10HMATTE IN C,I2,16H,FILLED FOR BLOW,I3)
  TURN OFF ADDITICN LIGHT
      CALL LIGHT(56+NCV,0)
      ISTAT(NCV)=1
      J    .500
  CHECK IF INPUT IS LEGAL OR NOT USING ARRAY IO(JIO,ISTAT(NCV))
  SPECIFY CONVERTER CONTINUING HIGH BLOW AFTER SLAG SKIM
 120 NCV=ATRIB(3)+.5
      J    .115
  SPARE EVENT CCDES
 125  CONTINUE
 125  CONTINUE
 130 CONTINUE
 135 CONTINUE
 140 CONTINUE
 145 CONTINUE
 150 CONTINUE
 155 CONTINUE
 160 CONTINUE
 165 CONTINUE
 4000 I=ISTAT(NCV)
      IJ=IO(JIO,I)
      GO TO (4100,4200,4300,4300,4300,4200),IJ
      TYPE ILLEGAL ADDITION REPORT
 4100 WRITE(IDIV,4110)
 4110 FORMAT(1X,70H,ILLEGAL ADDITION REPORT*/)
      J    .990
  MATERIAL ADDITICN DURING A SLAG STAGE
 4200 NN=1
      J    .5000
  MATERIAL ADDITICN DURING A COPPER STAGE
 4300 NN=2
      DETERMINE AMOUNTS OF CU2S,FES,SLAG,FREE CU,FREE FE
      IADD(JIO,1) TO IADD(JIO,4) ARE WEIGHTS AN PERCENTAGES
      OF CU,FE,S RESPECTIVELY IN INPUT CATEGORY JIO
 5000 F=ISAVE(NCR)
      SLG=F*FLOAT(IADD(JIC,1)*(100-IADD(JIO,3)-IADD(JIO,4)-IADD(JIO,2)
     1))*.01
      ALL FE AS FES
      IF(FLOAT(IADD(JIO,4))*.01.GE..00575*FLOAT(IADD(JIO,3)))GO TO
     *5310
      FES=F*FLOAT(IADD(JIO,1)*IADD(JIO,4))*.027
      CUS=0.0
      FCU=F*FLOAT(IADC(JIO,1)*IADD(JIO,2))*.01
      FFE=F*FLOAT(IADD(JIO,1)*IADD(JIO,3))*.01-FES*.0064
      NSN=1
      NFN=2
      J    .5601
 5310 FES=F*FLOAT(IADD(JIO,1)*IADD(JIO,3))*.0197
      ALL CU AS CU2S
      IF(FLOAT(IADD(JIO,4))*.01.GE..00575*FLOAT(IADD(JIO,3))*.00252
     **FLOAT(IADD(JIO,2))) GO TO 5320
      CUS=(FLOAT(IADC(JIO,4))*.01-.00575*FLOAT(IADD(JIO,3)))*4.960
     *FLOAT(IADC(JIO,1))*F
      FCU=(FLOAT(IADD(JIO,2))*.01-3.96*(FLOAT(IADD(JIO,4))*.01-.00575*
     *FLOAT(IADC(JIO,3))))*FLOAT(IADD(JIO,1))*F
      FFE=0.0
      NSN=2
      NFN=1
      J    .5601
 5320 CUS=FLOAT(IADD(JIO,1)*IADD(JIO,2))*.0125*F
      FCU=0.0
      FFE=0.0
 5601 GO TO(5600,5700), NN
  RESET ICS AFTER MATERIAL ADDITION TO SLAG STAGE
 5600 CALL QRRADR(CVI,ICV(NCV,3),2,IER)
      FES=(FES-.691*FCU)*.01
      SLG=(SLG+1.829*FFE+.0016*FCU)*.01
  RESET ICS AFTER MATERIAL ADDITIONS TO SLAG STAGE
  STARTA,THE IC FOR FES,DEPENDS ON BLOW NO. NEVC
      IF(FES.LT.0.) FES=0.
```

```
      NNEVC=NEVC(NCV)
C DETERMINE IF CONVERTER CAN GO HIGH
      CV(NCV,2)=CV(NCV,2)+(CUS+1.25*FCU)*.01
      IF(CV(NCV,2).LT.CV(NCV,1)) GO TO 5602
      IF(7.LT.NNEVC) GO TO 5602
      IF(CV(NCV,8).LT.CV(NCV,2)) GO TO 5633
C DECISION TO GC HIGH OR NOT
 5629 IF(.NOT.SENSW(7)) GO TO 5630
C
C  AUTOMATIC DECISION CN NEVC(NCV)
C
        TYPE5627,TCEND(NCV),TTNOW,NEVC(NCV)
 5627 FORMAT(1X,4HAUT.,2F7.2,I4/)
      IF(NEVC(NCV).LT.4) GO TO 5602
      IF((TCEND(NCV)-60.).LT.TTNOW) GO TO 5633
      IF(NEVC(NCV).GT.5) GO TO 5633
      J    .5602
C
C
C ASK OPERATOR FOR DECISICN
 5630 NNCV=NCV+2
      IER=CV(NCV,2)*.8/19.
      WRITE(IDIV,5631) NNCV,IER
 5631 FORMAT(1X,1HC,I2,5H CCNT,I3,21H LB CU, YE TO GO HIGH/)
      ACCEPT 5632,MANS
 5632 FORMAT(A2)
      IF(MANS.NE.MY) GO TO 5602
 5633 NEVC(NCV)=8
      NNEVC=8
      J    .5605
 5602 IF(JIO.EQ.3) GC TC 5604
 5605 CONTINUE
C DISTRUBUTE FES ACCORDING TC BLOW NO.
 5603 CV(NCV,3)=CV(NCV,3)+FES
      STARTA=RFR(NNEVC)*CV(NCV,3)/TPU(NCV)+CVI(1)
      CV(NCV,3)=CV(NCV,3)*(1.-RFR(NNEVC))
      J    .5610
C FES DISTRUBITION FCLLOWING TH ADDITICN
 5604 STARTA=(FES+CV(NCV,3)/3.)/TPU(NCV)
      CV(NCV,3)=CV(NCV,3)*2./3.+CVI(1)*TPU(NCV)
 5610 STARTB=CVI(2)+SLG/TPU(NCV)
      RFES=.016*CV(NCV,6)*CV(NCV,7)
      MT(NCV,13)=STARTA*TPU(NCV)/RFES
      RSL=.013*(1.+.419*FRAT)*CV(NCV,6)*CV(NCV,7)
      MT(NCV,19)=1.2*FLOAT(IADD(IO,1))/RSL
      IF(NNEVC.NE.8) GC TO 5692
C SCHEDULE PERMIT TO SKIM IN GOING HIGH BLOW
      ATRIB(1)=TTNOW+FLOAT(MT(NCV,19))
      ATRIB(2)=NCV+265
      ATRIB(3)=NCV
      CALL FILEM(1,NSET)
C UPDATE CV ARRAY
 5692 CV(NCV,4)=STARTB*TPU(NCV)
      CV(NCV,5)=0.
C CHECK IF BLOW BEING PROLCNGED
      IF(ISTAT(NCV).NE.14) GO TO 5800
      CV(NCV,3)=CV(NCV,3)+(STARTA/2.)*TPU(NCV)
      STARTA=STARTA/2.
      J    .5800
C RESET ICS AFTER MATERIAL ADDITION TO FINISH STAGE
 5700 CALL QRBACR(CVI,ICV(NCV,3),2,IER)
      CUS=(CUS+1.81*FES)*.01
      CU=(FCU-3.44*FES)*.01
      SLG=(SLG+1.829*FFE+1.16*FES)*.01
      STARTA=CVI(1)+CUS/TPU(NCV)
      STARTB=CVI(2)+CU/TPU(NCV)
      RCUS=.043*CV(NCV,6)*CV(NCV,7)
      MT(NCV,14)=STARTA*TPU(NCV)/RCUS
C UPDATE CV ARRAY
      CV(NCV,2)=STARTA*TPU(NCV)
      CV(NCV,3)=0.
      CV(NCV,4)=CV(NCV,4)+SLG
      CV(NCV,5)=STARTB*TPU(NCV)
 5800 IL=ICV(NCV,7)
      CALL LIGHT(IL,0)
      CALL QMJCAM(STARTA,ICV(NCV,1),IER)
      CALL QMJCAM(+STARTB,ICV(NCV,2),IER)
C RESET IA,IB TO HOLD VIA CLB
      IL=ICV(NCV,6)
      CALL LIGHT(IL,0)
      IL=ICV(NCV,7)
      CALL LIGHT(IL,1)
      GO TO (5810,5809),NN
 5809 IF(ISTAT(NCV).EQ.8) GO TO 5811
      J    .106
 5811 ISTAT(NCV)=5
      J    .900
 5810 RMTTE=STARTA*TPU(NCV)+CV(NCV,2)+CV(NCV,3)+CV(NCV,4)
      IF(CV(NCV,9).LT.RMTTE) GO TO 115
      ISTAT(NCV)=9
      ISET=7
      NNCV=NCV+2
      TYPE 5815,NNCV
 5815 FORMAT(1X,1HC,I2,1X,20HNEEDS MORE MATTE TO BLOW/)
      J    .900
C     CONVERTER STATUS REPORT
  990 JEST=ISTAT(NCV)
      IOC=IO(JIC,JEST)
      WRITE(IDIV,991) NCV,JEST ,JIO,IOC
      WRITE(IDIV,992)  (CV(NCV,I),I=1,9)
      J    .900
  991 FORMAT(5I13/)
  992 FORMAT(9F8.3/)
 2000 FORMAT(1X,5HEVENT,I4,9HAT TIME,F6.1)
 3001 FORMAT(1X,9HATRIB,2X,4F8.2)
      END
```

FOURTH SUBMODULE

```
C      DECK FOR FOURTH SUBMODULE
       SUBROUTINE CISK
C      THIS IS THE FOURTH SUBMODULE.   ITS FILE NAME IS UPS4CI.
C      IT IS USED FOR ACCEPTING AND INTERPRETING OPERATORS COMMANDS
      DIMENSION IW(14),JW(17),LUBR(3)
      LOGICAL SENSW,LOGVAL
      COMMON /SAVE/ NSET(6,60),MODSCT(13)
      COMMON /ARG/ IX,  ID1X,  ID2X,  YSET,  JSEQ(12),  JCMND(12),
     1  NCRAN(12),  ICC(5),  MODUL
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOG,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(6),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON/BLOK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTAT(3),  LOCDEV(31),  IWTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPS,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     4   ITHOL(2),   JJ,   II,
     *IOLE(3),KAPUT(3),NEXT(3),CSTR(3),NSLAG(3)
     5,IAFFL(3),IAWML(2),NEVC(5),IADO(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRPO(5,5),IAFG(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2) ,
     9,NCPIT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),PTREQ(5),MTAV(2),IREP(5,6)
      COMMON /BLOK2/ ISLAK(44) *
      COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIREX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),TH(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TCL(3),TCH(3),CAL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
      COMMON /BLCK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,ARE
      EQUIVALENCE (ID1,ID1X),(ID2,ID2X)
      DATA IW(1),  IW(2),  IW(3),  IW(4),  IW(5),  IW(6),  IW(7),  IW(8)
     1,IW(9),  IW(10),  IW(11),  IW(12),  IW(13),  IW(14)
     2 /2HPR,  2HDP,  2HPU,  2HGO,  2HGW,  2HRE,  2HCA,  2HBU,
     3  2HST,  2HOI,  2HIN,  2HPO,  2HLT,  2HPI  /
      DATA JW(1),  JW(2),  JW(3),  JW(4),  JW(5),  JW(6),
     1  JW(7),  JW(8),  JW(9),  JW(10),  JW(11),  JW(12),
     *   JW(13)
     2  /  2HSL,  2HMT,  2HTM,  2HCU,  2HPG,  2HSC,
     4  2HAS,  2HFD,  2HSK,  2HLD,  2HAS,   2H  ,
     4  2HCS/
      DATA LUBR(1),  LUBR(2),  LUBR(3)
     1  /2HL1,  2HL2,  2HL3 /
C       LOC   IW    JW                      LOC
C   1   A1    PR    SL                19    PP
C   2   A2    DP    MT                20    PB
C   3   A3    PU    TM                21    BB
C   4   A4    GC    CU                22    PI
C   5   A5    GW    PG                23    CP
C   6   A6    RE    SC                24    NE
C   7   B1    CA    AS                25    SE
C   8   B2    BU    FD                26    K1
C   9   C3    ST    SK                27    K2
C  10   C4    OI    LO                28    K3
C  11   C5    IN    AS                29    GP
C  12   C6    PO                      30    AL
C  13   C7    LT    CS                31    PT
C  14   F3    PI
C  15   F4
C  16   F5
C  17   F6
C  18   F7
      NCMND=0
      II=0
      IF(SENSW(5))WRITE(6,1000) ID1,ID2
      DO 400 I=1,12
      IF(IBUFX(I) .EQ. -29440) GO TO 401
      IF(IBUFX(I) .EQ. -21363) GO TO 402
  400 CONTINUE
  403 CALL CONRAD(IBUFX,12)
      DO 1 I=1,14
      IF(ID1 .EQ. IW(I)) GO TO 2
    1 CONTINUE
   15 PAUSE 1
   10 CALL CARD
      RETURN
C      REPLACE CARRIAGE RETURN-BLANK BY BLANK-BLANK
  401 IBUFX(I)=-24416
      J    .403
C      REPLACT CCMMA-CARRIAGE RETURN BY CCMMA-BLANK
  402 IBUFX(I)=-21344
      J    .403
    2 GO TO (3,4,5,6,7,8,8,25,70,27,28,50,90,87) , I
C      CASTING OR REFINING COMMAND
    8 READ(6,100) NFRNC
      IF(NFRNC .GT. 6) GO TO 15
    8 IF(NFRNC .LT. 4) GO TO 15
      IF( I .EQ. 6) GO TO 12
C      CASTING COMMAND
C      EVENT 36. START CASTING  FURNACE 4 OR 5 ON ANODE
C      WHEEL  6
      L=72
      IF(NFRNC .EQ. 6) L=73
      NW=74-L
      NF=NFRNC-3
      IF(IAFFL(NF) .NE. 3) GO TO 36
      IF(IAWML(NW) .EQ. 1) GO TO 37
      IAFFL(NF)=4
      CALL CLOCK(IATIME(NF))
      IAWML(NW)=1
      CALL LIGHT(L,1)
      CALL LIGHT(NFRNC+10,0)
      ATRIB(2)=37.
      ATRIB(3)=NFRNC
      IF(NFRNC-5) 1001,1006,40
 1001 A=TCST4A
      B=TCST4B
      J    .1007
   40 A=TCST6A
      B=TCST6B
      J    .1007
 1006 A=TCST5A
      B=TCST5B
 1007 CALL ANTIME(NFRNC,A,B)
 1002 CALL FILEM(1,NSET)
      J    .10
   36 TYPE 360,NFRNC
      J    .10
   37 TYPE 361,NW
      J    .10
C      REFINING COMMAND
C      EVENT 34. START ANODE FURNACE 4 OR 5 TO REFINE
   12 NF=NFRNC-3
      IF(IAFFL(NF) .NE. 6) GO TO 38
      IAFFL(NF)=2
      CALL CLOCK(IATIME(NF))
      CALL LIGHT(NFRNC+7,0)
      CALL LIGHT(NFRNC+4,0)
      ATRIB(2)=35.
      ATRIB(3)=ILADA(NF)
      ATRIB(4)=NFRNC
      IF(NFRNC-5) 1005,1003,1004
 1004 A=TAF6A
      B=TAF6B
      J    .1007
 1005 A=TAF4A
      B=TAF4B
      J    .1007
 1003 A=TAF5A
      B=TAF5B
      J    .1007
   38 TYPE 362,NFRNC
      J    .10
C      DOWT COMMAND
    6 READ(6,101) NCR,POS,TW,IPR
   20 NXEVT=0
      IF(IPR .EQ. 0) IPR=1
      IF(TW .LT. .1) TW=.1
      KODE=0
      CALL QWAIT(NSET)
      J    .10
C      DUMP
   25 ICMND=12
      ISEQ=2
      READ(6,103) NCR,TW,IPR
      I=21
      J    .53
C      PICK COMMAND
    3 ISEQ=1
      J    .24
C      DUMP COMMAND
    4 ISEQ=2
   24 DO 13 I=1,13
      IF(ID2 .EQ. JW(I)) GO TO 14
   13 CONTINUE
      J    .15
   14 ICMND=1
      IF(ISEQ .EQ. 3) GO TO 91
   23 READ(6,102) NCR,IPOS,B,TW,IPR
      CALL LOCFND(IPOS,I,31,IER)
      IF(IER .EQ. 1) GO TO 15
   21 IXX=1
   53 POS=LOCDEV(I)
   17 CALL CHECK(IER)
      IF( IER .EQ. 1) GO TO 10
      IF(KAPUT(NCR).EQ.1) GO TO 20
      NCOM=B+.1
      IF(ICMND .EQ. 5) GO TO 50
      IF(NCOM .LT. 1) NCOM=1
      B=0.
   60 DO 59 I=1,NCOM
      CALL CMNDG(K,ICMNC,ISEQ,IXX,NSET,NCMND)
   59 CONTINUE
      J    .10
   50 NCOM=1
      J    .60
C      GO AND WAIT AT SPECIFIC COMPONENT
    7 CALL LOCFND(ID2,I,31,IER)
      IF(IER .EQ. 1) GO TO 15
   19 READ(6,103) NCR,TW,IPR
      POS=LOCDEV(I)
      J    .20
C      COLLAR PULLING COMMAND
    9 ICMND=12
      ISEQ=1
      J    .23
C      START COPPER BLOW IN CONVERTER NFRNC
   20   TYPE 1009 ,NCR,TDOWN(NCR)
      J    .10
   27 DO 29 I=1,3
```

```
      IF(IO2 .EQ. LUBR(I)) GO TO 30
   29 CONTINUE
      J    .19
C      OILING COMMAND FOR CRANE  I
   30 ATRIB(2)=57.
   34 ATRIB(3)=I
      ATRIB(1)=TIME(TLAST)+.5
      J    .1002
   28 READ(6,104) IPCS,NCR
      CALL LOCFND(IPCS,I,31,IER)
      IF(IER .EQ. 1) GO TO 15
      IX=I
      J    .10
   50 I,EQ=1
      J    .24
   51 READ(6,105) NCR,IPOS,IPOS1,Q
      CALL LOCFND(IPCS,I1,31,IER)
      IF(IER .EQ. 1) GO TO 15
      CALL LOCFND(IPCS1,I2,31,IER)
      IF(IER .EQ. 1) GO TO 15
      NCOM=Q+.1
      IF(ICMND .EQ. 9) GO TO 61
      IF(NCOM .LT. 1) NCOM=1
      Q=0.
   62 DO 63 NCOMX=1,NCOM
      TW=0.
      IPR=0
      POS=LOCDEV(I1)
      CALL CHECK(IER)
      IF(IER .EQ. 1) GO TO 10
      IF(KAPUT(NCR) .EQ. 1) GO TO 26
      IF(I1 .LT. 7 .AND. ICMND .EQ. 11) TW=10.
      CALL CMNDG(Q,ICMND,1,I1,NSET,MCMND)
      IF(     MCMND .EQ. 99) GO TO 10
      POS=LOCDEV(I2)
      MCMND=1
      CALL CMNDG(Q,ICMND,2,I2,NSET,MCMND)
      IF(     MCMND .EQ. 99) GO TO 10
      GO TO (52,52,52,52,63,55,52,55,63,63,55,15,52) , ICMND
   55 I=I4
      J    .54
   52 I=I3
   54 POS=LOCDEV(I1)
      MCMND=2
      CALL CMNDG(Q,I,2,I1,NSET,MCMND)
   63 CONTINUE
      J    .10
   61 NCOM=1
      J    .62
C      TURN LIGHT NCR ON OR OFF
   56 READ(6,104) NCR,IPOS
      IF(IPOS .GT. 1) GO TO 10
      IF( 1 .LT. IPOS) GO TO 10
      CALL LIGHT(NCR,IPOS)
      J    .10
C      ERROR RECOVERY FOR CRANES
   57 READ(6,106) NCR
      IF(NCR .GT. 3) GO TO 15
      IF( 3 .LT. NCR) GO TO 15
      IF(NCR .LT. 1) GO TO 15
      GO TO (153,154,155) ,NCR
  153 PTO=100.
      J    .157
  154 PTO=300.
      J    .157
  155 PTO=200.
  157 CONTINUE
      JW(NCR)=0
      KW(NCR)=0
      PAUSE 190
      CALL GO(TINCR,PTO,0,1)
      J    .10
C      MAINTENANCE WORK ON CONVERTERS
   58 READ(6,107) IPOS,TW
      CALL LOCFND(IPOS,I1,31,IER)
      IF(IER .EQ. 1) GO TO 15
      NCV=I1-8
      IF(NCV .LT. 0) GO TO 15
      IF(5 .LT. NCV) GO TO 15
      ATRIB(1)=TIME(TLAST)+.5
      ATRIB(2)=254+NCV
      ATRIB(3)=NCV
      IF(TW .LT. .5) TW=.5
      ATRIB(4)=TW
      J    .1002
  100 FORMAT(I2)
  101 FORMAT(I5,2F6.0,I5)
  102 FORMAT(I5,A2,1X,2F6.0,I5)
  103 FORMAT(I5,F6.0,I5)
  104 FORMAT(A2,I1)
  105 FORMAT(I5,A2,1X,A2,1X,F6.0)
  106 FORMAT(2I5)
  107 FORMAT(A2,1X,F6.0)
  140 FORMAT(5HANCDE,I2,17HNOT READY TO CAST/)
  141 FORMAT(5HWHEEL,I2,9HNOT READY/)
  142 FORMAT(5HANCDE,I2,12HWRONG STATUS/)
  300 FORMAT(1X,2A2)
  190 FORMAT(1X,3HCRN,I3,9HDOWN TILL,F8.2/)
      END
```

```
      SUBROUTINE CMNDG(Q,ICMND,ISEQ,IXX,NSET,MCMND)
      LOGICAL SENSW
      LOGICAL LEGVAL
      COMMON ID,IM,INIT,JEVNT,MFA,MSTOP,NX,NNC,NOQ,SCALE,ISEED,TNOW,
     1TREG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),ENQ(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTAT(8(5)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IAUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  IWTNGT(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *    ITROL(2),   JJ,   II,
     4IDLE(3),KAPLT(3),NNXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IANHL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRPO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPR(6,2),IOP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IALC*(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NPRNC,NRE
      EQUIVALENCE (NXEVT,NEXT)
      NCV=IXX-8
      KST=ISTAT(NCV)
      IF(Q .LE.0.) Q=1.
      IF(Q .GT. 2.) Q=2.
      IF( 2. .LT. Q) Q=2.
   52 IF(TW .LE. 0.) TW=FLOAT(IWTNGT(ICMND,ISEQ))/60.
      IF(IPR .EQ. 0) IPR=IPRA(ICMND,ISEQ)
      IF(ISEQ .EQ. 2) GC TO 99
      GO TO (1,2,3,4,5,14,7,16,9,6,11,12,13) , ICMND
   99 GO TO (21,22,23,24,25,26,27,28,29,30,21,32,33,41, ICMND
CCCCCC  PICK-UP COMMANDS
C        SLAG
    1 L=174
      JIO=9
   50 IF(SENSW(7)) GO TO 97
      CALL CONCHK(JIO,KST,IER)
      IF(IER .EC. 1) GO TO 200
   97 NEXT=IXX-8+L
      CALL COCK(NEXT,170,199,IER)
   92 IF(IER .EC. 1) GO TO 1000
      J    .91
C        MATTE
    2 NEXT=IXX+39
      IF(ITROL(IXX-6) .EQ. 1) GO TO 201
      IF(IFLOR(IXX-6) .EQ. 0) GO TO 200
  201 CALL COCK(NEXT,46,47,IER)
      J    .92
C        TRANSFER MATTE
    3 IF(IXX .EQ. 31) GO TO 34 .
      L=179
      JIO=11
      J    .50
C        COPPER
    4 TW=TW*Q
      IF(IXX .EC. 31) GO TO 34
      L=184
      JIO=10
      J    .50
C        COPPER OR TW FRCM PROTOTYPE
   34 NEXT=9
      IF(IPTCU .NE. 1) GO TO 20C
      J    .91
C        COPPER PIG CR CCPPER CAKE
    5 KODE=Q+10.01
C        PIG BAY
      NEXT=23
C        ANODE FURNACE CCLLAR
      IF(IXX .LT. 7 .AND. IXX .GT. 3) GO TO 21C
C        PIG PILE
      IF(IXX .EQ. 19) GO TC 202
C        PILOT PLANT
      IF(IXX .EC. 22) GO TO 203
C        PROTOTYPE
      IF(IXX .EQ. 31) GC TO 204
      TW=TW*Q
      IF( NCOLP .LT. 1) GO TO 200
      J    .91
  202 IF( NPPP  .LT. 1) GO TO 200
      NEXT=26
      J    .91
  203 IF( NCTOT .LT. 1) GO TO 200
      NEXT=3
      J    .91
  210 NEXT=IXX+23
      IF(IXX .EQ. 6) NEXT=22
      IF(NACOL(IXX-3) .LT. 1) GO TO 200
      J    .91
  204 IF( NCPTT .LT. 1) GO TO 200
      NEXT=9
      J    .91
C        LADLE OR
C        RETURN EMPTY SCRAP BOAT
    6 NEXT=0
      J    .91
C        FLUEDUST OR SCRAP
   14 NEXT=12
      J    .91
C        REACTOR SLAG
    7 NEXT=6
      IF(IRESL .LT. 1) GO TO 200
      J    .91
C        SKULL
    9 NEXT=17
      IF( NSKL  .LT. 1) GO TO 200
      J    .91
C        ANODE SLAG
   11 NEXT=IXX+10
      CALL COCK(NEXT,14,16,IER)
      J    .92
C        COLLAR BEING PULLED
```

```
C     PICK UP COLLAR PULLER
   12 TW1=TW
      IPR1=IPR
      POS=LOCDEV(11)
      TW=FLOAT(INTNGT(11,1))/60.
      IPR=IPRA(11,1)
      NEXT=0
      CALL GWAIT(NSET)
C     MOVE TO CONVERTER AND EXECUTE 275 TO 279
      IF(IXX .LT. 7) GO TO 120
      NEXT=IXX+266
      CALL COCK(NEXT,275,279,IER)
      IF(IER .EQ. 1) GO TO 1000
  121 POS=LOCDEV(IXX)
      TW=0.2
      CALL GWAIT(NSET)
C     PULL COLLAR
      TW=TW1
      IPR=IPR1
      IF(IXX .LT. 7) GO TO 122
      NEXT=IXX+271
      CALL COCK(NEXT,280,284,IER)
      IF(IER .EQ. 1) GO TO 1000
  123 CALL GWAIT(NSET)
C     RETURN COLLAR PULLER
      POS=LOCDEV(11)
      TW=FLOAT(INTNGT(11,2))/60.
      IPR=IPRA(11,2)
      NEXT=0
      J     .91
  120 NEXT=IXX+16
      CALL COCK(NEXT,19,20,IER)
      J     .121
  122 NEXT=IXX+16
      J     ,123
C     CONVERTER SLAG
C     EVENTS 195 TO 199 TO REMOVE CS FROM FLOOR
C     EVENTS 170 TO 174 TO REMOVE CS FROM CONVERTER, AUTO MODE ONLY
   13 L=194
      IF(INGS(IXX-8) .LT. 1) GO TO 133
  134 JIO=12
      J     .50
  133 IF(.NOT. SENSW(7)) GO TO 134
      L=169
      J     .136
  136 TYPE 135, INGS
  135 FORMAT(1X,5HNO CS,5I4/)
      J     .205
   11 L=76
      JIO=8
   51 IF(SENSW(7)) GO TO 80
      CALL CONCHK(JIO,KST,IER)
      IF(IER .EQ. 1) GO TO 200
   80 NEXT=IXX-84
      CALL COCK(NEXT,60,119,IER)
      J     .92
CCCCC     DUMP COMMANDS
C     SLAG, CODES 49 TO 51
   21 NEXT=48+IXX
   72 CALL COCK(NEXT,49,51,IER)
      J     .92
C     MATTE
   22 L=89
      IF(IXX .LT. 9) GO TO 71
      JIO=1
      J     .51
   71 NEXT=IXX+43
      J     .72
C     TRANSFER MATTE
   23 L=96
      JIO=2
      J     .51
C     COPPER
   24 NEXT=IXX+26
      IF(IXX .NE. 20) GO TO 96
      NEXT=21
      IF(NPPB .GT. 26)  GO TO 200
      IF( 26 .LT. NPPB) GO TO 200
      J     .92
   96 IF(IAPPL(IXX-1) .NE. 0) GO TO 200
      CALL COCK(NEXT,30,32,IER)
      J     .92
C     COPPER PIGS OR COPPER CAKES
   25 L=69
      IF(IXX .NE. 19) GO TO 81
      NEXT=24
      J     .91
   81 JIO=3
      J     .51
C     SCRAP
   26 L=59 .
      IF(IXX .LT. 9) GO TO  6
      IF(IXX .GT. 13) GO TO 6
      IF( 17 .LT. IXX) GO TO 6
      JIO=4
      J     .51
C     REACTOR SLAG
   27 L=66
      JIO=5
      J     .51
C     FLUE DUST
   28 L=106
      JIO=6
      J     .51
C     SKULL
   29 L=109
      JIO=7
      J     .51
```

```
C     LADLE
C     PILOT PLANT
   30 IF(IXX .EQ. 22) GO TO 95
C     PROTOTYPE PLANT
      IF(IXX .EQ. 31) GO TO 35
C     DUMPING BLOCK
      IF(IXX .EQ. 21) GO TO 94
C     MATTE TUNNEL 1
      IF(IXX .EQ. 7)  GO TO 93
C     MATTE TUNNEL 2
      IF(IXX .EQ. 8) GO TO 90
      NEXT=IXX+181
      CALL COCK(NEXT,190,194,IER)
      J     .92
   95 NEXT=7
      J     .91
   94 NEXT=13
      J     .91
   93 NEXT=42
      J     .91
   90 NEXT=43
      J     .91
C     PROTOTYPE PLANT
   35 NEXT=10
      J     .91
  200 TYPE 1001
      I=NCR+1
      IF(NCMND-1) 205,206,207
  207 CALL RMOVE(MLE(I),I,NSET)
  206 CALL RMOVE(MLP(I),I,NSET)
  205 NCMND=99
      J     ..1000
C     DUMPING
   32 NEXT=11
   91 CALL GWAIT(NSET)
 1000 CONTINUE
      TW=0.
   92 CONTINUE
      RETURN
 1001 FORMAT(12HILLEGAL CMND/)
      END
```

```
      SUBROUTINE CONCHK(JIO,ISTAT,IER)
      DIMENSION I(12,4)
      DATA I(1,1),I(2,1),I(3,1),I(4,1),I(5,1),I(6,1),I(7,1),I(8,1),
     1I(9,1),I(10,1),I(11,1),I(12,1),
     2     I(1,2),I(2,2),I(3,2),I(4,2),I(5,2),I(6,2),I(7,2),I(8,2),
     3I(9,2),I(10,2),I(11,2),I(12,2),
     4     I(1,3),I(2,3),I(3,3),I(4,3),I(5,3),I(6,3),I(7,3),I(8,3),
     5I(9,3),I(10,3),I(11,3),I(12,3)
     6 /6,8,2,1,1,7,1,1,3,5,4,4,9,14,2,6,6,7,8,7,4,5,4,4,14,10,
     711,1,1,7,6,6,14,5,6,14/
      DATA I(1,4),I(2,4),I(3,4),I(4,4),I(5,4),I(6,4),I(7,4),I(8,4),
     8 I(9,4),I(10,4),I(11,4),I(12,4)
     9 /14,9,11,6,6,7,6,6,15,5,9,15/
      IER=0
      DO 1 J=1,4
      IF(I(JIO,J) .EQ. ISTAT) GO TO 2
    1 CONTINUE
      IER=1
    2 RETURN
      END
```

FIFTH SUBMODULE

```
 TCH FOR FIFTH SUBMODULE
 SUBROUTINE DISK
 THIS IS THE FIFTH SUBMODULE.   ITS FILE NAME IS UPSSCI
 IT PROVIDES INFORMATION ON VARIOUS MODEL COMPONENTS
 COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
   TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
   TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
 HMAX,SETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
 1,TA(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
 (3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
 (3),TOOWN(3),TDL(3),TOH(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
  TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
 ,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
 COMMON /SAVE/ NSET(6,60),MODSCT(13)
 COMMON /ARG/ IX,  IO1X,  IO2X,  ISET,  JSEQ(12),  JCMND(12),
   NCRAN(12),  ICO(5),  MODUL
 COMMON /BLOK4/ NCR,  POS,  IPR,  TH,  KODE,  NKEVT,  IS(10),
 LOGVAL,NFRNC,NRE
 CALL CLOCK(ITT)
 IF(IX .NE. 30) GO TO 33
 WRITE(6,2000) ITT
 DO 35 J=1,31
 I=J
 IF(I.EQ.2) GO TO 35
 IF(I.EQ.3) GO TO 35
 IF(I.EQ.8) GO TO 35
 IF(I.EQ.27.OR.I.EQ.28) GO TO 35
 N=(I-1)/2
 N=(I-2)/6
 IF(N .EQ. 2) GO TO 35
 IF(N .EQ. 2) GO TO 35
 IDIV=6
 IF(NCR .EQ. 9) IDIV=1
 CALL STATS(I,NSET,IDIV)
 CONTINUE
 ITT
 RETURN
 IDIV=1
 IF(IX.LE.3) IDIV=6
 IF(IX .EQ. 9) IDIV=6
 IF(25.LT.IX) IDIV=6
 IF(NCR .EQ. 9) IDIV=1
 CALL STATS(IX,NSET,IDIV)
 J    .10
 FORMAT(10X,29HINTERMEDIATE RESULTS AT TIME=,I5/)
 END
```

```
 SUBROUTINE STATS(I,NSET,IDIV)
 DIMENSION ANT(4)
 DIMENSION IAF(10),IRF(4)
 DIMENSION NSET(6,1)
 COMMON IO,IP,INIT,JEVNT,MFA,NSTOP,NX,NXC,NCQ,SCALE,ISEED,TNOW,
 TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
 COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
 ,TATRIB(5)
 COMMON/BLOK1/PO(3),WT(3),TP(3),SP(3),POD(3),WTD(3),IPO(3),NC(3),N
 1,  JWT(3),  KWT(3),  TCRAN(3)
 COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFR(12),  IBUFV(12),
   NRAN(3),  ISTRT(3),  LOCDEV(3),  INTNGT(17,2),  IPRA(17,2),
   NC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
   NTANC,  ICNTRW, NTER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
   ITROL(2),  JJ,  II,
 ,(3),KAPUT(3),NBRT(3),LSTR(3),NSLAG(3)
 IAFFL(3),IAWNL(2),NEVC(5),IADO(3,4),ICV(5,9)
 ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
 IMS(5),IRFO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
 IATIME(3),NTPPB,NCPT,IRESL,TPTCU,IPR(6,2),TDP(6,2)
 NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
 NPIS,BI,PTREQ(5),NTAV(2),IREP(5,6)
 COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
   TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
   TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
 HMAX,SETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
 ,TA(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
 3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
 3),TOOWN(3),TDL(3),TOH(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
 TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
 TNOW=TIME(TLAST)
 CALL CLOCK(ITT)
 GO TO (1,1,1,2,2,2,3,3,4,4,4,4,4,99,99,99,99,99,6,7,8,9,99,99,99,1
 0,10,11,99,31) , I
 BLISTER FURNACE
 WRITE(IDIV,100)
 WRITE (IDIV,101) RMLEV
 WRITE (IDIV,102) YMMAX
 WRITE (IDIV,103) YMMIN
 WRITE (IDIV,104) TOLA
 WRITE (IDIV,105) SLEV
```

```
       WRITE (IDIV,102) SMAX
       WRITE (IDIV,103) SMIN
       WRITE (IDIV,104) TOTSL
       WRITE (IDIV,107) TSLAG
       J    .99
C**** ANODE FURNACE
C******
C        IAFFL =-1   WHEN FURNACE IS NOT IN USE
C        IAFFL = 0   WHEN WAITING FOR COPPER
C        IAFFL = 1   WHEN FULL BEFORE SLAG SKIMMING
C        IAFFL = 2   WHEN REFINING
C        IAFFL = 3   WHEN FINISHED REFINING
C        IAFFL = 4   WHEN CASTING
C        IAFFL = 5   WHEN FINISHED CASTING
C        IAFFL = 6   WHEN FULL AFTER SLAG-SKIMMING
C        IAFFL = 7   WHEN COLLAR IS BEING PULLED
C        IAFFL = 8   WHEN COLLAR HAS BEEN PULLED
C
C        IAWNL(1) IS ASSOCIATED WITH FURNACE 6 AND ANODE WHEEL
C        IAWNL(2) IS ASSOCIATED WITH FURNACE 4 OR 5 AND ANODE WHEEL 4
C
C        IAWNL = 0   WHEN NOT CASTING
C        IAWNL = 1   WHEN CASTING
C
    2 WRITE(IDIV,203) NANCST,NTANC
      DO 22 N=1,3
   22 ANT(N)=.01*FLOAT(ICAPA(N))
      WRITE(IDIV,203)(ANT(N),N=1,3)
      ANT(4)=0.
      DO 20 N=1,3
      ANT(N)=.01*FLOAT(ILADA(N))
   20 ANT(4)=ANT(4)+ANT(N)
      WRITE(IDIV,201) ANT
      ANT(4)=0.
      DO 21 N=1,3
      ANT(N)=.01*FLOAT(NTLAD(N))
   21 ANT(4)=ANT(4)+ANT(N)
      WRITE(IDIV,202) ANT,IAFFL,IATIME,IAWNL(1),IAWNL(2),IAWNL
      J    .99
C**** MATTE TUNNEL
    3 WRITE(IDIV,300)
      WRITE (IDIV,301) IFLOR
      WRITE (IDIV,302) ITROL
      WRITE (IDIV,303) TOTW
      J    .99
C**** CONVERTERS
    4 N=I-6
C WRITE CONVERTER INPUT/OUTPUT STATISTICS
      WRITE(IDIV,410) ITT
      WRITE(IDIV,420)
      WRITE(IDIV,430)
      DO 40 NCO=1,5
      DO 41 JP=1,10
      IAF(JP)=FLOAT(IAFO(NCO,JP))*.01
   41 CONTINUE
      DO 42 JP=1,4
      IRF(JP)=FLOAT(IRFO(NCO,JP))*.01
   42 CONTINUE
      NIO=NCO+2
      WRITE(IDIV,440)NIO,(IAF(J),J=1,10),(IRF(J),J=1,4),ISTAT(NCO),NCV
     *NCO)
   40 CONTINUE
C CONVERTER TIMES
      WRITE(IDIV,480)TNOW
      WRITE(IDIV,481)
      WRITE(IDIV,482)
      WRITE(IDIV,483)
      WRITE(IDIV,484)
      DO 45 JC=1,5
C CALCULATE BLOWING TIME PER LADLE OF REVERB MATTE
      SRM=IAFO(JC,1)+IAFO(JC,2)
      NT(JC,17)=0
      IF(SRM.NE.0) GO TO 47
      J    .49
   47 NT(JC,17)=100.*FLOAT(NT(JC,5))/FLOAT(SRM)
   49 WRITE(IDIV,485)(NT(JC,JN),JN=1,17)
   45 CONTINUE
      WRITE(IDIV,450)
      WRITE(IDIV,460)
      DO 43 NCO=1,5
      NIO=NCO+2
      WRITE(IDIV,470)NIO,CV(NCO,2),CV(NCO,3),CV(NCO,4),CV(NCO,
   43 CONTINUE
      WRITE(IDIV,471)
      J    .99
C**** PIG PILE
    6 WRITE(IDIV,690) NPPP
      J    .99
C**** PIG BAY
    7 WRITE(IDIV,700) NPPB,NCOLP,NTPPB
      J    .99
C**** DUMPING BLOCK
    8 WRITE(IDIV,800) NCOOL,NSKL,NAPL
      J    .99
C**** PILOT PLANT
    9 WRITE(IDIV,900) NCTOT,NCPPL,TPISL,ANS
      J    .99
C**** CRANES
   10 N=I-25
      WRITE(IDIV,1000)
      DO 1010 J=1,3
      DEL=100.*SP(J)/TNOW
      PRACT=100.*TCRAN(J)/TNOW
 1010 WRITE(IDIV,1001) J,PO(J),POD(J),WT(J),WTD(J),TP(J),IPO(J),TOOWN(J)
     *,NBRT(J),LSTR(J),NRAN(J),PRACT,DEL
      WRITE(IDIV,1002)
      DO1011 J=1,3
      IFIL=NFE(J+1)
 1011 WRITE(IDIV,1003) J,(NSET(I,IF(L)),I=1,4)
      WRITE(IDIV,1004) KWT,JWT
      J   .99
C**** GASP VARIABLES
   11 WRITE (IOJV,1100) ATRIB,ISTRIB
      K=MFE(1)
      WRITE(IDIV,1102) (NSET(J,K), J=1,4)
      I5=ISTRIB(5)
      I1=ISTRIB(1)
      WRITE(IDIV,1101) TNOW,JEVNT,I1,I5
      DO 1105 J=1,ID
      JK=ID+1-J
      IF(NSET(I,JK) .NE. 0) GO TO 1106
 1105 CONTINUE
 1106 WRITE(IDIV,1110)
      DO 1107 K=1,JK
 1107 WRITE(IDIV,1111)K,(NSET(J,K) ,J=1,4)
      WRITE(IDIV,1112)
      WRITE(IDIV,1113) MFE
      J   .99
C**** PROTOTYPE PLANT
   31 WRITE(IDIV,3100) NCPTT,NCPT,IPTCU,LCPU
      J   .99
C
C
   98 IF(IDIV .EQ. 6) GO TO 99
      WRITE(1,2001)
 99   RETURN
 100  FORMAT(/23X,13HREV. FURNACES,//20X,2HR1,8X,2HR2,8X,2HR3/)
 101  FORMAT(1X,11HMATTE LEVEL,6X,3(F6.2,4X))
 102  FORMAT(1X,10HMAX. LEVEL,7X,3(F6.2,4X))
 103  FORMAT(1X,10HMIN. LEVEL,7X,3(F6.2,4X))
 104  FORMAT(1X,12HMATTE TAPPED,5X,3(F6.2,4X))
 105  FORMAT(/1X,10HSLAG LEVEL,7X,3(F6.2,4X))
 106  FORMAT(1X,11HSLAG TAPPED,6X,3(F6:2,4X))
 107  FORMAT(1X,13HSLAG RETURNED,4X,3(F6.2,4X))
  200 FORMAT(15X,16HANODE STATISTICS//,22X,2HA4,4X,2HA5,4X,2HA6,3X,
     * 5HTOTAL//,1X,11HANODES CAST,8X,4I6)
  201 FORMAT(1X,17HCOPPER IN FURNACE,2X,3F6.2,F6.1)
  202 FORMAT(1X,16HTOTAL LDLS ADDED,3X,3F6.2,F6.1/,1X,14HFURNACE STATUS,
     * 5X,3(5X,I1)/,1X,20HTIME OF STATUS START,I5,2I6/,1X,18HANODE WHEEL
     2 STATUS,6X,I1,2(5X,I1))
  203 FORMAT(1X,16HFURNACE CAPACITY,3X,3F6.2/)
 300  FORMAT(/22X,13HMATTE FUNNELS,/21X,3HMT1,10X,3HMT2,/)
 301  FORMAT(1X,11HMT ON FLOOR,8X,2(I3,9X))
 302  FORMAT(1X,9HMT ON CAR,10X,2(I3,9X))
 303  FORMAT(1X,17HMIN MT IS WAITING,1X,2(F6.2,7X)/)
  410 FORMAT(/1X,44H** CONVERTER INPUT/OUTPUT STATISTICS AT TIME,I5//)
 420  FORMAT(15X,9HADDITIONS,25X,8HREMOVALS,7X,6HSTATUS/)
 430  FORMAT(1X,74HCV  DM  MR  TM  PG  SC  RS  FD  SK  CS  A/B  SL  CU
     S TM  CS    IST  BLOW/)
 440  FORMAT(I3,I6,9I4,I5,3I4,2I5)
 450  FORMAT(1X,24H** CONVERTER CONTENTS **/)
 460  FORMAT(1X,33HCV    CU2S    FES     SLAG     CU/)
```

```
 470  FORMAT(I3,4F8.1)
 471  FORMAT(1X,//)
  480 FORMAT(1X,25H**CONVERTER TIMES AT TIME,F6.1/)
 481  FORMAT(1X,79HSTAGE *   MATTE BLOW  *   CU  *   BRK *  PULL * EP AT
     1 *PIG*   MIN**/)
 482  FORMAT(1X,5HSTART,21X,6HBLOW,19X,16H3*  2* ADD   PER/)
 483  FORMAT(2X,80HN  CU   B   E   SK  SHB  E  SB   B SHB   B SHB   M  CU
     1 INT LCU LM*/)
 484  FORMAT(2X,80H1   2   3   4   5   6   7   8   9  10  11  12  13  14
     1  15  16*/)
 485  FORMAT(17I4)
 600  FORMAT(15X,8HPIG PILE//1X,11HNO OF PIGS=,I6/)
 700  FORMAT(15X,7HPIG BAY//1X,10HNO OF PIGS,I6/1X,9HCOOL PIGS,I6/,
     *1X,18HTOTL. PIGS POURED=,I7/)
 800  FORMAT(15X,13HBUMPING BLOCK//1X,19HCOOLING LADLES=,I4/1X,7HSKULLS=
     1,I4/,1X,18HTOTL SKULLS PROD.=,I6/)
 900  FORMAT(15X,11HPILOT PLANT//1X,12HCAKES READY=,I4/1X,18HTOTAL NO OF
     1 CAKES=,I4/,1X,19HLADLES OF SLAG RDY=,I4,24HTOTAL LDLS OF SLAG PRO
     1D=,I4)
 1000 FORMAT(/25X,6HCRANES//1X,3HNCR,1X,3HPOS,3X,3HPOD,2X,3HWTT,2X,,3HWT
     *O,2X,3HIPR,2X,3HIPO,2X,5HITDOWN,2X,4HNEXT,1X,4HLSTR,1X,9HNUMANK,3HP,
     2USE,1X,7HP.DELAY/)
 1001 FORMAT(1X,I2,2F6.0,2F5.2,2(I3,2X),F6.2,I4,1X,I4,I5,2X,2F6.2/)
 1002 FORMAT(/1X,45HNCR   ATRIB(1)   ATRIB(2)   ATRIB(3)   ATRIB(4)   ,/)
 1003 FORMAT(1X,I3,3X,4(I7,3X)/)
 1004 FORMAT(15X,15HSTATE VARIABLES/1X,12HWAITING(KWT),6X,3(I5,5X)/,1X,
     *18HINTERFERENCE(JWT) ,3(I5,5X)/)
 1100 FORMAT(20X,14HGASP VARIABLES//1X,12HATRIB(1-4) =,4F8.2/1X,12HISTRI
     1B(1-5)=,5I8)
 1101 FORMAT(1P,12HPRES. EVENT=,F8.2,3I8/)
 1102 FORMAT(1X,12HNEXT EVENT= ,4I8)
 1110  FORMAT(20X,14HEVENTS IN NSET/(1X,2HNO,6X,3HPOS,6X,3HTW ,2X,7HLINK/1
     1P0,2X,10HCODE/METEV//)
 1111 FORMAT(5X,7I8)
 1112 FORMAT(/1X,10HFILE NAMES,15X,20HTIME      CR 1     CR 2     CR 3/)
 1113 FORMAT(1X,20HSTART OF FILES(MFE) ,4I8/)
 2001 FORMAT(1X,17HNO INFO AVAILABLE/)
 3100 FORMAT(15X,15HPROTOTYPE PLANT//1X,9HCAKE RDY=,I4,18HTOTAL NO OF CA
     1KES=,I4,/1X,15HLDLS OF CU RDY=,I4,22HTOTAL LDLS OF CU PROD=,I4/)
      END
```

SIXTH SUBMODULE

```
  ECK FOR SIXTH SUBMODULE
  SUBROUTINE CISK
THIS IS THE SIXTH   SUBMODULE,ITS FILE NAME IS UPSAC1,
IT CONSISTS OF EVENTS 170 TO 249,CONVERTER REMOVAL EVENTS
    LOGICAL LCGVAL,SENSW
    DIMENSICN  IO(4,15),CV1(2)
    DIMENSICN CHIN(4)
    COMMON IO,IP,INIT,JEVNT,MFA,MSTOP,NX,NXC,NCQ,SCALE,ISEED,TNOW,
   1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
    COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
   *,ISTRIB(5)
    COMMON/BLOK1/PO(3),MT(3),IP(3),SP(3),POD(3),MTO(3),IPO(3),NC(3),NP
   1C(3),  JMT(3),  KMT(3),  TCRAN(3)
    COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
   1  NRANK(3),  ISTRT(3),   LOCDEV(31),  IMTNGT(17,2),  IPRA(17,2),
   2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
   3  NTANC,  ICNTRM,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
   *   ITROL(2),   JJ,   II,
   4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
   5,IAFFL(3),IAMHL(2),NEVC(5),IADD(11,4),ICV(5,9)
   6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
   7,IMGS(5),INFO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
   8 IATIME(3),MTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2)
   *,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
   *MOP(5,6),MTREQ(5),MTAV(2),IREP(5,6)
    COMMON /BLOK3/  TP1A,  TP1B,  TPISLA,  TPISLB,  TPCOOL,
   1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
   2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
   3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RRAT(3
   4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RAT(3),SLEV
   5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
   6(3),TOOWN(3),TOL(3),TOM(3),CRL(3),CRM(3),TCST6A,TCST6B,TAFR6A,
   7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
   8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
    COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  TS(10),
   * LOGVAL,NFRNG,NRE
    COMMON /SAVE/ NSET(6,60),MODSCT(13)
    COMMON /ARG/ IX,  IO1X,  IO2X,  ISET,  JSEQ(12),  JCNNO(12),
   1  NCRAN(12),  ICO(5),  MODUL
    DATA CHIN(1),CHIN(2),CHIN(3),CHIN(4)/.08,.01,.01,.1/
    DATA IO(1,1),IO(1,2),IO(1,3),IO(1,4),IO(1,5),IO(1,6),IO(1,7),IO(1,
   18),IO(1,9),IO(1,10),IO(1,11),IO(1,12),IO(1,13),IO(1,14),IO(1,15)/
   21,1,2,4,1,1,1,1,1,1,1,1,1,2,5/
    DATA IO(2,1),IO(2,2),IO(2,3),IO(2,4),IO(2,5),IO(2,6),IO(2,7),IO(2,
   18),IO(2,9),IO(2,10),IO(2,11),IO(2,12),IO(2,13),IO(2,14),IO(2,15)/
   21,1,1,1,3,1,1,1,1,1,1,1,1,1,1/
    DATA IO(3,1),IO(3,2),IO(3,3),IO(3,4),IO(3,5),IO(3,6),IO(3,7),IO(3,
   18),IO(3,9),IO(3,10),IO(3,11),IO(3,12),IO(3,13),IO(3,14),IO(3,15)/
   21,1,1,1,1,2,1,1,2,1,1,1,1,1,1/
    DATA IO(4,1),IO(4,2),IO(4,3),IO(4,4),IO(4,5),IO(4,6),IO(4,7),IO(4,
   18),IO(4,9),IO(4,10),IO(4,11),IO(4,12),IO(4,13),IO(4,14),IO(4,15)/
   21,1,1,4,1,1,1,1,1,1,1,1,1,2,4/
    IDIV=1
    III=0
    IOI=0
    TTNOW=TIME(TLAST)
    IF(.NOT. SENSW(4)) WRITE(6,2000) IX,TTNOW
    I=IX-169
    GO TO(170,170,170,170,170,175,175,175,175,175,180,180,180,180,
   1180,185,185,185,185,185,190,190,190,190,190,195,195,195,195,195,
   2200,200,200,200,200,205,205,205,205,205,210,210,210,210,210,215,
   3215,215,215,215,220,220,220,220,220,225,225,225,225,225,230,230,
   4230,230,230,235,235,235,235,235,240,240,240,240,240,245,245,245,
   5245,245),I
101 PAUSE 5
    J   .900
 20 CALL FILEM(1,NSET)
900 CONTINUE
C  ATT CONVERTER REMOVAL IFFORMATION
    INFO(NCV,JIO)=INFO(NCV,JIO)+ISAVE(NCR)
990 IF(SENSW(3)) GO TO 1404
    WRITE(6,3001) ATRIB
 02 CONTINUE
    III=ISTAT(NCV)
    IOI=IO(JIO,III)
    WRITE(6,3002)NCV,III,IOI,JIO,JAD,NEN,NN,ISAVE(NCR),NCR
 02 FORMAT(1X,9I4/)
    WRITE(6,3003) OUT1,STARTA,OUT2,STARTB,OUTD,STARTD,CNEN,CV1(1),
   1CV1(2)
 03 FORMAT(9F8.3)
 04 CONTINUE
 05 IF(IX.EQ.9) GO TO 1405
 01 IX=0
 01 IF(.NOT.SENSW(7)) ISET=0
    RETURN
    REMOVE HIGH CU SLAG  FROM A CONVERTER,AUTO ONLY
    NCV=IX-169
    CALL LADLE
    JAD=9
    JIO=6
    J   .3999
    REMOVE REGULAR SLAG
    NCV=IX-174
    JIO=1
    JAD=10
    J   .3999
    REMOVE TRANSFER MATTE
    NCV=IX-179
    JIO=3
    JAD=9
    J   .3999
    REMOVE BLISTER COPPER
    NCV=IX-184
    JAD=11
    JIO=7
    J   .3999
  AN EMPTY SLAG LADLE TO C(IX-189)
    NCV=IX-189
    CALL LIGHT(61+NCV,0)
    J   .1401
  CL FROM FLCOR,MAN OR AUTO
    NCV=IX-194
         CALL LADLE
         ISAVE(NCR)=IMGS(NCV)
         IMGS(NCV)=0
         J    .900
C START BLOWING AFTER GM BLOW SLAG SKIM
C RESET REF TO OPERATE CONDITION
 200   NCV=ISTRIB(2)
       CALL ONFRL(ICV(NCV,9),.TRUE.,IER)
C IA,IB TO OPERATE
       JL=ICV(NCV,7)
       IL=ICV(NCV,6)
       MT(NCV,3)=TTNOW
       MT(NCV,6)=MT(NCV,6)+MT(NCV,3)-MT(NCV,4)
       WRITE(6,999) ICV(NCV,7),ICV(NCV,6),NCV,JL,IL,ICV(NCV,9)
 999   FORMAT(1X,6I8)
       CALL LIGHT(JL,1)
       CALL LIGHT(IL,1)
       J    .1400
C SPARE EVENT CODES
  205 CONTINUE
  210 CONTINUE
  215 CONTINUE
  220 CONTINUE
  225 CONTINUE
  230 CONTINUE
  235 CONTINUE
  240 CONTINUE
  245 CONTINUE
       J    .900
C  TURN ON EMPTY LADLE LIGHT
 3999  CALL LIGHT(61+NCV,1)
       CALL LADLE
C CHECK IF REMOVAL LEGAL OR NOT
 4000 I=ISTAT(NCV)
      IJ=IC(JIC,I)
      GO TO(4100,4200,4300,4400,4500),IJ
C      TYPE ILLEGAL REMOVAL REPORT
 4100 WRITE(IDIV,4110)
 4110 FORMAT(1X,15HILLEGAL REMOVAL/)
      ISAVE(NCR)=0
      J    .990
C      MATERIAL REMOVAL AT END OF A SLAG BLOW
C CHECK IF A MATTE TRANSFER REMOVAL
 4200  IF(JIO.NE.3) GO TO 4201
      NEN=4
      NN=1
C *****TRANSFER MATTE ********
      CALL ORBACK(CV1,ICV(NCV,1),2,IER)
      QF=CV(NCV,3)
      QC=CV(NCV,2)
      QS=CV1(2)*TPU(NCV)
      QT=QF+QC+QS
      QL=IADD(JIO,1)
      IF(QT.LE.0.)QT=.01
      QCR=QC/QT
      QFR=QF/QT
      QSR=QS/QT
      IADD(JIO,2)=QCR*80.
      IADD(JIO,3)=QFR*43.
      IADD(JIO,4)=QCR*20.+QFR*37.
      IF(QT.LT.QL)GO TC 4260
      NCRIT=2
      NEVC(NCV)=2
      J    .4265
 4260 NCRIT=1
      NEVC(NCV)=1
      QL=QT
 4265 OUT1=QFR*QL
      OUT2=QSR*QL
      OUTD=QCR*QL
      QT=QT-QL
      SA=CV1(1)
      CV(NCV,3)=QF-OUT1
      SB=(QS-OUT2)/TPU(NCV)
      CV(NCV,4)=SB-OUT2
      SD=(QC-OUTD)/TPU(NCV)
      CV(NCV,2)=QC-OUTD
      CV(NCV,5)=0.
      CNEN=CHIN(NEN)*TPU(NCV)
      IF(I.LT.NCRIT) GO TO 6060
      IF(QT.LT.CNEN) GO TO 4290
      J    .6060
 4290 NNCV=NCV+2
      FOL=QT/FLOAT(IADD(JIO,1))
      WRITE(IDIV,4295) FOL,NNCV
 4295 FORMAT(1X,F6.1,I3,19HLADS TM LEFT IN C ,I7/)
      J    .6060
 4201 NEN=1
 4202 OUT1=1.57*FLOAT(IADD(JAD,1)*IADD(JAD,3))*.01
      OUT2=FLOAT(IADD(JAD,1)*(100-IADD(JAD,3)-IADD(JAD,2)-IADD(JAD,4)
     1))*.01
      OUTD=1.25*FLOAT(IADD(JAD,1)*IADD(JAD,2))*.01
      J    .6300
C      MATERIAL REMOVAL AT END OF A COPPER BLOW
 4300 OUT1=0.0
      OUT2=FLOAT(IADD(JAD,2)*IADD(JAD,1))*.01
      OUTD=FLOAT(IADD(JAD,1)*(100-IADD(JAD,2)))*.01
      NEN=3
      J    .6000
C      MATERIAL REMOVAL AT END OF A SLAG STAGE
 4400 NN=2
      NEN=2
      J    .4202
C MATERIAL REMOVAL DURING DOING HIGH BLOW
 4500 NN=1
      NEN=1
C PUT IA,IB TO HELD
      IL=ICV(NCV,6)
      CALL LIGHT(IL,0)
      J    .6202
C     RESET ICS FOLLOWING MATERIAL REMOVAL
6000  CALL ORBADR(CVI,ICV(NCV,3),2,IER)
      SA=CVI(1)-OUT1/TPU(NCV)
      SB=CVI(2)-OUT2/TPU(NCV)
      GO TO (6002,6002,6004,6002), NEN
C     SLAG STAGE
6002  SD=(CV(NCV,2)-CUTD)/TPU(NCV)
      NN=1
      IF(NEN.NE.4) GO TO 6006
      ISTAT(NCV)=9
      MT(NCV,6)=FLOAT(MT(NCV,6)-MT(NCV,4))+TTNOW
      MT(NCV,4)=TTNOW
C ESTABLISH BLOW NO. FOR SOURCE OF TM
      NTM=SD*TPU(NCV)/FLOAT(IADD(3,1))
      IF(NTM.GE.1) GO TO 6003
      NEVC(NCV)=1
      J    .6006
6003  NEVC(NCV)=2
      J    .6006
C     FINISH STAGE
6004  SD=(CV(NCV,4)-CUTD)/TPU(NCV)
      NN=2
C     CHECK IF ALL MATERIAL REMOVED
C     CMIN(NEN) IS LIMITING VALVE FOR CRITICAL MATERIAL
6006  GO TO(6015,6010,6015,6020),NEN
6010  CRIT=SB
      CRIT1=SA
      CRIT2=SD
      J    .6030
6015  CRIT=SB
      CRIT1=SA
      CRIT2=SD
      J    .6030
6020  CRIT=SD
      CRIT1=SA
      CRIT2=SB
6030  CNEW=CMIN(NEN)
      IF(CRIT.GT.CNEW) GO TO 6040
C     LIMIT REACHED ON CRITICAL MATERIAL
      CALL LIGHT(66+NCV,0)
      NCRIT=1
      IF(CRIT.GE.C.) GO TO 6041
      GO TO(6031,6031,6032,6033),NEN
6031  OUT2=CV(NCV,4)
      SB=0.
      J    .6041
6032  OUT2=CV(NCV,5)
      SB=0.
      J    .6041
6033  CUTD=CV(NCV,2)
      SD=0.
6043  NCRIT=2
      GO TO(6052,6052,6054,6056),NEN
6052  OUTD=CV(NCV,2)
      SD=0.
      J    .6060
6054  CUTD=CV(NCV,4)
      SD=0.
      J    .6060
6056  OUT2=CVI(2)*TPU(NCV)
      SB=0.
6060  ISAVE(NCR)=(OUT1+OUT2+OUTD)/(FLOAT(IADD(JAD,1))*.01)
      IF(NEN.LT.4) GO TO 6069
      IAFO(NCV,10)=IAFO(NCV,10)-ISAVE(NCR)
6069  GO TO (6070,6079), NCRIT
6070  GO TO(6073,6074,6076,6078), NEN
C     START ANOTHER SLAG BLOW, ASK FOR MORE MATTE
6073  IF(14.LT.ISTAT(NCV)) GO TO 6666
      NEVC(NCV)=NEVC(NCV)+1
      ISTAT(NCV)=9
      IAFO(NCV,17)=0
6666  STARTA=SA
      STARTB=SB
      STARTD=SD
      NN=1
      IF(14.LT.ISTAT(NCV)) J    .6667
      CALL LIGHT(96+NCV,1)
      J    .6080
6667  ISTAT(NCV)=1
      J    .6080
C     END OF SLAG STAGE, START COPPER BLOW WITH FLUE DUST ADDITION
6074  ISTAT(NCV)=7
      STARTA=SD
      STARTB=0.0
      STARTD=SA
      NN=2
C DETERMINE NO.,FREQUENCY OF PIG ADDITIONS
      NPGS      =SD*TPU(NCV)/20.
      RCUS=.063*CV(NCV,6)*CV(NCV,7)
      MT(NCV,14)=SD*TPU(NCV)/RCUS
      MT(NCV,19)=MT(NCV,14)/NPGS
      IAFO(NCV,15)=0
C SWITCH TO COPPER STAGE RATE POTS
      CALL OWFRLI(ICV(NCV,9),.FALSE.,IER)
      CALL LIGHT(41+NCV,0)
      CALL LIGHT(66+NCV,1)
      CALL LIGHT(96+NCV,1)
      J    .6080
C     END OF FINISH STAGE, START SLAG BLOW WITH MATTE ADDITION
6076  ISTAT(NCV)=9
      STARTA=0.0
      STARTB=SD
      STARTD=SA
      NN=1
      MT(NCV,1)=TTNOW
      MT(NCV,6)=MT(NCV,6)+MT(NCV,1)-MT(NCV,7)
C SCHEDULE EVENT TO ZERO ARRAYS
      ATRIB(1)=TIME(TLAST)
      ATRIB(2)=295
      ATRIB(3)=NCV
      ATRIB(4)=2.
      IX=9
      CALL FILEM(1,NSET)
C *** MATTE ADDITION AT START OF NEW CV. CYCLE IN AUTOMATIC OPERATOI(
      ISET=8
C SWITCH TO SLAG STAGE RATE POTS
      CALL OWFRLI(ICV(NCV,5),.TRUE.,IER)
      CALL LIGHT(66+NCV,0)
      CALL LIGHT(41+NCV,1)
      CALL LIGHT(96+NCV,1)
C CALCULATE AVERAGE BLOW TIME/LADLE CU
      MT(NCV,16)=100*MT(NCV,8)/INFO(NCV,2)
      J    .6080
C     END OF SLAG BLOW, TRANSFER MATTE,REMOVED, ASK FOR MORE MATTE
6078  ISTAT(NCV)=9
      STARTA=SA
      STARTB=SB
      STARTD=SD
      MT(NCV,4)=TTNOW
      NN=1
C SCHEDULE EVENT TO ZERO ARRAYS
      ATRIB(1)=TIME(TLAST)
      ATRIB(2)=295
      ATRIB(3)=NCV
      ATRIB(4)=1.
      IX=9
       CALL FILEM(1,NSET)
      CALL LIGHT(96+NCV,1)
      J    .6080
C END OF SLAG BLOW,MORE SLAG OR TRANSFER MATTE TO BE REMOVED
6079  STARTA=SA
      STARTB=SB
      STARTD=SD
      IF(NEN .NE. 4) GO TO 6080
      NNCV=NCV+2
      TML=QT/QL
      WRITE(IDEV,6099)TML,NNCV
6099  FORMAT(2X,F6.2,25H LADLES OF TM REMAIN IN C,I2/)
C     RESET ICS
C     IA,IB, TO IC
6080  IL=ICV(NCV,7)
      CALL LIGHT(IL,0)
C     ENTER NEW ICS
      CALL ONJOAR(STARTA,ICV(NCV,1),IER)
      CALL ONJOAR(STARTD,ICV(NCV,2),IER)
      GO TO(6082,6084), NN
C     SLAG STAGE
C UPDATE CV ARRAY
6082  IF(1.LT.NEN) GO TO 900
      CV(NCV,2)=STARTD*TPU(NCV)
      CV(NCV,4)=STARTB*TPU(NCV)
      CV(NCV,5)=0.
      IF(14.LT.ISTAT(NCV)) GO TO 6083
      IF(1.LT.ISTAT(NCV)) GO TO 900
C SCHEDULE START OF GOING HIGH BLOW AFTER SKIMMING TIME
6083  ATRIB(1)=TTNOW+FLOAT(INTWST(1,1))/60.+3.5
      ATRIB(2)=300
      ATRIB(3)=FLOAT(NCV)+.5
      MT(NCV,5)=FLOAT(MT(NCV,5)-MT(NCV,3))+TTNOW
      MT(NCV,10)=FLOAT(MT(NCV,10)+MT(NCV,11))-TTNOW
      MT(NCV,4)=TTNOW
      J    .1000
C     FINISH STAGE
C UPDATE CV ARRAY
6084  CV(NCV,7)=STARTA*TPU(NCV)
      CV(NCV,3)=0.
      CV(NCV,4)=STARTD*TPU(NCV)
      CV(NCV,9)=STARTD*TPU(NCV)
C     RETURN TO 900 WITH IA, IB IN IC.CA REFERENCE ON PCSITIVE LOCK
6090  IF(ISTAT(NCV).NE. 7) GO TO 900
C     ALARM OPERATOR THAT CRANE HAS HIGH CU SLAG TO GO TO A CONVERTE(
C     REMOVE EVENT FROM CRANE FILE ASKING FOR SLAG TO REVERB
C     CHECK WHICH CONVERTER CAN TAKE HIGH CU SLAG
C     PERMISSIBLE STATES ISTAT ARE 1,6,10
3012  IF(SENSW(7)) GO TO 900
      JCV=0
      DO 3010  J=1,3
      IF(ISTAT( J).EQ.1) GO TO 3011
      IF(ISTAT( J).EQ.6) GO TO 3011
      IF(ISTAT( J).EQ.10) GO TO 3011
3010  CONTINUE
      J  . .3020
3011  JCV=J
      JJCV=JCV+2
3020  IF(JCV.NE.0) GO TO 3022
      NNCV=NCV+2
      TYPE 3023,NNCV
3023  FORMAT(1X,17HHIGH CU SLAG AT C,I4/)
      J    .3013
3022  TYPE 3024, JJCV
3024  FORMAT(1X,20HHIGH CU SLAG DUMPED IN C,I4/)
      NNEVT=T6+JCV
      L=13
C     *** SET UP POSITION, WAITING TIME AND PRIORITY
3031  POS=LOCREV(JCV+8)
      TW=FLOAT(INTWST(L,2))/60.
      IPR=IPRS(L,2)
C     *** FIND DUMP SLAG COMMAND
      DO 3000 I=1,3
      P=LOCREV(I)
```

SEVENTH SUBMODULE

```
      CALL FINDIF,NCR+1,1,KCOL,NSET)
      IF(KCOL .NE. 0) GO TO 3030
 3000 CONTINUE
C   *** COMMAND NOT FOUND.   FILE NEW COMMAND AT END OF CRANE FILE.
      CALL GNASTINSET)
      J    .500
C   *** ENTER POSITION OF CONVERTER
 3030 NSET(1,KCOL)=LOCDEV(JCV+8)*10
C   *** ENTER WAITING TIME
      NSET(2,KCOL)=100.*TW+.05
C   *** FIND OLD RANK
      NR=NSET(3,KCOL)/10
C   *** FIND OLD KODE
      KD=NSET(4,KCOL)/1000
C   *** ENTER NEW PRIORITY
      NSET(3,KCOL)=IPR+10*NR
C   *** ENTER NEW NEXT EVENT
      NSET(4,KCOL)=NXEVT+1000*KD
      J    .500
C****** DEPOSIT LADLE OF HIGH GRADE SLAG NEAR CONVERTER ICV
 3013 IHGS(NCV)=ISAVE(NCR)
      ISAVE(NCR)=0
      L=10
      NXEVT=0
      J    .3031
 3001 FORMAT(1X,5HATRIB,2X,4F8.2)
 2300 FORMAT(1X,5HEVENT,I4,7HAT TIME,F8.3)
  990 CONTINUE
      J    .500
      END
```

```
 DECK FOR SEVENTH MODULE
    DECK FOR SEVENTH SUBMODULE
      SUBROUTINE DISK
C     THIS IS   SUBMODULE SEVEN     . ITS FILE NAME IS UPS7CI
C IT CONSISTS OF EVENTS 250 TO 299,OTHER THAN I/O EVENTS
      LOGICAL LOGVAL,SENSW
      DIMENSION CV1(2) ,MUL(5,8),CMIN(4)
      COMMON IO,IW,INIT,JEVNT,NPA,NSTOP,NX,NNC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON/BLOK1/PO(3),WT(3),IP(3),SP(3),POB(3),WTO(3),IPO(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IOUPX(12),  IOUPY(12),
     1 NRANK(3),  ISTRT(3),   LOCDEV(33),  IWTWGT(17,2),  IPOA(17,2),
     2 NLC,  NFL,  ASKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTCT,  NCPPL,
     3 NTANC,  ICNTRW,  IER,  ISAVE(5),  NAPL,  NOIL(3),  IFLUR(2),
     *   ITROL(2),   JJ,   II,
     4IOLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IAMML(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRPO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NIPPB,NCPT,IRPSL,IPTCU,IPK(6,2),IOP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),INLCN(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RNLEV(3),YMMAX(3),YMMIN(3),RMAT(
     43),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTW(2),TRAIL(2),RAT(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TCL(3),TDH(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCCNT(9)
      COMMON /BLOK4/ NCR,  PCS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRAC,NRE
      COMMON /SAVE/ NSET(4,40),NOBSCT(13)
      COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEC(12),  JCMND(12),
     1  NCRAN(12),  ICO(5),  NMOUL
      DATA MUL(1,1),MUL(1,2),MUL(1,3),MUL(1,4),MUL(1,5),MUL(1,6),
     1MUL(1,7),MUL(1,8)/4,3,2,2,1,1,1,1/
      DATA MUL(2,1),MUL(2,2),MUL(2,3),MUL(2,4),MUL(2,5),MUL(2,6),
     1MUL(2,7),MUL(2,8)/4,3,2,2,1,1,1,1/
      DATA MUL(3,1),MUL(3,2),MUL(3,3),MUL(3,4),MUL(3,5),MUL(3,6),
     1MUL(3,7),MUL(3,8)/4,3,2,2,1,1,1,1/
      DATA MUL(4,1),MUL(4,2),MUL(4,3),MUL(4,4),MUL(4,5),MUL(4,6),
     1MUL(4,7),MUL(4,8)/4,3,2,2,1,1,1,1/
      DATA MUL(5,1),MUL(5,2),MUL(5,3),MUL(5,4),MUL(5,5),MUL(5,6),
     1MUL(5,7),MUL(5,8)/5,4,3,2,2,1,1,1/
      DATA CMIN(1),CMIN(2),CMIN(3),CMIN(4)/.00,.08,.01,.1/
      IDIV=1
      FRAT=1.1
      TTNOW=TIME(TLAST)
      III=0
      IF(.NOT. SENSW(4)) WRITE(6,2000) IX,TTNOW
      IF(IX .GT. 295) GO TO 100
      I=IX-249
      GO TO(250,250,250,250,250,255,255,255,255,255,260,260,260,260,
     1260,265,265,265,265,265,270,270,270,270,270,275,275,275,275,275,
     2280,280,280,280,280,285,285,285,285,285,290,290,290,290,290,295),I
  100 PAUSE 4
      J    .500
  101 PAUSE 5
      J    .500
 1000 CALL FILEP(1,NSET)
  500 CONTINUE
      IX=0
 1400 IF(SENSW(3)) GO TO 1401
      WRITE(6,3001) ATRIB
 1402 CONTINUE
      III=ISTAT(NCV)
      IOI=777
      WRITE(6,3002)NCV,III,ICI,JIO
 3002 FORMAT(1X,7OHNCV,III,ICI,JIO*/1X,4I4)
      WRITE(6,3003)(CV(NCV,J),J=1,9)
 3003 FORMAT(9F8.3)
 1401 IF(.NOT.SENSW(7)) ISET=0
      RETURN
C     END OF SLAG STAGE IN C(IX-249), SLAG TO BE REMOVED
  290 ISTAT(NCV)=4
      ISET=4
      CALL ONPRL(ICV(NCV,5),.FALSE.,IER)
      CALL BELLS(3,IX)
      CALL LIGHT(NCV+51,1)
      CALL LIGHT(46+NCV,1)
      NT(NCV,20)=0
      NNCV=NCV+2
      WRITE(IDIV,3005) NNCV
 3005 FORMAT(1X,7OHEND OF SLAG STAGE IN C*,I3/)
      J    .500
C     START CONVERTER BREAKDOWN
  255 NCV=IX-254
      JSTAT(NCV)=ISTAT(NCV)
      ISTAT(NCV)=13
      IL=ICV(NCV,8)
      CALL LIGHT(IL,0)
      CALL ONPRL(ICV(NCV,9),.FALSE.,IER)
      CALL LIGHT(41+NCV,1)
      CALL LIGHT(46+NCV,1)
      CALL ONPRL(ICV(NCV,9),.FALSE.,IER)
      NT(NCV,9)=TTNOW
C     SCHEDULE END OF BREAKDOWN EVENT
C CHECK IF OPERATOR STOP
      IF(0..LT.ATRIB(4)) GO TO 2950
      ATRIB(1)=TTNOW+20.
      ATRIB(2)=FLOAT(290+NCV)
      ATRIB(3)=FLOAT(NCV)
      ATRIB(4)=FLOAT(JSTAT(NCV)).
      J    .1000
 2950 ATRIB(1)=TTNOW+ATRIB(4)
      ATRIB(2)=FLOAT(290+NCV)
      ATRIB(3)=-FLOAT(NCV)
      ATRIB(4)=FLOAT(JSTAT(NCV))
      J    .1000
C     END OF CONVERTER BREAKDOWN
  260 NCV=IABS(ATRIB(3))
      ISTAT(NCV)=ATRIB(4)
      JIST=ISTAT(NCV)
      NT(NCV,10)=FLOAT(NT(NCV,10)-NT(NCV,9))+TTNOW
      GO TO(2601,2611,2600,2610,2610,2601,2610,2610,2600,2600,2611,
     12600,2610,2600,2601),JIST
 2601 NT(NCV,5)=FLOAT(NT(NCV,5)+NT(NCV,9))-TTNOW
      NT(NCV,10)=FLOAT(NT(NCV,10)+NT(NCV,9))-TTNOW
      MT(NCV,6)=FLOAT(MT(NCV,6)-MT(NCV,9))+TTNOW
 2620 CALL LIGHT(46+NCV,0)
      J    .2620
 2615 MT(NCV,8)=FLOAT(MT(NCV,8)+MT(NCV,9))-TTNOW
 2610 CALL LIGHT(41+NCV,0)
C     SCHEDULE NEXT CONVERTER BREAKDOWN
 2620 IF(ATRIB(3).LT.0.) GO TO 2625
      ATRIB(1)=TTNOW+400.
      ATRIB(2)=FLOAT(294+NCV)
      ATRIB(3)=NCV
      CALL FILEP(1,NSET)
 2625 IF(JIST.LT.3) GO TO 2630
      IF(JIST.NE.4) GO TO 500
C     RESUME BLOWING IN C(NCV)
 2630 CALL OWFRL(ICV(NCV,9),.TRUE.,IER)
      IL=ICV(NCV,6)
      CALL LIGHT(IL,1)
      CALL LIGHT(51+NCV,0)
      J    .500
C     COLD COPPER NEEDED AT C(IX-264)
 265  NCV=IX-264
      IF(ISTAT(NCV).EQ.2) GO TO 2655
      IF(ISTAT(NCV).NE.11) GO TO 500
 2655 ISTAT(NCV)=11
      ISET=11
      IAFO(NCV,15)=IAFO(NCV,15)+1
      JPG=IAFO(NCV,15)
      IF(JPG.LE.0) GO TO 2651
      CALL BELLS(2,IX)
      CALL LIGHT(56+NCV,1)
 2651 IF(JPG.LT.2) GO TO 2650
C DEMAND COPPER PIGS
      NNCV=NCV+2
      WRITE(IDIV,3010) JPG,NNCV
 3010 FORMAT(2X,3HADD,I3,1X,9HPIGS TO C,I2/)
 2650 CONTINUE
C SCHEDULE ANOTHER REQUEST FOR PIGS
      ATRIB(1)=TTNOW+FLOAT(MT(NCV,15))
      ATRIB(2)=NCV+264
      ATRIB(3)=NCV
      J    .1000
C SLAG SKIN PERMITTED IN GOING HIGH BLOW
  270 NCV=IX-269
      IF(14.LT.ISTAT(NCV)) GO TO 2700
      IF(ISTAT(NCV).NE.1) GO TO 500
      CALL LIGHT(66+NCV,1)
      ISTAT(NCV)=15
C SCHEDULE ANOTHER CALL FOR SLAG SKIN
 2700 ATRIB(1)=TTNOW+FLOAT(MT(NCV,19))
      ATRIB(2)=NCV+269
      ATRIB(3)=NCV
      J    .1000
C     START COLLAR PULL AT CONVERTER IX-274
 275  NCV=IX-274
      CALL LIGHT(41+NCV,1)
      CALL LIGHT(46+NCV,1)
      MT(NCV,11)=TTNOW
      MT(NCV,20)=0
      IL=ICV(NCV,6)
      CALL LIGHT(IL,0)
      NSTAT(NCV)=ISTAT(NCV)
      ISTAT(NCV)=12
      J    .500
C     FINISH COLLAR PULL AT CONVERTER IX-279
  280 NCV=IX-279
      IL=ICV(NCV,6)
      CALL LIGHT(IL,1)
      ISTAT(NCV)=NSTAT(NCV)
      ISS=ISTAT(NCV)
      MT(NCV,12)=FLOAT(MT(NCV,12)-MT(NCV,11))+TTNOW
      GO TO(283,284,281,281,282,283,281,282,281,281,284,281,281,281,
     1283),ISS
C TURN OFF CU LIGHT
  283 MT(NCV,5)=FLOAT(MT(NCV,5)+MT(NCV,11))-TTNOW
      MT(NCV,6)=FLOAT(MT(NCV,6)-MT(NCV,11))+TTNOW
  281 CALL LIGHT(46+NCV,0)
C COLLAR PULL RESULTS IN SKULL ADDITION
      ATRIB(1)=TTNOW
      ATRIB(2)=NCV+109
      ATRIB(3)=NCV
      J    .1000
C TURN OFF MATTE BLOW LIGHT
  284 MT(NCV,8)=FLOAT(MT(NCV,8)+MT(NCV,11))-TTNOW
      MT(NCV,18)=FLOAT(MT(NCV,18)-MT(NCV,11))+TTNOW
  282 CALL LIGHT(41+NCV,0)
      J    .500
C     END POINT OF A BLOWING PERIOD
  285 NCV=IX-284
      CALL ORBADR(CVI,ICV(NCV,3),2,IER)
      MT(NCV,13)=C
      MT(NCV,14)=C
C REVERSE REF TC CA
      CALL OWFRL(ICV(NCV,9),.FALSE.,IER)
C MATCH CA
      IL=ICV(NCV,6)
      CALL LIGHT(IL,0)
      IF(ISTAT(NCV).EQ.2)GO TO 290
      IF(ISTAT(NCV).EQ.11) GO TO 290
C END OF A SLAG BLOW IN C(IX-289)
      CV(NCV,1)=CVI(1)*TPU(NCV)+CV(NCV,3)
      CV(NCV,4)=CVI(2)*TPU(NCV)
      NNCV=NCV+2
      FACT=.01*FLOAT(100-IADD(10,2)-IADD(10,3)-IADD(10,4))
      SLG=CV(NCV,4)/(FLCAT(IADD(10,1))*FACT)
      MT(NCV,4)=TTNOW
C CALCULATE FLUX ADDED IN THIS BLOW
      MT(NCV,10)=FLOAT(MT(NCV,18)-MT(NCV,3))+TTNOW
      RFLX=.03ECT*FEAT*CV(NCV,6)*CV(NCV,7)
```

```
      FLUX=FLOAT(MT(NCV,18))*RFLX
      MT(NCV,18)=C
      MT(NCV,5)=FLOAT(MT(NCV,5)-MT(NCV,3))+TTNOW
      WRITE(IDIV,3004) NNCV, SLG ,FLUX
 3004 FORMAT(1X,8HEND SB C,I2,F7.2,9H LAD SLAG,F7.1,10H TONS FLUX/)
      IF(7.LT.NEVC(NCV)) GO TO 290
      IF(CV(NCV,8).LE.CV(NCV,2)) GO TO 290
      CALL LIGHT(66+NCV,1)
      CALL LIGHT(NCV+51,1)
      WLDLE=CPIA(1)*TPU(NCV)
      CALL BELLS(1,IX)
      IF(CV(NCV,4).LT.WLDLE) GO TO 2850
      ISET=3
      MBL=NEVC(NCV)
      MAT=IAFO(NCV,10)/100
      MMUL=MUL(NCV,MBL)
      IF(MMUL.LE.MAT) GO TO 2848
      ISTAT(NCV)=14
C *** SET ISET ACCORDINGLY ***
      J    .500
 2848 ISTAT(NCV)=3
      IAFO(NCV,10)=0
      J    .500
C END SLAG BLOW,NOT ENOUGH SLAG TO SKIN
 2850 ISTAT(NCV)=9
      ISET=7
      CALL LIGHT(56+NCV,1)
      CALL LIGHT(NCV+66,0)
      J    .500
C  END OF A FINISH STAGE IN C(IX-289)
 290  ISTAT(NCV)=8
      ISET=5
      CV(NCV,2)=CVI(1)*TPU(NCV)
      CV(NCV,5)=CVI(2)*TPU(NCV)
      MT(NCV,7)=TTNOW
      MT(NCV,8)=MT(NCV,8)+MT(NCV,7)-MT(NCV,2)
      MT(NCV,4)=TTNOW
      EV=FLOAT(NCV+264)+.001
      CALL FIND( EV,1,2,KCOL,NSET)
      WRITE(6,12345) EV,KCOL
12345 FORMAT(1X,F12.2,I7)
      IF(KCOL.EQ.0) GO TO 2900
      CALL RMOVE(KCOL,1,NSET)
 2900 CALL LIGHT(NCV+51,1)
      CALL LIGHT(56+NCV,1)
      CALL BELLS(2,IX)
      NNCV=NCV+2
      CU=CV(NCV,5)/FLOAT(IADD(11,1))
      CUS=CV(NCV,2)
      SLG=CV(NCV,4)
      WRITE(IDIV,3006) NNCV, CU
 3006 FORMAT(1X,17HEND CU STAGE IN C,I2,F6.1,13H LADLES OF CU/)
      J    .500
C ZERO IAFO,INFO ARRAYS PRIOR TO START OF NEW CHARGE
  295 NCCV=FLOAT(ISTRIB(1))+.5
      NCV=NCCV
      DO 2950 JII=1,15
      IAFO(NCCV,JII)=0
 2950 CONTINUE
      DO 2951 JCI=1,5
      INFO(NCCV,JCI)=0
 2951 CONTINUE
C ATRIB(4)=1.,TP,=2.,CU
      JCL=20
      IF(ATRIB(4).LT.2.) JCL=19
      DO 2952 JCI=1,JCL
      MT(NCCV,JCI)=0
 2952 CONTINUE
      ISTAT(NCV)=9
      MT(NCV,4)=TTNOW
      NEVC(NCCV)=1
      J    .500
C     CONVERTER STATUS REPORT
 990  JEST=ISTAT(NCV)
      IOC=777
      WRITE(IDIV,991) NCV,JEST ,JIO,IOC
      WRITE(IDIV,992)  (CV(NCV,I),I=1,9)
      J    .500
  991 FORMAT(5I10/)
  992 FORMAT(9E8.3/)
 2000 FORMAT(1X,5HEVENT,I4,7HAT TIME,F8.1)
 3001 FORMAT(1X,5HATRIB,2X,4F8.2)
      END
```

EIGHTH SUBMODULE

```
C     DECK FOR EIGHTH SUB MODULE
      SUBROUTINE DISK
C
C     THIS IS THE EIGHTH SUB MODULE.   ITS FILE NAME IS UPSBCT.
C     IT CONSISTS OF THE PREVIOLS EVNTS ROUTINE FOR EVENTS 39 TO 59
C
C
      LOGICAL LOGVAL,SENSW
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCROR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON/BLOK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),EPO(3),NC(3),NP
     1(3),  JMT(3),  KMT(3),  TCRAN(3)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTAT(3),   LOCDEV(3),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPR,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IANML(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRPO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATTME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIREX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TOL(3),TOM(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
      COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,NRE
      COMMON /SAVE/ NSET(6,60),MODSCT(13)
      COMMON /ARG/ IX,  IDIX,  ID2X,  JSET,  JSEQ(12),  JCMND(12),
     1  NCRAN(12),  ICO(5),  MODUL
```

```
************************************************************************

LIGHT MODEL STATE WHEN  LIGHT IS ON                   SW ON IN  SW OFF IN

  19  M TUN FULL LADLE OF MATTE NEAR MATTE TUN1 52          46
  20  M TUN FULL LADLE OF MATTE NEAR MATTE TUN2 53          47
  21  REVERB EMPTY LADLE REQUIRED AT MATTE T1    46         42
  22  REVERB FULL LADLE WAITING AT  TUNNEL 1     44         46
  23  REVERB EMPTY LADLE RQU. AT MATTE TUNNEL 1 47          43
  24  REVERB FULL LADLE WAITING AT M TUNNEL 2  45           47
  25  REVERB MATTE AVAILABLE IN REVERB 1        41          44
  26  REVERBMATTE LEVEL TOO HIGH IN REV. 1      41          44
  27  REVERB SLAG LEVEL TOO HIGH IN REVERB 1    48 OR 49    48
  28  REVERB MATTE AVAILABLE IN REVERB 2        41          44
  29  REVERBMATTE LEVEL TOO HIGH IN REV. 2      41          44
  30  REVERB SLAG LEVEL TOO HIGH IN REVERB 2    48 OR 50    48
  31  REVERB MATTE AVAILABLE IN REVERB 3        41          44
  32  REVERBMATTE LEVEL TOO HIGH IN REV. 3      41          44
  33  REVERB SLAG LEVEL TOO HIGH IN REVERB 3    48 OR 51    48
  34  REVERB SAVED FOR POSSIBLE THIRD MATTE TUNNEL
  35  REVERB SAVED FOR POSSIBLE THIRD MATTE TUNNEL
  36  CRANE CRANE 1 OPERATIONAL                 55 OR 56    55 OR 56
  37  CRANE CRANE 1 BUSY                        CRA1        GASP
  38  CRANE CRANE 2 OPERATIONAL                 55 OR 56    55 OR 56
  39  CRANE CRANE 2 BUSY                        CRA2        GASP
  40  CRANE CRANE 3 OPERATIONAL                 55 OR 56    55 OR 56
  41  CRANE CRANE 3 BUSY                        CRA3        GASP

************************************************************************
```

```
      **    E V E N T S   O F   S U B M O D E L S   **

      **    REVERBERATORY FURNACES -  **

      41    UPDATING OF MATTE LEVEL IN R1,R2,R3
      41    CHANGE OF MATTE PRODUCTION IN  R1,R2,R3
      42    EMPTY LADLE IS RETURNED TO MATTE TUNNEL 1
      43    EMPTY LADLE IS RETURNED TO MATTE TUNNEL 2
      44    FULL LADLE FROM MT 1 ARRIVES AT CONVERTER AISLE
      45    FULL LADLE FROM MT 2 ARRIVES AT CONVERTER AISLE
      46    CRANE PICKS FULL LADLE FROM MT 1 OR FROM FLOOR
      47    CRANE PICKS FULL LADLE FROM MT 2 OR FROM FLOOR
      48    SLAG LEVEL IS UPDATED IN R1,R2,R3
      48    REVERBERATORY SLAG IS SKIMMED FROM R1,R2,R3
      49    CONVERTER SLAG IS RETURNED TO R1
      50    CONVERTER SLAG IS RETURNED TO R2
      51    CONVERTER SLAG IS RETURNED TO R3
      52    LADLE OF MATTE DUMPED ON FLOOR AT MT1
      53    LADLE OF MATTE DUMPED ON FLOOR AT MT2

      **    OVERHEAD CRANES   **

      55    CRANE FAILURE GENERATED IF KAP=1
      55    CRANE REPAIR  GENERATED IF KAP=0
      56    CRANE MAINTENANCE IS SCHEDULED FOR CR1,CR2,CR3
      57    CRANE MAINTENANCE IS EXECUTEDD FOR CR1,CR2,CR3

      **    OTHER EVENTS   **

      40    SCHEDULING AND FORECASTING FOR AUTOMATIC OPERATION
            IS TRIGGERED AT FIXED TIME INTERVALS
      54    START OF NEW CONVERTER CYCLE IN AUT. OPERATION
      58    END OF 8-HOUR SHIFT,GIVES COMPLETE MODEL STATUS
            AND STATISTICS ON PERFORMANCE
      59    SIMULATION DATE IS UPDATED  EVERY 24 HOURS
      59    SIMULATION TIME DISPLAY IS UPDATED  EVERY 60 MINUTES

************************************************************************
```

```
C
      TTNOW=TIME(TLAST)

      CALL CLOCK(IER)
      IF(.NOT. SENSW(2)) WRITE(6,2000) IX,IER
      I=IX-38
      GO TO (101,40,
     541,  42,  43,  44,  45,  46,  47,  48,  49,  50,
     6 51,  52,  53,   54,  55,  56,  57,  58,  59 ) , I
  101 PAUSE 5
      J    ,500
 1000 CALL FILEM(1,NSET)
  500 CONTINUE
C     SENSE SWITCH 7 IS SET  FOR AUTOMATIC OPERATION
      IF(.NOT. SENSW(7)) ISET=0



      IX=0
      IF(I .EQ. 20) IX=30
  501 IF(SENSW(2)) GO TO 1400
C *** WRITE STATEMENTS
      WRITE(6,984) RMLEV,TOLA,TAW,SLEV,TOTSL
 1400 CONTINUE
 1401 RETURN
C
C**   EVENT CODE 40
C     SCHEDULING AND FORECASTING FOR AUTOMATIC OPERATION
C     IS TRIGGEREC AT FIXED TIME INTERVALS
   40 WRITE(6,400)TTNOW
  400 FORMAT(1X,19HSTATISTICS AT TIME=,F6.2//)
      ATRIB(1)=TTNOW+ FLOAT(ISTRIB(1))
      ATRIB(2)=40.0
      ATRIB(3)=ISTRIB(1)
      IF(SENSW(7)) MODUL=11
      J    ,1000
C***  EVENT CODE 41
C***  UPDATING OF MATTE LEVEL IN R1,R2,R3
   41 NRE=ISTRIB(1)
      LIG=22+NRE*3
      RMLEV(NRE)=RMLEV(NRE)+1.0
      IF(RMLEV(NRE).GE.YMMIN(NRE)) CALL LIGHT(LIG,1)
      IF(RMLEV(NRE).GE.YMMAX(NRE)) CALL LIGHT(LIG+1,1)
      J=0
      K=0
      L=0
      DO 410 I=1,3
      IF(RMLEV(I).GE.YMMAX(I)) J=J+1
      IF(RMLEV(I).LT.(YMMAX(I)-3.0)) K=K+1
      IF(RMLEV(I).LT.YMMIN(I)) L=L+1
  410 CONTINUE
      KT=0
      IF(L.GE.1) KT=KT+1
      IF(J.EQ.3) KT=KT+1
C     IF(K.GE.2)   KT=KT+1
      IF(KT.EQ.0) GO TO 411
      TYPE 412,(RMLEV(I),I=1,3),(YMMAX(I),I=1,3),(YMMIN(I),I=1,3)
      TYPE 4140,RMAT,(ADD(1,2),IADD(1,2),IADD(2,2)
      TYPE 4141
  417 TYPE 418
      ACCEPT 420,IK,PERC,NOB
      IF(NOB.GT.3.OR.NOB.LT.1)  GO TO 430
      IF(IK.EQ.1) GO TO 413
      IF(IK.EQ.2) GO TO 414
  430 TYPE 416
      J    ,417
C**   INCREASE MATTE GRADE
  413 IF(PERC.GT.0.) GO TO 4130
      RMAT(NOB)=RMAT(NOB)*100./(100.-4.*PERC)
      J    ,4131
 4130 RMAT(NOB)=RMAT(NOB)*(100.+PERC*4.0)/100.
 4131 N=1
```

```
      IF(NOB.EQ.3) N=2
      IADD(N,2)=FLOAT(IADD(N,2))*PERC
      IADD(N,3)=IADD(N,3)
      IADD(N,4)=IADD(N,4)
 413  MG=IADD(1,2)
      IF(RMAT(NOB).LE.0.) RMAT(NOB)=60.
      IF(NOB.EQ.3) MG=IADD(2,2)
      TYPE 419,NOB,MG,RMAT(NOB)
      IF(NOB.NE.3) GO TO 417
      J    .411
C***  DECREASE MATTE PRODUCTION RATE ONLY - I.E. DECREASE FIRE
 414  IF(PERC.LT.0.) GO TO 4135
      RMAT(NOB)=RMAT(NOB)*100./(100.+PREC)
      J    .4136
4135  RMAT(NOB)=RMAT(NOB)*(100.-PERC)/100.
4136  J    .415
 412  FORMAT(1X,10HHIGH MATTE,3F7.2/6X,4HMAX=,3F7.2/6X,4HMIN=,3F7.2)
 416  FORMAT(1X,5HERROR/)
 418  FORMAT(1X,37H1 CR 2,+ - PERC. CHANGE/HR., REV. NO./)
 419  FORMAT(/1X,13HPT GRADE OF R,I2,1H=,I5,7HMT RATE,F7.2)
 420  FORMAT(I2,F8.0,I2)
4141  FORMAT(1X,26HCHANGE GRADE(1) OR RATE(2),)
4140  FORMAT(1X,15HMT.RATE(MIN/IN),3F7.2/2X,14HMT.GRADE(PERC),3I7/)
 411  TR(NRE)=TINCH+RMAT(NRE)
      ATRIB(1)=TR(NRE)
      ATRIB(2)=41.
      ATRIB(3)=NRE
      J    .1000
C***  EVENT CODE 42
C***  EMPTY LADLE IS RETURNED TO MATTE TUNNEL 1
  42  IF(ITROL(1) .NE. 0) GO TO 900
      ITROL(1)=-1
      CALL LIGHT(21,0)
      IF(RMLEV(1).GE.YMMIN(1).AND.RMLEV(2).GE.YMMIN(2)) GO TO 421
      IF(RMLEV(1).GE.YMMIN(1)) GO TC 422
      IF(RMLEV(2).GE.YMMIN(2)) GO TO 423
      ATRIB(1)=TR(2)+RNORM(TLO(1),THI(1))+TRAIL(1)
      IF(TR(1).LE.TR(2))ATRIB(1)=TR(1)+RNORM(TLO(1),THI(1))+TRAIL(1)
      ATRIB(2)=44.
      NP(1)=ATRIB(1)+THI(1)
      J    .1000
 421  IF(RMLEV(1).GE.YMMAX(1)) GO TO 422
      IF(RMLEV(2).GE.YMMAX(2)) GO TO 423
 425  ATRIB(1)=TTNOW+RNORM(TLO(1),THI(1))+TRAIL(1)
      IAP=1
      J    .424
 422  IF(RMLEV(2).GE.(YMMAX(2)-1.5)) GO TO 425
      ATRIB(1)=TTNOW+2.*RNORM(TLO(1),THI(1))+TRAIL(1)
      IAP=2
      J    .424
 423  IF(RMLEV(1).GE.(YMMAX(1)-1.5))  GC TO 422
      ATRIB(1)=TTNOW+2.*RNORM(TLO(2),THI(2))+TRAIL(1)
      IAP=3
 424  ATRIB(2)=44.
      ATRIB(3)=IAP
      NP(1)=ATRIB(1)
      J    .1000
C***  EVENT CODE 43
C***  EMPTY LADLE IS RETURNED TO MATTE TUNNEL 2
  43  IF(ITROL(2) .NE. 0) GO TO 900
      ITROL(2)=-1
      CALL LIGHT(23,0)
      IF(RMLEV(3).GE.YMMIN(3)) GO TO 431
      ATRIB(1)=TR(3)+RNORM(TLO(2),THI(2))+TRAIL(2)
      ATRIB(2)=45.
      NP(2)=ATRIB(1)+THI(2)
      J    .1000
 431  ATRIB(1)=TTNOW+RNORM(TLO(2),THI(2))+TRAIL(2)
      ATRIB(2)=45.
      NP(2)=ATRIB(1)
      J    .1000
C***  EVENT CODE 44
C***  FULL LADLE FROM MATTE TUNNEL 1 ARRIVES AT CONVERTER AISLE
  44  CALL LIGHT(22,1)
      ISET=1
C**   NSLAG(1) STORES UNASSIGNED MATTE
      NSLAG(1)=NSLAG(1)+1
      CALL BELLS(2,44)
      ITROL(1)=1
      ITU=ISTRIB(1)
      GO TO (441,442,443),ITU
 441  RMLEV(2)=RMLEV(2)-RAT(2)
      IF(RMLEV(2).LT.YMMIN(2)) CALL LIGHT(28,0)
      IF(RMLEV(2).LT.YMMAX(2)) CALL LIGHT(29,0)
      TOLA(2)=TOLA(2)+1.0
      J    .445
 442  RMLEV(1)=RMLEV(1)-RAT(2)
      IF(RMLEV(1).LT.YMMIN(1)) CALL LIGHT(25,0)
      IF(RMLEV(1).LT.YMMAX(1)) CALL LIGHT(26,0)
      TOLA(1)=TOLA(1)+1.0
      J    .445
 443  DO 444 I=1,2
      RMLEV(I)=RMLEV(I)-0.5*RAT(I)
      IF(RMLEV(I).LT.YMMAX(I)) CALL LIGHT(25+I*2,0)
      IF(RMLEV(I).LT.YMMIN(I)) CALL LIGHT(I*2+24,0)
 444  TOLA(I)=TOLA(I)+0.5
 445  TAW(1)=TNOW
      CALL LADLE
      J    .900
C***  EVENT CODE 45
C***  FULL LADLE FROM MATTE TUNNEL 2 ARRIVES AT CONVERTER AISLE
  45  CALL LIGHT(24,1)
      ISET=2
C**   NSLAG(2) STORES UNASSIGNED MATTE
      NSLAG(2)=NSLAG(2)+1
      CALL BELLS(2,45)
      ITROL(2)=1
      RMLEV(3)=RMLEV(3)-RAT(3)
      IF(RMLEV(3).LT.YMMIN(3)) CALL LIGHT(31,0)
      IF(RMLEV(3).LT.YMMAX(3)) CALL LIGHT(32,0)
      TOLA(3)=TOLA(3)+1.0
      CALL LADLE
      TAW(2)=TNOW
      J    .900
C***  EVENT CODE 46
C***  CRANE PICKS FULL LADLE FROM MATTE TUNNEL 1 OR FROM FLOOR
  46  ITUN=1
 461  ISTAT(NCR)=ITUN
      IF(ITROL(ITUN).LE.0) GO TO 462
      CALL LIGHT(20+ITUN*2,0)
      CALL LIGHT(19+ITUN*2,1)
      TOTW(ITUN)=TOTW(ITUN)+TIME(TLAST)-TAW(ITUN)
      ITROL(ITUN)=0
 465  ISAVE(NCR)=100
      J    .900
 462  IF(IFLOR(ITUN).LE.0) GO TO 463
      IFLOR(ITUN)=IFLOR(ITUN)-1
      IF(IFLOR(ITUN) .EQ. 0) CALL LIGHT(ITUN+18,0)
      IF(IFLOR(ITUN).GT.0) TYPE 464,ITUN,IFLOR(ITUN)
 464  FORMAT(1X,9HLAD .AT M,I2,2H =,I2/)
      J    .465
 463  TYPE 464, ITUN
      ISAVE(NCR)=C
      J    .900
C***  EVENT CODE 47
C***  CRANE PICKS FULL LADLE FROM MATTE TUNNEL 2 OR FROM FLOOR
  47  ITUN=2
      J    .461
C***  EVENT CODE 48
C***  SLAG LEVEL IS UPDATED AND SLAG IS SKIMMED FROM R1,R2,R3
  48  DO 482 NSL=1,3
      LIG=24+NSL*3
      SLEV(NSL)=SLEV(NSL)+RSLG(NSL)
      IF(SLEV(NSL).LT.SMIN(NSL)) GO TO 481
      IF(RMLEV(NSL).GT.YMMAX(NSL)) GO TO 481
      SLG=UNFRM(STAPL(NSL),STAPH(NSL))
      SLEV(NSL)=SLEV(NSL)-SLG*POT(NSL)
      TOTSL(NSL)=TOTSL(NSL)+SLG
 481  IF(SLEV(NSL).GT.SPAX(NSL)) CALL LIGHT(LIG,1)
      IF(SLEV(NSL).LT.SMAX(NSL)) CALL LIGHT(LIG,0)
 482  CONTINUE
      ATRIB(1)=TTNOW+30.0
      ATRIB(2)=48.
      J    .1000
C***  EVENT CODE 49
C***  CONVERTER SLAG IS RETURNED TO R1
  49  ISR=1
 491  QSL=.01*FLOAT(ISAVE(NCR))
      IF(SLEV(ISR).LT.SMAX(ISR)) GO TO 492
      TYPE 493,ISR
 493  FORMAT(1X,1HR,I2,19H CANNOT ACCEPT SLAG/)
      J    .900
 492  ISAVE(NCR)=0
C  ** VINCR ACCOUNTS FOR THE CHANGE IN SP. WT.OF CV SLAG AND REV.SLAG
      VINCR=1.14
      SLEV(ISR)=SLEV(ISR)+SLRA(ISR)*QSL*VINCR
      IF(SLEV(ISR).GT.SMAX(ISR)) CALL LIGHT(24+ISR*3,1)
      TSLAG(ISR)=TSLAG(ISR)+QSL
      J    .900
C***  EVENT CODE 90
C***  CONVERTER SLAG IS RETURNED TO R2
  90  ISR=2
      J    .491
C***  EVENT CODE 91
C***  CONVERTER SLAG IS RETURNED TO R3
  91  ISR=3
      J    .491
C***  EVENT CODE 92
C**   DUMP LADLE OF MATTE ON FLOOR AT MT1
  92  CALL LIGHT(19,1)
      IFLOR(1)=IFLOR(1)+1
      IF(2.LT.IFLOR(1)) TYPE 530
      J    .900
C***  EVENT CODE 93
C**   DUMP LADLE OF MATTE ON FLOOR AT MT2
  93  CALL LIGHT(20,1)
      IFLOR(2)=IFLOR(2)+1
      IF(2.LT.IFLOR(2)) TYPE 530
      J    .900
C***  EVENT CODE 94
C *** MATTE ADDITION AT START OF NEW CV. CYCLE IN AUTOMATIC OPERATION
  94  ISET=7
      NCV=ISTRIB(1)
      J    .900
C***  EVENT CODE 95
C***  CRANE FAILURE AND REPAIR TIMES ARE GENERATED
  95  NKR=ISTRIB(1)
      KAP=ISTRIB(3)
      LIG=NKR*2+34
      IF(KAP.EQ.1) GC TO 551
      IF(KAP.EQ.0) GO TO 552
      PAUSE 95
 551  CALL LIGHT(LIG,0)
      KAPUT(NKR)=1
      CALL BELLS(4,95)
      IHOR=TNOW/60.
      DOW=RNORM(TCL(NKR),TDM(NKR))
      FA=(TNOW+DOW)/60.
      IB=FA
      PC=FLOAT(IB)+0.6*(FA-FLOAT(IB))
      TDOWN(NKR)=HOURS+PC-FLOAT(IHOR)
5509  TYPE 5510,NKR,TDOWN(NKR)
5510  FORMAT(1X,5HCRANE,I2,9H DOWN TILL,F6.2/)
      ATRIB(1)=TTNOW+DOW
      ATRIB(2)=95.
      ATRIB(3)=NKR
```

NINTH SUBMODULE

```
      ATRIB(4)=0.0
      J   .1000
 552  CALL LIGHT(LIG,1)
      KAPUT(NKR)=0
      CALL BELLS(2,55)
      CALL GONT(NKR,POSL,.01)        SAME FOR OTHER CRANES
C***  CALL GONT(NKR,PC(NKR),0.06)
      ATRIB(1)=TTNOW+UNFRM(CALINKR),CRM(NKR))
      ATRIB(2)=55.
      ATRIB(3)=NKR
      ATRIB(4)=1.0
      J   .1000
C***  EVENT CODE 56
C***  CRANE MAINTENANCE IS SCHEDULED
  56  NCE=ISTRIB(1)
      LIG=NCE*2+34
      IF(NOIL(NCE).EQ.2) GO TO 500
      IF(NOIL(NCE).EQ.1) GO TO 562
      CALL LIGHT(LIG,0)
      TYPE 561,NCE
 561  FORMAT(1X,5HCRANE,I3,23HIS DUE FOR MAINTENANCE/)
      CALL LIGHT(LIG,1)
      CALL BELLS(4,56)
      ATRIB(1)=TTNOW+30.
      ATRIB(2)=56.
      ATRIB(3)=NCE
      J   .1000
 562  ATRIB(1)=TTNOW+3.0*8.0*3.0*60.0
      ATRIB(2)=56.
      ATRIB(3)=NCE
      NOIL(NCE)=2
      CALL LIGHT(LIG61)
      KAPUT(NCE)=C
      CALL BELLS(2,56)
      CALL GONT(NCE,PC(NCE),0.06)
      J   .1000
C***  EVENT CODE 57
C***  CRANE MAINTENANCE IS EXECUTED
  57  NCC=ISTRIB(1)
      LIG=NCO*2+34
      NOIL(NCO)=1
      KAPUT(NCO)=1
      CALL LIGHT(LIG,C)
      JHOUR=TIME(TLAST)/60.0
      FA=(TIME(TLAST)+30.0)/60.0
      IB=FA
      FC=FLOAT(IB)+0.6*(FA-FLOAT(IB))
      TDOWN(NCO)=HOURS+FC-FLOAT(JHOUR)
      TYPE 9510,NCC,TDOWN(NCO)
      ATRIB(1)=TTNOW+90.0
      ATRIB(2)=56.
      ATRIB(3)=NCO
      J   .1000
C      END OF SHIFT EVENT
  58  ATRIB(1)=TTNOW+480.
      ATRIB(2)=58.
      TYPE 1406
      CALL BELLS(9,58)
      J   .1000
C***  EVENT CODE 59
C**   SIMULATION TIME IS UPDATED
  59  CALL CONTRL(6,1)
      HOURS=ISTRIB(1)
      IF(INHOUR .EQ. -1) INHOUR=HOURS+.5
      IF(HOURS.GT.23.1) ISTRIB(5)=ISTRIB(5)+100
      DATE=FLOAT(ISTRIB(5))/100.+.001
      IF(HOURS.GT.23.1) HOURS=0.0
      TYPE 591,HOURS,DATE
 591  FORMAT(1X,7HTIME = ,F6.2,3X,7HDATE = ,F5.2,5H,1973/)
      SETDA=-HOURS/100.0-0.0006
      CALL OWJCAR(SETDA,10,IER)
      ATRIB(1)=TTNOW+60.0
      ATRIB(2)=59.
      ATRIB(3)=HOURS+1.0
      ATRIB(4)=ISTRIB(5)
      CALL CONTRL(6,C)
      J   .1000
 444  FORMAT(1X,14HNO MATTE AT MT,I3/)
 510  FORMAT(1X,13HNO MT ON FLOOR,)
 904  FORMAT(1X,TIMRMLEV,RMLEV,RMLEV TOLA,TOLA,TOLA,TAM,TAM,SLEV,SLEV,SL
     1EV,TOTS,TOTS,TOTS,/1X,3F6.1,3F5.1,2F4.1,3F5.1,3F6.1)
1406  FORMAT(12HEND OF SHIFT/)
2000  FORMAT(1X,5HEVENT,I4,7HAT TIME,I5)
      END
```

```
C      DECK FOR NINTH SUBMODULE
       SUBROUTINE DISK
C
C       THIS IS THE NIN TH SUBMODULE  ITS FILE NAME IS UPS9CI
C      IT GENERATES CCMMANDS IN THE AUTOMATIC OPERATIONAL MODE
C
       DIMENSION LOCDEF(1)
       DIMENSION CRIN(4)
       DIMENSICN JFROM(12),JTO(12),JQ(12),JPR(12)
       LOGICAL SENSW
       COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,NX,NXC,NOO,SCALE,TSEED,TNOW,
      1TREG,TFIN,NXX,NPRNT,NCRDR,OUT
       COMMON ATRIB(4),IHN(4),KRANK(4),MFE(4),MLC(4),MLF(4),NQ(4)
      *,ISTRIB(5)
       COMMON/BLOK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
      1(3),  JWT(3),  KWT(3),  TCRAN(3)
       COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
      1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTRGT(17,7),  IPRA(17,2),
      2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
      3  NTANC,  ICNTRW,  IER,  ISAVE(5),  NAPL,  NOIL(3),  IPLOR(2),
      *   ITROL(2),   JJ,   II,
      4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
      5,IAPPL(3),IANNL(7),NEVC(5),IADD(11,4),ICV(5,9)
      6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(5)
      7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
      8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPR(6,2),IDP(6,2)
      9,NCPTT,LCPU,LAS,IAS,IPRIT(5),INHOUR,NACOL(9),ITCOR(15),IMLC=(15),
      *NOP(5,6),NTREQ(5),NTAV(2),IREP(5,6)
       COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
      1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST6A,  TCST6B,
      2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
      3XMAX,BETA,TMAX,TINEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
      4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),AATI(3),SLEV
      5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLA(3),SLRD(3),POT
      6(3),TDOWN(3),TDL(3),TDM(3),CRL(3),CRM(3),TCST6A,TCST6B,TAFR6A,
      7 TAFR6B,HOURS,TSLAG(3),CY(5,9),TPU(5)
      8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCCNT(5)
       COMMON /BLDRA/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
      * LOGVAL,NPRNC,NME
       COMMON /ARG/ IX,  IDIX,  IDZX,  ISET,  JSEC(12),  JCMND(12),
      1  NCRAN(12),  ICO(9),  MODUL
       COMMON /SAVE/ NSET(6,60),NODSCT(13)
       EQUIVALENCE (IBUFX(1),JFROM(1)),(IBUFY(1),JTO(1))
       EQUIVALENCE (IPR(1,1),JQ(1)),(IDP(1,1),JPR(1))
       DATA CRIN(1),CRIN(2),CRIN(3),CRIN(4)/.08,.01,.01,.1/
       NCRANE=NCR
       IFLAG=0
       IF(ISET-5COO) 3999,3979,3981
 3979  ISET=14
       J   .3999
 3981  ISET=ISET-5000
       IFLAG=1
 3999  CONTINUE
       ISEQ=3
       IPR=0
       IQ=0
       DO 3000 I=1,12
       NCRAN(I)=0
       JPR(I)=0
       JQ(I)=0
 3000  CONTINUE
       IF(SENSW(0)) WRITE(6,4010) ISET,NCV,NPRNC,ISTAT
 4010  FORMAT(1X,4HISET,3I4/1X,5HISTAT,5I4)
       IF(25. LT .ISET) GO TO 4000
       IF(ISET.LT.1) GO TO 4000

       GO TO(1,2,3,4,5,6,7,8,30,30,11,12,13,14,15,16,17,18,19,20,30,30
      1,30,30,25) ,ISET
```

```
4000 PAUSE 17
     J    .6000
5000 NO=2
     CALL CRANE(NC,LOCDEV(IFROM),LOCDEV(ITO),NBEST)
     NCR=NBEST
     IF(NCR.EQ.0) GO TO 6000
5002 JSEQ(1)=ISEQ
     NCRAN(1)=NCR
     JCMND(1)=ICMND
     JFROM(1)=IFROM
     JTO(1)=ITC
     JPR(1)=IPR
     JQ(1)=10
5001 MODUL=4
     IF(NCRAN(1).EQ.0) GO TO 6000
     J    .9000
7000 TYPE 7001
7001 FORMAT(1X,16HER.AT 7000 IN 9/)
6000 MODUL=0
     TYPE 6001, ISET
6001 FORMAT(1X,16HNO EV. FOR ISET=,I3/)

9000 ISET=0
     IF(IFLAG.EQ.1)  GO TO 14
     IF(NCRAN(12).NE.0) GO TO 7000
     NCR=NCRANE
     RETURN
 50  J    .6000
C*** POMT .M .C .
 1   IFROM=7
     J    .100
 2   IFROM=8
 100 IT=0
     ICMND=2
     DO 105 I=1,5
     IF(8.LT.NEVC(I)) NEVC(I)=8
     IF(ISTATE(I).NE.9) GO TO 105
     IT=IT+1
     ICO(IT)=I
 105 CONTINUE

     NBEST=ICO(1)
     IF(O.LT.IT) GO TO 107
     CALL TISTAT(6,14,IT,ISTAT,ICO)
 101 IF(IT.EQ.C) GO TC 103
     NBEST=ICO(1)
     IF(1.LT.IT) CALL CONV(1,IT,NBEST,MUL)
 107 ITO=NBEST+8
C*** LOOK FOR ADD MATTE EVENTS
     DO 111 JQ=2,4
     IVAL=NBEST+89
     CALL FNDCEV(IVAL,JQ,4,KCOL,NSET)
     IF(KCOL.NE.0) GO TO 6000
 111 CONTINUE
     NSLAG(ISET)=NSLAG(ISET)-1
     J    .106
 103 ITO=IFROM
     IF(IFLOR(ISET).LT.3) GO TO 106
     TYPE 109,ISET
 109 FORMAT(1X,10H3 LD ON FL,I2/)
     J    .6000
 106 J    .5000

C**** POSL .C .R .
 3   CALL TWOCR(JC1,JC2,KAPUT,NO,NCV)
     FI=IADD(10,1)
     FRCT=100-IADD(10,2)-IADD(10,3)-IADD(10,4)
     FACT=.01*FRCT
     SLG=CV(NCV,4)/(FI*FRCT)
     ISLG=SLG+.00001
     FSLG=ISLG
     FOVER=(SLG-FSLG)*FI
     TT=CHIN(1)*TPU(NCV)
     IF(ISET.EQ.4) TT=CHIN(2)*TPU(NCV)
     IF(TT.LT.FOVER) ISLG=ISLG+1
     IF(ISLG.LT.1)  GO TO 315
     DO 300 I=1,ISLG
     JSEQ(I)=3
     JCMND(I)=3
     JFROM(I)=NCV+8
     CALL REV(NBEST,IRS)
     JTO(I)=NBEST
     IF(ISET.EQ.4) GO TO 350
     IF(I.EQ.ISLG) GO TO 301
 350 JPR(I)=8
     JQ(I)=0
     IPCX=JC1
     KS=(I/2)*2
     IF(KS.EQ.I)  IPCX=JC2
     NCRAN(I)=IPCX
     IOTH=JC1
     IF(IPCX.EQ.JC1) IOTH=JC2
 300 CONTINUE
     IF(ISET.EQ.4) GO TO 410
C*** POSL JC2
 301 NCRAN(I)=JC2
     JQ(I)=20+JC1
     JPR(I)=7
     J    .316
C*** POMT JC1
 315 I=0
 316 J=I+1
     CALL MATTE(NTUN)
     IF(NTUN.EQ.0) GO TO 302
     JFROM(J)=NTUN
 310 JSEQ(J)=3
     JCMND(J)=2
     JTO(J)=NCV+8
```

```
     NCRAN(J)=JC1
     JQ(J)=JC2+23
C***   ONLY MATTE IS ADDED
     IF(J.LT.2) JQ(J)=0
     JPR(J)=8
     J    .304
 302 WRITE(1,399)
 399 FORMAT(1X,15HNC MT,SKIM ONLY/)
     NCRAN(J)=0
     IF(J.LT.2) GO TO 6000
     JQ(J-1)=0
 304 CONTINUE
     J    .5001
C**** POSL ,C ,C , (HIGH GRADE SLAG)
  4  GO TO 3
C*** CHOOSE CORRECT CONVERTER HERE
 410 JS=I-1

     JCMND(JS)=13
     CALL TISTAT(6,1,IT,ISTAT,ICO)
     IF(IT.EQ.0) GO TO 402
     NBEST=ICO(1)
     IF(1.LT.IT) CALL CONV(1,IT,NBEST,MUL)
     JTO(JS)=NBEST+8
     J    .404
 402 TYPE 403
 403 FORMAT(1X,14HNO CONV.FOR CS/)
     JTO(JS)=JPRCH(JS-1)
C**   NO CONVERTER TO TAKE CS PLACE ON FLOOR
 404 JQ(JS)=20+ICTH
     JPR(JS)=6
C**  POFD .C .C .
 415 JS=JS+1
     JSEQ(JS)=3
     JCMND(JS)=8
     NCRAN(JS)=IOTH
     JFROM(JS)=NCV+13
     JTO(JS)=NCV+8
     JQ(JS)=23+IPCX
     JPR(JS)=7
     J    .5001
C**** POCU .C .PR-AF.
C    CU WILL BE POURED IF 15 MOULDS ARE AV. AND THERE ARE LESS T 35 PGS

  5  CALL TWOCR(JC1,JC2,KAPUT,NQ,NCV)
     CUNT=IADD(11,1)
     CU=CV(NCV,5)/CUNT
     ICU=CU+.00001
     FCU=ICU
     PLEFT=(CU-FCU)*CUNT
     TT=CHIN(3)*TPU(NCV)
     IF(TT.LT.PLEFT) ICU=ICU+1
     IF(ICU.LT.1)  GC TO  6000
     ICU=ICU+1
     NPGS=10
     LEFTP=26
     NOURY=NPPB
     NUM=NPPP+NPPB
C*** POSK JC1
     JSEQ(1)=3
     JCMND(1)=9
     NCRAN(1)=JC1
     JFROM(1)=21
     JTO(1)=NCV+8
     JQ(1)=50+JC2
     JPR(1)=6
C    IF(NSKL.GT.C) GO TO 520
     IF(0.LT.NSKL) GO TO 520
     WRITE(1,521)
 521 FORMAT(1X,8HNO SKULL/)
     ISET=16
     J    .9000
C*** POCU JC1 AND JC2
 520 DO 500 I=2,ICU
     JSEQ(I)=3
     JCMND(I)=4
     JFROM(I)=NCV+8
     JPR(I)=8
     JQ(I)=0
     IPCX=JC1
     KS=(I/2)*2
     IF(KS.EQ.I) IPCX=JC2
     NCRAN(I)=IPCX
     IOTH=JC1
     IF(IPCX.EQ.JC1) IOTH=JC2
     IF(I.EQ.2) JQ(2)=13+IOTH
     IF(I.EQ.2) JPR(2)=7
C*** DETERMINE JTO(I)
C    IF(NUM.GE.NPGS) GO TO 501
     IF(NPGS.LT.NUM) GO TO 501
 506 NOURY=NOURY+4
     JTO(I)=20
     J    .500
 501 IT=0
     DO 505 JR=1,3
     IF(IAFPL(JR).NE.0) GO TO 505
     IT=IT+1
     ICO(IT)=JR
 505 CONTINUE
     IF(1.LT.IT) GO TO 502
     IF(IT.EQ.1) GO TO 503
C    IF(NOURY.LE.LEFTP) GO TO 506
     IF(NOURY.LT.LEFTP) GO TO 530
     WRITE(1,504) NCV
 504 FORMAT(1X,18HNO PL FOR CU FR CV,I3/)
     ISET=15
     J    .9000
 503 JTO(I)=ICO(1)+3
```

```
      J    .503
C**** CALL ANODE
  502 CALL ANODE1(IT,NBEST)
      JTJ(I)=NBEST+3
  503 CONTINUE
C***  PONT JC1
C  ** REMOVED
  512 J    .5001
C***  POSK     ETC. IF REQUIRED

    6 CONTINUE
      CALL TISTAT(8,6,IT,ISTAT,ICO)
      IF(IT.LE.0) GO TO 6000
      IF(IT.EQ.1) NCV=ICO(1)
      IF(1.LT.IT) TYPE1810,NCV
 1810 FORMAT(1X,3HCV ,I3,15HIS EMPTIED ONLY/)
      ISET=5
      J    .3999
C**** PONT ,MT,C ,
    7 ICMND=2
      CALL MATTE(IFROM)
      IF(IFROM.EQ.0) GO TO 6000
      NBEST=NCV
      J    .107
C****  SCHEDULE EVENT TO ADD MATTE TO NCV AFTER CU-STAGE
    8 ATRIB(1)=TIME(TLAST)+15.
      ATRIB(2)=54
      ATRIB(3)=NCV
      CALL FILEP(1,NSET)
      J    .9000
C**** POPG ,PB-PP,C ,
   11 ICMND=5
      IQ=1
      ITO=NCV+8
      IF(I.LT.NCOLP) GO TO 1101
      IF(I.LT.NPPP) GO TO 1100
      TYPE 1102
      J    .6000
 1101 IFROM=20
      J    .5000

 1100 IFROM=19
      J    .5000
 1102 FORMAT(1X,7HNO PIGS/)
C**** POPG ,PI,PP,
   12 ICMND=5
      IQ=1
      ITO=19
      IFROM=22
      J    .5000
C**** PORS ,PI,C ,
   13 ICMND=7
      IFROM=22
      CALL TISTAT(6,1,IT,ISTAT,ICO)
      IF(IT.EQ.0) GO TO 6000
      NBEST=ICO(1)
      IF(1.LT.IT) CALL CONV(1,IT,NBEST,NUL)
      ITO=8+NBEST
      J    .5000
C**** POLD ,C ,BR
C***  POLD ,C ,BB,  AND BUMP
   14 CONTINUE
C* *   CHECK IFLAG
      IF(IFLAG.NE.1) GO TO 1403
      IFLAG=0
      DO 1402 I=1,12
      IF(NCRAN(I).EQ.0) GO TO 1401
 1402 CONTINUE
      J    .7000
 1403 I=1
 1401 JSEQ(I)=3
      JCMND(I)=10
      JFROM(I)=9
      JTO(I)=21
      CALL CRANE(2,LOCDEV(9),LOCDEV(21),NBEST)
      NCRAN(I)=NBEST
      IF(1.LT.I) NCRAN(I)=NCRAN(I-1)
      J=I+1
      JSEQ(J)=2
      JCMND(J)=12
      JTO(J)=21
      NCRAN(J)=NCRAN(I)
      J    .9001
C**** POPG ,PB,PP,
   15 ICMND=5
      IQ=2
      IFROM=20
      ITO=19
C **  IF PB IS EMPTIED IT SHOULD BE DONE BY CR3
       IF(NCR.EQ.3)GO TO 9002
      J    .9000
C***  POAS ,AF,C ,  AND  BE AF
   16 CONTINUE
      ICMND=11
      IFROM=NFRNC+3

      CALL TISTAT(6,16,IT,ISTAT,ICO)
      ITO=ICO(1)+8
      IF(IT.EQ.0) GO TO 6000
      J    .9000
 **** CAAF
   17 ISEQ=7
      NN=2
      IF(NFRNC .EQ. 3) NN=1
      IF(IAMPL(NN).EQ.1) GO TO 6000
      ITO=NFRNC+3
      NCR=1
```

```
      J    .5002
C**** POSL C ,R ,    IN GOING HIGH BLOW
   18 ICMND=1
      IFROM=NCV+8
      CALL REV(ITC,IRS)
      J    .5000

C**** CAA4 OR A5
   19 ISEQ=7
      ITO=NFRNC+3
      NOTHER=1
      IF(NFRNC.EQ.1) NOTHER=2
      IF(IAPPL(NOTHER).EQ.3)  GO TO 1900
      J    .6000
 1900 ITO=NOTHER+3
      J    .5002
C***  REAF
   20 CONTINUE  ,
      ISEQ=6
      ITO=NFRNC+3
      NCR=1
      J    .5002
C***  PULL C ,
   25 ISEQ=1
      ICMND=12
      IFROM=NCV+8
      J    .5000
      END
```

```
      SUBROUTINE TWOCR(JC1,JC2,KAPUT,NQ,NCV)
C**   SELECTION OF TWO CRANES IF SENSE SWITCH IS NOT SET
      LOGICAL SENSW
      DIMENSION KAPUT(3),NQ(1)
      COMMON/BLOK1/PO(3),WT(3),IP(3),SP(3),POD(3),WTO(3),IPO(3),NC(3),N
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      JC2=3
      JC1=1
      IF(SENSW(6))GO TO 308
    3 IF(KAPUT(1).EQ.1) GO TO 305
      IF(KAPUT(3).EQ.1) GO TO 306
      IF(2.0.LT.WT(1)) GO TO 303
      IF(2.0.LT.WT(3)) GO TO 304
      IF((NQ(2)-NQ(4)).GE.3) JC1=3
      IF((NQ(4)-NQ(2)).LT.3) GO TO 307
  304 JC2=1
      JC1=1
      J    .307
  303 JC1=3
      J    .307
  305 JC1=3
      JC2=2
      J    .307
  306 JC2=1
  307 RETURN
  308 IF(NCV.EQ.4.OR.NCV.EQ.2) GO TO 303
      J    .304
      END
```

```
      SUBROUTINE CRANE(NO,LOCA,LOCB,NBEST)
C**   SELECT CRANE ON BASIS OF LOCATION ONLY
      COMMON ID,IM,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POD(3),WTO(3),IPO(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),  LOCDEV(31),  IWTHET(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANK,  ICNTON,  IER,  ISAVE(9),  NAPL,  NOIL(3),  IPLOW(2),
     4   ITROL(2),  JJ,  II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPPL(3),IAMPL(2),NEVC(5),IABD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IMBS(5),IRPO(5,5),IAPO(5,15),WT(5,20),NCV,NTLAD(3),NAMCST(3),
     8 IATIM(3),NTPPB,NCPT,IRBSL,IPTCU,IPR(6,2),IDP(6,2)
     9,NCPTT,LCPU,ARS,IRS,IPRLT(5),INHOUR,NACOL(3),ITCOM(15),IBLCU(15),
     *NOP(5,6),PTREQ(5),RTAV(2),IREP(5,6)
      COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  ADDE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,NRF
      COMMON /BRG/ IT,  IDLX,  IQ2,  ISET,  JSEQ(12),  JCMND(12),
     1  NCRAN(12),  ICO(5),  NODUL
      IF(NO.EQ.3) GO TO 20
C     IF(LOCA.LT.290.AND.LOCB.LT.290) GO TO 10
      IF(LOCA.LT.300.AND.LOCB.LT.210) GO TO 10
    9 NBEST=3
      IF(NQ(2).LT.NQ(4).AND.2.LT.NQ(4))  NBEST=1
      IF(2.0.LT.WT(3)) NBEST=1
      IF(KAPUT(3).EQ.1)NBEST=1
      J    .90
   10 NBEST=1
      IF(NQ(4).LT.NQ(2).AND.3.LT.NQ(2)) NBEST=3
      IF(2.0.LT.WT(1)) NBEST=3
      IF(KAPUT(1).EQ.1) NBEST=3
      J    .90
   20 IF(LOCA.LT.190.AND.LOCB.LT.160) GO TO 10
      IF(LOCA.LT.300.AND.LOCB.LT.300) GO TO 9
      NBEST=2
      IF(2.0.LT.WT(2)) NBEST=3
      IF(KAPUT(2).EQ.1) NBEST=3
   90 RETURN
      END
      SUBROUTINE CONV(JZZZ,IT,NBEST,MUL)
      DIMENSION MUL(5,8)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFPL(3),IAMNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RAT(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TOL(3),TON(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
      COMMUN /ARG/ IX,  IO1X,  IO2X,  ISET,  JSEQ(12),  JCMNO(12),
     1  NCRAN(12),  ICO(5),  MODUL
C**   JZZZ=1  ADDITICN OF MATTE,REACTOR SLAG, HIGH GRADE CONV. SLAG
C**   JZZZ=2  FIND CCNVERTER TO RECEIVE TR.
C**   JZZZ=3  FIND CCNVERTER TO SUPPLY TN.
C**   JZZZ=4  FIND CCNVERTER TO SUPPLY COPPER
      GO TO (1,7,7,7)  ,JZZZ
    7 PAUSE  21
    1 DO 10 N=1,IT
      M=ICO(N)
      IF(M.LT.0.OR.5.LT.M) TYPE111,M
      IF(8.LT.NEVC(M)) NEVC(M)=8
      KR=NEVC(M)
      IF(KR.LT.0.CR.8.LT.KR) TYPE 112,KR
      IPRIT(M)=IAFO(M,10)/100
      IF(IPRIT(M).LT.0) IPRIT(M)=0
   10 CONTINUE
      CALL SORT(IPRIT,IT,MA,MI)
      NBEST=MI
      J    .100
  100 IF(0.LT.NBEST.AND.NBEST.LT.6)  GO TO 200
      NBEST=ICO(1)
  200 RETURN
  111 FORMAT(1X,10HER CCNV.NO,I3/)
  112 FORMAT(1X,10HER.BLOW NO,I3/)
      END


      SUBROUTINE REVEN(NBEST,IRS)
      COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOCL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B/  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RAT(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TOL(3),TON(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
      CCMMON /ARG/ IX,  IO1X,  IO2X,  ISET,  JSEQ(12),  JCMNO(12),
     1  NCRAN(12),  ICO(5),  MODUL
      NBEST=IRS+1
      IF(NBEST.LT.0.OR.3.LT.NBEST)NBEST=1
      IRS=NBEST
      RETURN
      END


      SUBROUTINE MATTE(MTUR)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFPL(3),IAMNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      MTUR=0
      WRITE(6,121)NSLAG
  121 FORMAT(1X,5HNSLAG,3I7)
      IF(NSLAG(1).GT.0) GO TO 7
      IF(NSLAG(2).GT.0) GO TO 8
   10 RETURN
    7 MTUR=7
      IF(ITROL(1).LT.1.AND.ITROL(2).EQ.1) GO TO 8
      NSLAG(1)=NSLAG(1)-1
      J    .10
    8 MTUR=8
      NSLAG(2)=NSLAG(2)-1
      J    .10
      END


      SUBROUTINE ANODE(IT,NBEST)
      COMMON /ARG/ IX,  IO1X,  IO2X,  ISET,  JSEQ(12),  JCMNO(12),
     1  NCRAN(12),  ICO(5),  MODUL
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(1:),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPL R(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFPL(3),IAMNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANC T(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),I LCH(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      CALL SORT(ILADA,IT,MA,MI)
      NBEST=MI
      IF(NBEST.LT.0.OR.3.LT.NBEST) TYPE92 ,NBEST
   92 FORMAT(1X,1CHER,AFR. NO,I3/)
      IF(3.LT.NBEST.CR.NBEST.LT.1) NBEST=ICO(1)
      RETURN
      END


      SUBROUTINE SORT(JU,K,MA,MI)
C     THIS ROUTINE RETURNS THE MAX. AND MIN. VALUCS OF JU AS M  AND MI
C     VALUES OF JU ARE BETWEEN 0  AND  ITOP
      DIMENSION JU(5)
      COMMON /ARG/ IX,  IO1X,  IO2X,  ISET,  JSEQ(12),  JCMNO(12),
     1  NCRAN(12),  ICO(5),  MODUL
      MIC=32767
      MAC=0
      IF(K.LT.2) TYPE 82,K
   82 FORMAT(1X,9HER.K SORT,I6/)
      DO 70 I=1,K
      L=ICO(I)
      IF(L.LT.1.OR.L.GE.6)  TYPE 81,L,K
   81 FORMAT(1X,10HER. L SORT,2I6/)
      IF(JU(L)-MAC)67,66,66
   66 MAC=JU(L)
      MA=L
   67 IF(JU(L)-MIC)68,70,70
   68 MIC=JU(L)
      MI=L
   70 CONTINUE
      WRITE(6,80) MA,MI
   80 FORMAT(1X,5HMA,MI,2I7/)
      RETURN
      END


      SUBROUTINE TESTAT(IFI,JFI,IT,ISTAT,ICO)
      DIMENSION ICO(1),ISTAT(1)
      INO=IFI
      JNO=JFI
  300 IT=0
      DO 400 I=1,5
      IF(ISTAT(I).NE.INO) GO TO 400
      IT=IT+1
      ICO(IT)=I
  400 CONTINUE.
      IF(IT.NE.0) GO TO 900
      IF(INO.EQ.JNO) GC TO 900
      INO=JNO
      J  - .300
  900 RETURN
      END


      SUBROUTINE PROCEV(IVAL,JO,JATT,KCOL,NSET)
      DIMENSION NSET(6,1)
      COMMON IO,IP,INIT,JEVNT,NPA,NSTOP,NX,NRC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(6),ENQ(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(1 ),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTCT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPL(R(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFPL(3),IAMNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAFO(5,15),NT(5,20),NCV,NTLAD(3),NANC T(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),I LCH(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      KBEST=0
      NEXTR=MFE(JC)
      IF(NEXTR) 16,1,20
C  16 CALL ERROR(99,NSET)
   16 PAUSE 75
    1 KCOL=KBEST
      IF(NSET(JO-1).EQ.IEVNTW) KCOL=[illegible]
    2 IIINCO=JO-1
      IF(KCOL.NE.0) TYPE 98,IIINCO
   98 FORMAT(1X,8HNT CN CO,I6)
      RETURN
   20 ICVNTW=NSET(JATT,NEXTR)-1000*(NSET(JATT,NEXTR)/1000)
      IF(IEVNTW-IVAL) 6,15,6
    6 NEXTR=NSET(MS,NEXTR)
      IF(NEXTR-7777) 20,1,1
   15 KCOL=NEXTR
      J   .2
      END
```

```
DECK FOR TENTH SUBMODULE
SUBROUTINE CISK

THIS IS THE TENTH SUBMODULE.   ITS FILE NAME IS UP10CI
IT GENERATES CRANE COMMANDS WHICH ARE INITIATED BY MODULE 9.

LOGICAL SENSW
DIMENSION JFROM(12),JTO(12),JQ(12),JPR(12)
COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
+,ISTRIB(5)
COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
1  NRANK(3),  ISTRT(3),  LOCOEV(31),  INTNGT(17,2),  IPRA(17,2),
2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
+   ITROL(2),   JJ,   II,
4IDLE(3),KAPUT(3),NEXT(3),LSTA(3),NSLAG(3)
5,IAFFL(3),IAWML(2),NEVC(5),IADD(11,4),ICV(5,9)
6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
7,INGS(5),IAFO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPR(6,2),IDP(6,2)
9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
+NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
COMMON /BLOK3/  TP1A,  TP1B,  TP1SLA,  TP1SLB,  TPCOOL,
1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RAT(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
6(3),TDOWN(3),TOL(3),TDM(3),CAL(3),CRM(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
+ LOGVAL,NFRNC,NRE
COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
1  NCRAN(12),  ICD(5),  MODUL
COMMON /SAVE/ NSET(6,50)
EQUIVALENCE (IBUFX(1),JFROM(1))
EQUIVALENCE (IBUFY(1),JTO(1))
EQUIVALENCE (IPR(1,1),JQ(1)),(IDP(1,1),JPR(1))
NCRANE=NCR
DO 99 LOOP=1,12
IF(NCRAN(LOOP) .EQ. 0) GO TO 999
NCR=NCRAN(LOOP)
ISEQ=JSEQ(LOOP)
ICMND=JCMND(LOOP)
IFROM=JFRCM(LOOP)
ITO=JTO(LOOP)
QTY=JQ(LOOP)
IPR=JPR(LOOP)
IQ=JQ(LOOP)
IF(SENSW(8)) WRITE(6,100) ISEQ,ICMND,NCR,IFROM,ITO,QTY,JPR(LOOP)
MCMND=0
GO TO (1,2,3,4,5,6,7,8) , ISEQ
 PICK COMMANDS
1 IXX=IFRCM
40 POS=LOCOEV(IXX)
TW= 0.
CALL CMNDG(CTY,ICMND,ISEQ,IXX,NSET,MCMND)
J   .99
 DUMP COMMANDS
2 IXX=ITO
J   .40
 PICK AND DUMP CCMMANDS
3 TW=0.
IPR=JPR(LOOP)
POS=LOCOEV(IFRCM)
IF((IQ/10).EQ.1) KODE=IQ-10
CALL CMNDG(CTY,ICMND,1,IFRCM,NSET,MCMND)
KODE=0
IF(MCMND .EQ. 99) GO TO 99
POS=LOCOEV(ITO)
TW=0.
MCMND=1
IPR=JPR(LOOP)
IF((IQ/10).EQ.2) KODE=IQ-20
CALL CMNDG(QTY,ICMND,2,ITO,NSET,MCMND)
KODE=0
IF(MCMND .EQ. 99) GO TO 99
GO TO (52,52,52,52,99,55,52,55,55,99,55,15,52) , ICMND
 RETURN LADLE
52 I=10
54 POS=LOCOEV(IFRCM)
TW=0.
IPR=JPR(LOOP)
MCMND=2
```

```
      IF((IQ/10).EQ.3) KODE=IQ-30
      CALL CMNDG(CTY,1,2,IFRCM,NSET,MCMND)
      KODE=0
      J   .99
C       RETURN BCAT
   55 I=14
      J   .54
   15 PAUSE 1
    4 CONTINUE
    5 CONTINUE
C       REFINE CCMMAND
C       REFINING CCMMAND
C       EVENT 34, START ANODE FURNACE 4 OR 5 TO REFINE
    6 NFRNC=ITO
   12 NF=NFRNC-3
C     IF(IAFFL(NF) .NE. 6) GU TO 38
      IAFFL(NF)=2
      CALL CLOCK(IATIME(NF))
      CALL LIGHT(NFRNC+7,0)
      CALL LIGHT(NFRNC+4,0)
      ATRIB(2)=35.
      ATRIB(3)=ILADA(NF)
      ATRIB(4)=NFRNC
      IF(NFRNC-5) 1005,1003,1004
 1004 A=TAF6A
      B=TAF6B
      J   .1007
 1005 A=TAF4A
      B=TAF4B
      J   .1007
 1003 A=TAF5A
      B=TAF5B
      J   .1007
   38 TYPE 362,NFRNC
      J   .99
C       CAST COMMAND
    7 NFRNC=ITO
      L=72
      IF(NFRNC .EQ. 6) L=73
      NW=74-L
      NF=NFRNC-3
      IF(IAFFL(NF) .NE. 1) GO TO 36
      IF(IAWML(NW) .EQ. 1) GO TO 37
      IAFFL(NF)=4
      CALL CLOCK(IATIME(NF))
      IAWML(NW)=1
      CALL LIGHT(L,1)
      CALL LIGHT(NFRNC+10,0)
      ATRIB(2)=37.
      ATRIB(3)=NFRNC
      IF(NFRNC-5) 1001,1006,1008
 1001 A=TCST4A
      B=TCST4B
      J   .1007
 1008 A=TCST6A
      B=TCST6B
      J   .1007
 1006 A=TCST5A
      B=TCST5B
 1007 CALL ANTIME(NFRNC,A,B)
 1002 CALL FILEM(1,NSET)
      J   .99
   36 TYPE 360,NFRNC
      J   .99
   37 TYPE 361,NW
      J   .99
C       DUMP COMMAND
    8 CONTINUE
   99 CONTINUE
  999 NCR=NCRANE
      RETURN
  100  FORMAT(1X,I1,4I3,F4.1,3I2)
  360 FORMAT(5HANODE,I2,17HNOT READY TO CAST/)
  361 FORMAT(5HWHEEL,I2,9HNOT READY/)
  362 FORMAT(5HANODE,I2,12HWRONG STATUS/)
      END
```

ELEVENTH SUBMODULE

```
 DECK FOR ELEVENTH SUBMODULE
 DECK FOR ELEVENTH SUBMODULE
 SUBROUTINE CISK

 THIS IS THE ELEVENTH SUBMODULE  ITS FILE NAME IS UP11CI
 IT IS USED FOR FORECASTING AND AUTOMATICALLY SCHEDULING
 ACTIVITIES IN THE CONVETER AISLE

 LOGICAL SENSW
 DIMENSION TASK(10),IN(5),TTC(9),KLD(9),KBL(9),MUL(5,8),KCVRT(5)
 COMMON /SAVE/ NSET(6,60),MODSCT(13)
 COMMON/BLCK1/PO(3),MT(3),LP(3),SP(3),POO(3),MTO(3),IPO(3),NC(3),NP
1(3),  JMT(3),  KMT(3),  ICRAN(3)
 COMMON /BLOK2/ ICAPA(3),  TLADA(3),  IBUFX(12),  IBUFY(12),
1  NHANK(3),  ISTRT(3),  LOCDEV(31),  INTNGT(17,2),  IPMA(17,2),
2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
3  NTANC,  ICNTPW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
*  ITROL(2),   JJ,   II,
4IDLF(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
5,IAPFL(3),IAWHL(2),NEVC(5),IADD(11,4),ICV(5,9)
6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
7,IHGS(5),IAPO(9,5),IAPO(9,19),MT(9,20),NCV,NTLAD(3),NANCST(3),
8 IATIME(3),NTPPW,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
9,NCPTT,LCPU,LRS,IPS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
*MOP(5,8),PTREQ(5),MTAV(2),IREP(5,6),MTOR(15),ICNO(15)
 COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3SMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),VMMAX(3),VMMIN(3),RRAT(3
4),TH(3),TLO(3),TH(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
6(3),TOOWN(3),TOL(3),TOM(3),CRL(3),CRM(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
 COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
1  NCRAN(12),  ICD(5),  MODUL
 DATA TASK(1),  TASK(2),  TASK(3),  TASK(4),  TASK(5),
*      TASK(6),  TASK(7),  TASK(8),  TASK(9),  TASK(10)
*      /4.60,  3.35, 4.60, 4.35, 0.25, 4.05, 6.0, 2.05, 5.0, 2.25/
* CONSP,PO    SL     MT     TH     CU     PG     SC     RS    FO     SK    LD

 DATA MUL(1,1),MUL(1,2),MUL(1,3),MUL(1,4),MUL(1,5),MUL(1,6),
1MUL(1,7),MUL(1,8)/4,3,2,2,1,1,1,1/
 DATA MUL(2,1),MUL(2,2),MUL(2,3),MUL(2,4),MUL(2,5),MUL(2,6),
1MUL(2,7),MUL(2,8)/4,3,2,2,1,1,1,1/
 DATA MUL(3,1),MUL(3,2),MUL(3,3),MUL(3,4),MUL(3,5),MUL(3,6),
1MUL(3,7),MUL(3,8)/4,3,2,2,1,1,1,1/
 DATA MUL(4,1),MUL(4,2),MUL(4,3),MUL(4,4),MUL(4,5),MUL(4,6),
1MUL(4,7),MUL(4,8)/4,3,2,2,1,1,1,1/
 DATA MUL(5,1),MUL(5,2),MUL(5,3),MUL(5,4),MUL(5,5),MUL(5,6),
1MUL(5,7),MUL(5,8)/5,4,3,3,2,1,1,1/


 LMW=15
 TTNOW=TIME-TLAST
 DO 500 KX=1,5
 IF(NEVC(KX).GE.9) NEVC(KX)=8
 NB1=NEVC(KX)
 NB2=NB1+1
 IPTS=ISTAT(KX)
 IF(8.LT.NB2) NB2=1
 IF(IPTS.GE.16) GO TO 400
 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) ,IPTS
 PAUSE222
 SP  NO MATTE REQ.FOR PRESENT BLOW ONLYES +CS
 CONTINUE
 SLAD=CV(KX,4)/FLOAT(IADD(10,1))
 TREQ(KX)=FLOAT(MT(KX,3)+MT(KX,13))*SLAD*TASK(1)
 IF(NB2.GE.8) GO TO 41
 MTREQ(KX)=MUL(KX,NB2)
 J   .500
 CS  MATTE REQ. ONLY WHEN STARTING NEW MATTE STAGE
 CONTINUE
 TT=TASK(5)*FLOAT(PT(KX,16)/MT(KX,15))
 CLAD=(CV(KX,5)+CV(KX,2)*.8)/FLOAT(IADD(11,1))
 TREQ(KX)=FLOAT(MT(KX,14)+MT(KX,2))*CLAD*TASK(4)+TASK(9)+TT
 MTREQ(KX)=MUL(KX,1)
      J   .500
C***** SB  REMOVE SLAG
C      ADDITION OF MT  OR TH  OR  REMOVAL OF TH
  3   CONTINUE
      SLAD=CV(KX,4)/FLOAT(IADD(10,1))
      TREQ(KX)=SLAD*TASK(1)+TTNOW
      MTREQ(KX)=MUL(KX,NB2)
      J   .500
C***** SB  GOING ON CU-BLOW , NO MATTE FOR LONG WHILE
  41  TT=TREQ(KX)
      J   .42
  4   CONTINUE
      TT=TTNOW
  42  SLAD=CV(KX,4)/FLOAT(IADD(10,1))
      CLAD=CV(KX,2)*.8/FLOAT(IADD(11,1))
      RCUS=.043*CV(KX,6)*CV(KX,7)
      TREQ(KX)=CV(KX,2)/RCUS+TASK(8)+CLAD*TASK(4)+TASK(9)+SLAD*TASK(1)
     *+TT
      MTREQ(KX)=MUL(KX,1)
      J   .500
C***** END OF CS - REMOVE CU
  5   CONTINUE
      TT=TTNOW
  51  CLAD=CV(KX,5)/FLOAT(IADD(11,1))
      TREQ(KX)=CLAD*TASK(4)+TT
      MTREQ(KX)=MUL(KX,1)
      J   .500
C***** SB - MATTE REQUIRED IMMEDIATELY
  6   CONTINUE
      TT=0.
  63  TREQ(KX)=TTNOW+TT
      MTREQ(KX)=MUL(KX,NB1)-IAPO(KX,10)/100
      IF(MTREQ(KX).LE.0) MTREQ(KX)=0
      J   .500
C***** FO REQUIRED
  7   CONTINUE
      CLAD=CV(KX,2)*.8/FLOAT(IADD(11,1))
      RCUS=.043*CV(KX,6)*CV(KX,7)
      TREQ(KX)=CV(KX,2)/RCUS+CLAD*TASK(4)+TASK(8)+TTNOW+TASK(9)
      MTREQ(KX)=MUL(KX,1)
      J   .500
C***** CS -SK REQUIRED
  8   CONTINUE
      TT=TASK(9)+TTNOW
      J   .51
C***** SS - REMOVE TH
  9   CONTINUE
      J   .6
C***** SS - ADD TH
  10 ,CONTINUE
      TREQ(KX)=TTNOW+400.
      MTREQ(KX)=0
      J   .500
C***** CS - ADD PIG
  11  CONTINUE
      J   .2
C***** PULL COLLAR
  12  CONTINUE
      IPTS=NSTAT(NCV)
      J   .399
C***** REBRICKING
  13  CONTINUE
      TREQ(KX)=TTNOW+400.
      MTREQ(KX)=0
      J   .500
C****  ADD MT WITHOUT SKIMMING SLAG IF MUL(BLOW) ..GT. IAPO(10,BLOW)
  14  CONTINUE
      SLAD=CV(KX,4)/FLOAT(IADD(10,1)(
      TT=SLAD*TASK(1)
      J   .63
C****  SLAG CAN BE SKIPPED INLAST BLOW BEFORE GOING ON COPPER
  15  CONTINUE
      J   .1
 500  CONTINUE

C****  WRITE PRESENT CONVERTER STATUS AND FIRST MATTE REQUIREMENT

      IF(.NOT. SENSW(8)) GO TO 199
      WRITE(6,502) TTNOW
 502  FORMAT(10X,29HCONVERTER SCHEDULE AT TIME = ,F6.2//
     *10X,4 HCCNV.   STATUS  BLOW NO.    MT.REQ.   TIME REQ./)
      DO 505 I=1,5
      WRITE(6,503) I,ISTAT(I),NEVC(I),MTREQ(I),TREQ(I)
 503  FORMAT(10X,I3,5X,I3,5X,I3,8X,I3,6X,F7.2)
  505 CONTINUE

C******   WRITE PRESENT MATTE TUNNEL STATUS

  199 DO 19 I=1,2
      MTAV(I)=0
      TAV(I)=0
      IF(ITROL(I).NE.-1) GO TO 190
      TAV(I)=-TTNOW+FLOAT(NP(2))
C*****     THIS VALUE IS SUPPLIED BY MT - TUNNEL
      MTAV(I)=1
 190  IF(ITROL(I).EQ.1) MTAV(I)=1
      MTAV(I)=MTAV(I)+IFLOR(I)
      TAV(I)=TTNOW+TAV(I)
  19  CONTINUE
      IF(.NOT. SENSW(8)) GO TO 536
      WRITE(6,906)
 906  FORMAT(//30X,24HMT.    MT.AV.     TIME AV./)
      DO 510 I=1,2
 510  WRITE(6,907)I,MTAV(I),TAV(I)
 907  FORMAT(10X,I2,6X,I3,7X,F7.2)

C*****  DETERMINE EXPECTED MATTE REQUIREMENT TO THE END OF EACH CU-CYCLE
```

```
 536 DO 540 I=1,LMN
     TCOR(I)=0.
     ICNO(I)=0
     IBLOW(I)=0
 540 MTOR(I)=0
     IF(.NOT. SENSW(8)) GO TO 537
     WRITE(6,521)
 521 FORMAT(1X,50HCCNV.   T1   T2   T3   T4   T5   T6   T7   T8   T9/)
 537 LP=0
     DO 520 K=1,5
     TCEND(K)=0
     KT=(IAFO(K,1)+IAFO(K,2))/100
     CALLCVFORC(TREQ,MTREQ,NEVC,KT,TTC,KLD,KBL,TCEND,K,MUL,ISTAT(K),MT
     IF(.NOT. SENSW(8)) GO TO 538
     WRITE(6,522)K,TTC
 522 FORMAT(2X,I2,4X,9(F4.0,1X))
     WRITE(6,523) KLD,KBL
 523 FORMAT(2X,5HMATTE,9(I3,2X)/2X,5HBLOW ,9(I3,2X)/)
 538 DO 527 MKR=1,9
     IF((TTC(MKR)-TTNOW).GE.71.) GO TO 524
     IF(TTC(MKR).LT..1.AND.TTNOW.NE.0.) GO TO 524
     IF(ISTAT(K).EQ.13) GO TO 524
     LP=LP+1.
     IF(LMN.LT.LP) GO TO 524
     MTOR(LP)=KLD(MKR)
     TCOR(LP)=TTC(MKR)
     IBLOW(LP)=KBL(MKR)
     ICNO(LP)=K
 527 CONTINUE
 524 CONTINUE
 520 CONTINUE
     IF(LMN.LT.LP)WRITE(1,519) LP
 519 FORMAT(1X,8HLP IN 11,I7/)

C**** SCHEDULE REQUIREMENTS WITHIN THE NEXT HOUR

     CALL TCSORT(TCOR,ITCOR,LMN,-1)
     DO 550 I=1,LMN
     LOS=ITCOR(I)
 550 IF(TCOR(LOS).GE..001)WRITE(6,551) TCOR(LOS),ITCOR(I),ICNO(LOS),
    *MTOR(LOS)
 551 FORMAT(5X,F6.2,5X,3I6)

C**** WRITE EXPECTED START OF CU-STAGES

     IF(.NOT. SENSW(8)) GO TO 539
     WRITE(6,526)TCEND
 526 FORMAT(1X,/5F5.0)

C**** DETERMINE DELAY OR ADVANCE OF COPPER BLOW

 539 LLL=0
 578 LLL=LLL+1
     CALL TCSORT(TCEND,IN,5,-1)
     DO 560 IRT=1,5
     IRTT=IRT+1
C**  TEST FOR I
     I=IN(IRT)
     IF(IRT.EQ.5) KCVRT(I)=0
     IF(ISTAT(I).NE.13) GC TO 559
     KCVRT(I)=0
     J    .560
C**  TEST FOR K
 559 K=IN(IRTT)
     IF(ISTAT(K).EQ.13) IRTT=IRT+2
     IF(5.LT.IRTT) GC TO 562
     K=IN(IRTT)
     T1=TCEND(I)
     T2=TCEND(K)
     KCVRT(I)=0
     IF((T1+100.).GE.T2) KCVRT(I)=-1
 560 CONTINUE
 562 CONTINUE
     IF(.NOT. SENSW(8)) GO TO 568
     WRITE(6,561) KCVRT
 561 FORMAT(5X,5I6)

C**** DELAY OR ADVANCE COPPER BLOW  -- SPECIFY TIME IN TCEND

 568 DO 570 IL=1,4
     I=IN(IL)
     IF(ISTAT(I).EQ.13) GO TO 570
     ILL=IL+1
     K=IN(ILL)
     IF(ISTAT(K).EQ.13) ILL=IL+2
     IF(ILL.GE.6) GO TO 571
     K=IN(ILL)
     IF(KCVRT(I).EQ.-1) TCEND(I)=TCEND(I)-20.
 570 CONTINUE
 571 CONTINUE
     IF(.NOT. SENSW(8)) GO TO 589
     WRITE(6,526)TCEND
 589 IF(LLL .LT. 5) GO TO 578
     MODUL=12
     RETURN
     END
```

```
      SUBROUTINE CVFORC(TREQ,MTREQ,NEVC,KT,TTC,KLD,KBL,TCEND,KX,MUL,ISS,
     *MT)
      DIMENSION NEVC(1),TTC(1),KLD(1),KBL(1),TCEND(1),MTREQ(1),TREQ(1)
      DIMENSION ITIM(9),MUL(5,8),MT(5,20)
      DATA ITIM(1),ITIM(2),ITIM(3),ITIM(4),ITIM(5),ITIM(6),ITIM(7),ITIM(
     *8),ITIM(9)/10,14,18,20,20,20,20,20,10/
      KTOTLD=KT
      DO 10 J=1,9
      TTC(J)=0.
      KBL(J)=0
 10   KLD(J)=0
      I=1
      IST=NEVC(KX)+2
      IF(ISS.EQ.13)   GO TO 83
      IF(ISS.EQ.6) IST=NEVC(KX)+1
      TTC(I)=TREQ(KX)
      KLD(I)=MTREQ(KX)
      KTOTLD=KTOTLD+KLD(I)
      IF(IST.LT.2) IST=2
      IF(IST.GE.10) IST=2
      KBL(1)=IST-1
      IF(IST.EQ.9) GO TO 64
      DO 70 IQTX=IST,8
      IQ=IQTX-1
      I=I+1
      J=I-1
      TTC(I)=FLOAT(KLD(J)*(ITIM(IQ)+5))+TTC(J)
      KLD(I)=MUL(KX,IQTX)
      KBL(I)=IQTX
      KTOTLD=KTOTLD+KLD(I)
      IF(I.EQ.2) GO TO 69
      J    .70
 69   IF(ISS.EQ.6.OR.ISS.EQ.14) TTC(2)=FLOAT(MT(KX,1)+MT(KX,15)+KLD(J)*(
     *ITIM(IQ)+5))
 70   CONTINUE
 65   J=I
 64   CONTINUE
      I=I+1
      TTC(I)=TTC(J)+FLOAT(KTOTLD*ITIM(9)+30)+FLOAT(KLD(J)*(ITIM(8)+5))
      KLD(I)=MUL(KX,1)
      KBL(I)=1
      TCEND(KX)=TTC(J)+FLOAT(KLD(J)*(ITIM(8)+5))
      RETURN
 83   TCEND(KX)=0.
      RETURN
      END
```

```
      SUBROUTINE TCSORT(A,I,N,M)
      DIMENSION A(1),I(1)
      LOGICAL MILO,TIME1
      MILO=M.LT.0
      N1=N+1
      N2=N1/2
      DO 10 J=1,N
 10   I(J)=-1
      DO 11 K=1,N2
      TIME1=.TRUE.
      DO 12 J=1,N
      IF(I(J).GT.0) GO TO 12
      IF(.NOT.TIME1) GO TO 14
      TIME1=.FALSE.
      BIG=A(J)
      SMALL=BIG
      JB=J
      JS=JB
      J    .12
 14   IF(A(J).GT.SMALL) GO TO 13
      SMALL=A(J)
      JS=J
      J    .12
 13   IF(A(J).LT.BIG) GO TO 12
      BIG=A(J)
      JB=J
 12   CONTINUE
      L=N1-K
      I(JB)=IABS(I(JB))
      I(JS)=IABS(I(JS))
      IF(MILO) GO TO 15
      I(L)=ISIGN(JS,I(L))
      I(K)=ISIGN(JB,I(K))
      J    .11
 15   I(L)=ISIGN(JB,I(L))
      I(K)=ISIGN(JS,I(K))
 11   CONTINUE
      RETURN
      E N D
```

TWELFTH SUBMODULE

```
C     DFCK FOR TWELFTH SUBMODULE
      SUBROUTINE DISK
C
C      THIS IS THE TWELFTH  SUBMODULE  ITS FILE NAME IS UP12CI
C     CONTINUATION OF MODULE 11
C
      LOGICAL SENSW
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(5),  NAPL,  NOIL(3),  IPLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFPL(3),TAWHL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRFO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
     *MOP(5,8),MTREQ(5),MTAV(2),IREP(5,6),MTOR(15),ICNO(15)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),VMMAX(3),VMMIN(3),RRAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RAT(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TOOWN(3),TOL(3),TOM(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CY(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCCNT(5)
      COMMON /ARG/ IX,  ID1X,  ID2X,  ISET,  JSEQ(12),  JCMND(12),
     1  NCRAN(12),  ICO(5),  MODUL
      DIMENSION TSTORE(2)
C*****  DETERMINE MATTE ALLOCATICN

      LMN=15
      TTNOW=TIME(TLAST)
      MTT=MTAV(1)+MTAV(2)
      TSTRT=TTNOW
      TSTORE(2)=0.
      TSTORE(1)=0.
      LMNO=LMN-1
      DO 580 IB=1,LMNC
      I=ITCOR(IB)
      K=ITCOR(IB+1)
      IF(TCOR(I).LT.0.01) GO TO 580
      IF(I.LT.1.OR.LMN.LT.I) WRITE(6,585)I
      IF(K.LT.1.OR.LMN.LT.K) WRITE(6,585)K
  585 FORMAT(1X,15HER.AT 585 IN 11,I6/)
      MT1=MTOR(I)
      MT2=MTOR(K)
      NOC1=ICNO(I)
      NOC2=ICNO(K)
      IF(NOC1.LT.1.OR.5.LT.NOC1) WRITE(6,587)NOC1
      IF(NOC2.LT.1.OR.5.LT.NOC2) WRITE(6,587)NOC2
  587 FORMAT(1X,15HER.AT 587 IN 11,I6/)
      IBL1=IBLOW(I)
      IBL2=IBLOW(K)
      IF(IBL1.LT.1.OR.8.LT.IBL1) WRITE(6,588)IBL1
      IF(IBL2.LT.1.OR.8.LT.IBL2) WRITE(6,588)IBL2
  588 FORMAT(1X,15HER.AT 588 IN 11,I6/)
      IF(8.LT.IBL1) IBL1=1
      IF(8.LT.IBL2) IBL2=1
      T1=TCOR(I)
      CALL  MTFORC(T1,TSTRT,MTAV,TAV,MTT,TSTORE)

      IF(.NOT. SENSW(8)) GO TO 598
      WRITE(6,677) MTT,MT1,MT2,TSTRT,T1
  677 FORMAT(1X,3I6,5X,2F6.0/)

  598 TSTRT=T1
      IF(MT1.LE.0) GC TO 586
      IF(MTT.GE.(MT1+MT2)) GO TO 586
      IF(TCOR(K).GE.(TCOR(I)+20.)) GO TO 586
      IF(MTT.GE.MT1.AND.MT2.LE.1) GO TO 586
      IF(IBL1.EC.1) GO TO 589
      MT1=MT1/2
      J    .586
  589 IF(3.LT.MTT) MT1=3
  586 IF(MTT.LE.MT1) MT1=MTT
      MOP(NOC1,IBL1)=MT1
      MTT=MTT-MT1
  580 CONTINUE

C*****   WRITE PROPOSED MATTE ALLOCATION

      IF(.NOT. SENSW(8)) GO TO 599
      WRITE(6,590)((MOP(I,M),I=1,5),M=1,8)
  590 FORMAT(1X,8(5I6/)/)
  599 MODUL=0
      RETURN
      END
      SUBROUTINE MTFORC(TPC,TSTRT,MTAV,TAV,MTT,TSTORE)
      DIMENSION MTAV(2),TAV(2),ARRTM(2),TSTORE(2)
      DATA ARRTM(1),ARRTM(2)/12.,20./
      M=0
      DO 10 I=1,2
      T=TAV(I)
      IF(TSTORE(I).GE..001) TSTRT=TSTORE(I)
      IF(TSTRT.GE.T) T=TSTRT
      MAT=(TPC-T)/(ARRTM(I)+5.)
C  **  ARRTM + 5   =TIME BETWEEN ARRIVALS OF MATTE AT MT
      IF(MAT.LT.0) MAT=0
      TSTORE(I)=0.
      IF(MAT.EQ.0) TSTORE(I)=TSTRT
   10 M=M+MAT
      MTT=MTT+M
      RETURN
      END
      SUBROUTINE ANTIME(N,A,B)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  IWTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANG,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IAWHL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRFO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),BRAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(5),TOL(3),TOH(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCCNT(5)
      ATRIB(1)=TIME(TLAST)+.01*FLOAT(ILADA(N-3))*UNFRM(A,B)
      RETURN
      END
```

```
      SUBROUTINE LOCFND(IPOS,I,IEND,IER)
      DIMENSION LOC(31)
      DATA  LOC(1),  LOC(2),  LOC(3),  LOC(4),  LOC(5),  LOC(6),
     1  LOC(7),  LOC(8),  LOC(9),  LOC(10),  LOC(11),  LOC(12),
     2  LOC(13),  LOC(14),  LOC(15),  LOC(16),  LOC(17),  LOC(18),
     3  LOC(19),  LOC(20),  LOC(21),  LOC(22),  LOC(23),  LOC(24),
     5  LOC(25),  LOC(26),  LOC(27),  LOC(28),  LOC(29),  LOC(30),
     *  LOC(31)
     6  / 2HR1,  2HR2,  2HR3,  2HA4,  2HA5,  2HA6,
     7  2HM1,  2HM2,  2HC3,  2HC4,  2HC5,  2HC6,
     8  2HCT,  2HF3,  2HF4,  2HF5,  2HF6,  2HFT,
     9  2HPP,  2HPB,  2HBB,  2HP1,  2HCP,  2HNE,
     *  2HSE,  2HK1,  2HK2,  2HK3,  2HGP,  2HAL,
     *  2HPT/
      IER=0
      DO 1 I=1,IEND
      IF(IPOS .EQ. LOC(I)) GO TO 2
    1 CONTINUE
      IER=1
    2 RETURN
      END
```

```
      SUBROUTINE COCK(NEXT,LL,LM,IER)
      IER=0
      IF(NEXT .LT. LL) GO TO 1
      IF(NEXT .GT. LM) GO TO 1
    2 RETURN
    1 PAUSE 1
      IER=1
      J    .2
      END
```

```
      SUBROUTINE FIND(XVAL,JQ,JATT,KCOL,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      NVAL=XVAL*SCALE+.01
      KBEST=0
      NEXTK=MFE(JQ)
      IF(NEXTK) 14,1,20
   14 CALL ERROR(89,NSET)
   16 PAUSE 76
    1 KCOL=KBEST
    2 RETURN
   20 IF(NSET(JATT,NEXTK)-NVAL) 4,19,4
    4 NEXTK=NSET(MX,NEXTK)
   16 IF(NEXTK-7777) 20,1,1
   19 KCOL=NEXTK
      J    .2
      END
```

```
*************
      SUBROUTINE GETEV(NSET)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
```

```
     *,ISTRIB(5)
      COMMON/BLCK1/POL(3),NT(3),IP(3),SP(3),POP(3),NTO(3),IPO(3),NC(3),N
     1I(3),  JNT(3),  KNT(3),  TCRAN(3)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  IWTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANG,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IAWHL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRFO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLCK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,NRE
      NFILE=NCR+1
      CALL RMOVE(MFE(NFILE),NFILE,NSET)
C     SCALE IN GASP MUST BE GREATER THAN 10.0 TO RETAIN DECIMAL FIGURE
      POS=ATRIB(1)
      TW=.1*ATRIB(2)
C     PRIORITY HAS GOT VALUES BETWEEN 0 - 10
      LSTR(NCR)=ATRIB(3)+.01
      NDIFF=LSTR(NCR)*10
      IPR=ATRIB(3)*10.0 -FLOAT(NDIFF)+.01
C     EVENTS HAVE VALUES BETWEEN 1 - 999
      KODE=ATRIB(4)/100.0
      DIFF=1000.*FLOAT(KODE)
      ISTRIB(NFILE)=KODE
      NXEVT=ATRIB(4)*10.0 -DIFF+.01
      RETURN
      END
```

```
      SUBROUTINE SLTEST(KCMPAR)
      LOGICAL LOGVAL,SENSW
      DIMENSION KCMPAR(6)
      COMMON /BLCK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NFRNC,NRE
      DO 3 I=1,10
    3 IS(I)=0
      CALL ORCPL(KOMPAR(6),LOGVAL,IER)
      IF(LOGVAL) IS(1)=1
      DO 1 I=1,3
      CALL ORSLL(I,LOGVAL,IER)
      CALL ORSLL(I,LOGVAL,IER)
      J=I+1
      IF(LOGVAL) IS(J)=J
    1 CONTINUE
      DO 2 I=1,5
      J=I+4
      CALL ORCPL(KCMPAR(I),LOGVAL,IER)
      IF(.NOT. LOGVAL) IS(J)=J
    2 CONTINUE
      IF(SENSW(1)) IS(10)=10
      RETURN
      END
```

```
      SUBROUTINE TESTPL(NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEE ,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
  550 IF(MFE(1) .LT.1) RETURN
      KXXX=MFE(1)
      TTEST=FLOAT(NSET(1,KXXX))/SCALE+.001
      IF(TTEST.LT.TNOW) GO TO 600
    1 RETURN
  600 ATRIB(1)=TNOW
      ATRIB(2)=JEVNT
      ATRIB(3)=ISTRIB(1)
      ATRIB(4)=ISTRIB(5)
      CALL FILEM(1,NSET)
      CALL RMOVE(KXXX,1,NSET)
      CALL SETIME(NSET)
      J    .1
      END
```

```
      SUBROUTINE SETIME(NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  IWTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANG,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IAWHL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,IHGS(5),IRFO(5,5),IAFO(5,15),MT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATE(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TOL(3),TON(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
      IER=0
 511  TNOW = ATRIB(1)
      JEVNT = ATRIB(2)
      ISTRIB(1)=ATRIB(3)
      ISTRIB(5)=ATRIB(4)
C*****SET COMPCNENTS FOR TIME CIRCUIT
 400  TREL=TNOW-TLAST
      IF(TREL.LE.C.0) TREL=0.05
      IF(TREL .LT. 0.05) TREL=.05
      IF(TIMEX .LT. TREL) PAUSE 11
      IF(TREL .GT. TIMEX)  PAUSE 11
      TREL=TREL/TIMEX
      CALL CONTRL(5,0)
      CALL QWJDAR(TREL,9,IER)
      CALL QSOLY(500)
      CALL CONTRL(5,1)
      RETURN
      END



      SUBROUTINE CRA1(PCS,TW,IPR,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),ENN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),   IWTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IAWWL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
     6(3),TDOWN(3),TOL(3),TON(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HCURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)

      CALL LIGHT(37,1)
      POO(1)=PO(1)
      WTO(1)=WT(1)
      IPO(1)=IP(1)
      IF(POS.LT.0.0) POS=0.0
      XM2W=XMAX-2.0*W
      IF(POS.GT.(XMAX-2.0*W))  POS=XMAX-2.0*W
      IF(XM2W .LT. PGS) POS=XM2W
      PO(1)=POS
      WT(1)=TW
      IP(1)=IPR
      JJ=0
C     CHECK INTERFERENCE WITH CRANE 3
      IF((PO(3)-W).GE.PO(1)) GO TO 99
      IF(IP(3).EQ.0)          GO TO 71
      CALL ORBACR(T3,9,1,IER)
      T3=T3*TMAX
      IF(IP(1) .LE. IP(3)) GO TO 82
      IF((WT(3)-T3).LE.TCL)  GO TO 81
C     CRANE 3 IS MOVED IF CRANE 2 ALLOWS THIS
 80   WAIT=WT(1)
 81   JJ=2
      CALL CRA3(PO(1)+W,WAIT,IP(1),NSET)
      JJ=3
      IF(II.EQ.2)    GO TO 98
C     CRANE 3 IS DELAYED AT POS1+W
C     FILE EVENT AFTER TWS WHEN CRANE 3 COMPLETES ITS TASK
 97   IF(IP(3).EQ.0) GO TO 99
      IP(3)=IP(3)+1
      IF(IP(3).GT.9) IP(3)=9
      IF( 9 .LT. IP(3)) IP(3)=9
      CALL CFILE(3,POO(3),WTO(3)-T3,IPO(3),LSTR(3),ISTRIB(4),
     1 NEXT(3),1,NSET)
      JWT(3)=1
      IP(3)=0
      J    .99
 98   JWT(1)=2
 96   IP(1)=IP(1)+1
      IF(IP(1).GT.9) IP(1)=9
      IF( 9 .LT. IP(1)) IP(1)=9
      CALL CFILE(1,PO(1),WT(1),IP(1),LSTR(1),ISTRIB(2),NEXT(1),1,NSET)
 83   CALL CONT(1,PO(3)-W,.1)
      IP(1)=0
      J    .1
 71   WAIT=0.1
      J    .61
 81   IF(T3.LT.0.C6) GC TO 80
 82   JWT(1)=3
      J    .96
C     SET ANALCG COMPONENTS OF CRANE MODELS BY CALLING CONT(NC*,POS,TW)
 99   CALL CONT(1,PO(1),WT(1))
    1 RETURN
      END



      SUBROUTINE CRA2(POS,TW,IPR,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),   IWTNGT(17,2),  IPRA(17,2),
     2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRN,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IPLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAFFL(3),IAWWL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),PTREQ(5),NTAV(2),IREP(5,6)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCCOL,
     1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAW(2),TOTW(2),TRAIL(2),RATI(3),SLEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TCTSL(3),RSLG(3),SLRA(3),PCT
     6(3),TDOWN(3),TOL(3),TON(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
     7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCCNT(5)
C
      CALL LIGHT(38,1)
      POO(2)=PO(2)
      WTO(2)=WT(2)
      IPO(2)=IP(2)
C     IF(POS.GT.XMAX)  POS=XMAX
      IF(XMAX .LT. PCS) POS=XMAX
      X2W=2.0*W
      IF(POS.LT.X2W)    POS=X2W
      PO(2)=POS
      WT(2)=TW
      IP(2)=IPR
      JJ=0
C     CHECK INTERFERENCE WITH CRANE 3
      IF((PO(3)+W).LE.PO(2)) GO TO 99
      IF(IP(3).EQ.0)          GO TO 71
      CALL ORBACR(T3,9,1,IER)
      T3=T3*TMAX
      IF(IP(2) .LE. IP(3)) GO TO 82
      IF((WT(3)-T3).LE.TOL)  GO TO 81
 80   WAIT=WT(2)
 61   JJ=1
      CALL CRA3(PO(2)-W,WAIT,IP(2),NSET)
      JJ=0
      IF(II.EQ.1)  GC TO 98
 97   IF(IP(3).EQ.0) GO TO 99
      IP(3)=IP(3)+1
C     IF(IP(3).GT.9) IP(3)=9
      IF( 9 .LT. IP(3)) IP(3)=9
      CALL CFILE(3,PCO(3),WTO(3)-T3,IPC(3),LSTR(3),ISTRIB(4),
    1 NEXT(3),1,NSET)
      JWT(3)=2
      IP(3)=0
      J    .99
 98   JWT(2)=1
 96   IP(2)=IP(2)+1
C     IF(IP(2).GT.9) IP(2)=9
      IF( 9 .LT. IP(2)) IP(2)=9
      CALL CFILE(2,PO(2),WT(2),IP(2),LSTR(2),ISTRIB(3),NEXT(2),1,NSET)
 83   CALL CONT(2,PO(3)+W,.1)
      IP(2)=0
      J    .1
 71   WAIT=0.1
      J    .61
 81   IF(T3.LT.0.C6) GO TO 80
 82   JWT(2)=3
      J    .96
C     SET ANALOG COMPONENTS OF CRANE MODELS BY CALLING CONT(NC*,POS,TW)
 99   CALL CONT(2,PO(2),WT(2))
    1 RETURN
      END



      SUBROUTINE CRA3(POS,TW,IPR,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
     *,ISTRIB(5)
      COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
     1(3),  JWT(3),  KWT(3),  TCRAN(3)
      COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),   IWTNGT(17,2),  IPRA(17,2),
2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
*   ITROL(2),   JJ,   II,
4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
5,IAFFL(3),IAWHL(2),NEVC(5),IADD(11,4),ICV(5,9)
6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
7,IHGS(5),IRFO(5,5),IAFO(5,15),NT(5,20),NCY,NTLAD(3),NANCST(3),
8 IATIME(3),NTPPR,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2)
9,NCPTT,LCPU,LAS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
*NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
 COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
41,TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RATI(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
6(3),TDOWN(3),TOL(3),TOM(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)

 CALL LIGHT(41,1)
 POO(3)=PO(3)
 WTO(3)=WT(3)
 IPO(3)=IP(3)
 XMIW=XMAX-W
 IF(POS.GT.(XMAX-W))  POS=XMAX-W
 IF(XMIW .LT. PCS) POS=XMIW
 IF(POS.LT.W) POS=W
 PO(3)=POS
 WT(3)=TW
 IP(3)=IPR
10  II=0
 IF(JJ.EQ.1.OR.JJ.EQ.2) GO TO 8
 IF(JJ .EQ. C) GO TO 9
 PAUSE 41
8  IF(JJ-1) 9,1,2
 IS CRANE 1  OR CRANE 2  APOSSIBLE INTERFERENCE
9  IF(PO(3).GE.POO(3)) GO TO 2
 TEST INTERFERENCE WITH CRANE 1
1  IF(PO(3).GE.(PO(1)+W))  GO TO 99
 IF(IP(1).EQ.0)           GO TO 71
 CALL ORBADR(T1,3,1,IER)
 T1=T1*TMAX
 IF(IP(3) .LE. IP(1)) GO TO 84
 IF((WT(1)-T1).LE.TOL)  GO TO 81
61  IP(1)=IP(1)+1
 IF(IP(1).GT.9) IP(1)=9
 IF(9 .LT. IP(1)) IP(1)=9
 CALLCFILE(1,PO(1),WT(1)-T1,IP(1),LSTR(1),ISTRIB(2)NEXT(1),1,NSET)
 JWT(1)=3
 IP(1)=0
71  CALL GONT(1,PO(3)-W,0.100)
 J   .99
81  IF(T1.LT.0.06) GO TO 61
 IF(IP(3).GT.9) IP(3)=9
 IF( 9 .LT. IP(3)) IP(3)=9
 CALL CFILE(3,PO(3),WT(3),IP(3),LSTR(3),ISTRIB(4),NEXT(3),1,NSET)
51 CALL GONT(3,PPP,.1)
 IP(3)=0
 J   .50
98  II=1
52 PO(3)=POO(3)
 WT(3)=WTO(3)
 IP(3)=IPO(3)
 J   .50

 TEST INTERFERENCE WITH CRANE 2

2  IF(PO(3).LE.(PO(2)-W))  GO TO 99
 IF(IP(2).EQ.0)           GO TO 72
 CALL ORBADR(T2,4,1,IER)
 T2=T2*TMAX
 IF(IP(3) .LE. IP(2)) GO TO 83
 IF((WT(2)-T2).LE.TOL)   GO TO 82
62  IP(2)=IP(2)+1
 IF(9.LT.IP(2)) IP(2)=9
 CALLCFILE(2,PO(2),WT(2)-T2,IP(2),LSTR(2),ISTRIB(3),NEXT(2),1,NSET)
 JWT(2)=3
 IP(2)=0
72 CALL GONT(2,PO(3)+W,.1)
 J   .99
82  IF(T2.LT.0.06) GO TO 62
83 IF(JJ .EQ. 2) GO TO 98
 JWT(3)=2
 PPP=PO(2)-W
 J   .85
90  II=2
 J   .52
 SET ANALCG COMPONENTS OF CRANE MODELS BY CALLING GONT(NCR,POS,TW)
99  CALL GONT(3,PO(3),WT(3))
 IF(IPO(3).EQ.0.AND.JJ.GT.0) IP(3)=0
 II=3
50 RETURN
 END



 SUBROUTINE GONT(NCR,PPOS,TTW)
 LOGICAL LOGVAL,L1
 COMMON/BLCK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
1(3),  JWT(3),  KWT(3),  TCRAN(3)
```

```
 COMMON /BLCK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RATI(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
6(3),TDOWN(3),TOL(3),TOM(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
 COMMON /BLCK4/ NCR,  POS,  IPR,  TW,  KDJE,  NXEVT,  IS(10),
* LOGVAL,NFRAC,NRE
C
 NCR=NCR
 POS=PPOS
 TW=TTW
 IF(TW.LE.0.C5) TW=0.05
C IF(TW.GT.TMAX) TW=TMAX
 IF( TMAX .LT. TW) TW=TMAX
 CALL ORBACR(XCR,NCR-1,1,IER)
 PO(NCR)=PCS
 WT(NCR)=TW
 TW=TW/TMAX
 POS=POS/XMAX
 I=3*NCR-2
 CALL CONTRL(NCR+1,0)
 CALL QWJOAR(POS,I,IER)
 CALL QWJOAR(TW,I+1,IER)
 S=XCR-POS
 K1=S*NCR-1
C IF(S .GT. .CO12) GO TO 1
 IF(.0012 .LT. S) GO TO 1
 IF(S .LT. -.0012) GO TO 2
4 CALL CONTRL(NCR+1,1)
99  RETURN
1 L1=.TRUE.
 J   .3
2 L1=.FALSE.
3 CALL QWFRL(K1,L1,IER)
 J   .4
 END



 FUNCTION TIME(TLAST)
 COMMON/BLOK1/PO(3),WT(3),IP(3),SP(3),POO(3),WTO(3),IPO(3),NC(3),NP
1(3),  JWT(3),  KWT(3),  TCRAN(3)
 COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
1  NRANK(3),  ISTRT(3),  LOCDEV(3)1,  INTNCT(17,2),  IPMA(17,2),
2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
*   ITROL(2),   JJ,   II,
4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
9,NCPTT,LCPU,LAS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
*NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
 COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCCCL,
1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3XMAX,BETA,TMAX,TIMEX,TLASS,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RATI(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
6(3),TDOWN(3),TOL(3),TOM(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCCNT(5)
 CALL ORBADR(TAM,6,1,IER)
 TIME=TLAST+TAM*TIMEX
 RETURN
 END



 SUBROUTINE CHAI(NSET)
 LOGICAL LOGVAL
 COMMON /BLCK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
1  NRANK(3),  ISTRT(3),  LOCDEV(3)1,  INTNCT(17,2),  IPMA(17,2),
2  NLC,  NFL,  NSKL,  NCOOL,  NPPB,  NCOL*, ,NPPP,  NCTOT,  NCPPL,
3, NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
*   ITROL(2),   JJ,   II,
4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
5,IAFFL(3),IAWHL(2),NEVC(5),IADD(11,4),ICV(5,9)
6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
7,IHGS(5),IRFO(5,5),IAFO(5,15),NT(5,20),NCY,NTLAD(3),NANCST(3),
8 IATIME(3),NTPPR,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2)
9,NCPTT,LCPU,LAS,IRS,IPRIT(5),INHOUR,NACOL(3),ITCOR(15),IBLCH(15),
*NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
 COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
1  TAF6A,  TAF6B,  TAF5A,  TAF5B,  TAF4A,  TAF4B,  TCST4A,  TCST4B,
2  TCST5A,  TCST5B,  TAFR4A,  TAFR4B,  TAFR5A,  TAFR5B,
3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,RMLEV(3),YMMAX(3),YMMIN(3),RMAT(3
4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),RATI(3),SLEV
5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLG(3),SLRA(3),POT
6(3),TDOWN(3),TOL(3),TOM(3),CRL(3),CRH(3),TCST6A,TCST6B,TAFR6A,
7 TAFR6B,HOURS,TSLAG(3),CV(5,9),TPU(5)
8,TREQ(5),TAV(2),TCEND(5),TCOR(15),TCONT(5)
 COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KDJE,  NXEVT,  IS(10),
* LOGVAL,NFRAC,NRE
 CALL CHECK(IER)
 IF( IER .EQ. 1) RETURN
 IF(KAPUT(NCR).EQ.1) GO TO 20
 NRANK(NCR)=NRANK(NCR)+10
 IF(NRANK(NCR) .LT. 10) NRANK(NCR)=10
```

```
      CALL CFILE(NCR,PCS,TW,IPR,NRANK(NCR),KODE,NXEVT,0,NSET)
 2    RETURN
20    TYPE 1004 ,NCR,TDOWN(NCR)
1004  FORMAT(1X,3HCRN,I3,9HDOWN TILL,F8.2)
      J    .2
      END
```

```
      SUBROUTINE CHECK(IER)
      LOGICAL LOGVAL
      COMMON /BLOK4/ NCR,  POS,  IPR,  TW,  KODE,  NXEVT,  IS(10),
     * LOGVAL,NPRNC,NRE
      IF(IPR .GT. 9) IPR=9
      IF( 9 .LT. IPR) IPR=9
      IF(IPR .LT. 0) IPR=0
      IER=0
      IF(NCR .GT. 3) GO TO 1
      IF( 3 .LT. NCR) GO TO 1
      IF( NCR .LT. 1) GO TO 1
      J    .2
    1 PAUSE 3
      IER=1
    2 RETURN
      END
```

```
      SUBROUTINE LADLE
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNST(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPPL(3),IANNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INNDUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      COMMON /ANG/ IX,IDIX,IDZX,ISET
      NLC=NLC+1
      IF(NLC .LT. NPL) RETURN
      CALL LIGHT(3,1)
      IJK=NLC-NPL*(NLC/NPL)
    1 IF(IJK .EQ. 0) ISET=ISET+5000
      IJK=IJK-NPL
      IF(IJK .LT. 0) RETURN
      J    .1
```

```
      SUBROUTINE LIGHT(N,M)
      LOGICAL SENSW
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNST(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPPL(3),IANNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INNDUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      IF(SENSW(2)) GO TO 4
      IF(M-1) 1,2,3
    1 WRITE(6,100) N
      J    .4
    2 WRITE(6,101) N
      J    .4
    3 WRITE(6,102)
    4 CONTINUE
C RESET CONTROL LINES 0,1, 8 TO 15
      LA   ICNTRW
      AND  /'37400
      OCT  9042
      STA  ICNTRW
C CALC DIGITS
      IT = (N/10)
      IU = N-10*IT
      IT = IT*16
C SET BITS 0,1
      LA   /'2
      S    M
      OCT  27407
      LA   /'3
      OCT  26216
C MAKE UP CONTROL LINES CONTROL WORD
      OR   IT
      OR   IU
      OR   ICNTRW
      OCT  9042
      RETURN
100   FORMAT(1X,I3,6HT. OFF)
101   FORMAT(1X,I3,5HT. ON)
102   FORMAT(1X,13HLIGHTS RESET)
      END
```

```
      SUBROUTINE CONTRL(NCNTRL,N)
C  RESETS,SETS CONTROL LINE NCNTRL IF N=0,1
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNST(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPPL(3),IANNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INNDUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
C  ENSURE 0<NCNTRL<15
  10  LA   NCNTRL
      AND  /'17
C  BUILD SHIFT CMD
      TCA
      A    /'26217
      STA  .20
C  PRODUCE MASK FOR OPERATIONS
      OCT  20000
  20  OCT  777777
      STA  MASK
      OCA
      STA  NMASK
      LA   N
      OCT  24100
      J    .30
      LA   ICNTRW
      AND  NMASK
      STA  ICNTRW
      J    .40
  30  LA   ICNTRW
      OR   MASK
      STA  ICNTRW
  40  OCT  9042
  50  OCT  27400
      RETURN
      END
```

```
      SUBROUTINE BELLS(N,L)
      LOGICAL SENSW
      INTEGER BELL,OUT(4)
      DATA BELL,OUT(1),OUT(3),OUT(4),IN /'103640,2H( ,2HA1,2H) ,2H00/
      OUT(2)=IN+N
      TYPE OUT,(BELL,I=1,N)
      TYPE 100,L
      IF(SENSW(7)) GO TO 1
      ACCEPT 101, IER
    1 RETURN
  101 FORMAT(I1)
  100 FORMAT(1X,5HEVENT,I3/)
      END
```

```
      SUBROUTINE CPILE(NCR,POS,TW,IPR,NRANK,KODE,NXEVT,INPUT,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IA,INIT,JEVNT,NPA,NSTOP,NN,NNC,NOB,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),NRANK(4),NPE(4),NLC(4),NLE(4),NQ(4)
     0,ISTATS(9)
      COMMON/BLOCK1/PB(3),UT(3),IP(3),SP(3),POB(3),UTD(3),IPO(3),NC(3),NP
     1(3),  JUT(3),  KDT(3),  TCRAM(3)
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTNST(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPB,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANC,  ICNTRW,  IER,  ISAVE(3),  NAPL,  NOIL(3),  IFLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPPL(3),IANNL(2),NEVC(5),IADD(11,4),ICV(5,9)
     6,ISTAT(5),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INGS(5),IRPO(5,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPB,NCPT,IRESL,IPTCU,IPK(6,2),IDP(6,2)
     9,NCPTT,LCPU,LRS,IRS,IPRIT(5),INNDUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),NTREQ(5),NTAV(2),IREP(5,6)
      IF(INPUT .EQ. 1) NRANK=NRANK-1
      ATRIB(1)=POS
      ATRIB(2)=10.*TW+.00
C     SCALE IN GASP MUST BE GREATER THAN 10.0 TO RETAIN DECIMAL FIGURE
      ATRIB(3)=.1*(FLOAT(IPR)+.01)+FLOAT(NRANK)
      ATRIB(4)=.1*(FLOAT(NXEVT)+.01)+FLOAT(KODE*100)
      CALL FILEM(NCR+1,NSET)
      IF(INPUT .EQ. 1) NEXT(NCR)=0
      RETURN
      END
```

```
      FUNCTION RNORM(RMEAN,STDV)
C     ISEED IS COMMON
      COMMON ID,IA,INIT,JEVNT,NPA,NSTOP,NN,NNC,NOB,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),NRANK(4),NPE(4),NLC(4),NLE(4),NQ(4)
```

```
*,ISTRIB(9)
 A=0.0
 DO 1 I=1,12
 S=DRAND(ISEED)
1  A=A+S
 RNORM=(A-6.)*STDV+XMEAN
 RETURN
 END
```

```
 FUNCTION UNFRM(A,B)
 COMMON IO,IF,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
 COMMON ATRIB(4),ENQ(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
 UNFRM=A+(B-A)*DRAND(ISEED)
 RETURN
 END
```

```
  FUNCTION DRAND(IX)
  LA  IX
  M   /259
  OCT 24100
  J   .1
  OCT 26540
  J   .2
1 OCT 26540
  OCT 26440
  OCT 20100
2 STA IX
6 YFL=IX
  DRAND=YFL*.3051758E-4
  RETURN
  END
  SUBROUTINE EXEC(N)
  LOGICAL SENSW
  COMMON /SAVE/ NSET(6,60),MODSCT(13)
  DATA LAST /0/
  IF( .NOT. SENSW(2)) TYPE 100, N
C    CHECK TO SEE IF MODULE IS ALREADY LOADED
  IF(N .NE. LAST) GO TO 2
3 CALL DISK
  RETURN
2 LAST=N
C    LOAD ZONE ZERO FROM 0 TO 400
C    LOAD PROGRAM
  CALL FETCH(MODSCT(N),ISNXT)
  J   .3
100 FORMAT(I3/)
  END
```

```
      SUBROUTINE CLOCK(ITIME)
C      SUBROUTINE TO FIND THE PRESENT TIME IN HOURS AND MINUTES
C      IN FORMAT   HHMM   GIVEN THAT THE SIMULATION BEGAN AT INHOUR .
      COMMON /BLOK2/ ICAPA(3),  ILADA(3),  IBUFX(12),  IBUFY(12),
     1  NRANK(3),  ISTRT(3),   LOCDEV(31),  INTMST(17,2),  IPRA(17,2),
     2  NLC,  NPL,  NSKL,  NCOOL,  NPPS,  NCOLP,  NPPP,  NCTOT,  NCPPL,
     3  NTANK,  ICNTRN,  IER,  ISAVE(9),  NAPL,  NOIL(3),  IPLOR(2),
     *   ITROL(2),   JJ,   II,
     4IDLE(3),KAPUT(3),NEXT(3),LSTR(3),NSLAG(3)
     5,IAPFL(3),IANNL(2),NEVC(5),IADD(11,4),ICV(9,9)
     6,ISTAT(9),JSTAT(5),NSTAT(5),NAPLC(3)
     7,INQS(5),IRPO(9,5),IAPO(5,15),NT(5,20),NCV,NTLAD(3),NANCST(3),
     8 IATIME(3),NTPPS,NCPT,IRESL,IPTCU,IPK(6,2),IOP(6,2)
     9,NCPTT,LCPU,LAS,IRS,IPRIT(9),INHOUR,NACOL(3),ITCOR(15),IBLOW(15),
     *NOP(5,8),NTREQ(5),NTAV(2),INEP(9,6)
      COMMON /BLOK3/  TPIA,  TPIB,  TPISLA,  TPISLB,  TPCOOL,
     1  TAP6A,  TAP6B,  TAP5A,  TAP5B,  TAP4A,  TAP4B,  TCST4A,  TCST4B,
     2  TCST5A,  TCST5B,  TAPR4A,  TAPR4B,  TAPR5A,  TAPR5B,
     3XMAX,BETA,TMAX,TIMEX,TLAST,TOL,W,OMLEV(3),VMMAX(3),VMMIN(3),RMAT(3
     4),TR(3),TLO(3),THI(3),TOLA(3),TAM(2),TOTM(2),TRAIL(2),MATI(3),ELEV
     5(3),SMAX(3),SMIN(3),STAPL(3),STAPH(3),TOTSL(3),RSLB(3),SLRA(3),POT
     6(3),TDOWN(3),TDL(3),TDN(3),CRL(3),CRN(3),TCST6A,TCST6B,TAPR6A,
     7 TAPR6B,HOURS,TSLAG(3),CV(9,9),TPU(5)
     8,TREQ(5),TAV(2),TCEND(3),TCOR(15),TCONT(5)
      TTIME=TIME(TLAST)
      IF(TTIME .LT. 0.) TTIME=0.
      JHOUR=TTIME/60.
      MINUTE=TTIME-60.*FLOAT(JHOUR)+.5
      JHOUR=JHOUR+INHOUR
    2 IF(JHOUR .GT. 23) GO TO 1
      ITIME=100*JHOUR+MINUTE
      RETURN
    1 JHOUR=JHOUR-24
      J   .2
      END
```

```
*  **COREAD,BUFWRI,CARD**
*
*
*  THIS PROGRAM REPLACES THE LINKAGE
*  TO .CAS BY A LINK TO BUFWRI
*  THE CARD READER I/C IS DISABLED BUT
*  THE READ(6,XXX) STATEMENT CAN BE
*  USED TO DECODE ANY CATA IN THE BUFFER
*  IA OF SIZE N
*  N IS LIMITED TC 528 WORDS
*
*     USE* CALL COREAD(IA,N) - TO INITIALIZF
*     FOLLOWED BY REAC(6,XXX) STATMENT
*     THEN CALL CARD TO RESTORE
*
 REL 0
 EXTERN .CAS
 EXTERN .FAT
 NAME COREAD,BUFWRI,CARD
IA OCT 0
N OCT 0
COREAD OCT 0
 L .FAT
 ADR IA
 OCT 2
 LA CALCAS
 AND MASK
 STA ZZADR
 EX
 LA,X 0
 STA CASAD
 LA CORAD
 STA,X 0
 J,I COREAD
CALCAS CALL .CAS
MASK OCT 377
CASAD OCT 0
CORAD ADR BUFWRI
ZZADR OCT 0
*  THIS ROUTINE LOADS BUFFER IA INTO I/O BUFFER
BUFWRI OCT 0
 LA,I *-1
 STA BVADR
 LA,I N
 SSP
 S ONE
 EX
BVLBR LA,IX IA
 STA,IX BVADR
 DCX 1
 J BVLBR
 ADR BUFWRI
 J,I BUFWRI
BVADR OCT 0
ONE OCT 1
*
*  THIS ROUTINE REPLACES ZONE ZERO POINTR TO .CAS
CARD OCT 0
 LA COREAD
 C ZERO
 SNE
 J,I CARD
 LA CASAD
 STA,I ZZADR
 J,I CARD
ZERO OCT 0
 END 0
```

GASP FILE

```
      SUBROUTINE CATAN(NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NCQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
C
C*****DATA CARD TYPE TWO
C
    5 READ(6,803)ID,IM,NOQ,MXC,SCALE
  803 FORMAT (4I5,F10.2)
  103 FORMAT (10I5)
C
C*****DATA CARD TYPE FOUR
C*****SPECIFY KRANK=RANKING ROW
C
   41 READ(6,103) KRANK
C
C*****DATA CARD TYPE FIVE
C*****SPECIFY INN=1 FOR LVF, INN=2 FOR HVF
C
   42 READ(6,103) INN
C
C*****DATA CARD TYPE SEVEN.  THE NEP VALUE IS FOR THE NEXT RUN.  SET
C*****JSEED GREATER THAN ZERO TO SET TNOW EQUAL TO TBEG.
C
   43 READ(6,104) MSTOP,TBEG,TFIN,JSEED
  104 FORMAT(I5,2F10.3,I4)
      IF (JSEED) 27,26,27
   27 ISEED=JSEED
      RNUM = DRAND(ISEED)
      TNOW = TBEG
   26 JNNIT = 0
C
C*****INITIALIZE NSET
C*****SPECIFY INPUTS FOR NEXT RUN
C*****READ IN INITIAL EVENTS
C
  299 DO 300 JS = 1,ID
C
C*****DATA CARD TYPE 8
C*****INITIALIZE NSET BY JQ EQUAL TO A NEGATIVE VALUE ON FIRST EVENT
C*****CARD
C*****READ IN INITIAL EVENTS.  END INITIAL EVENTS AND ENTITIES WITH JQ
C*****EQUAL TO ZERO
C
      READ(6,1110) JQ,ATRIB
 1110 FORMAT (I10,4F10.4)
      IF(JQ) 44,15,320
   44 INIT=1
      CALL SET(1,NSET)
      J.  ,300
  320 CALL FILEM(JQ,NSET)
  300 CONTINUE
   15 RETURN
      END
```

```
      SUBROUTINE FILEM (JQ,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IM,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
C
C*****TEST TO SEE IF THERE IS AN AVAILABLE COLUMN FOR STORAGE
C
      IF (MFA - ID ) 2,2,3
C   3 WRITE(6,4)
C   4 FORMAT (//24H OVERLAP SET GIVEN BELOW/)
C     CALL ERROR (87,NSET)
    3 PAUSE 77
C
C*****PUT ATTRIBUTE VALUES IN FILE
C
    2 DO 1 I = 1,IM
      DEL =.000001
C     IF (ATRIB(I)) 5,1,1
C   5 DEL = -.000001
      IF(ATRIB(I) .LT. 0.) DEL=-.000001
    1 NSET(I,MFA)=SCALE*(ATRIB(I)+DEL)
C
C*****CALL SET TO PUT NEW ENTRY IN PROPER PLACE IN NSET
C
      CALL SET (JQ,NSET)
      RETURN
      END
```

```
      SUBROUTINE RMOVE  (KCOL,JQ,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IP,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
C     IF (KCOL) 16,16,2
C  16 CALL ERROR(97,NSET)
      IF(KCOL .LT. 1) PAUSE 76
    2 MLC(JQ) = KCOL
C
C*****PUT VALUES OF KCOL IN ATTRIB
C
      DO 3 I = 1,IM
    3 ATRIB(I)=FLOAT(NSET(I,KCOL))/SCALE
C
C*****SET OUT=1 AND CALL SET TO REMOVE ENTRY FROM NSET
C
      OUT = 1.
      CALL SET (JQ,NSET)
      RETURN
      END
```

```
      SUBROUTINE SET(JQ,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IM,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,OUT
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
      DATA KOL,KOF,KLE/ 7777,8888,9999/
C
C*****INIT SHOULD BE ONE FOR INITIALIZATION OF FILE
C
      IF (INIT-1) 27,28,27
C
   28 MX=IM+1
      MXX = IM+2
C
C*****INITIALIZE POINTING CELLS OF NSET AND ZERO OTHER CELLS OF NSET
C
      DO 1 I = 1,ID
      DO 2 J = 1,IM
    2 NSET(J,I) = 0
      NSET(MXX,I) = I-1
    1 NSET(MX,I) = I + 1
      NSET(MX,ID) = KOF
C     ****************************************************************
C       TEMPORARY PROGRAM - TO REPLACE DC 1 LOOP
      DO 3 K=1,16
    3 MFE(K)=0
C     DO 3 K = 1,400
C     NQ(K)=0
C     MLC(K)=0
C     MFE(K)=0
C   3 MLE(K)=0
C*****************************************************************
C
C*****FIRST AVAILABLE COLUMN = 1
C
      MFA = 1
      INIT = 0
      OUT = 0.0
  101 RETURN
C
C*****MFEX IS FIRST ENTRY IN FILE WHICH HAS NOT BEEN COMPARED WITH ITEM
C*****TO BE INSERTED
C
   27 MFEX = MFE(JQ)
C
C*****KNT IS A CHECK CODE TO INDICATE THAT NO COMPARISONS HAVE BEEN MADE
C
      KNT = 2
C
C*****KS IS THE ROW ON WHICH ITEMS OF FILE JQ ARE RANKED
      KS = KRANK(JQ)
C*****TEST FOR PUTTING VALUE IN OR OUT
C*****IF OUT EQUALS ONE AN ITEM IS TO BE REMOVED FROM FILE JQ.  IF OUT
C*****IS LESS THAN ONE AN ITEM IS TO BE INSERTED IN FILE JQ
C
      IF (OUT-1.0) 6,5,100
C
C*****PUTTING AN ENTRY IN FILE JQ
C*****MFA IS THE SUCCESSOR COLUMN OF THE FIRST AVAILABLE COLUMN FOR
C*****STORING INFORMATION
C*****THE ITEM TO BE INSERTED WILL BE PUT IN COLUMN MFA
```

```
C
    8 NXFA = NSET(MX,MFA)
C
C*****IF INN(JQ) EQUALS TWO THE FILE IS A HVF FILE. IF INN(JQ) IS
C*****ONE THE FILE IS A LVF FILE. FOR LVF FILES TRY TO INSERT
C*****STARTING AT END OF FILE.  MLEX IS LAST ENTRY IN FILE WHICH HAS
C*****NOT BEEN CCMPARED WITH ITEMS TO BE INSERTED.
C
      IF (INN(JQ)-1) 100,7,6
    7 MLEX=MLE(JQ)
C
C*****IF MLEX IS ZERC FILE IS EMPTY.  ITEM TO BE INSERTED WILL BE ONLY
C*****ITEM IN FILE.
C
      IF (MLEX) 100,10,11
   10 NSET(MXX,MFA)=KLE
      MFE(JQ) = MFA
C
C*****THERE IS NO SUCCESSOR OF ITEM INSERTED.  SINCE ITEM WAS INSERTED
C*****IN COLUMN MFA THE LAST ENTRY OF FILE JQ IS IN COLUMN MFA.
C
   17 NSET(MX,MFA) = KOL
      MLE(JQ) = MFA
C
C*****SET NEW MFA EQUAL TO SUCCESSOR OF OLD MFA.  THAT IS NXFA.  THE
C*****NEW MFA HAS NO PREDECESSOR SINCE IT IS THE FIRST AVAILABLE COLUMN
C*****FOR STORAGE.
C
   14 MFA = NXFA
      IF (MFA-KCF) 237,238,238
  237 NSET(MXX,MFA) = KLE
C
C*****UPDATE STATISTICS OF FILE JQ
C
  238 NQ(JQ)=NQ(JC)+1
      MLC(JQ) = MFE(JQ)
      J    .101
C
C*****TEST RANKING VALUE OF NEW ITEM AGAINST VALUE OF ITEM IN COLUMN
C*****MLEX
C
   11 IF(NSET(KS,MFA)-NSET(KS,MLEX))12,13,13
C
C*****INSERT ITEM AFTER COLUMN MLEX.  LET SUCCESSOR OF MLEX BE NSU.
C
   13 NSU = NSET(MX,MLEX)
      NSET(MX,MLEX) = MFA
      NSET(MXX,MFA) = MLEX
      GO TO (18,17),KNT
C
C*****SINCE KNT EQUALS CNE A COMPARISON WAS MADE AND THERE IS A
C*****SUCCESSOR TO MLEX, I.E., NSU IS NOT EQUAL TO KOL.  POINT COLUMN
C*****MFA TO NSU AND VICE VERSA.
C
   18 NSET(MX,MFA) = NSU
      NSET(MXX,NSU) = MFA
      J    .14
C
C*****SET KNT TC CNE SINCE A COMPARISON WAS MADE.
C
   12 KNT = 1
C
C*****TEST MFA AGAINST PREDECESSOR OF MLEX BY LETTING MLEX EQUAL
C*****PREDECESSOR OF MLEX.
C
      MLEX = NSET(MXX,MLEX)
      IF(MLEX-KLE) 11,16,11
C
C*****IF MLEX HAD NO PREDECESSOR MFA IS FIRST IN FILE.
C
   16 NSET(MXX,MFA) = KLE
      MFE(JQ) = MFA
C
C*****SUCCESSOR OF MFA IS MFEX AND PREDECESSOR OF MFEX IS MFA. (NOTE AT
C*****THIS POINT MLEX = MFEX IF LVF WAS USED).
C
   26 NSET(MX,MFA) = MFEX
      NSET(MXX,MFEX) = MFA
      J    .14
C
C***** FOR HVF OPERATION TRY TO INSERT ITEM STARTING AT BEGINNING OF
C*****FILE JQ.
C*****IF MFEX IS 0, NO ENTRIES ARE IN FILE JQ.  THIS CASE WAS CONSIDERED
C*****PREVIOUSLY AT STATEMENT 10.
C
    6 IF (MFEX) 100,10,19
C
C*****TEST RANKING VALUE OF NEW ITEM AGAINST VALUE OF ITEM IN COLUMN
C*****MFEX.
C
   19 IF(NSET(KS,MFA)-NSET(KS,MFEX))20,21,21
C
C*****IF NEW VALUE IF LOWER, MFA MUST BE COMPARED AGAINST SUCCESSOR OF
C*****MFEX.
C
   20 KNT = 1
C
C*****LET MPRE = MFEX AND LET MFEX BE THE SUCCESSOR OF MFEX.
C
      MPRE = MFEX
      MFEX = NSET(MX,MFEX)
      IF (MFEX-KOL) 19,26,19
C
C*****IF NEW VALUE IS HIGHER, IT SHOULD BE INSERTED BETWEEN MFEX AND ITS
C*****PREDECESSOR.
C*****IF KNT = 1, MFEX HAS NO PREDECESSOR.  GO TO STATEMENT 16.  IF KNT
C***** 1, A COMPARISON WAS MADE AND A VALUE OF MPRE HAS ALREADY BEEN
C*****OBTAINED ON THE PREVIOUS ITERATION.  SET KNT = 2 TO INDICATE THIS.
```

```
C
   21 GO TO (22,16),KNT
   22 KNT = 2
C
C*****MFA IS TO BE INSERTED AFTER MPRE.  MAKE MPRE THE PREDECESSOR OF
C*****MFA AND MFA THE SUCCESSOR OF MPRE.
C
   24 NSET(MXX,MFA) = MPRE
      NSET(MX,MPRE) = MFA
C
C*****IF KNT HAS NOT RESET TO 2, THERE IS NO SUCCESSOR OF MFA.  POINTERS
C*****ARE UPDATED AT STATEMENT 17.  IF KNT = 2, IT HAS RESET AND THE
C*****SUCCESSOR OF MFA IS MFEX.
C
      GO TO (17,26), KNT
C
C*****REMOVAL OF AN ITEM FROM FILE JQ.
C
    9 OUT = 0.0
C
C*****UPDATE POINTING SYSTEM TO ACCOUNT FOR REMOVAL OF MLC (JQ).  COLUMN
C*****REMOVED IS ALWAYS SET TO MLC(JQ) BY SUBROUTINE ABOVE.
C
      MMLC = MLC(JQ)
C
C*****RESET OUT TO 0 AND CLEAR COLUMN REMOVED.  LET JL EQUAL SUCCESSOR
C*****OF COLUMN REMOVED AND JK EQUAL PREDECESSOR OF COLUMN REMOVED.
C*****IF JL = KOL, MLC WAS LAST ENTRY.  IF JK = KLE, MLC WAS FIRST ENTRY
C*****MLC WAS NCT FIRST OR LAST ENTRY.  UPDATE PCINTERS SC THAT JL IS
C*****SUCCESSOR OF JK AND JK IS PREDECESSOR OF JL.
C
      DO 32 I=1,IP
   32 NSET(I,MMLC) = 0
      JL = NSET(MX,MMLC)
      JK= NSET(MXX,MMLC)
      IF (JL-KOL) 33,34,33
   33 IF (JK-KLE) 35,36,35
   35 NSET(MX,JK) = JL
      NSET(MXX,JL) = JK
C
C*****UPDATE POINTERS.
C
   37 NSET(MX,MMLC) =MFA
      NSET(MXX,MMLC) = KLE
      IF (MFA-KCF) 234,235,235
  234 NSET(MXX,MFA) = MMLC
  235 MFA= MLC(JQ)
      MLC(JQ) = MFE(JQ)
C
C*****UPDATING FILE STATISTICS
C
      NQ(JQ) = NQ(JQ)-1
      J    .101
C
C*****MLC WAS FIRST ENTRY BUT NOT LAST ENTRY.  UPDATE PCINTERS.
C
   36 NSET(MXX,JL) = KLE
      MFE(JQ) = JL
      J    .37
   34 IF (JK-KLE) 38,39,38
C
C*****MLC WAS LAST ENTRY BUT NOT FIRST ENTRY.  UPDATE PCINTERS.
C
   38 NSET(MX,JK) = KOL
      MLE(JQ) = JK
      J    .37
C
C*****MLC WAS BOTH THE LAST AND FIRST ENTRY. THEREFORE, IT IS THE ONLY
C*****ENTRY.
C
   39 MFE(JQ) = 0
      MLE(JQ) = 0
      J    .37
C 100 CALL ERROR(88,NSET)
C     CALL EXIT
  100 PAUSE 75
      END
```

```
      SUBROUTINE ERROR(J,NSET)
      DIMENSION NSET(6,1)
      COMMON ID,IM,INIT,JEVNT,MFA,MSTOP,MX,MXC,NOQ,SCALE,ISEED,TNOW,
     1TBEG,TFIN,MXX,NPRNT,NCRDR,NEP
      COMMON ATRIB(4),INN(4),KRANK(4),MFE(4),MLC(4),MLE(4),NQ(4)
      WRITE(6,100) J
C 100 FORMAT(///17H1ERROR EXIT, TYPE,I3,7H ERROR.)
C   6 CALL EXIT
  100 FORMAT(17X,5HERROR,I3/)
    5 RETURN
      END
```

```
      SUBROUTINE EXIT
      PAUSE 100
      RETURN
      END
```

```
************ D A T A   C A R D S   F O R   12  T O  8  S H I F T   *************
************ M A R C H   3   1973                                  *************


1450155C17000OCC15001050
0,0,0,0,0,0,0,0,0,0,10,10,10,1,2,3,30,106,182,0,
42,216,62,141,264,324,152,376,96,296,351,129,351,126,469,308,244,450,126,500,
0,0,0,0,0,0,430,134,45,134,80,25,38,55,38,30,45,87,515,133,
10,10,10,10,37,37,37,62,39,31,93,31,158,30,80,60,37,10,10,10,
10,6,9,7,6,1,1,1,4,4,1,1,1,4,1,1,1,1,5,8,
7,5,3,2,7,2,2,4,1,1,3,1,1,1,1,17,21,1,1,15,
15,20,0,0,0,0,0,0,0,0,0,0,0,0,1,1,-1,-1,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,-1,4,0,0,1,
7,8,8,2,2,19,19,23,5,5,13,10,7,15,13,19,35,31,64,98,
25,13,32,36,13,4,58,28,33,6,0,50,0,13,22,0,2,0,25,28,
20,0,0,0,15,21,2,2,0,C,6,12,14,16,3,11,13,15,17,7,
9,11,13,15,8,10,12,14,16,94,64,49,19,69,90,91,92,93,94,95,
96,97,98,99,4,5,6,7,8,74,99,79,104,109,3,2,13,6,6,
0,0,0,0,0,0,0,0,0,6056598464620,0,0,0,0,120014500,
3C0,300,0,C,0,0,0,0,200,0,500,
0,0,800,800,0,300,40C,400,500,0,200,300,0,0,0,0,0,500,

0,0,0,0,0,0,0,100,10C,0,200,300,

0,0,0,0,0,0,0,C,0,0,0,0,0,0,0,0,0,285,290,
65,85,95,140,0,1C,40,0,0,0,0,0,0,120,

C,0,0,0,0,0,0,0,C,0,0,0,0,0,0,0,0,24,22,0,
13,12,0,0,0,0,0,40,0,0,0,0,12,13,0,5,7,


0,0,0,0,-1,



175.,530.,90.,240.,180.,9.08,10.08,8.23,9.23,18.,
11.,34.55,35.55,31.12,12.12,120.,121.,1.,5.,900.,
20,20,60,0,3,20,18,19,18,21,
21,25,16,16,10,52,48,46,0,0,0,
0,9,39,21,46,0,2,53,5,3,0,0,0,0,
0,0,0,0,2,2.,77.,79.,68,54.,
53,49,57,56,57,43,43,43,1.6,1.7,
.8,2.4,2.5,1.2,C,0,0,.67,.67,.14,
.73,.76,.72,.78,.8,.76,0,0,0,131.2,
10.,59.8,10.,10.,10.,200.,1000.,200.,3000.,3000.,
3000.,30.65,31.65,1.,5.,0.,0.,0.,0.,72.,
72.,72.,72.,72.,95.,5.5,0.,45.,63.,2.6,
1.,0.,38.,34.8,14.,10.,0.,17.,32.,0.,
95.6,0.,0.,0.,22.5,23.7,0.,23.4,26.2,.96,
1.,1.,1.,.98,140.,140.,140.,140.,150.,50.,
50.,50.,50.,70.,150.,150.,150.,150.,150.,



60.,4,4,0,10.,
1,3,3,3,
1,1,1,1,
0,0.,0.,.833,
-1,
1,240.,37.,5.,0.,                              ANODE FURNACE & FINISHED CASTING
1,6.,44.,1.,
1,4.,45.,2.,                                   MATTE READY AT TUNNEL 72
1,0,0,59,0,0,0,0503...                          INITIALISE TIME GRATE
1,400.,58.,
1,10.,48.,                               INITIALISE SLAG LEVEL IN 48
1,15.,41.,1.,                                   UPDATING MATTE LEVEL IN R1
1,15.,41.,2.,                                   UPDATING MATTE LEVEL IN R2
1,15.,41.,3.,                                   UPDATING MATTE LEVEL IN R3
1,15.,270.,1.,                               SLAG READY INCONVETER 3
1,0.,40.,60.,                                   INITIALISE FORECASTING
0,
```