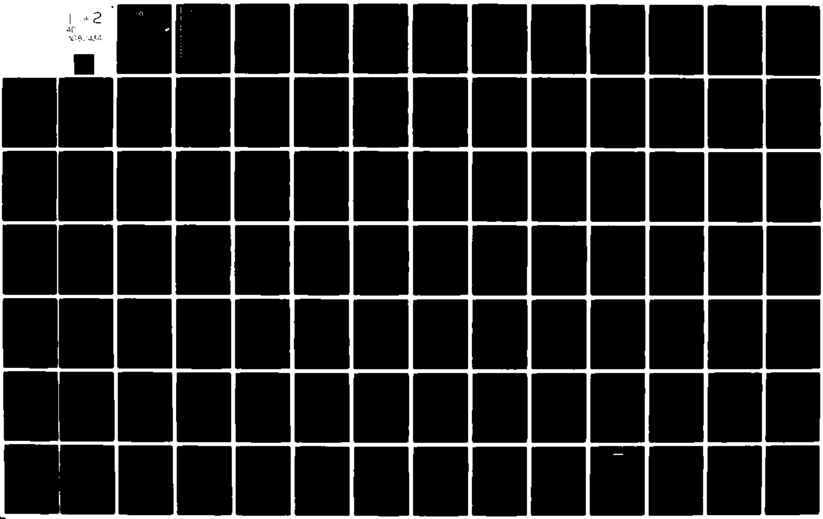
AD-A082 434 OHIO STATE UNIV RESEARCH FOUNDATION COLUMBUS F/G 9/2
FINITE STATE MODELS OF MANNED SYSTEMS: VALIDATION, SIMPLIFICATI--ETC(U)
NOV 79 R A MILLER AFOSR-79-0015
UNCLASSIFIED AFOSR-TR-80-0216 NL

1 of 2
AD
A082434



AFOSR-TR- 80-0216

LEVEL

RF Project 761348/711544

Report No.

ADA082434

ANNUAL

REPORT

By

THE OHIO STATE UNIVERSITY
RESEARCH FOUNDATION

1314 KINNEAR RD.
COLUMBUS, OHIO 43212

To U.S. AIR FORCE
Air Force Office of Scientific Research
Building 410, Bolling A.F.B.
Washington, D.C. 20332
Grant No. AFOSR-79-0015

On FINITE STATE MODELS OF MANNED SYSTEMS: VALIDATION, SIMPLIFICATION, AND EXTENSION

For the period October 1, 1978 - September 30, 1979

Submitted by Richard A. Miller
Industrial and Systems Engineering

Date November, 1979

80 3 20 094

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFOSR-TR- 80-0216	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): FINITE STATE MODELS OF MANNED SYSTEMS: VALIDATION, SIMPLIFICATION, AND EXTENSION		5. TYPE OF REPORT & PERIOD COVERED: Annual 10/1/78-9/30/79
		6. PERFORMING ORG. REPORT NUMBER: ~~761348/711544~~ ✓
7. AUTHOR(s): Richard A. Miller		8. CONTRACT OR GRANT NUMBER(s): AFOSR-79-0015
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The Ohio State University Research Foundation, 1314 Kinnear Road Columbus, Ohio 43212		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2312 A6
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Air Force Air Force Office of Scientific Research Washington, D.C. 20332		12. REPORT DATE: November 1979
		13. NUMBER OF PAGES: 137
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

Annual rept. 1 Oct 78 - 30 Sep 79

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

2312

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

discrete control, coordination, operator teams, validity

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Three basic research objectives were pursued. An assessment of the robustness and validity of previously developed discrete control models resulted in a comprehensive examination of the conceptual and methodological issues associated with validity. This work is included in the report.

Work on computer assisted discrete control analysis focused primarily on the problems associated with representing continuous data in a form suitable for use by existing analysis routines.

(continued)

DD FORM 1473 1 JAN 73

211360

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block #20 (Abstract) -- Continued

Additional work on modelling discrete control tasks, including modelling coordination in small trams was also performed and resulted in two papers which are included as appendices to the report.

FINITE STATE MODELS OF MANNED SYSTEMS:
VALIDATION, SIMPLIFICATION, AND EXTENSION

Richard A. Miller, Christine Mitchell, Rajendra Nalavade,
Anant Misal and Chang Feng-Chang

The Ohio State University

for the period
October 1, 1978 - September 30, 1979

Department of the Air Force
Air Force Office of Scientific Research
Bolling Air Force Base, DC 20332

Grant No. AFOSR-79-0015

November 1979

TABLE OF CONTENTS

Table of Contents (Cont'd)

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This research had three main objectives: 1) An assessment of the robustness of previously developed discrete control models, 2) design and implementation of computer-aided procedures for discrete control analysis and 3) theoretical work on the use of heterarchical systems to represent discrete control tasks. Significant progress was made toward ach objective and the major technical results are reported here. This work is an extension of work founded under a previous grant, AFOSR-79-0015, and the final report on that work (Miller, 1979). provides additional background information.

The report is organized as follows. The research task on validation has resulted in a significant piece of technical and conceptual work. This work which includes both definitional and methodological aspects is currently being refined for possible publication. Chapter 2 is the report on this research task.

The task on computer-aiding has emphasized problems associated with representing continuous data in a compact, useful form, as well as the implementation of procedures which enable the analyst to easily reconfigure data bases to better match the immediate analysis task. This work has also enabled the use of a smaller, less expensive computer system (a Digital Equipment Corporation PDP 11/34 owned by the Department of Industrial and Systems Engineering at The Ohio State University) which makes feasible interactive analysis including the use of graphics. The work on continuous data approximation, for example, would not have been possible without the availability of interactive graphics.

The work on heterarchical structures and modelling operator behavior resulted in two papers. These papers document this work and are included as appendices.

2. ON THE CONCEPT OF VALIDITY

This section is a report on the attempt to clarify what validity means at a theoretical level, and to establish the methodological issues associated with the validation of discrete control models.

2.1 INTRODUCTION

Model validity, though a crucial issue in any modelling exercise, is a very poorly understood concept. Perhaps due to its nebulous nature, it has received little attention in the literature. A quick review of the modelling and simulation literature yields the following insights into some conceptions of model validity.

. Forrester (1968) states that model validity is a relative matter. That is, validity should be judged in the context of mental or descriptive models which would be employed, in the absence of a formal or mathematical model, to assist in understanding or describing the behavior of the real system.

. Naylor (1971) defines a valid model as a "true" model. Thus, a set of criteria to differentiate between "true" models and models which are not true as well as the means to apply these criteria are necessary in a procedure to determine model validity. As the concept of "truth" is so illusive as to render the problems of validity overwhelming, Naylor suggests that the focus shift from validation to confirmation. A series of tests of the model are conducted; few or no negative results increase the degree of confirmation of the model with respect to the real system.

.Shannon (1975) suggests that validity is a concept of degree rather than an either-or notion. The degree of validity corresponds to the level of confidence in an inference drawn from the model about the real system.

.Ziegler (1976) characterizes model validity as an issue of how well the model represents the real system. One measure of validity is the extent of agreement between the real system data and model generated data. A second measure of validity, structural validity, is the degree to which the model structure reflects the structure of the real system.

Each author contributes attributes that a valid model is expected to have. In a primarily subjective vein, Forrester and Shannon point out that a valid model is one which is at least as adequate in providing insight and understanding of the real system as any candidate model and inspires an adequate degree of confidence in the model's resemblance to the real system. Ziegler, a bit more specific about the nature of the resemblance, suggests that a valid model is one which is behavior and structure preserving. Naylor, admittedly abandoning the philosophical and abstract character of validity, is concerned with the more traditional perspective of measures of merit and goodness of fit between model generated and real system generated data.

These characterizations all concern a relation between three objects: the real system, the model, and the collection of properties and behaviors of the real system. Validity is fundamentally an issue of the preservation of behavior, structure, or properties exhibited by or hypothesized about the real system in the model. The extent or completeness of the preservation determines validity; but, the required or expected preservation needed for validity is subjective, varying among modelling exercises.

2.2 A CONTEXT FOR VALIDITY

The three objects fundamental to any discussion of validity (the model, the real system, and the collection of properties and behaviors characterizing the real system) are very different entities. A model, as used in this context, is a formal, well-defined, set-theoretic structure, an artifact. The real system, however, in some amorphous entity in the world which is of interest. The real system is never exactly known. At best, attributes and properties of the real system can be identified. In all likelihood, these characteristics are hypotheses about the real system rather than "truths". Often sets of behavior or data are identified with the real system; but, these collections of characteristics and behaviors do not constitute the real system. They are merely a description of the real system.

Validity entails the preservation of the collection of real system characteristics and behaviors in the model. The real system is so intangible that its only relation to the model is through the collections of properties, attributes, and data which define an interface. Model validity, then, is a concept which relates formal models to the collections of real system characteristics and behaviors, while the real system is an entity that gives rise to the characteristics and behaviors.

Concepts from model theory provide some precise language to talk about the relationships of models, real systems, and collections of characteristics of a real system.

A theory is a collection of statements in some language. A model of a theory is a structure in which the statements of the theory are interpretted as true. It is useful to differentiate models into two classes: intensional and extensional.

An intensional model is one which gives rise to or generates a theory: therefore, the theory is construed as a linguistic abstraction of the properties, behaviors, or characteristics of the intensional model. The extensional model is one which embodies the statements of the theory. That is, given a theory, the extensional model is a structure created or selected so that the statements of the theory are interpretted as true in the model. The extensional model is an instanciation of the theory. The relationships between an extensional model, a theory, and an intensional model is illustrated in Figure 1 where the arrow denotes the relation "gives rise to".

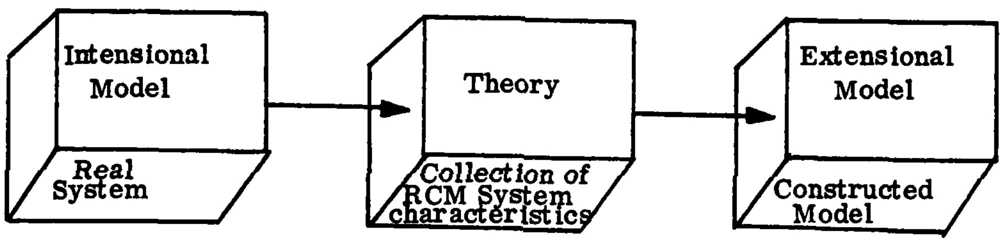


Figure 1.

Within the framework of a given modelling process the real system, a term designating the amorphous entity of interest, corresponds to the intensional model. The theory generated by the intensional model corresponds to the collection of real system characteristics. The distinction between intensional model and theory is instructive in understanding the distinction between the real system and the collection of real system characteristics. The collection of real system characteristics is not the real system anymore than the theory is the intensional model. Rather, the collection of system characteristics is a linguistic abstraction, i.e., a set of statements describing the real system, based on experience with that system. The experience with the system is filtered and modified by individual perceptions and organizational processes, limited perhaps by time or by measurement instruments. The flavor is that the collection of real system characteristics is a theory which attempts to explain, summarize or capture, with varying degrees of success, the behavior or structure of the real system using the available knowledge at a given point in time.

Given a theory of the real system, the extensional model corresponds to the formal model constructed in an effort to understand the real system. The extensional, or constructed, model of the theory is a formal, set-theoretic structure which embodies the system characteristics posited by the theory.

In addition to the set of real system characteristics, the constructed model seeks to preserve or replicate the system behaviors as contained in the data collected from the real system. The collection of data sets or behaviors from the real system is called the data structure of the real system. The data structure is a part of the real system description, and, as such, is distinct from the real system itself as well as the constructed model of the real system. The data structure is a fragmented collection of behaviors manifested by the real system. It is set-theoretic in nature, a rudimentary model of the real system.

The description of the real system has two components: the collection of characteristics (theory) and the collection of behaviors (data structure). The characteristics and behaviors are also distinct. The characteristics embodied in the theory are a set of statements about real system structure or behavior;

characteristics and behaviors are also distinct. The characteristics embodied in the theory are a set of statements about real system structure or behavior; they are not behaviors themselves. Given a data structure, conjectured properties about behavior or relationships in the data become statements of the theory.

In terms of the objects and relations depicted in Figure 1, the data structure is derived from the real system. It, in turn, impacts or helps to give rise to both the theory and the extensional model. In addition to, or sometimes in lieu of, the inclusion of data properties in the theory, and, thus, eventual incorporation into the extensional model, the data sets may be used directly, in their most primitive forms, to assist in the formulation of the extensional model. As a result, one component of model validity must address the preservation of observed behavior in the model.

Figure 2 depicts the relationships between the data structure and the other components of a modelling exercise.

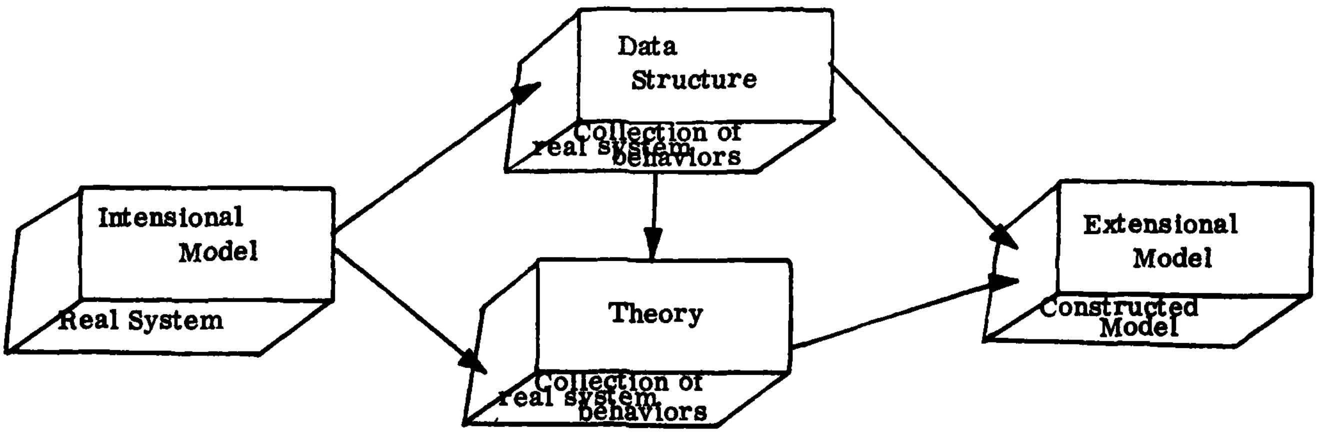


Figure 2

This characterization of the relation between the real system, the description of the real system, and the constructed model is helpful in exploring some of the ambiguity associated with validity.

The intuitive notion of validity involves a relation between the intensional and extensional models. That is, a valid model is one which exhibits structure and behavioral characteristics of the real system. However, the only explicit knowledge of the real system is that which is contained either in the theory or in the data structure. As it is only through the theory and data structure that the constructed model "knows" the real system, the constructed model can only be expected to preserve those behaviors and properties which have been made explicit by or are logical implications of theory postulate. At a pragmatic level, validity is an issue concerning the preservation, in the constructed model, of the characteristics and behaviors postulated by the theory and indicated by the data structure. Validity, at this level, is concerned with preservation of known attributes; model validity cannot be judged with respect to real system attributes and behaviors which are currently unknown. Using this conceptualization, validity is a relative issue depending on the current state of knowledge of the real system and the representation of that knowledge in the extensional model.

In this discussion, pragmatic and intuitive notions of validity represent two levels of validity issues. The first, designated level I, concerns the relationship between the constructed model and the knowledge of the real system as embodied in the theory and data structure; the second, designated level II, concerns the relationship between the real system (intensional) model and the extensional (constructed) model. Clearly, level II issues are heavily constrained by level I issues. The validity of the constructed model with respect to the real system is constrained by the shophistication, cogency, or depth of the theory. A good model of a bad theory provides little level II validity. Likewise, poor data gives meaning to the old adage: garbage in, garbage out.

2.3 LEVEL II VALIDITY

Level II validity concerns the relation between the real system and the constructed model. It is this relation that underpins the intuitive notion of validity. The amorphous, intangible nature of the real system makes it difficult to relate it to the formal model in any precise way. It is this lack of precision which accounts, at least in part, for the ambiguity and nebulosity characterizing most discussions of validity. Precision is possible in relating the description of the real system to the constructed model; the next chapter provides some formal procedures to describe such relationships. This chapter will outline two conceptions of level II validity, ways that the constructed model is judged against the unknowable real system. However, it will have the same lack of formal structure and precision that most validity discussions exhibit.

A. The Tacit Component

Michael Polanyi (1966) suggests that "we know more than we can tell", that is, all knowledge has two components: an explicit dimension and a tacit dimension. The theory and data structure of the real system constitute the explicit dimension. They are collections of behaviors and of statements describing properties that the real system is known or believed to possess. Intuition represents the tacit dimension. In the course of observing or manipulating the real system and in formulating the theory, the researcher or others connected with the process develop a "sense" of the system which extends beyond the description embodied in the theory.

For the most part this "sense" of the system will act as a warning or detection system for level II validity issues. The constructed model may produce behavior or contain structures which are counterintuitive, challenging the modeller's "sense" of the system. In such cases, an opportunity is available to expand the theory by attempting to make explicit those real system characteristics which are being violated by the current constructed model.

In some cases it might be argued that in addition to producing non-counterintuitive results, the constructed model must produce strictly intuitive results as a necessary condition for level II validity. The point here is that the model results must confirm the intuitive sense of what out to be; the model can produce no surprises, unacceptable or otherwise. This is a much stronger condition and presupposes the existence of a set of characteristics for intuitive results At least some suggest (Forrester, 1968) that a major purpose of the modelling exercise is to fill in the large gap between a set of assumptions about the real system and the consequences of those assumptions, i.e., the relationship between a linguistic theory and a model of that theory. Such an approach allows for the existence of a tacit framework to detect unacceptable results but requires no fixed range for valid results.

Adopting the more conservative approach, the admittedly subjective criteria of non-violation of the "sense" of the real system is proposed as a screening mechanism for level II validity. Additional criteria are described in the following section.

B. Predictive Capability

Level II validity issues concern the relation between the constructed model and the real system. One helpful characterization of level II validity is the definition for predictive validity given by Ziegler (1976). A model is said to be predictively valid "when it can match data before data are acquired from the real system (or at least "seen" by the model) (p. 5)." Thus, at a specific point in time, given a theory of a real system, a constructed model which realizes the theory, and a new set of behaviors or properties obtained from the real system, the constructed model is considered predictively valid if the model behavior or structure is consistent with the new knowledge of the real system.

This formulation of validity is also directly tied to the relation between the description of the real system and the constructed model. Level II validity suggests that the constructed model is more robust or complete than the data structure from which it was derived. The new real system behavior had not been incorporated into the description giving rise to the model, yet the model "anticipated" the behavior.

Failure of a constructed model to be predictively valid frequently necessitates revision of the theory. Using the broad concept employed in this paper, a description of the real system, at a given point in time, consists of both characteristics and behaviors. These two components constitute the current state of knowledge of the system. Thus, when new data are obtained, regardless of the predictive ability of the constructed model, the data structure is revised to include the additional behaviors. In those cases where the model fails to be predictively valid, tenets of the theory which characterize system behavior based on the data may have to be updated to re-establish consistency between theory tenets and the data structure. Any change in the theory may require a corresponding change in the model which interprets it.

Predictive validity provides insight into the dynamic characteristic of validity. A model which is deemed valid at one point may become invalid with the advent of additional real system characteristics. Particularly with theories or data structures, which are considered incomplete and real systems viewed as poorly understood, it is expected that model validity is a transient property. It is only through the cycle of manipulation or analysis of the intensional model. formulation of more cogent theories,and construction of richer models that the potential for, and duration of, model validity is achieved.

2.4 LEVEL I VALIDITY

Level I validity concerns the degree to which the constructed model preserves the behaviors defined in the data structure and the properties postulated by the theory. It is the thesis of this discussion that all formal procedures to ascertain validity must addreww the relationship between the description of the real system and the extensional model, rather than the relationship between the intensional and extensional models. The following sections outline the tools needed to specify in a precise, formal manner the relationship between the model and the description.

A. Models for Completion and Restructuration

A theory has been defined as a collection of statements in a language. In particular, it is a set of statements which constitute the current state of knowledge about the real system, the intensional model giving rise to the theory. Knowledge about the real system may be very complex or very elementary; it may be organized in a cogent way or be very fragmented and discontinuous. Frequently, the coherence of the theory determines the relation between the model and the theory.

A model of a theory is a realization, a set-theoretic structure, in which all the valid statements of the theory are satisfied. The valid statements include both the explicit tenets of the theory as well as all logical consequences of those tenets.

Apostle (1961) differentiates between models constructed to complete a theory and models constructed to restructure a theory. Models constructed for completion permit extension of the theory by means other than experimentation or analysis of the real system. Such models incorporate the properties and behaviors postulated by the theory as well as some which are not; the completion models are realizations of theories extending or completing the original theory. The original theory is extended or completed by adding one or more axioms. The new axioms were not previously included in the statements of the theory nor are they logical consequences of the original set of axioms. The assumption is that in attempting to complete or add to a theory in this manner, it is an easier task to choose between models than it is to choose between the extended linguistic structures; the distinctions among models constructed from the theories are clearer than the distinctions among the theories themselves. Typically, this type of modelling exercise involves a theory which is well confirmed and well developed and is associated with a real system considered well understood. The validity issue for completion models is the strict preservation of the entire set of postulates of the core theory in the constructed models.

Models created to assist in the restructuration of a theory are not precisely models under the classical definition of model. Restructuration models satisfy only a portion of the statements of the theory, perhaps the best confirmed or most intuitive; they fail to satisfy other less central fractures of the theory. The assumption is that the partial correspondence and partial discrepancy between model and theory may lead to the reformulation of the initial theory. In addition, various theory tenets can be relaxed to create a spectrum of partial models (partial discrepancies, partial correspondences, partial inconsistency among models), none satisfying the whole initial theory, potentially leading to a reformulation of the theory to account for the newly acquired insights. Modelling to restructure a theory is likely to be associated with theories which are less coherent and cohesive, generated by an intensional model which is poorly understood. Obviously the validity issue for restructuration models is a bit different than the strict preservation of the behavior and properties postulated by the theory. The validity concerns can only be directed to those theory statements which were selected for incorporation into the extensional mode; this new set of statements is called the partial theory.

The key point is that the stage of development of theory influences the type of modelling excersize undertaken. One must be explicit in defining the purpose of the model with respect to the statements of the theory before the validity questions concerning the preservation of theory tenets can be addressed.

The concept of restructuration models permits amendment of Figure 2 to incorporate the new relations:

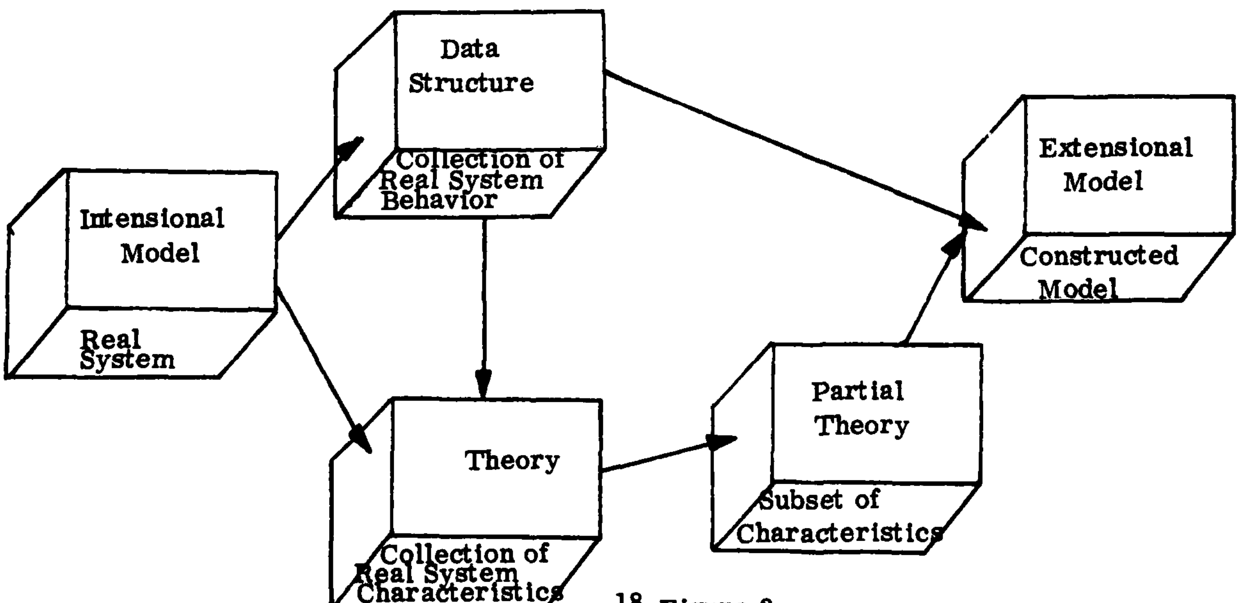


 Figure 3

B. The Base Model

The level I notion of validity concerns the preservation, in the constructed model, of behavior and properties that the real system is known or hypothesized to possess. A natural interpretation of the idea of preservation is that of a map which represents the objects or elements of one structure in another so that the properties and relations defined on the objects in the first structure are preserved in the second.

For the purpose of this discussion the description of a real system has two components: the data structure and the theory. Intuitively, validity requires the preservation, in the extensional model, of the properties and behaviors given in the description. This presents a technical problem; the theory is not a structure but a set of linguistic expressions. To circumvent this difficulty the idea of a base model is used. The base model is a set-theoretic structure in which the statements of the theory are strictly interpretted. In addition, the base model is defined so that the data structure, a set-theoretic entity originally, is a proper subset of the base model.

The theory used in the construction of the base model refers to the partial theory; that is, that subset of statements of the original theory which are to be incorporated into the constructed model. There is no requirement that the set of statements of the partial theory be a proper subset of the statements of the original; in the case of models for completion the two sets of statements are exactly the same.

The theory of the real system consists of properties characterizing the system; these properties may be conjectures about system behavior or structure. In most cases there is no reason to expect that the data sets or the collection of inferences are complete or form a coherent description. The base model, a strict realization of the theory and a superset of the data structure, will be as fragmented, rudimentary, and irregular in form as the theory and data structure from which it is derived. It may consist of several incomplete data sets; moreover, there is no requirement that relations defined on the structure be functional or possess any other mathematically tractable properties. For example, bits of information about particular state transitions given a current state and input or isolated conjectures about internal structure will be structurally interpretted in the base model; but there is no extension to a well-defined state transition function nor elaboration of internal structure in the base model. The base model is a very primitive model of the theory; it merely provides the mechanism to define more precise concepts for level I validity, i.e., the preservation of theory postulates and observed behaviors in the extensional model. The regularity, coherence, and, perhaps, mathematical tractability are found in the extensional model, the completion of the base model.

C. Hierarchy of System Specifications: A Review

This section provides a review of formal structures, called system models, used to represent the constructed and base models. The review is organized as a hierarchy of levels of system model specifications, where each level can be viewed as a level of description or a level of knowledge of the real system.

Various assumptions will be made about the attributes of the system models. The general background will be that of formal structures used to describe discrete control systems. A comprehensive development of the structure can be found in Miller (1976, 1979). The use of a discrete control context requires the inclusion of some limiting assumptions; however, a good deal of the discussion is immediately generalizable.

C.1. Behavioral Description

To formalize the notion of a dynamic system model a time set is needed. A time set is some set T together with a binary relation defined on T which linearly orders the set. If "model time" is discrete, so is T; continuous time is represented by a set corresponding to a subset of the non-negative real numbers. In the following discussion time is considered continuous.

Input and output values for the model are required. These values are provided by two sets: A, the input alphabet; and, B, the output alphabet. Neither A nor B need be first order sets; either can be relations of the form.

$$A \subset A_1 \times A_2 \times \ldots \times A_n$$

and

$$B \subsetneq B_1 \times B_2 \times \ldots \times B_m$$

In a discrete control context the input alphabet corresponds to discrete pieces of information about the system or the environment; thus, A is assumed to be a finite set. The output alphabet, corresponding to the available decision alternatives, is also a finite set.

Input and output sequences are defined as sequences, over time, of input and output values. The notion of sequences or trajectories is formalized as:

$$A^T \equiv \{x \mid x: T \longrightarrow A\}$$

$$B^T \equiv \{y \mid y: T \longrightarrow B\}$$

A^T is the set of all possible input sequences. An element of A^T is a function where $x(t)$, $t \in T$ is the input value to the system model at time t. B^T is defined similarly where $y(t)$ is the value of the output trajectory at time, $t \in T$.

A behavioral description of a system model, S, is

$$S \subseteq A^T \times B^T$$

To simplify notation, let $X = A^T$ and $Y = B^T$ so that the behavioral description can be rewritten as:

$$S \subseteq X \times Y$$

An element of S, (x, y), is a pair of trajectories, input and output, representing one system appearance. At the behavioral level of description the system model is simply a collection of such pairs.

The behavioral description is the most primitive level of description and represents a fairly rudimentary knowledge of the real system. Additional knowledge about structure and specific behavioral responses are represented at the higher levels of system specification.

C.2. Constructive Specification

The usual means of describing, at least technically, how or why a particular input trajectory produces an output trajectory is to employ a state representation of the system model.

Let C denote the state space or set. The set of state trajectories is

$$Z = C^T = \{z \mid z: T \longrightarrow C\}$$

For the discussion, the state set, C, is assumed to be finite.

An important task in constructing a system model is to define a state space. On a conceptual level, a state should summarize the past history of the system so that the knowledge of the future output can be predicted using knowledge of the current state and future input. On a structural level, the state space "functionalizes" the behavioral representation of the system, permitting specification of output as a function of input and state.

Given sets A, B, C and T, three functions are needed to complete the constructive specification of a system model: the state transition function, the input event transition function, and output assignment function.

The two transition functions require a more explicit notion of the state space and state trajectories. A state trajectory is a sequence of states together with the times at which the stat is occupied. Figure 4 is a hypothetical state trajectory.

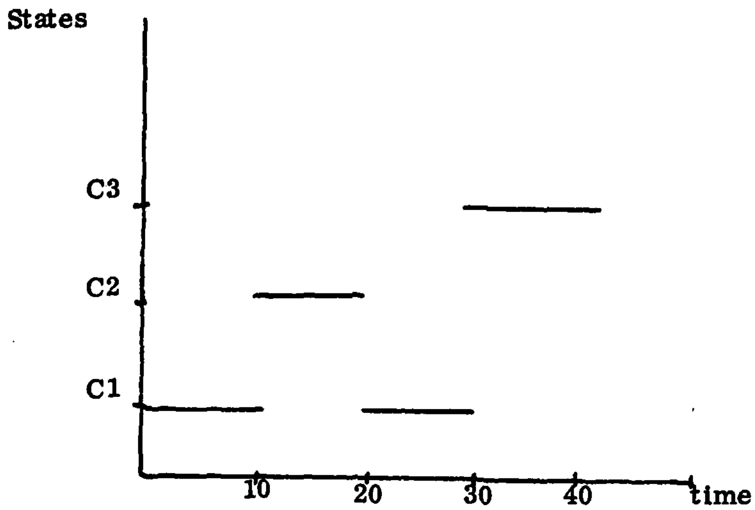


Figure 4.-- State Trajectory

Three pieces of information are provided by a state trajectory:

1. The state occupied at any point in time.
2. The time at which occupancy began.
3. How long ago the occupancy began.

This set of information defines the abstract state of the system model. A system model is called time homogeneous if the time at which occupancy began, item 2, is unnecessary information. Time homogeneity is assumed in this discussion.

This system model is characterized by two kinds of events: input events and state change events. An input event occurs when the input changes; likewise, a state event is said to occur when the state changes. The occurrance of an event updates the abstract state of the system model. An input event updates the state occupancy duration component by incrementing it by the time since the last event. A state change event changes the state of the abstract state and resets the occupancy duration time to zero, i.e., the occupancy just began.

The input event transition, *updating the abstract state*, is defined as:

$$\Psi : C \times T \times A \times T \longrightarrow C \times T$$

so that

$$\Psi(c, t_1, a_1, t_2) = (c, t_1 + t_2)$$

where (c,t) is the abstract state at the previous event, the time since that event is t, and the new input, at the current time, is a. The new abstract state has the same state as before but the occupancy duration is increased to account for the time since the last event.

The state transition function updates in a probabilistic manner the state component of the abstract state. Consider a set:

$$M = \left\{ m \mid m: C \rightarrow [0,1] \text{ where } \sum_{C} m(c) = 1 \text{ and } m(c) \geq 0 \right\}$$

This set is interpreted as the set of state occupancy distributions. The state transition function is given by:

$$\Phi : C \times T \times A \times T \rightarrow M$$

where Φ (c,t,a,s) M is a state occupancy distribution given the following information:

1. The abstract state at the last event was (c,t).
2. The input since that time is fixed at a.
3. The next state event occurs within s units of the last.

(N.B. m(c), where c is the state at the last event, is the probability that no state event occurs in the s units following the last event.

Thus, Φ establishes the probability of state occupancy at the next state change (assuming that the next state change occurs within a time units) given the abstract state at the last event and a constant input since that time.

One interesting form of the function occurs when s = 0; this function gives the state occupancy distribution at the occurrence of an input event. Conceptually, an input event signals the need to update the abstract state, using Ψ , and then to establish the state, using Φ. Thus, if the system were in abstract state (c,t) and a new input a was applied at t_2 then

$$\Phi\ (\ \psi(c_1,\ t_1,\ a_1,\ t_2),\ a)\ =\ (c_1,\ t_1,\ t_2,\ a,\ o)$$

gives the state occupancy distribution at the occurrence of the input.

Finally, an output assignment function is required.

$$\lambda : C \rightarrow B$$

This function determines the output given a state.

The constructively specified system model is an entity

$$S = \langle A, B, C, T, \Psi, \Phi, \lambda \rangle$$

which describes in a more structured way the behavior of the system model.

C.3 Network Specification

Both the behavioral description and the constructive specification of the system model are macro or wholistic representations of the real system, basically input/output descriptions. There is little attempt to represent the internal structure. One mechanism to model the internal structure of the real system is through the use of networks of system models. This procedure uses the entire network to represent the real system and the individual system models to represent important or well-understood subsystems of the real system.

Structurally, a network is a directed graph with nodes representing system components and arcs consisting of communication links.

Let:

$$N = \{ i \mid i \text{ is a name of a system model component} \}$$

and

$$G \subseteq N \times N$$

so that

$$(i\ j) \in G$$

if and only if there is an arc originating at i and terminating at j.

A system model named i is given as

$$S_i \subseteq A_i^T \times B_i^T$$

with

$$A_i \subseteq A_{i1} \times A_{i2} \times \ldots \times A_{in}$$

and

$$B_i \subseteq B_{i1} \times B_{ij} \times \ldots \times B_{im}$$

To specify the interconnections among the system models, define a relation L so that

$$L \subseteq G \times (I \times I)$$

where I is the set of positive integers, and

$$((i,j), (l,r)) \in L$$

if and only if output of system i is input k of system j; that is,

$$B_{i_e} = A_{j_k}$$

In the hierarchy of system mo del specifications the network is the highest level; it represents structural as well as behavioral properties of the real system. Given a network specification a behavioral description can always be obtained by observing only the highest (input) and lowest (receiving or output)nodes. Moreover, if each of the components is constructively specified a constructively specification is easily derived.

C.4. The Hierarchy

The hierarchy of system specification is given in Table 1. Conceptually, the network specification represents the most detailed level of knowledge of the real system. On a technical level the network model may be a bit more easily constructed than a direct attempt to constructively specify the general input/output behavior of the system.

The most important, and frequently most difficult task, in the constructive specification development is the definition of a technically feasible and conceptually appealing state space. For a network model a state space is defined for each component subsystem. Decomposing the system model into simpler subsystems often makes the definition of the state space for each component a significantly easier task. In addition, the network construction reduces the absolute number of state transition probabilities which must be defined (see Miller, 1979 for further details). Given a network with each component constructively specified having state space, C_i, the state space for the network is

$$C = \underset{i}{X}\ C_i$$

the cartisian product of the component state spaces. In a similar manner, the transition functions and the output assignment functions of the components can be cancontatonated to derive the system model functions required for the system model specification of hierarchy level II.

Given a constructive specification it is a simple manner to strip off the additional information provided by the functions and reproduce the behavioral description of level I.

Table 1. -- Hierarchy of System Specifications

Level	Specification	Formal Object
I	Behavioral Description	$S = \langle A, B, T \rangle$
II	Constructive Specification	$S = \langle A, B, C, T, \Psi, \Phi, \lambda \rangle$
III	Network Specification*	$S = \langle G, N, L, \; S_i \rangle$ where I N and $S_i = \langle A_i, B_i, C_i, T, \Psi, \Phi, \lambda, \rangle$

*This assumes that each component system model is constructively specified. If this is not the case, S_i is a behavioral description, i.e., $S_i = A_i, B_i, T$

D. Input/Output Validity

Knowledge and hypotheses about the real system can be divided into two rough categories: behavioral and structural. Behavioral knowledge is that which describes the macro or overall behavior of the system. At this level the system is a black box with behavior viewed as input/output responses. Except for attempting to structure the gross input/output behavior via state representation and transition functions there is little concern with the internal structure of the system. The validity issues at this level are designated input/output validity issues. Those issues which concern the preservation of internal structure, that is, how the system produces such behavior, are called structural validity issues and will be discussed in the next section.

Input/Output validity is concerned with system model specifications at hierarchy levels I and II, the behavioral levels. If the system model specification is at level III, the network level, the behavioral description is abstracted from it in order to reduce the model to either a behavioral description or a constructive specification.

Input/Output validity has two dimensions, one at the behavioral description level, called behavioral equivalence, and the second at the constructive specification level, called transition/output equivalence. The hierarchical level of specification in the base model will determine the input/output (I/O) validity dimension to be investigated in order to determine the validity of the system model.

In the subsequent discussion, two set-theoretic structures are employed; they are the base model and the extensional model. To simplify the descriptions these systems are designated S_1 and S_2 respectively, and called systems. This short hand does not imply any blurring of the distinction between the real system, the description of the system, and system models, however.

D. 1. Behavioral Equivalence

Comparing S and S_2 at a behavioral level necessitates that both systems be specified at the behavioral description level. That is,

$$S_1 \subseteq X_1 \times Y_1 \qquad \text{thus,} \quad S_1 = \langle A_1, B_1, T \rangle$$

and

$$S_2 \subseteq X_2 \times Y_2 \qquad \text{thus,} \quad S_2 = \langle A_2, B_2, T \rangle$$

Now the intuitive notion of validity, the preservation of the theorized properties and behaviors, can be formalized by a map from one structure to the other, i.e., a map f so that

$$f: \quad S_1 \longrightarrow S_2$$

The map, f, represents the elements of S_1 in S_2 so that the properties postulated by the theory and structured in the base model are preserved in the extensional model.

If those properties are preserved exactly, with no modification, abstraction, or aggregation f might take the form of an identity map, an inclusion relation.

$$S_1 \hookrightarrow S_2$$

Such a restriction is a fairly sweeping assumption, and the more general notion of map will be used instead.

Although the function, f, interprets the intuitive idea of validity as property preservation, maps which are much more specific about model input and output preservations are needed to characterize behavior equivalence.

In particular, weak behavioral equivalence of the two systems, S_1 and S_2 is defined by maps f_1 and f_2 where

$$f_1: \quad X_1 \rightarrow X_2$$

and

$$f_2: \quad Y_1 \rightarrow Y_2$$

S_2 is said to be behaviorally equivalent to S_1 with respect to the "interpretations" f_1 and f_2 if and only if,

for each $(x_1, y_1) \in S_1$

$(f_1(x_1), f_2(y_1)) \in S_2$

Thus, every behavior found in S_1 has an image in S_2 and S_2 preserves the behaviors found in S_1.

A stronger notion of behavioral equivalence which allows preservation and interpretation of the instantaneous behavior over time of the base model in the extension model requires three maps g_1, g_2 and g_3 so that:

$$g_1: \quad T_1 \longrightarrow T_2$$

$$g_2: \quad A_1 \longrightarrow A_2$$

That is, S_2 is (strong) behaviorally equivalent to S_1 if and only if given $(x_1, y_1) \in S_1$ there is an element $(x_2, y_2) \in S2$ so that

$$x_2 \circ g_1 = g_2 \circ x_1$$

and

$$y_2 \circ g_1 = g_3 \circ y_1$$

(Recall, in general, $x: T \to A$ and $y: T \to B$)

Clearly, strong behavioral equ'valence implies weak behavioral equivalence. The three maps defining strong behavioral equivalence identify the sequence of inputs and outputs that occur in S_2 corresponding to a given pair of sequences in S_1. That is, given a particular input value at time t, say $x_1(t) \in A$, the map g_2 interprets the input value in S_2, and $g_1(t) \in T_2$ identifies the time point in S_2 which corresponds to t in S_1. Thus, given $(x_1, y_1) \in S_1$ the corresponding input sequence in S_2 is found by evaluating x, at each element of T_1, and identifying its corresponding input value in A_2. By concatonating the pairs $(g_1(t), g_2(x(t)))$, for each $t \in T$, an input function can be identified. The output function is similarly constructed.

This type of equivalence is particularly helpful in the case of system models with finite input and output alphabets, system models where the input and output is assumed to remain constant over a "chunk" of time. Strong behavioral equivalence requires specification of the preservation of individual input and output values in the extensional model as well as the more general preservation of gross trajectory inputs and responses, i.e., behaviors.

D.2. Transition/Output Equivalence

The second dimension of I/O validity is at a higher level of the specification hierarchy, the constructive specification level. Thus, it is assumed that the base model S_1 is specified by

$$S_1 = \langle A_1, B_1, C_1, T_1, \Psi_1, \Phi_1, \lambda_1 \rangle$$

where

$$\lambda_1: \quad C_1 \longrightarrow B_1$$

$$\Psi_1: \quad C_1 \times T_1 \times A_1 \times T_1 \longrightarrow C_1 \times T$$

$$\Phi_1: \quad C_1 \times T_1 \times A \times T_1 \longrightarrow M_1$$

where

$$M_1 = \left\{ m \mid m: C_1 \longrightarrow [0,1] \text{ so that } \sum_c m/c) = 1,\ m/c \geq o \right\}$$

S_2, the extensional model, is specified by

$$S_2 = \langle A_2, B_2, C_2, T_2, \Psi_2, \Phi_2, \lambda_2 \rangle$$

As implied by the definitions of S_1 and S_2 the sets and maps specifying the models may be different; however, this is not required. For example, the time sets may well be the same.

Transition/Output equivalence is given in terms of five maps which interpret the state transition and output assignment behaviors of the base model in the extension model. The maps are:

$$g_1: \quad T_1 \longrightarrow T_2$$

$$g_2: \quad A_1 \longrightarrow A_2$$

$$g_3: \quad B_1 \longrightarrow B_2$$

$$g_4: \quad C_1 \longrightarrow C_2$$

$$g_5: \quad M_1 \longrightarrow M_2$$

Given these maps, S_2, is the equivalent with respect to the state transition and output behavior to S_1 if and only if the following relations hold.

The map

$$g_1: T_1 \longrightarrow T_2$$

has the following property

$$\forall t_1, t_2 \in T_1$$

$$g_1(t_1 + t_2) = g_1(t_1) + g_1(t_2)$$

(Recall: The time set is a set, T_1, with a binary relation, which linearly orders the set).

For each $x_1 \in X_1 = A_1^{T_1}$ there exists $X_2 \in {}_2 = A_2^{T_2}$ so that

$$\begin{array}{ccc} & x_1 & \\ T_1 & \longrightarrow & A_1 \\ g_1 \downarrow & & \downarrow g_2 \\ & x_2 & \\ T_2 & \longrightarrow & A_2 \end{array}$$

i.e. for each $t_1 \in T_1$

$$g_2(x_1(t_1)) = x_2(g_1(t_1))$$

$$\begin{array}{ccc} & \Phi_1 & \\ C_1 \times T_1 \times A_1 \times T_1 & \longrightarrow & M_1 \\ g_4 \downarrow \quad g_1 \downarrow \quad g_2 \downarrow \quad g_1 \downarrow & & \downarrow g_5 \\ & \Phi_2 & \\ C_2 \times T_2 \times A_2 \times T_2 & \longrightarrow & M_2 \end{array}$$

i.e., for each $(c_1, t_1, g_1, s_1) \in C_1 \times T_1 \times g_5(\Phi_1(c_1, t_1, g_1, p_1)) =$
$\Phi 2(g_4(c_1), g, (t_1), g_2(g_1), g_1(s_1))$

$$
\begin{array}{ccccc}
 & C_1 & \xrightarrow{\lambda_1} & B_1 & \\
g_4 & \downarrow & & \downarrow & g_3 \\
 & C_2 & \xrightarrow{\lambda_2} & B_2 &
\end{array}
$$

Thus, transition/output equivalence implies the existence of strong behavioral equivalence. Transition/Output equivalence preserves and interprets not only the instantaneous inputs and outputs of S_1 in S_2 but also the state behavior. The state transition behavior provides a "higher" level of knowledge of the system; the preservation of this knowledge in the extensional model is an important characteristic of the intuitive notion of validity.

D.3. The Nature of Maps

Level II validity requires the preservation of observed behavior and postulated properties, as realized in the base model, in the extensional model. The only requirement of the map defining the relationship between the base and extensional models is that it be a function. That is, given a set of behaviors in the base model, a corresponding set in the extensional model is uniquely identified by the map.

In the strictest sense of preservation, the map might be an identity or inclusion map. In this case, the behavior of the base model is identical to its corresponding behavior in the extensional model. Blurring the distinct behaviors of the base model somewhat, the requirement of the map can be relaxed so that it is a one to one function. Thus, there is an image of S_1 in S_2 which is, in a loose sense, isomorphic to S_1. Each behavior in S_1 is interpretted, rather than strictly preserved, in S_2 and each behavior in the image of S_1 represents a unique behavior or property of S_1. These relations between models give a clear sense of the extensional model as a "richer" or more complete model than the base model. The behaviors and properties of the base model are presumed or interpretted and subsequently augmented and enriched to form the extensional model.

In the least restrictive sense of map, behaviors and properties of the base model may be aggregated, condensed, or "lumped" (Ziegler; 1976) by the interpretation. Mathematically, this implies that the map is not one to one. Conceptually, the map partitions the system model S_1 into sets of behaviors which are preserved in some form by one element of S_2; thus, the map can be said to induce an equivalence relation on the elements of S_1 with equivalence classes consisting of the partitioned sets. As in the case of inclusion and one to one maps, the extensional model, S_2, is expected to be at least as rich and complete as S_1.

E. Structural Validity

Input/Output validity is concerned with the preservation of behavioral properties that the real system is believed to possess. The preservation of state and state transition information and properties begins to organize or "functionalize" the behavior of the black box, the current representation of the real system; however, the preservation of state transition and output information remains primarily a matter of external validity. Internal validity addresses the problem of the contents of the black box; that is, to what extent are the conjectures and information about the internal structure of the real system, as defined in the base model, preserved in the extensional model.

In order to talk about structural validity at all it is necessary that the base model be given a network specification. That is,

$$S = \langle G, N, L, \{Si\} \rangle$$

where

N is the name of the system model components.
B defines the communication links between components.
L defines the exact communication channels between components.
$\{S_i\}$ is a set, $i \in N$, consisting of system model components.

The individual components may be specified behaviorally or constructively; but for each component, $i \in N$, the behavioral description is

$$(1) \quad S_i \subseteq X_i \times Y_i$$

where

$$X_i = A_i^T \text{ and } Y_i = B_i^T$$

The input and output alphabets can be further decomposed as

$$A_i = A_{i_1} \times A_{i_2} \times \ldots \times A_{i_m} = \mathop{X}_{j=1}^{n} A_{i_j}$$

and

$$B_i = B_{i_1} \times B_{i_2} \times \ldots \times B_{i_m} = \mathop{X}_{j=1} B_{i_j}$$

Thus, equation (1) can be rewritten as

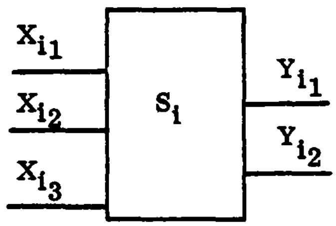
$$(2) \quad S_i \subseteq (a_{i_1} \times \ldots \times A_{i_m})^T \times (B_{i_1} \times \ldots \times B_{i_m})^T$$

Thus, an input trajectory for system model i can be considered as a sequence of ordered m-triples defining the inputs to the system at $t \in T$: the output trajectory can be decomposed in a similar manner.

For example, consider a system model component with three inputs and two outputs

$$S_i \subseteq (A_{i_1} \times A_{i_2} \times A_{i_3})^T \times (B_{i_1} \times B_{i_2})^T$$

This component can be graphically portrayed as



where for each t T

$$X_{i1}(t) \in A_{i1}$$

$$X_{i_2}(t) \in A_{i_2}$$

$$X_{i_3}(t) \in A_{i_3}$$

$$Y_{i_1}(t) \in B_{i_1}$$

$$Y_{i_2}(t) \in B_{i_2}$$

As the function of structural validity is the preservation of the input/output links between the internal components it is necessary to work with the more complete behavioral description of equation (2) rather than equation (1).

The relation, L, defines the communication links between components. Given

$$L \subseteq G \times (I \times I)$$

where

$$G \subseteq N \times N$$

and I is the set of positive integers and N is the set of names of system model components.

Thus,

$$((i,j), (s,t)) \in L$$

if and only if output of component i is input t to component j. Therefore, there exists components S_i and S_j where

$$S_i \subseteq (A_{i_1} \times \ldots \times A_{i_{m_i}})^T \times (B_{i_1} \times \ldots \times B_{i_s} \times \ldots \times B_{i_{m_i}})^T$$

and

$$S_j \subseteq (A_{j_1} \times \ldots \times A_{j_t} \times \ldots \times A_{j_{m_j}})^T \times (B_{j_1} \times \ldots \times B_{j_{m_j}})^T$$

where

$$B_{i_s} = A_{j_t}$$

Let the base model be denoted

$$S^1 = \langle G_1, N_1, L_1, \{S_i^1\} \rangle$$

and the extensional model as

$$S^2 = \langle G_2, N_2, L_2, \{S_i^2\} \rangle$$

Structural equivalence of S_2 to S_1 requires the following maps:

(1) $f: N_1 \longrightarrow N_2$

(2) $\{g_i \mid i \in N_1\}$

where

$$g_i: I \longrightarrow I \cup \{0\}$$

so that g_i (k) = o if k is not a name of an input to S_i^1

else g_i (k) $\in$ I so that g_i (k) is an input to $S^2_{f(i)}$

(3) $\{h_i | i \quad N_1\}$

h_i: I $\longrightarrow$ I U $\{0\}$

so that h_i (k) = 0 if k is not an output name for S_i^1

else h_i (k) $\in$ I and g (k) is an output name for $S^2_{f(i)}$

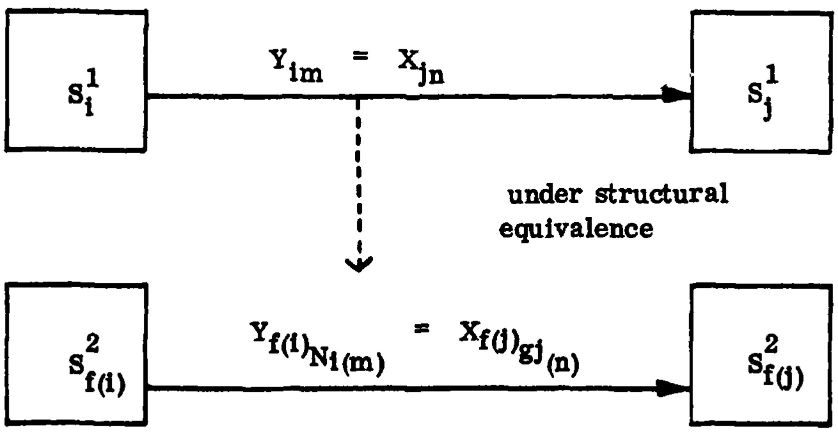
The map, f, interprets the components of the base model in the extensional model. Given the definition for the relationships between components, the maps g_i and h_i interpret the respective individual inputs and outputs of S_i^1 in $S^2_{f(i)}$. Thus, given a specific relation between components i and j in S^2, i.e.

$$((f/i), f/j)), (h_i (m), g_j (m))) \quad L_2$$

so that

$$B_{f(i)_{h_{i(m)}}} = A_{f(j)_{g_{j(n)}}}$$

Graphically, this is portrayed as



where, for each $t \in T$

$$Y_{i_m}(t) \in B_{i_m}$$

$$X_{j_n}(t) \in A_{j_n}$$

$$Y_{f(i)_{h_i(m)}}(t) \in B_{f(i)_{h_i(m)}}$$

$$X_{f(j)_{g_i(m)}}(t) \in A_{f(j)_{g_i(m)}}$$

This example illustrates the difference between input/output validity and structural validity. Structural validity is concerned with linkages between components but ignores the input/output behavior (i.e., specific input/output values) of the components. Structural validity guarantees the existence of a unique output (input) channel in $S^2_{f(i)}$ corresponding to a given specific output (input) channel in S^1_i. However, there is no mechanism provided by the structural equivalence maps to interpret the information that passes through the channel. Given an output sequence from component S^1_i, structure preserving maps show where to look for the corresponding output sequence in $S^2_{f(i)}$ but provide no dictionary to translate the values of the responses from one model to the other. That is the function of the input/output preservation maps.

In most validity discussions there is a need for both types of preservation. Typically, input/output issues receive the bulk of the attention. As the real system is assumed better understood, the focus of attention often shifts to more structural issues. Substantive differences between structural and input.output validity concerns are discussed in Chapter 5.

F. Summary of Level I Validity Dimensions

Given a base model which structurally interprets the theory and the data structure, level I validity is defined in terms of maps from the base to the extensional model. Level I validity concerns the preservation of the behaviors defined by the data structure and the properties postulated by the theory. Such concerns can be divided into two categories: Internal and external.

External validity or input/output validity addresses the preservation of behavioral responses of the system model. In its most rudimentary form weak behavioral validity presumes the existence of maps interpretting input and output trajectories from the base model in the intensional model. At a more detailed level, strong behavioral equivalence between the two models requires maps which interpret individual input and output values. Behavioral equivalence relates system specified at the lowest level of the specification hierarchy.

For constructively specified models, validity concerns include state transition and output behavior preservation. Models which are equivalent with respect to state transaction and output behavior require maps which interpret individual input and output values as well as state values and state transition distributions. Input/Output validity is hierarchical in that the highest level validity, transition/output, implies the next highest, strong behavioral equivalence, which, in turn, implies weak behavioral equivalence.

If the internal structure of the base model has been specified, i.e., it has a network specification, structural validity requires maps which relate components and linkings among components in the base model to those in the extensional model.

Structural validity and input/output validity are distinct yet complimentary concepts. The former concerns the internal linkages among subsystems - the communication channels within the system; the latter concerns the content of the messages which pass through the channels. Input/output validity may be an issue at the system level - thus, ignoring all but the highest level input channels and the lowest level output channels, or it may concern the behavior of individual subsystems.

The validity dimensions delineated in this discussion are summarized in Table 2. The major point is that one needs to be very precise about what is meant by validity, or what type of validity is required. A model may be externally valid (input/output) and fail to be internally (structurally) valid, or visa versa. Or a model may be valid in the sense that general behaviors are preserved but individual response values cannot be identified. Furthermore, given two extensional models of a common base model determining which is "more valid" is far from trivial task. Some of the methodological problems inherent in such an exercise are discussed in the next chapter.

Table 2

Validity Level	Associated Maps	Specification Level
I. Input/Output Validity		
1. Behavioral Equivalence a) Weak	$f_1: X_1 \rightarrow X_2$ $f_2: Y_1 \rightarrow Y_2$	Behavioral Description $S = \langle A, B, T \rangle$ where $X = A^T$ and $Y = B^T$
b) Strong	$g_1: T_1 \rightarrow T_2$ $g_2: A_1 \rightarrow A_2$ $g_3: B_1 \rightarrow B_2$	
2. Transition/Output Equivalence	$g_1: T_1 \rightarrow T_2$ $g_2: A_1 \rightarrow A_2$ $g_3: B_1 \rightarrow B_2$ $g_4: C_1 \rightarrow C_2$ $g_5: M_1 \rightarrow M_2$	Constructive Specification $S = \langle A, B, C, T, \Psi, \phi, \lambda \rangle$
II. Structural Validity	$f: N_1 \rightarrow N_2$ $\{g_i / g_i: I \rightarrow I \cup \{0\}\ i \in n_1\}$ $\{h_i / h_i: I \rightarrow I \cup \{0\}\ i \in n_1\}$	Network Specification $S = \langle G, N, L, \{Si\} \rangle$

2.5 METHODOLOGICAL ISSUES

A. The Question

Methodologically the issues concerning validity are pointed: When can a model be deemed valid? How is validity to be measured? Given two or more models, how are they to be compared with respect to validity dimensions?

The preceding portion of this paper provides a framework within which model validity can be more precisely discussed. In the most general sense, a valid model is defined as a model possessing the required properties Further, it was demonstrated that the required properties are those embedded in the theory and data structure constituting the description of the real system--the modeller's explicit knowledge and hypotheses of the system Validity then, is primarily an issue concerning the preservation in the model of those properties and behaviors which have been made explicit in the description.

In its most intuitive form, validity is a purely subjective, nonquantitative judgment call. It entails an act of faith by the modeller or user attesting to the model's validity. Comparison among models is possible using an ordinal scale which structures such perceptions. This form of validity, though perhaps the most common, is the least desirable. There is little ability to communicate the reasons for the judgment; thus widespread utility of such models is severely circumscribed.

It is always desirable and frequently possible to be more specific about the validity of a model. However, the subjective component of validity can never be completely eliminated; it is simply diminished in importance through augmentation with more replicable or well reasoned procedures. Furthermore, model validity is a relative issue. Forrester (1968), a foremost advocate of the relativistic view of model validity and utility, suggests that validity must be measured on a relative scale--relative to what is known about the real system and relative to the purpose for the particular modelling exercise.

The former dependence, relative to the knowledge of the real system, is explicitly incorporated into the definition of validity presented in this discussion. The latter is formalized by the kinds of equivalence maps established between the base model and the extensional model. If a model reflecting the behavior of the system, as viewed externally (the exterior of the black box), is desired, then validity requires the existence of input/output behavior preserving maps and measures. If the focus is on the internal structure then structure preserving maps and measures are needed.

The following sections address some of the problems and issues involved in the quantitative assessment of validity for both external and internal dimensions.

B. *Measures of Worth: "Does It Behave the Way It Is Supposed to Behave?"*

The most frequent means of making a quantitative determination of the validity of a model employs figures of merit or measures of fit. In almost all cases, validity determined by such procedures concerns the relation between behaviors or data produced by the model and the real system behaviors as represented in the data structure or characterized in the statements of the theory, i.e., external validity.

This perspective on validity is cogently presented by Zeigler.

> . . . we presume to explicate the ordinary usage of the term "valid" when applied in relation to a real system. We are asserting that validity can be established, if at all, only on empirical grounds (i.e., by comparison of the data collected from the real system with those generated by the model).

The underlying assumption which few discussions based on this view of validity make explicit (Zeigler does state this assumption) is that the world in general, and the real system of interest in particular, is nothing more than a source of potentially acquirable data. Given this predisposition, it is a natural consequence to focus the preponderance of validity questions and measures on the data matching capabilities of a model.

This interpretation of validity was formalized in the previous chapter by means of the input/output equivalence maps. Specifically, a constructed model is valid with respect to external or input/output behavior if and only if there exists maps defining either weak behavioral equivalence, strong behavioral equivalence, or transition/output equivalence of the constructed model with respect to the base model. Zeigler posits an even stronger condition. He requires that the maps at the weak behavioral level be invertible identity maps. Thus,

$$X_1 = X_2 \text{ and } Y_1 = Y_2$$

so that input/output validity requires

$$S_1 = S_2$$

This property may well be desirable in some cases, but is needlessly restricting in others.

To this point it has been assumed that validity is an either-or proposition: the extensional model is either valid, as evidenced by the existence of behavior or structure preserving maps, or it is not. However, in almost any modelling exercise a looser notion of validity, employing approximation, is needed and used.

The concepts developed for approximation are sketched at the weak behavioral equivalence level. The ideas generalize to other input/output validity levels; moreover, any model specified at a higher level of the specification hierarchy can be reduced to a behavioral description.

Strict validity requires that input/output behaviors in the base model as interpreted by the equivalence maps, are also behaviors in the constructed model. That is, for each $(x_1, y_1) \in S_1$, there exists some $(x_2, y_2) \in S_2$ so that

$$f_1(x_1) = x_2 \text{ and } f_2(y_1) = y_2$$

This strict agreement between the base and constructed models is frequently undesirable or impossible. Approximate models are often quite useful so that the additional resources and costs required to establish equality may be unwarranted. Equality may be an illusion even when established; measurement error or other biases may have tainted the data structure. For constructed models with stochastic structure rigid equality requirements are unreasonable. Thus, figures of merit or measure of fit are employed to determine how well the sets of model behaviors agree Such measures, frequently called goodness of fit statistics, are fundamentally important and somewhat subjective choices made by the modeller in the effort to establish the validity of a model.

Abstractly, the notion of approximation is characterized through the use of a metric which in some sense "measures the amount of error (deviation)" between the interpreted behavior of the base model and the corresponding behaviors in the constructed model. Given a set Z, a metric is defined as a map d so that

$$d: Z \times Z \longrightarrow R$$

where (1) $d(z, z) = 0, \; z \in Z$

(2) $d(z, y) = d(y, z), \; y, z \in Z$

(3) $d(x, y) + d(y, z) \geq d(x, z), \; x, y, z \in Z$

The construction of such metrics is not a trivial task. Additional details for the abstract development of metrics used in approximation are given by Zeigler (1976) and Miller (1976). Examples of applications of specific metrics are easier to present. The use of statistical measures of fit, such as the chi-square statistic and the coefficient of multiple determination (R^2), is widespread in model testing and confirmation and is well documented in the literature. Suppes (1969b) provides a helpful conceptualization of the use of such statistical measures of fit in model validation. Given a theory of a set of data, Suppes suggests statistical methods, such as those mentioned above, are used to determine if a realization of data is a model of the theory of the data.

The type of validity determined by measures of merit is that which concerns the relation of behavior produced by the model to those contained in the data structure or characterized by the theory--at level I it is replicative validity and at level II it is predictive validity (Zeigler, 1976).

Approximation, as structured by measures of merit, is a basic issue in validity and opens up the whole subjective infrastructure below the quantitative methods dimension. The choice of measures, the acceptable or tolerance values for the measures are all subjective judgments and frequently depend on what is known or what is sought in the modelling exercise. Thus, validity, even at its most "objective," is highly subjective, a matter of degree and intent, clearly not an either-or determination.

Validity questions which can be measured by "objective" techniques and models which lend themselves to such measures tend to overshadow any alternatives. This is precisely due to the existence of a procedure to measure. There is security in replicable procedures which produce a number. Thus, the focus is frequently on the development of models which have a high degree of external or behavioral (i.e., input/output) validity. Engineering approaches frequently produce behaviorally descriptive models using "curve fitting exercises in which parameters are adjusted to fit the specific situation" (Miller, 1979b) with little concern for the underlying process producing the behavior. Taken to an extreme, the result is Milton Friedman's (1953) controversial "as if" criteria for model validity: a necessary and sufficient criterion of model validity is the capability of the model to produce behavior as if it was the real system. Friedman challenges the merit of any other internal or structural validity criteria. Taking a less extreme stance it seems appropriate to solely employ external criteria only if the sole intent is to construct a descriptive model which produces behaviors similar to those of the real system. If, however a more explanatory model is required, a model which attempts to represent portions of the underlying processes of the real system structural validity issues must also be addressed.

C. Internal Validity: "What Is Inside the Black Box?"

A model possessing internal or structural validity is, intuitively, one which successfully incorporates some or all of the underlying processes in the real system.

Zeigler (1976) defines a structurally valid model as a model which not only "reproduces the observed real system behavior, but truly reflects the way in which the real system operates to produce this behavior (p. 5)."

Formally, a structurally valid model is one in which the relations (communication channels) between the subsystems of the base model are preserved in subsystems of the constructed model. It is the theory, or more precisely, the partial theory, which contains statements making conjectures concerning the actual processes employed by the real system to produce the observed behaviors. The base model is defined as a primitive, set-theoretic structure interpreting, in a very rudimentary way, the statements of that theory and containing, as a superset, the behaviors in the data structure. In a loose sense, input/output validity addresses the relationship between the data structure and the model, whereas internal validity addresses the relation between the theory and the model. Suppes (1969b) develops this idea even further by means of a hierarchy of theories of the real system, the experiments, and the data with a model realizing the theory at each level.

This distinction between behavior preservation and structure preservation is helpful in understanding the distinction frequently made between descriptive and explanatory models. A descriptive model is one which produces behaviors identical, within some standard of approximation, to real system generated behaviors without representing the processes in the real system generating those behaviors, a strictly input/output valid model. Such data oriented models often sacrifice structural fidelity and contextural sensitivity in order to be mathematically tractable (e.g., most engineering models of the human operator, see Kelley, 1968).

Explanatory models, on the other hand, put a high priority in representing hypothesized processes of the real system. Such theory based models attempt to incorporate the results of substantive theoretical as well as empirical knowledge of the system. Unfortunately, truly explanatory models are often mathematically intractable or cumbersome, and in many cases severely limited by a lack of suitable mathematical tools. Kelley, in describing the Laplace transform model of the human operator, suggests that the terms in the model represent descriptions, as opposed to explanations, "of processes more veridically represented by quite different forms of mathematical descriptions, which remain to be developed." *(More will be said in the next section on models of the human operator.)*

In addition to cumbersome or nonexistent mathematics, explanatory models are limited by a lack of quantitative measures to ensure internal validity. In most cases, the "inside" of the real system cannot be seen or examined. At best, the processes and structure of the real system can be inferred either from its behavior or by means of related theories. This lack of well defined, quantitative measures of internal validity explains, in part, the modellers who forsake internal validity issues and replace the real system by a concept of "data source," thus reducing all the validity questions to data matching questions.

Even the validity framework developed in the previous chapters offers a way to side-step meaninful structural considerations. A valid model, possessing validity, is one in which the statements of the partial theory are interpreted as true. It is technically possible to posit a substantively shallow or vacuous partial theory using a rich initial theory. Models for restructuration realize theories for which one or more tenets were dropped. In the extreme case, all theory tenets can be deleted creating a substantively void partial theory. The partial theory and data structure are interpreted by the base model; structural validity, as formally defined, merely requires that the structure of the base model be presumed in the extensional model. Thus, a substantively shallow partial theory yields a base model closely resembling the data structure for which it is a superset; the modelling processes may be reduced to a curve fitting exercise with the resultant model structurally valid, by definition--if there is no conjectured structure in the partial theory then any model of the theory is structurally valid. Such a procedure, though technically feasible, violates the spirit of the validity discussion as well as the intent of most modelling endeavors.

For the case of models for completion or a model for meaningful restructuration the questions still remain: What constitutes an internally valid model? What are the rules of evidence used in an internal validity determination? Though not well defined and, as a set, certainly not sufficient, there are a number of criteria commonly employed in such determinations.

At a type I level, a necessary condition for internal validity requires both input/output (behavioral) and structure preserving maps which define the equivalence between the constructed and base models. The existence of these maps as well as the specific type (e.g., identity, one to one, onto) and the allowed tolerance for approximation are partial determinants of the internal validity of the model.

The next set of criteria involve the relation between the theory and partial theory. An internally valid model requires a high degree of fidelity of the partial theory with respect to the original theory. Tenets discarded for reasons of pure expedience are contraindicators of a high degree of internal validity.

A widely used criteria of the internal validity of a model is the incorporation, into both the original theory and the partial theory, of statements of hypotheses which are compatible with existing and related theories for either the system of interest or related systems. Thus, a model which interprets statements consistent with other theories is judged to be a more internally valid structure than one which fails to do so.

Two commonly employed corrollaries to the last criterion utilize the judgment of the audience or scholarly community who have an interest in the system or are familiar with related theories and models. The first is the notion presented under the rubric of level II validity; that is, the behavior and structure of the model may not violate the intuition of the community at large. The second criterion has a somewhat long term time frame; it says that the internal validity of a model can be ascertained by the eventual acceptance of the model as internally valid by the interested community.

A note of caution is required concerning the application of such criteria. On occasion, a model or perhaps the theory on which the model is based may initially be found unacceptable by the interested community, at odds with its current theories. Such response does not automatically negate the internal validity of the model. Many fundamental theories which have revolutionized science were initially received by a hostile audience (see Kahn, 1970). In the time from public hostility to eventual acclaim the internal validity did not change, merely the perception changed. Thus, validity determinations which rest on the opinion of the relevant audience are neither necessary nor sufficient conditions for internal validity; they are merely indications.

The subjective nature of the criteria used to determine internal validity should be judged in light of the, at times, equally subjective, though quantitative, criteria used in input/output validity determinations. Consider the chi-square test for goodness of fit frequently used for stochastic models. As generally applied, the null hypothesis, conjecturing the underlying distribution for a set of data, is the hypothesis which the researcher would like to confirm, as opposed to the typical hypothesis testing approach where the research hypothesis is given as the alternative hypothesis. In goodness of fit applications the significance level of the test is increased, often allowing the probability of committing a type I error to exceed 30 percent, in order to make easier to reject the null hypothesis; intuitively, this makes the test more conservative. Statistically, however, failing to reject the null hypothesis does not imply that it has been confirmed, only that there did not exist enough evidence to reject it. Though numeric and repeatable this statistical procedure is merely an indicator, similar to many of those used in internal validity determinations, of the behavioral validity of the model.

A final and obvious point should be noted about the internal versus external validity issue. In most cases, external validity is a comparatively easy attribute with which to endow a model. Techniques and methods to ensure reasonable external validity are fairly well defined and the mathematics is usually available. This is rarely the case when a high degree of internal fidelity is required. Thus, the purposes of modelling should determine the types of validity requirements. If a descriptive model will suffice there is little reason to expend the additional effort development of an explanatory model will likely require. If, however, the purposes of modelling include finding answers to "why" questions about the real system, then internal validity issues must be addressed.

The final section illustrates some of the differences in form and focus between descriptive and explanatory models by describing two approaches to modelling the behavior of human operators.

D. Descriptive versus Explanatory Models: An Example

This section will use two models to illustrate some of the issues involved in validity determination. The first model will be one that is used to represent the linear portion of the human operator's response in manual tracking; this description is based on a discussion by Kelley (1968) who provides an excellent introduction to frequently employed engineering models of the human operator. The second is the descrete control model developed by Miller (1978, 1979) of a simulated antiaircraft artillery (AAA) system.

Kelley presents the control model as follows (p. 186).

> The Laplace transform model of the human operator that represents the linear portion of the human operator's response in manual tracking systems is most often expressed in the form
>
> (1) $$H(s) = \frac{Ke^{-\tau s}(1 + T_L s)}{(1 + T_N s)(1 + T_I s)}$$
>
> where
>
> $H(s)$ = linear portion of the human operator transfer function
> K = operator gain
> $e^{-\tau s}$ = reaction-time delay
> T_L = lead time constant
> T_N = neuromuscular lag time constant
> T_I = compensatory lag time constant
>
> The above function of the Laplace operator s corresponds to the following time domain expression, in which θ_o and θ_i are the operator's input and output, respectively:
>
> (2) $$\theta_0(t) + (T_N + T_I)\frac{d\theta_0(t)}{dt} + T_N T_I \frac{d^2\theta_0(t)}{dt^2} = K\left[\theta_i(t - \tau) + T_L \frac{d\theta_i(t - \tau)}{dt}\right]$$
>
> Thus in the "standard" human operator model, the weighted sum of the operator's output, rate of change of output, and output acceleration are proportional to a weighted sum of his input plus its rate of change, both of the latter being taken τ sec previously.

Kelley cautions that there is no reason to suppose that equation (2) explains how the human operation performs the task. It is merely an analytic description which produces behavior "as if" it were the human operator. By definition the model only provides a linear description of the human operator's response. Current mathematical tools do not permit the extension of this model to include the well known and extensive nonlinear responses evidenced in most manual tracking studies. This limitation affects both the external and internal validity of the model.

The transform model of the human operator has been useful in describing the response of the human operator. Such models "have served to outline with a nice precision of expression certain boundaries of human performance. They have employed a convenient means of expressing data about the human operator." (Kelley, 1968) They serve as an excellent source of behavioral data. When only data and description is required such models with their concise mathematical representation provide a good model of control activities for various systems. However, in the framework of the validity conceptions developed in this discussion, the transform model has little internal validity; it is the proverbial black box.

Kelley, in the table reproduced below, gives an excellent critique of the model with respect to the common "theory" of the human operator. The deficiencies identified in the table are primarily related to the internal validity requirements of a model of a human operator.

The discrete control methodology (DCM) used in the model development of the AAA system (see Miller, 1978 and 1979) was formulated to allow the black box to be less opaque. For system design and evaluation problems requiring explanatory context sensitive models, DCM is a far more suitable approach. The AAA model

Comparison of Describing Function Model with Actual Human Operator

Describing Function Model	Human Operator
Input Narrowness	
Input has same number of dimensions as output	Input typically has more dimensions than output; operator reduces data
One display	Multiple displays
Assumes impoverished display format (compensatory or pursuit tracking)	May use highly sophisticated multidimensional displays (contact analog, predictor display, or direct view of environment)
Lacks Internal Task Representation	
Does not include any explicit representation of task or environment	Operation is vitally affected by understanding of task and environment
Cannot adapt to changes in task save through arbitrary parameter adjustments	Veridical changes in internal representation of task result in changed predictions and, hence, a different nonarbitrary form of adaptation
Point-in-Time Limitation	
Restricted to present error, fixed exponential weightings of past, and derivatives	Response based on remembered past and predicted future
Cannot remember; can only summarize signals via integration (lag)	Can remember, modify response, or change internal task representation in consequence of past experience
Cannot predict input or output; response is an arbitrary weighting of error, lead, and lag terms	Can predict; response can be formed and modified to minimize future (predicted) error. Can preview or anticipate input as well as predict output, and plan response based on both of these "excursions from present time"

explicitly incorporates multiple data sources and displays, modelling their use and effects by means of a hierarchical/heterarchical state space approach. The methodology was developed, in part, as a mechanism to represent the internal model of the operator or team of operators as well as their communication channels, information flow, and decision points.

A major problem in utilizing the DCM in modelling human-machine behavior, however, is the cumbersome state of the methods. At present clear cut, well defined procedures to use the methodology are lacking. This contrasts sharply with the Laplace transform model, whose concise mathematical representation is straightforward, if opaque.

These two modelling approaches represent, at least on the validity dimension, opposite ends of the spectrum and serve to illuminate the problems and issues inherent in model validity. In the last analysis the choice of a type of model or modelling approach depends upon the purpose of the modelling exercise. Given a purpose, the types of validity requirements can be specified. This discussion has offered a framework within which those requirements can be defined and evaluated.

Bibliography

Allwood, Jens, Lars-Cunnar Andersson, and Osten Dahl, Logic in Linguistics, Cambridge Textbooks in Linguistics, Cambridge, 1977.

Apostel, Leo, "Towards the Formal Study of Models in the Non-Formal Sciences", The Model in Mathematics, B. H. Kazemier and D. Vuysje (ed.), D. Reidel Publishing Co., Dordrecht-Holland, 1961.

Forrester, Jay W., Principles of Systems, MIT Press, Cambridge, MA, 1968.

Friedman, Milton, Essays in Positive Economics, University of Chicago Press, Chicago, 1953.

Kelley, Charles R., Manual and Automatic Control, John Wiley & Sons, Inc., N. Y., 1968.

Kuhn, Thomas S., The Structure of Scientific Revolutions, (2nd Edition), University of Chicago Press, Chicago, 1970.

Mackay, D. M., "The Mechanics of Knowing", IEEE Transactions in Systems, Man, and Cybernetics, January 1974, Vol. SMC-4, No. 1, pp. 94-95.

Miller, R. A., ISE 760 Text, unpublished, 1976.

Miller, R. A., Identification of Finite State Models of Human Operators, Final Report, Grant No. AFOSR-77-3152, 1979a.

Naylor, Thomas H., Computer Simulation Experiment with Models of Economic Systems, John Wiley & Sons, Inc., N. Y., 1971.

Naylor, Thomas H., Joseph L. Balentfy, Donald S. Burdick, and Kong Chu, Computer Simulation Techniques, John Wiley & Sons, Inc., 1968.

Polanyi, Michael, The Tacit Dimension, Doubleday & Company, Inc., Garden City, N. Y., 1966.

Przelecki, Marian, The Logic of Empirical Theories, Humanities Press, N. Y., 1969.

Shannon, Robert E., Systems Simulation, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975.

Stoll, Robert R., Set Theory and Logic, W. H. Freeman and Co., San Francisco, 1963.

Suppes, Patrick, "A Comparison of the Meaning and Uses of Models In Mathematics and the Empirical Sciences", Studies in the Methodology and Foundations of Science, D. Reidel Publishing Company, Humanities Press, N. Y., 1969a.

Suppes, Patrick, "Models of Data", Studies in the Methodology and Foundations of Science, D. Reidel Publishing Company, Humanities Press, N. Y. 1969b.

Tarski, Alfred, Introduction to Logic, Oxford University Press, N. Y., 1965.

Torgerson, Warren S., Theory and Methods of Scaling, John Wiley & Sons, Inc., N. Y., 1958.

Zeigler, Bernard P., Theory of Modelling and Simulation, John Wiley & Sons, Inc., N. Y., 1976.

3. PIECEWISE LINEAR APPROXIMATION TO SAMPLED CONTINUOUS DATA

The need for functional approximation of continuous data collected in any tracking exercise cannot be underestimated. Such approximation helps not only to reduce the data storage requirements, but also is very useful for identifying states, events and strategies of the human operator for discrete control modelling.

With the proposed method, continuous time series data are presumed to be stored in discrete time form. To this set of discrete data, a functional approximation is undertaken leading to data compression and a natural representation for discrete control analysis.

3.1 PROBLEM

The problem can be stated as follows: given a set S of discrete points, it is desired to split the set into proper subsets $U_1, U_2, U_3 \ldots U_n$ such that

$$S = \bigcup_{i=1}^{n} U_i$$

and that for each set u_i a functional approximation can be undertaken so that the total error for all u_i is minimized.

Here the functional approximation was limited to a linear approximation ---

$y_{ij} = a_i + b_i x_i$, $j=1, \ldots, n_i$, where n_i is the number of points in U_i.

Two error norms were tried, total squared error, and mean squared error. If t_{ij} is the j^{th} point in the set U_i, then

Total Square Error

$$TSE = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \left[t_{ij} - (a_i + b_i * j) \right]^2$$

Mean Square Error MSE= $$\sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} \left[t_{ij} - (a_i + b_i * j) \right]^2$$

The problem is to find the subsets $u_1, u_2, \ldots u_n$ given the set S, so as to minimize TSE or MSE.

The problem thus is a case of discrete optimization.

3.2 ANALYSIS

Many procedures exist in the literature to solve the above problem. (See Pavlidis 1974 for example.) The area of functional approximation, spline fitting has many algorithms, and one of them was selected to perform the task. (Pavlidis, 1973.) It was modified so as to suit the task at hand and implemented.

3.3 ALGORITHM:

Assume that at the k^{th} iteration, we have the subsets U_i^k, $i = 1, \ldots, n$:

Also $U_i^k \cap U_j^k = \phi \quad i \neq j, \quad \forall i, \forall j$.

Let the first element in U_i^k be $t_{i_1}^k$ and the last element be $= t_{in_i}^k$, since Ui has a cardinality of n_i.

$$\text{Hence} \qquad t_{in_i}^k = t_{i+1,o}^k \quad \text{and}$$

$$t_{in_i+1}^k = t_{i+1,1}^k \qquad \forall i = 1, \ldots, n.$$

Let $e^k = g(a,b)$ be the error norm in use, given (a,b) the end points.

$$e_i^k = g(t_{i1}^k, t_{in_i}^k) = \sum_{j=1}^{N_i} \left[t_{ij} - (a_i + b_i * j) \right]^2$$

if LSE is used.

Consider the following procedure.

3.4 PROCEDURE (Adapted from Pavlidis (1973))

Step 1. Set flag = 0.

for i = 1, 3, 5, . . . Compare the error norms E_i^k and E_{i+1}^k

If $E_i^k > E_{i+1}^k$ then

set $U_i = U_i^k - t_{in_i}^k$

and $U_{i+1} = U_{i+1}^k \cup \{t_{in_i}^k\}$

i.e. transfer the last point from the i^{th} segment to the $(i+1)^{st}$ segment.

If $E_i^k < E_{i+1}^k$, then

set $U_i = U_i^k \cup \{t_{i+1,1}^k\}$

$U_{i+1} = U_{i+1}^k - \{t_{i+1,1}^k\}$.

i.e. transfer the first point from the $(i + 1)^{st}$ segment to the i^{th} segment.

Step 2. Evaluate the new error norms.

E_i^{k1} and E_{i+1}^{k1} for the sets U_i^1, U_{i+1}^1

If $\max(E_i^{k1}, E_{i+1}^{k1}) < \max(E_i^k, E_{i+1}^k)$

set $U_i^{k+1} = U_i$ $\quad E_i^{k+1} = E_i^{k1}$

$U_{i+1}^{k+1} = U_{i+1}$, $\quad E_{i+1}^{k+1} = E_{i+1}^{k1}$

flag = 1.

else $U_i^{k+1} = U_i^k$, $\quad E_i^{k+1} = E_i^k$

$U_{i+1}^{k+1} = U_{i+1}^k$, $\quad E_{i+1}^{k+1} = E_{i+1}^k$

Step 3.

Repeat steps 1 and 2 for i = 2, 4, 6, . . . using the new values for U_i's.

Step 4. If flag = 1, go to step 1. Else terminate.

This procedure assumes an initial segmentation of the data set S into sets U_i^o i = 1...n. To undertake this initial segmentation, the following procedure is used.

Step 0.

Specify a cut-off minimum error E_e for a segment. No segment will have an error higher than this after the execution of the previously mentioned procedure.

Step 1.

Start a segment U_i with two points - t_{i_1}, t_{i_2} and error $e_i^2 = 0$.

Step 2.

Add a point t_{i_j}, $j = 3, 4, 5, \ldots$ to the set U_i and evaluate the error e_i^j. If no new point exists, terminate.

$$e_i^j = f(e_i^{j-1}, t_{i_j}, j)$$

where e_i^j, is the error in the ith segment with j points in it.

If $e_i^j < E_e$, go to step 2.

If $e_i^j > E_e$, then go to step 3.

Step 3.

$$U_i = U_i - \{tij\} \qquad N_i = j-1$$

$$i = i + 1$$

Go to step 1.

The error norm used in step 2, was TSE or MSE as defined earlier.

Computer subroutines have been developed -

INSEG for formation of the initial segments given E_e

RESEG for optimizing the segments to minimize TSE or MSE.

Once the slopes a_i, b_i and n_i are known for each U_i, $i = 1, \ldots n$, it is possible to reconstruct the data for various purposes -

-testing the accuracy of this procedure

-predicting values

-further analysis

3.5 COMMENTS

This above procedure has the following advantages:

a. powerful data compression - about 1:8 reduction in storage

b. very good reproduction of the actual data.

It has the following limitations:

a. the number of piecewise segments is governed directly by E_e. Thus, choice of E_e is critical.

b. The regenerated data was discontinuous - since $U_i \cap U_j = \phi \quad i \neq j, \forall i, j$.

To make the linear approximation 'look' more continuous, a modification was undertaken. By adjusting the segments as follows:

$$U_i \cap U_{i+1} \neq 0, \; i = 1, 2, \ldots n$$
$$= \{t_{ini}\}$$
$$t_{ini} = t_{i+1,1}, \quad i = 1, 2, \ldots n$$

One element is placed in common between two adjacent segments. This increases the TSE or MSE but makes the approximation continuous.

3.6 DATA COMPRESSION

An algorithm for approximating time series data with spline functions was described in the previsous sections. Use of the previously mentioned algorithms is demonstrated in this section. The test data used here is human subject data collected by Dr. Richard Jagacinski while performing research funded by Grant No. AFOSR-78-3697. The following documents in part are preliminary efforts to analyze certain coordinative aspects of target acquisition using discrete control methods.

In order to compare data and a spline representation of it, a plotting routine was used to plot both the data and the spline version for various values of TSE, MSE. Figure 1 shows such plots for a typical trial.

After visual comparison of such plots, it was decided that a spline fit with MSE = 50 represents the data, and the patterns in it, quit well compared to other error norms.

However, when spline fit routine was applied to all the data, for certain trials MSE = 50 criteria did not work satisfactorily. For example, it can be seen from figure 2 that the data and from spline fit representation, are the same and the number of segments is quite large. This was attributed to various noise factors including:

1. Data recording
2. Digitation
3. Data amplification
4. Jitter on the sticks used by the human subjects

Thus, a noise attenuating scheme was necessary. In order to achieve this, a two stage, low pass filter using trapezoidal approximation was implemented through software. The exact form of the filter is given below:

$$\underline{Z}\,(t+\Delta t) = e^{-\alpha\Delta t}\begin{bmatrix} 1 + \Delta t & \Delta t \\ -\alpha^2\,\Delta t & 1-\alpha\Delta t \end{bmatrix} Z(t)$$

$$\begin{bmatrix} \frac{2}{\alpha\cdot\Delta t} - \left(2+\alpha\,\Delta t + \frac{2}{\alpha\cdot\Delta t}\right) e^{-\alpha\Delta t} \\ -\frac{1}{\Delta t} + \left(\alpha^2\Delta t + \alpha + \frac{1}{\Delta t}\right) e^{-\alpha\Delta t} \end{bmatrix} x(t)$$

$$\begin{bmatrix} 1 - \frac{2}{\alpha\Delta t} + \left(1 + \frac{2}{\alpha\cdot\Delta t}\right) e^{-\alpha\Delta t} \\ \frac{1}{\Delta t} - \left(\alpha + \frac{1}{\Delta t}\right)\cdot e^{-\alpha\Delta t} \end{bmatrix} x(t+\Delta t)$$

Where

x(t) is the input signal

$z(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}$ $z_1(t)$ is the smoothed signal, $z_2(t) = \frac{d}{dt} z_1(t)$

Δt is the sampling interval

α is the cutoff frequency.

Using this smoothing scheme, data was filtered with cut off frequency of 60 rad/sec. and 100 rad/sec. Again from the plots it was decided that smoothing at 100 H_z is quite satisfactory prerequisite for spline fitting.

This filter was incorporated in the spline fit routine and with cutoff frequency = 100 rad/sec. and MSE = 50 spline fit routine was used to represent smoothed data in a very compact form.

At this stage a data base is available in the following form which can be stored very economically on a secondary storage medium, (e.g. floppy disk cartridge).

Record Type	File Structure Variables in the Record	
1	IWIDE	
2	RUN DELIMITOR	
3	IRUN, ICH, IFIL (ICH) ISMAX, JMAX	
4	TIME FRAMES, INTERCEP SLOPE	ISMAX NO of Segments
4		
4		
3	IRUN, ICH, IFIL(ICH), ISMAX, JMAX	
4	TIME FRAMES INTERCEPT, SLOPE	
4		ISMAX No of Segments
4		
3	IRUM, ICH, IFIL(ICH), ISMAX, JMAX	
4	TIME FRAMES INTERCEPT SLOPE	
4		ISMAX No of Segments
4		
2	RUN DELIMITOR	

where

IRUN	-	Run No.
ICH	-	Channel #, 1, 2, 3
IFIL(ICH)	-	1, 0 represents whether data filtered or unfiltered
ISMAX	-	No of segments for run no IRUN on the Channel no ICH
JMAX	-	No of time frames in the run
IWIDE	-	1, 2 indicates narrow or wide target
RUN DELIMITOR	-	4444 in 315 format

This filter is a digital implementation of a two stage low pass filter with cutoff frequency .

These data bases can be directly processed by existing discrete control analysis routines. Part of our current effort is focusing on the problem of modelling and interpreting this data.

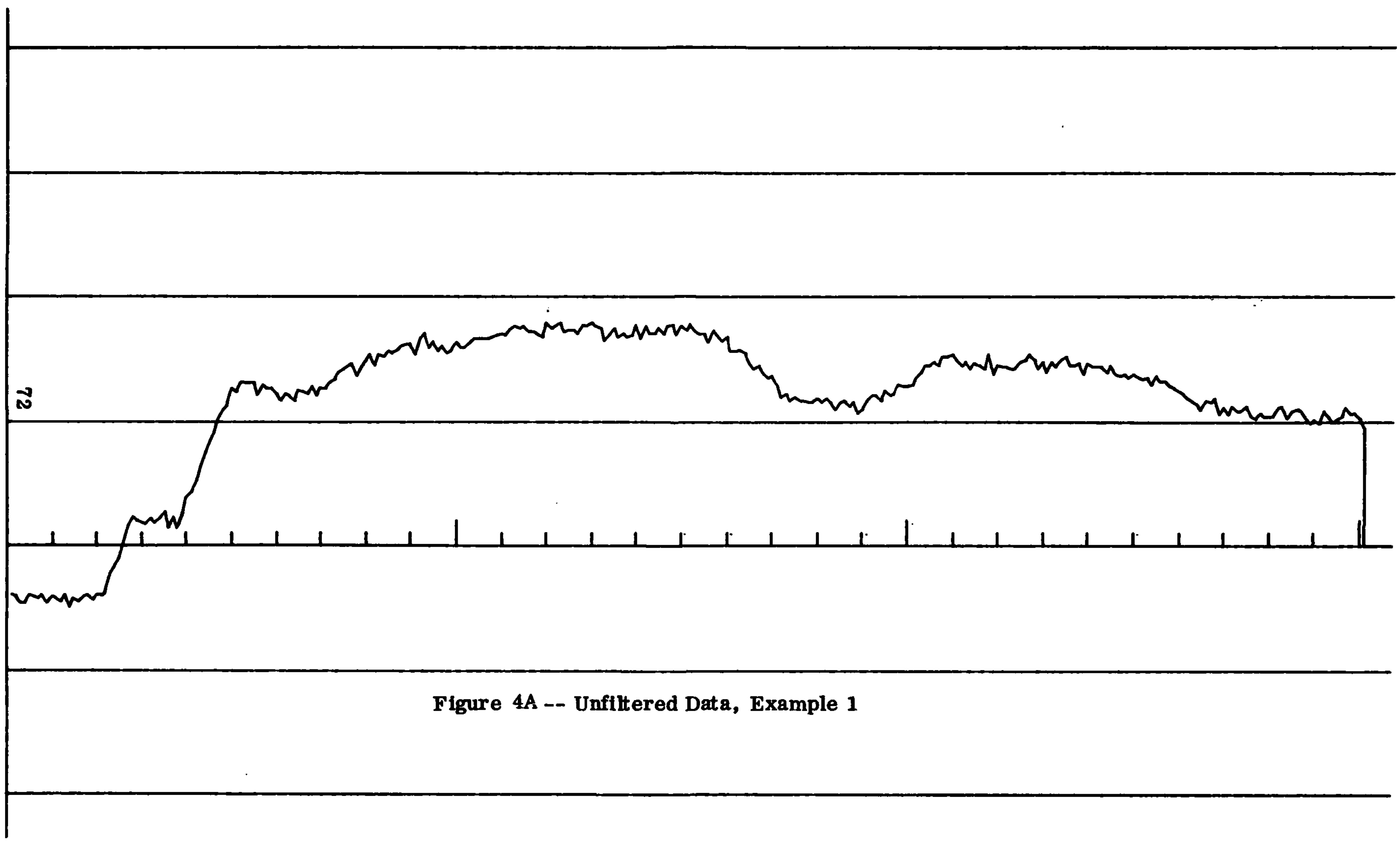


Figure 4A -- Unfiltered Data, Example 1

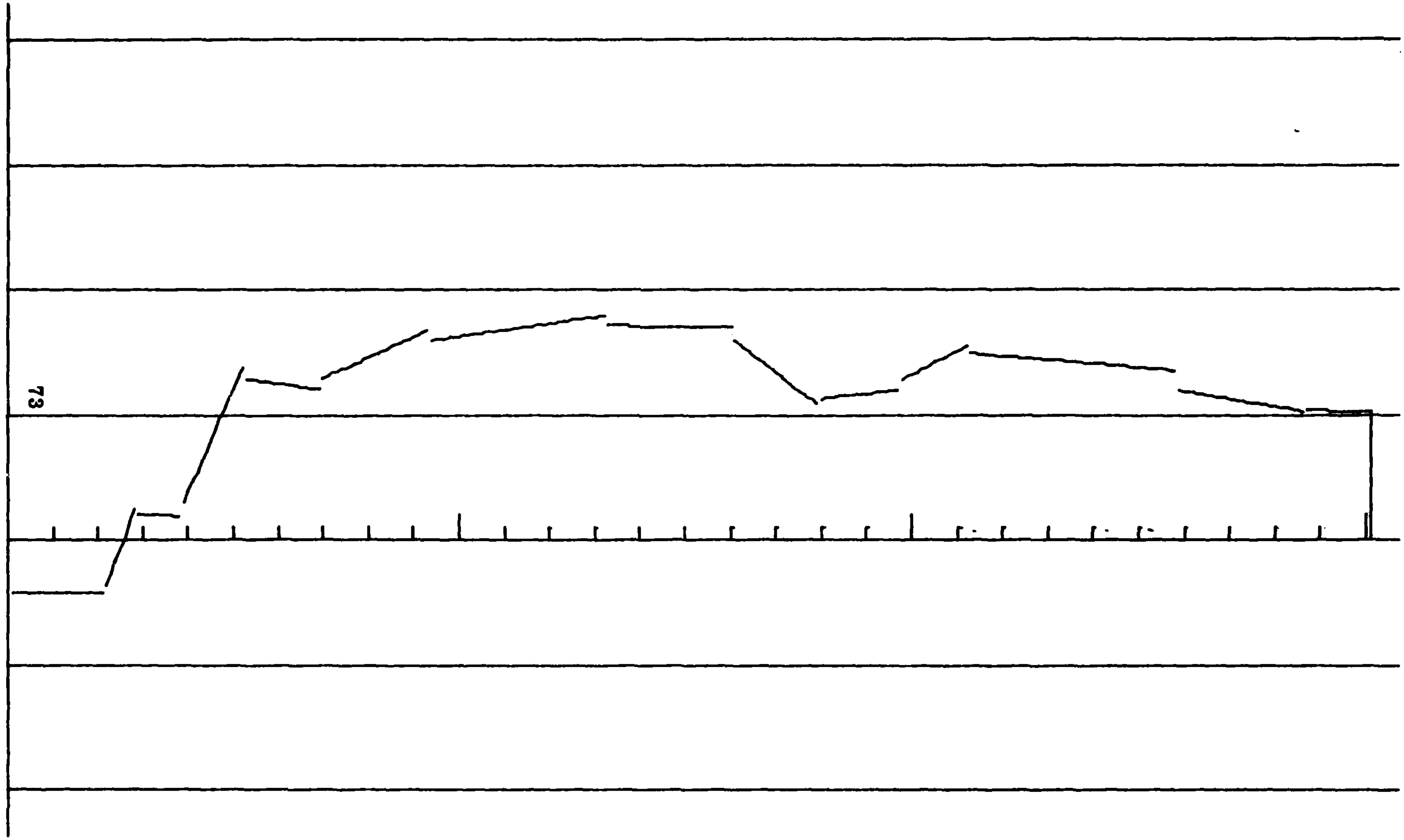


Figure 4B -- Spline Approximation, Unfiltered Data,
Example 1, MSE = 50

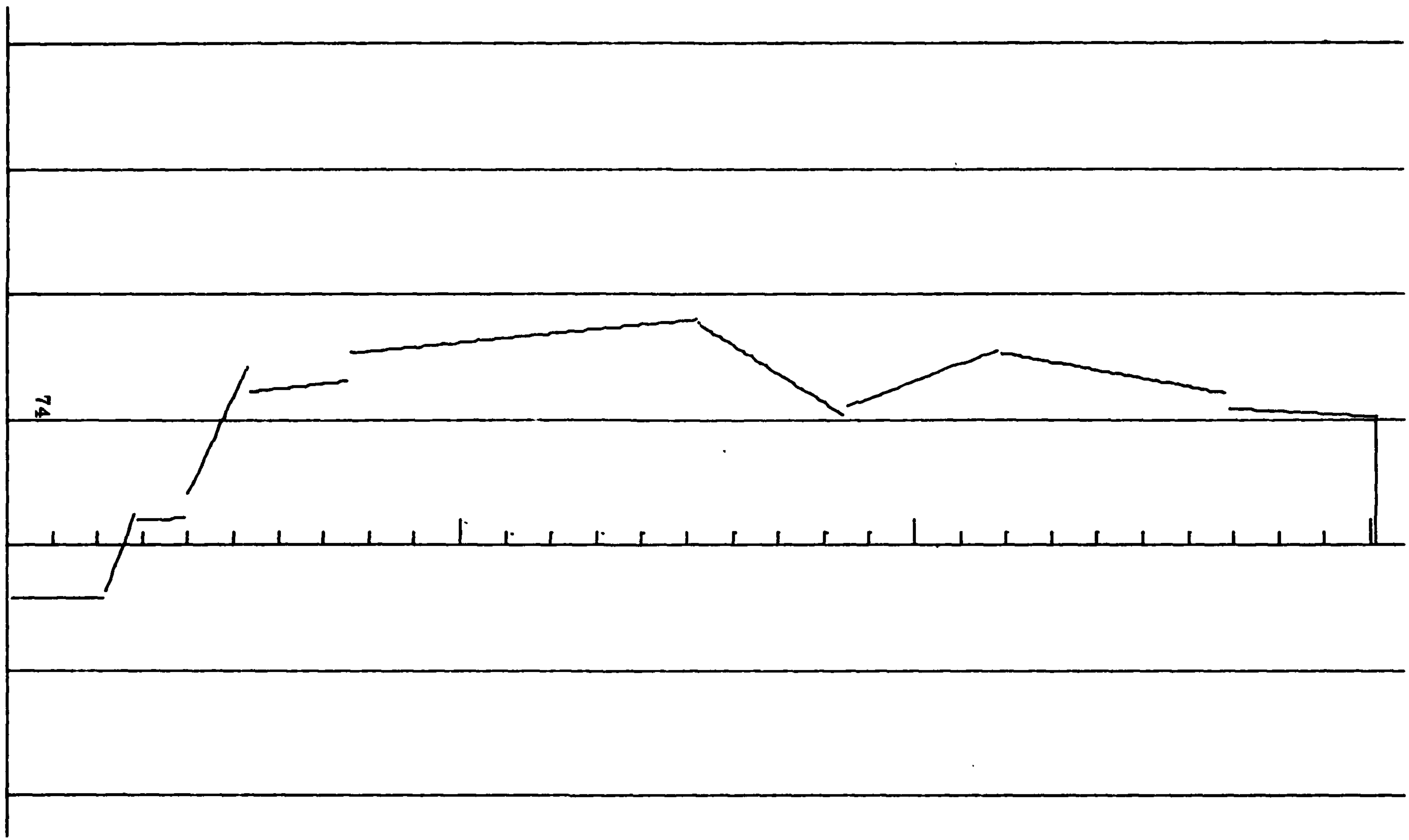


Figure 4C -- Spline Approximation, Unfiltered Data,
MSE = 100, Example 1

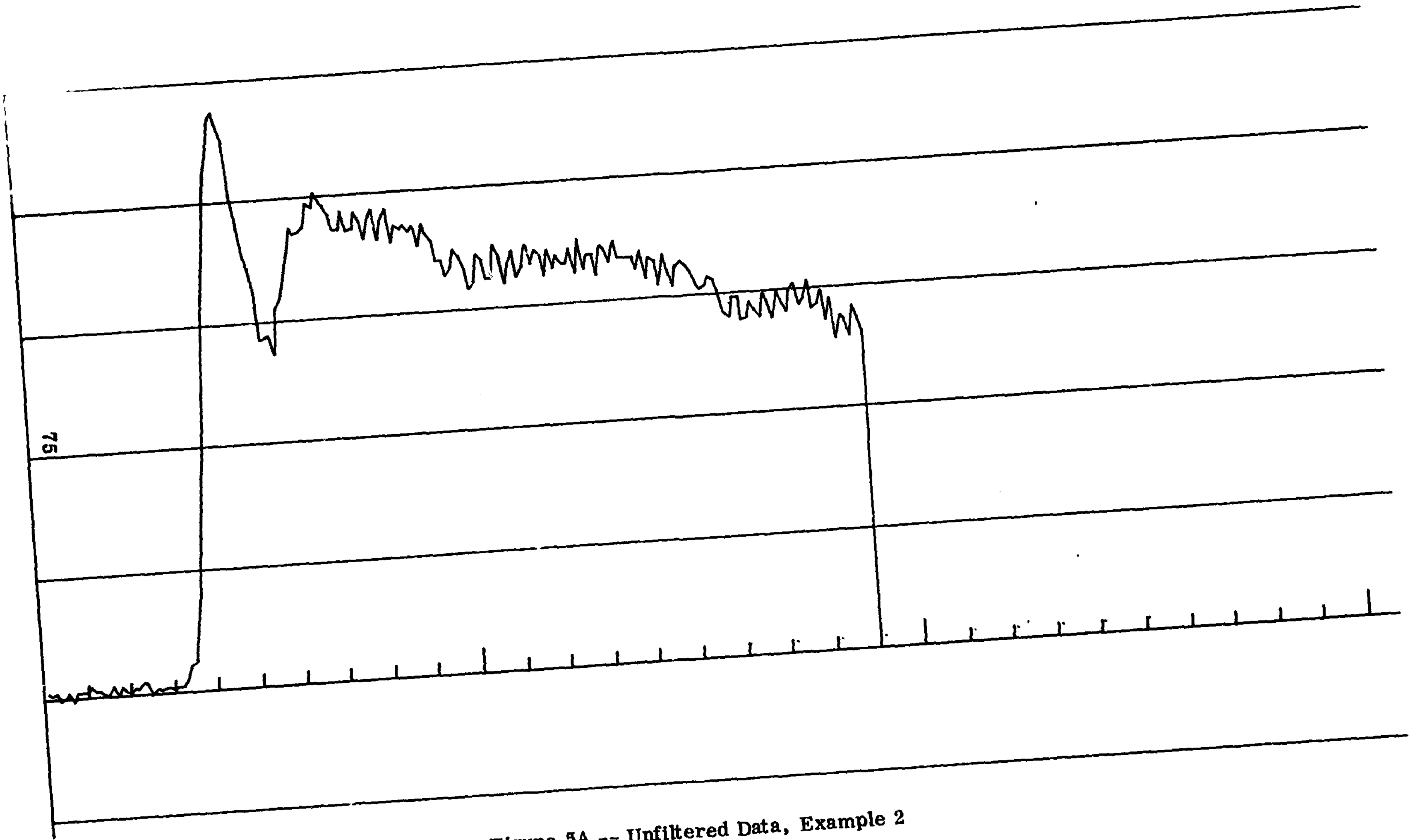


Figure 5A -- Unfiltered Data, Example 2

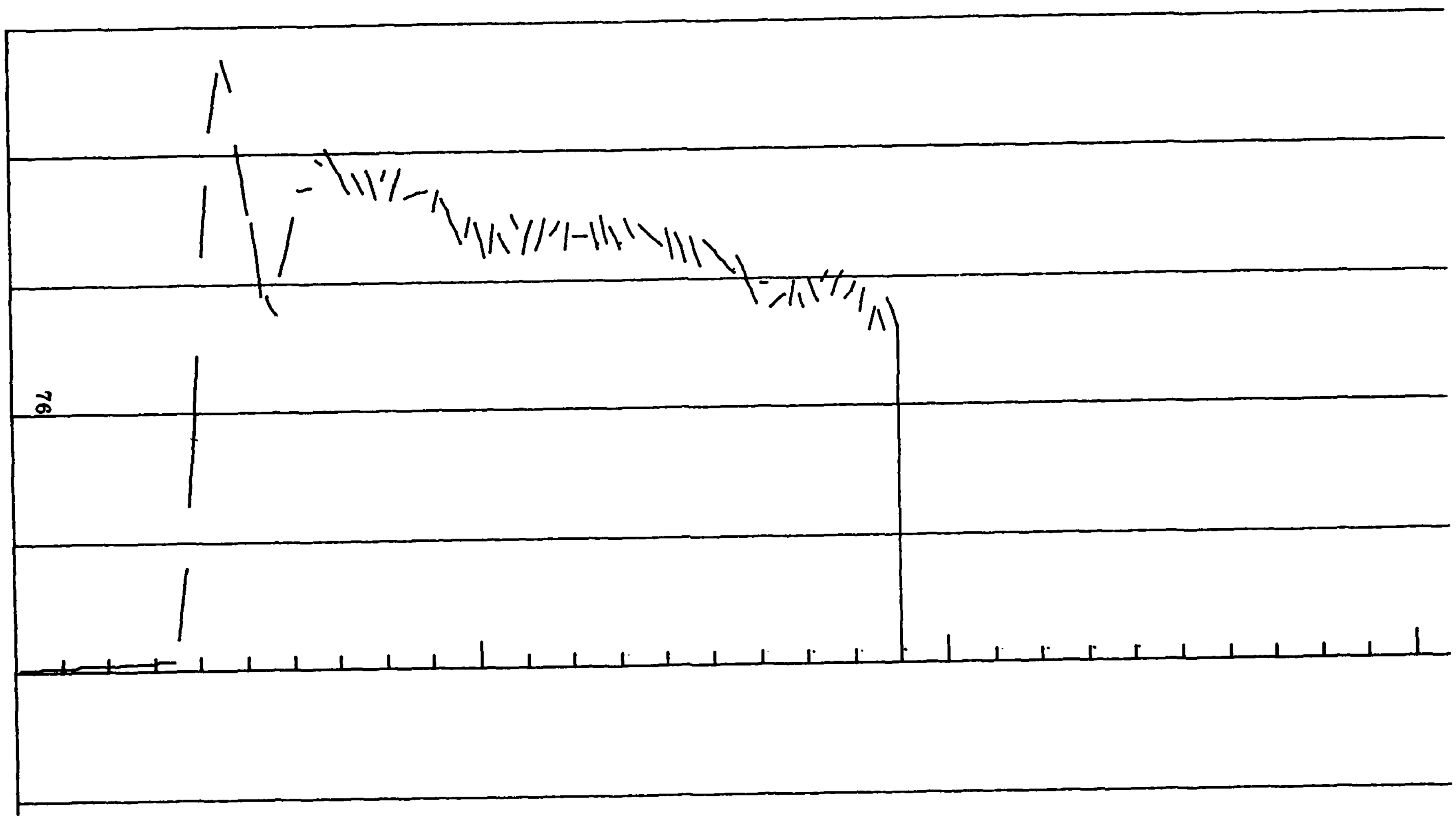


Figure 5B -- Spline Approximation, Unfiltered Data, MSE = 50, Example 2

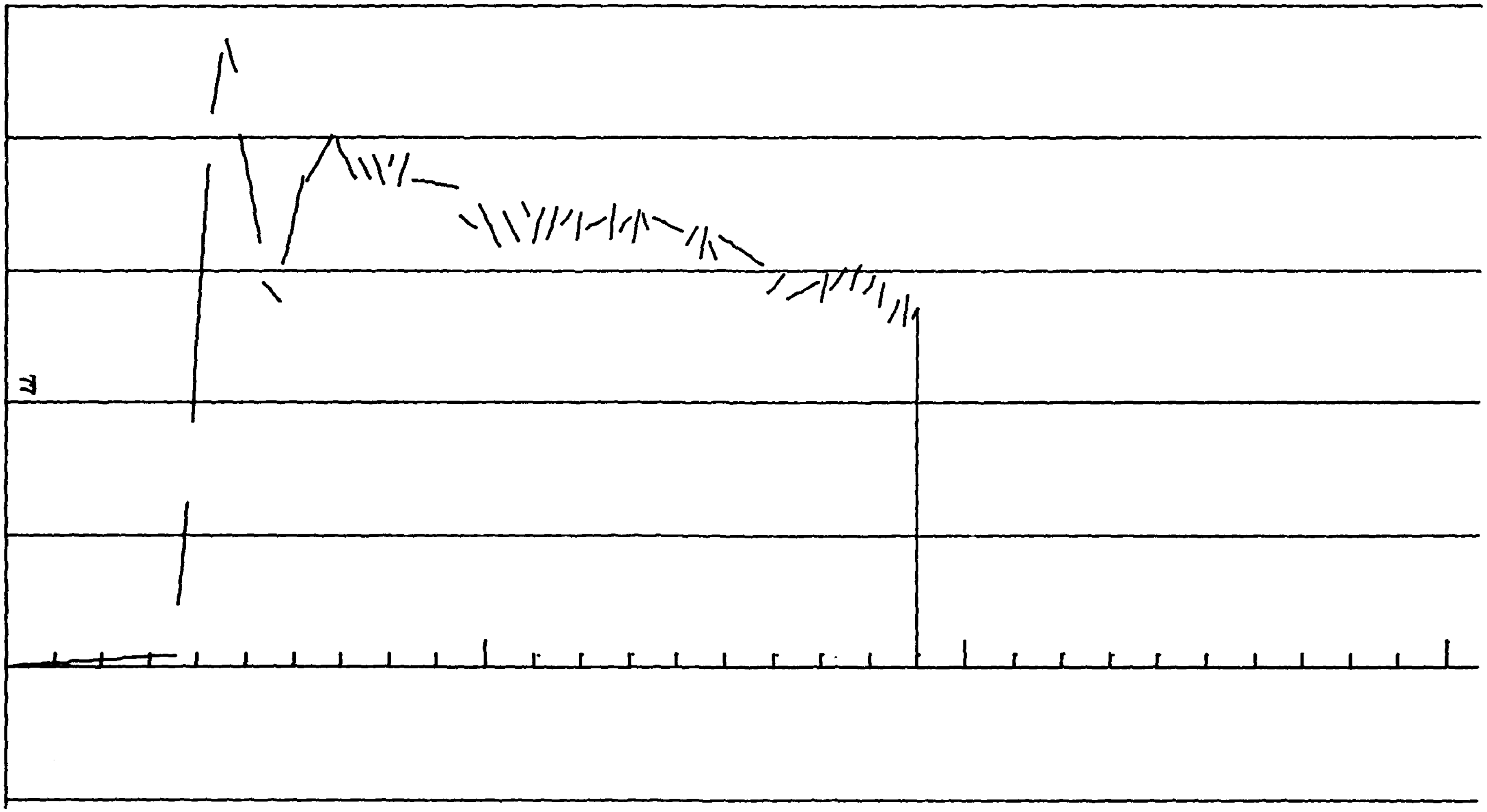


Figure 5C -- Spline Approximation, Unfiltered Data,
MSE = 100, Example 2

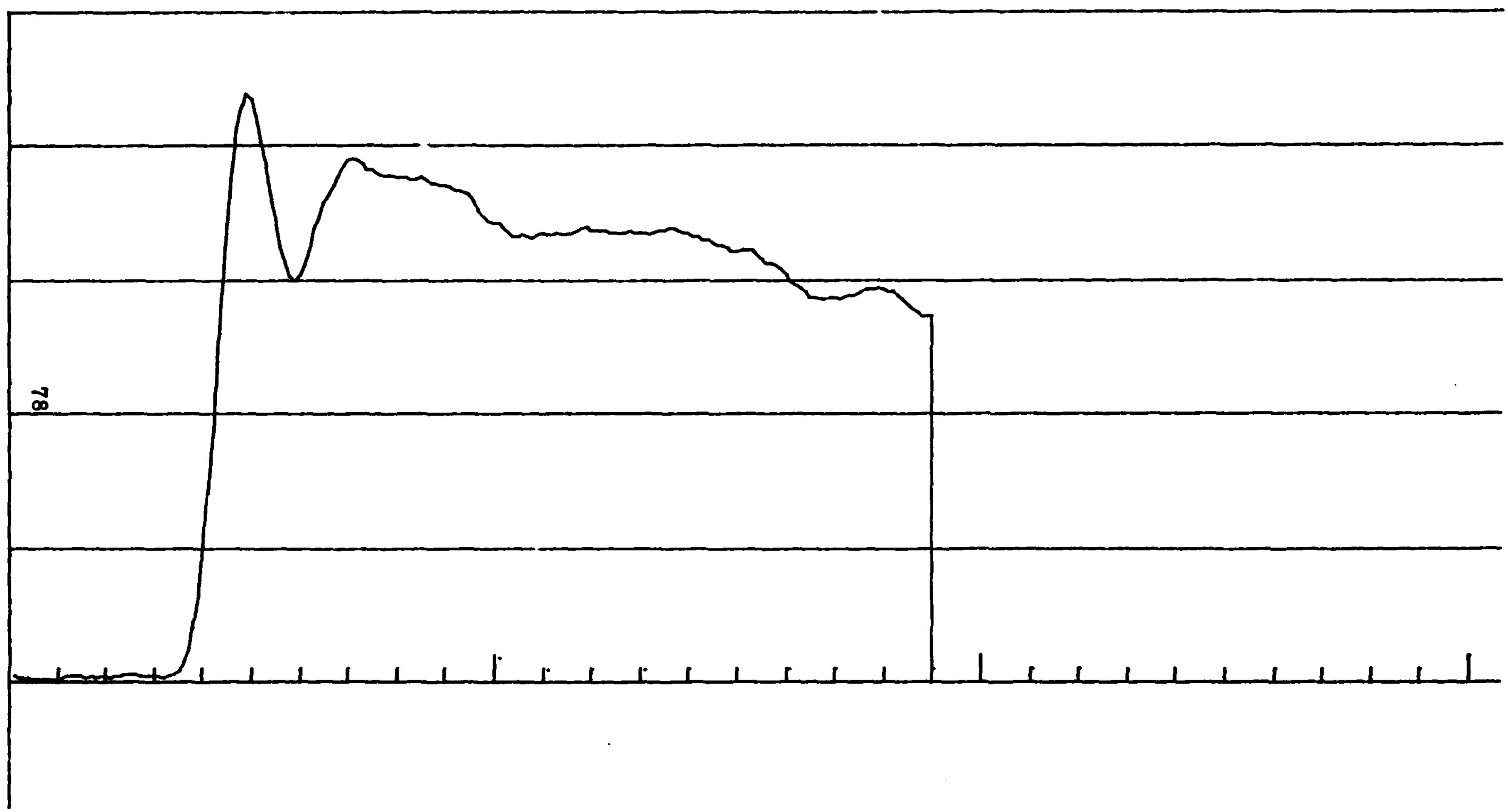


Figure 5D -- Filtered Data, Cutoff Freq. = 60 Rad/Sec, Example 2

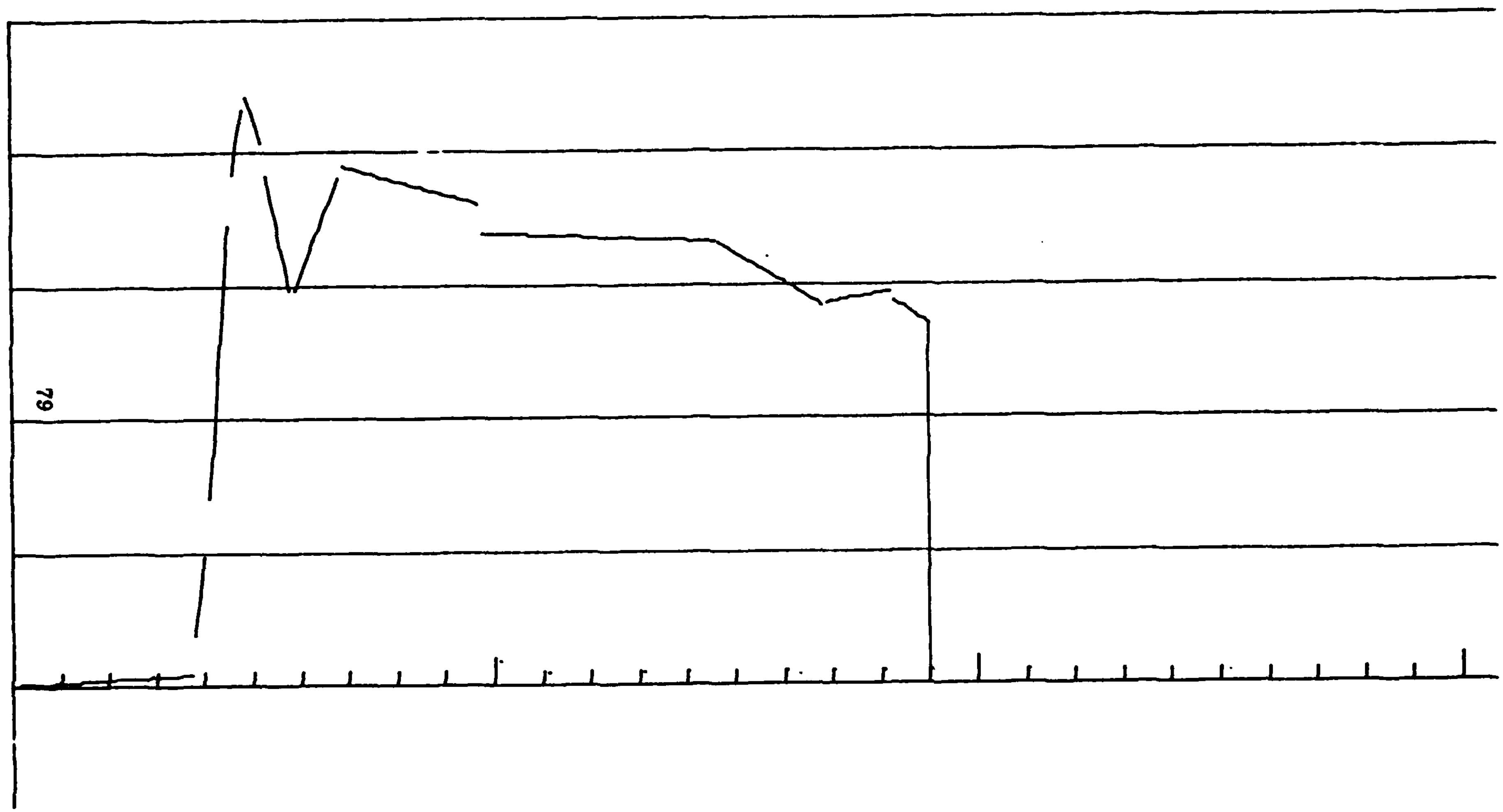


Figure 5E -- Spline Approximation, Filtered Data, Cutoff Freq. = 60 Rad/Sec, MSE = 50, Example 2

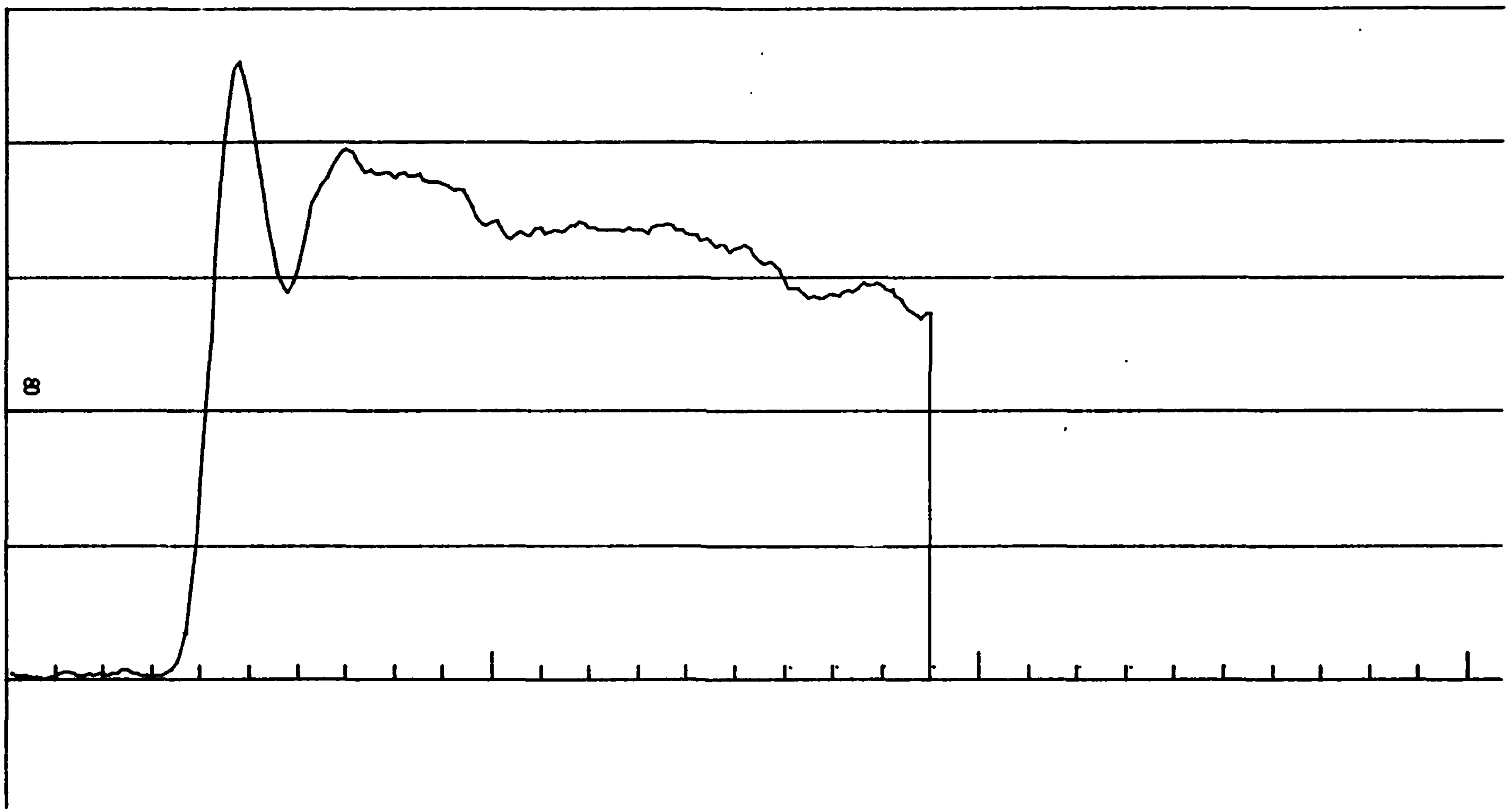


Figure 5F -- Filtered Data, Cutoff Freq. = 100 Rad/Sec., Example 2

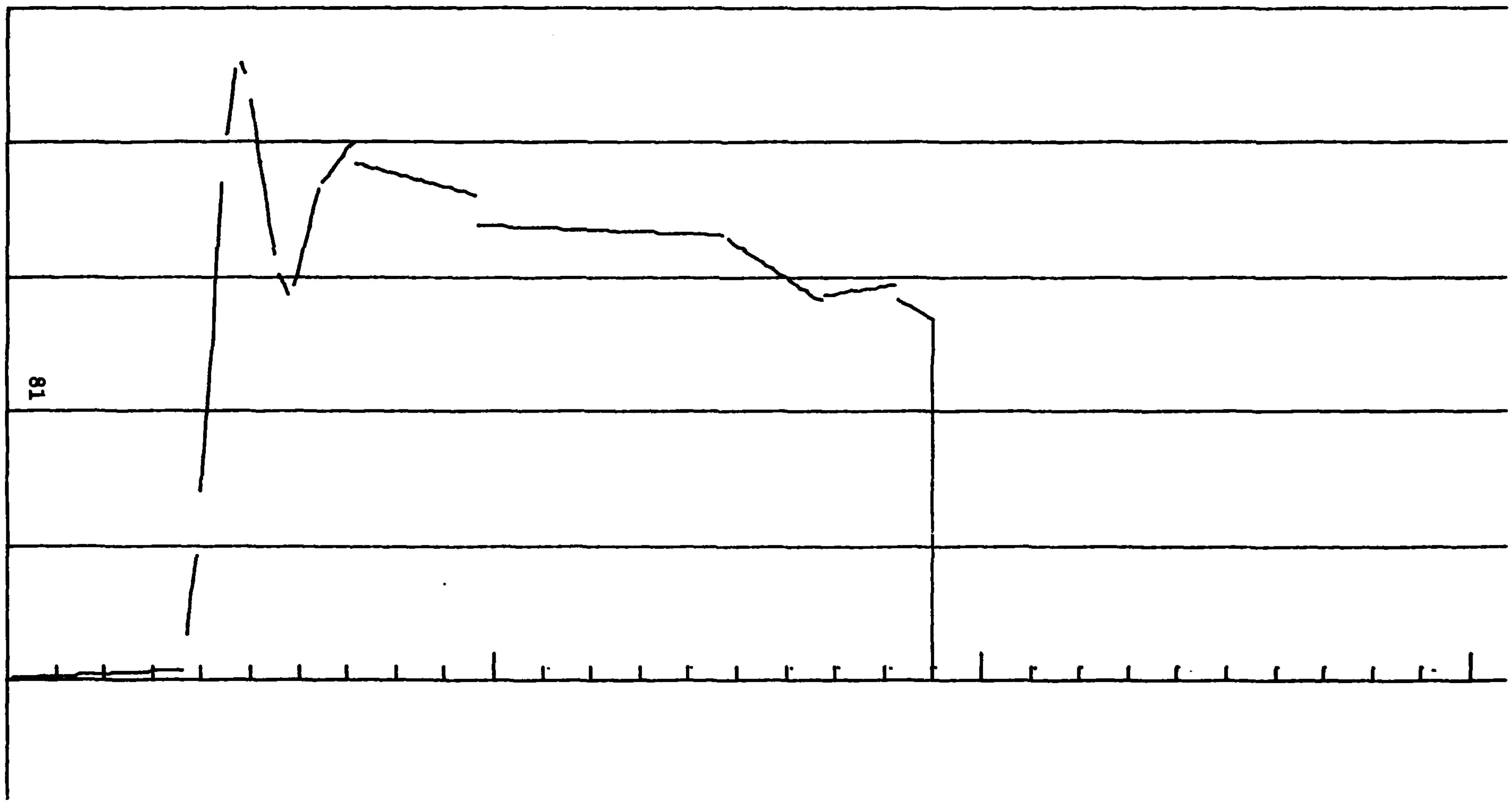


Figure 5G -- Spline Approximation, Filtered Data, Cutoff Freq. = 100 Rad/Sec, MSE = 50, Example 2.

References:

Pavlidis, T and S. L. Horowitz (1974). IEEE Trans. on Computers, Vol. 23, p. 23.

Pavlidis, T (1973). IEEE Trans. on Computers, Vol. 22, p. 689.

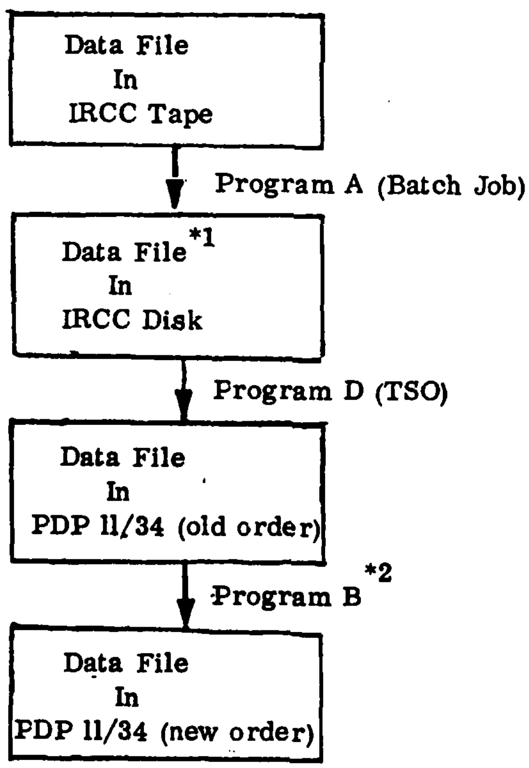
4. RESTRUCTURING PROCEDURES FOR DISCRETE CONTROL DATA FILES

This section documents work which was performed in support of the effort to make the discrete control analysis procedures more interactive. It also documents the existing anti-aircraft artillery data base.

4.1. INTRODUCTION:

The tracking data files of AAA (Anti-Aircraft Artillery) team performance are stored in eleven tapes, which are currently in the Ohio State University Instruction and Research Computer Center Tape Library. These data files can only be read by PL/1 programs, which are read into the IRCC computer through batch jobs (or TSO). In order to use the ISE Department PDP 11/34 to perform interactive analysis, the data files had to be reconstructed. This report describes the reconstruction procedures and the format of the new data files.

4.2 FLOW DIAGRAM OF THE RECONSTRUCTING PROCEDURES:



---

*1 because not enough space has been reserved in IRCC, these files must be transferred down to the PDP 11/34 disk before 5 a.m. each day; otherwise they would be deleted.

*2 reason for using this procedure will be explained later in this report.

4.3. ORIGINAL TRACKING DATA FILES (IN IRCC TAPES):

A. GENERAL INFORMATION:

These files are binary using 16 bit words (i.e. 2 bytes/word * 8 bits/byte = 16 bits/word).

B. GENERAL FORMAT:

1. Run Header
2. Engagement Header, Target #1
3. Data File of Engagement 1
4. Engagement Header, Target #2
5. Data File of Engagement 2

.
.
.

EOF (of this run)

C. SPECIFIC CONTENT:

1. Run Header (10 words)

WORD #	DESCRIPTION
1	Team #
2	Run #
3	# of targets in run
4	Day (#)
5	Month (#)
6-10	Not Assigned

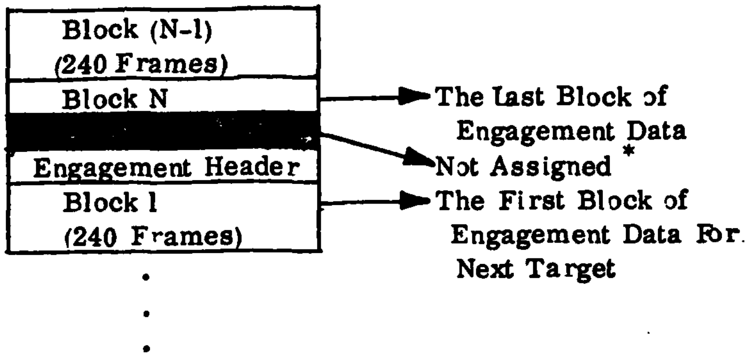
2. Engagement Header:

WORD#	DESCRIPTION
1	Target # within run
2	Trajectory #
3	Length of engagement (# of frames)
4	Not assigned

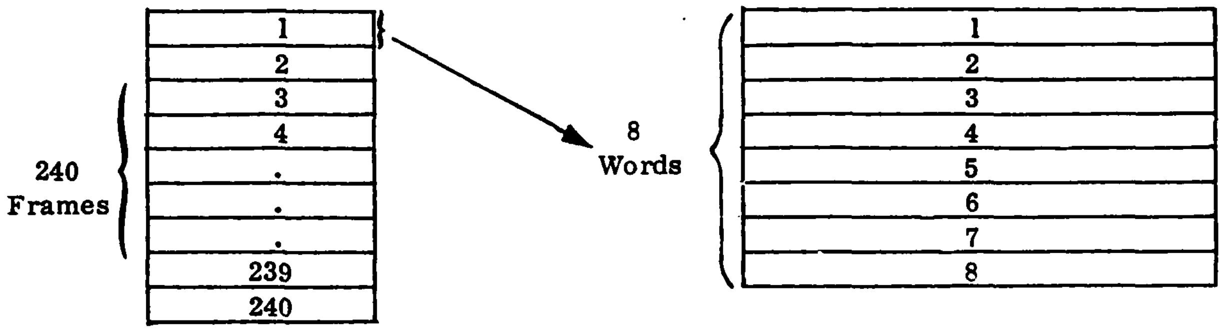
3. Engagement Data Record:

Block 1 (240 Frames)
Block 2 (240 Frames)

.
.
.



* For every trajectory data file, if its total frames cannot be divided by 240 (i. e. BLOCKSIZE =240 (Frames/Block) X 16 (Bytes/ Frames) = 3840 Bytes/Block), blank are used to fill the remaining space of the last block.



WORD #	DESCRIPTION
1	Frame Count (i. e. Sample #)
2	Mode Word #1 } *1
3	Mode Word #2 }
4	Azimuth Tracking Error } 2
5	Elevation Tracking Error }
6	Range Tracking Error
7	Upper Counter Reading
8	Lower Counter Reading

---

*1 The contents of these two mode words will be listed on the next page.

*2 The units of Azimuth and Elevation tracking error are "Milli Radians"; the unit of Range tracking error is "meters".

CONTENTS OF "MODE" WORDS

WORD 1 Bit #	WORD 1 Contents	WORD 2 Bit #	WORD 2 Contents
0	Antenna Horn	0	Target on Screen
1	Auto Circular Scan	1	Azimuth Error
2	Radar Mode	2	Elevation Error
3	. .	3	Range Error
4	Gun Servo Mode	4	Tracer Evaluation
5	0° Lead Enable	5	Optics Disturbance
6	Lead Enable	6	PPI Disturbance
7	Display In Use	7	Range and PPI Blanked
8	Mode Switch	8	Range Disturbance
9	Computer Shunt	9	Sight Selector
10	Data Ready Indicator	10	Sight Magnification
11	Coolant	11	Sight Filter
12	Trigger	12	. .
13	Azimuth Control	13	Range Tracking Control
14	Elevation Control	14	Lower Barrel Enable
15	Target Introduced	15	Upper Barrel Enable

4.4. PROGRAM A:

This program, which is written in PL/1 language, is used to reconstruct the tracking data files of AAA team performance, so that the new data file:

(1) can be processed by FORTRAN IV program;
(2) can provide all necessary information about the status of the system throughout the whole test;
(3) occupies only about 1/25 space, if compared with the space of the original data.

The basic unit of data, that could be processed by this program, is the data file of "run".

4.5. OUTPUT OF PROGRAM A:

A. General Format:
1. Run Header
2. Engagement Header, Target #1
3. Data File of Engagement 1
4. Engagement Header, Target #2
5. Data File of Engagement 2

.
.
.

EOF (of this run)

B. SPECIFIC CONTENT:

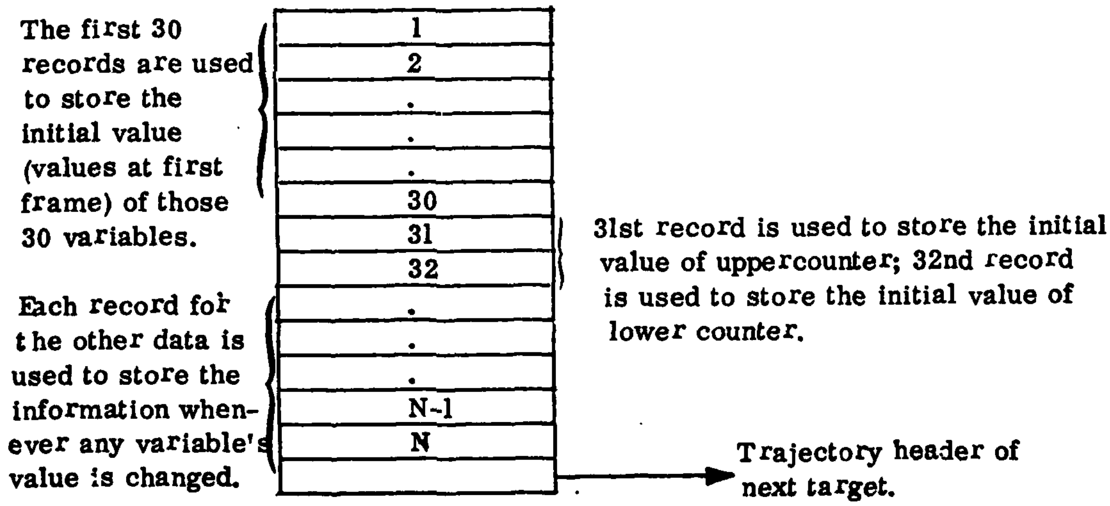
1. Run Header (10 Bytes)

BYTE #	FORMAT	DESCRIPTION
1-2	I2	Team #
3-4	I2	Run #
5-6	I 2	# of Targets in Run
7-8	I2	Day (#)
9-10	I2	Month (#)

2. Engagement Header (10 Bytes):

BYTE #	FORMAT	DESCRIPTION
1-2	I2	Target # Within Run
3-5	I3	Trajectory #
6-10	I5	Length of Engagement (# of frames)

3. Engagement Data Record:



BYTE #	FORMAT	DESCRIPTION
1-2	I2	Variable #
3-7	I5	Frame #
8-10	I3	Variable Value

4.6 PROCEDURES OF TRANSFERING DATA FILES FROM IRCC DISK TO PDP-11/34 DISK:

1. Log on the PDP-11 terminal.

2. Enter account name and password to communicate with PDP-11/34.

3. Run TSO: >Run TSO <CR> (Wait for "READY" Sign)

4. Log on: LOGON TS ϕ871/DEC111 F (SE85) ID (ID#) <CR>

5. Check the files in IRCC disk: LISTCAT <CR>

6. Allocate the dataset which is to be transferred: (e. g. test· dat)
   ALLOC DA (TEST· DATA) FI(FTϕIFϕϕI) <CR>

7. Open the PDP-11 file for input from TSO: (e. g. test· dat)
   *OPEN SY: TEST· DAT <CR>
   The computer should print: *** INPUT FILE OPEN

8. Run the TSO transfer program: RUN TD1ϕ. FORT*1 <CR>
   the file should be transferred, preceded and followed by '@@@@ . . .', the PDP-11 should then print: *** INPUT FILE CLOSED
   To transfer another dataset, type: FREE DA (TEST· DATA)
   then go to step 6.

9. Logoff: LOGOFF <CR>

10. Stop the TSO program: (CONTROL/Z) ↑Z

4.7 PROGRAM B

In section 4.3 of this report, we listed the contents of those two "Mode" words. In program A, Bit 1 of word 1 (i. e. ANTENNA HORN) is variable 1; Bit 2 of word 1 (i. e. AUTO CIRCULAR SCAN) is variable 2; Bit 3 and Bit 4 of word 1 (i. e. RADAR MODE) is variable 3, etc. These variable #s are not consistent with the original definitions of variables that were defined and listed on page 46 and page 47 of Miller (1979). We don't carry out this change in program A, because it would almost double the cost of running the program. The purpose of this program (PROGRAM B) is to change the variable numbers, so that they would match the original definitions.

AD-A082 434 OHIO STATE UNIV RESEARCH FOUNDATION COLUMBUS F/G 9/2
FINITE STATE MODELS OF MANNED SYSTEMS: VALIDATION, SIMPLIFICATI--ETC(U)
NOV 79 R A MILLER AFOSR-79-0015

UNCLASSIFIED AFOSR-TR-80-0216 NL

2 of 2

AD A082434

END
DATE FILMED
5 80
DTIC

4.8. NEW TRACKING DATA FILES: (IN PDP-11 DISK)

The formats of these data files are the same as those in section 4.5 except:

1. one blank byte is put on the first byte of every record (for the purpose of carriage control), so the record size is 11 bytes, instead of 10 bytes

2. the order of the first 30 records in "ENGAGEMENT DATA" have been changed.

Reference:

Miller (1979). Identification of Finite State Models of Human Operators, Final Report, Grant #AFOSR77-3152 Ohio State University.

5. PUBLICATION BOARD

The following papers and presentations have resulted from the research reported here.

"A Discrete Control Analysis of Coordination Activities in a Simulated AAA System". To appear in the Proceedings of the 15th Annual Conference on Manual Control, Wright State University, Dayton, OH, March 1979. (See appendix A).

"A Finite State System Model of Coordination in Multi-Person Teams", Proceedings of the 1979 International Conference on Cybernetics & Society, Denver, Colorado, pp. 210-216, October 1979. (See appendix B).

"Coordination in Small Teams" invited lecture presented at NASA-Ames Research Center, Moffett Field, California, June 1979.

"Approximation of Stochastic Automata with ℓth Order Markov Processes", a paper to be submitted to IEEE. Transactions on Systems, Man & Cybernetics.

"On the Concept of Validity", with Christine Mitchell, to be submitted to International Journal of General Systems. (This paper is based on Chapter 2 of this report.)

6. RESEARCH PERSONNEL

The following individuals were supported in part by this grant:

R. A. Miller, Principal Investigator

Christine M. Mitehell, Ph. D. student

Korhan Sevenler, M. S. student

Gregory Jewett, M. S. student

Chang Feng-Chang, M. S. student

Rajendra Nalavade, Ph. D. student

Anant Misal, M. S. student

APPENDIX A

This paper was presented at the 15th Annual Conference on Manual Control and will appear in the forthcoming proceedings.

A DISCRETE CONTROL ANALYSIS OF COORDINATION ACTIVITIES IN A SIMULATED AAA SYSTEM

by
R. A. Miller
Department of Industrial & Systems Engineering
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

The discrete control modelling methods used to represent and analyze performance data obtained from a three person AAA tracking simulation are discussed. The basic model structure, a hierarchical network of communicating finite systems, is presented and selected state transition function estimates are also presented. From the analysis several generalizations about the strategies employed by the subject teams are identified.

1. INTRODUCTION

The general class of systems of interest is that in which the operator (operators) of the system has (have) available only a finite number of control or decision alternatives and these are used to directly or indirectly control the behavior of the system over time. An operator might also have other tasks including continuous control tasks to perform, but the issue here is the set of discrete decisions by which the system configuration and mode of operation is established, and the procedures by which the team members' activities are coordinated.

The modelling questions focus on the problems of capturing in some mathematical representation the way in which team members might decompose a complex problem into simpler parts and how they then manage to coordinate their individual activities and configure the system so that acceptable overall system performance is achieved. The basic questions therefore are questions of knowledge representation, information flow and communication in a complex system. A general hierarchical/heterarchical structure which allows for structural coordination of subsystems by upper level components and which utilizes a heterarchical control structure to shift the focus of control to the proper subsystem at the proper time has been developed. This general structure was used to guide the analysis of a simulated anti-aircraft artillery system. A specific realization of the structure was constructed and the data obtained from experiments using the above mentioned simulator were interpreted using this structure and specially designed analysis routines. The model and its key properties form the major part of this paper.

2. A DESCRIPTION OF DISCRETE CONTROL MODELLING

The only property established by the definition of discrete control cited above is that a finite number of control alternatives are available for use by the operators. The operators presumably change the alternative (control) selected from time to time in response to changing requirements or a changing environment. Basically, the purpose behind constructing a discrete control model is to explain how specific selections are reached from information about the system, environment and the context of the situation.

It is assumed throughout this development that the information about the controlled system and the environment which is displayed to the operator, or otherwise provided him, is discrete. The assumption is that such data are naturally discrete or are used by him in discrete form. It is assumed that any continuous information can be categorized in some way; e.g., slow, medium, fast. This process of representing continuous information in a discrete qualitative format in some sense corresponds to a feature extraction or abstraction process performed by an operator when encoding and internalizing the information provided him.

Abstractly then the discrete control context consists of an operator or team of operators receiving information in the form of event sequences and producing some sequence of control selections in response. A behavioral representation of a discrete controller (operator or operators) is then a system S,

$$S \subseteq A^T \times B^T$$

where T denotes the time set, A the input alphabet (set of possible inputs to the operators, here assumed finite), and B the output alphabet (set of decision alternatives, also assumed finite). The set A^T is the set of possible input functions with domain T and co-domain A. B^T is similiarly defined. The system S therefore is the relation consisting of the pairs of input-output behaviors which are possible. This relation can be thought of as the sample space from which all data items in a discrete control experiment are selected.

The task in modelling is to provide a more detailed and constructive description of the relation S. In systems theoretic terms this is usually accomplished through a state decomposition of the system and the construction of state transition and output assignment functions. Since the input and output alphabets are both finite sets in discrete control situtations, an event based state description is most useful. Let C denote the state space, a finite set, and define

$$M = \{m \mid m : C \rightarrow [0,1], \sum_{C} m(c) = 1\}.$$

M should be interpreted as the set of all state occupancy probability distributions defined on state space C. Now a function of the form

$$\Phi : C \times T \times A \times T \rightarrow M$$

is a state transition function of a very special kind. Specifically,

$$\Phi(c, t, a, s) \quad M$$

defines a conditional state occupancy distribution with the conditioning provided by the following information:

1) The state at the last event occurrence was c.
2) State c had been occupied t units at that time.
3) The next state change event occurs within s time units.
4) The input over the interval from the last event to the next event is fixed at a.

The function Φ then establishes the probability of state occupancy within a specified length of time given certain state and input information.

An output assignment function is simply a function of the form

$$\lambda : C \rightarrow B$$

and it simply assigns the appropriate output symbol to each state. Together the state transition and output assignment functions provide a stochastic representation of the system S. More precisely, when modelling one must find a state space C and functions Φ and λ which provide an adequate representation of a given system S.

The concept of state and state transition used above is a highly abstract one and for any complex system requires some additional refinement. Experience with discrete control modelling has shown that a very powerful way of constructing representations is through the use of networks of simple systems.

The diagram shown in Figure 1 is a simple illustration. Each system in the network is a finite state system represented by the usual objects: an input alphabet, output alphabet, state space, transition function and output assignment function. In other words, each system in the network inherits all of the properties discussed above.

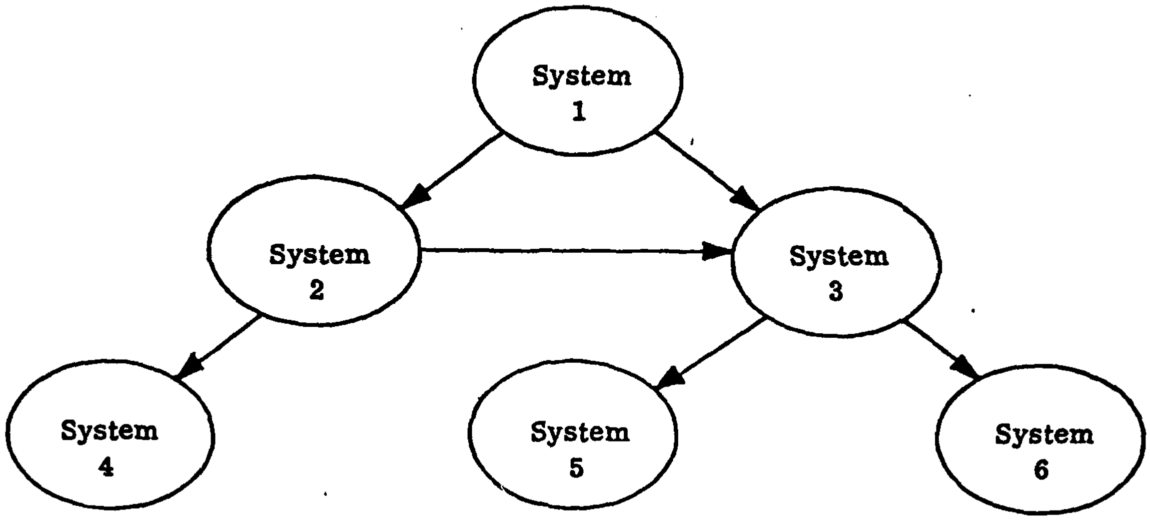


Figure 1.--Network of Systems.

The state space of the complete system; i.e., the network, is then the cartesian product of the state spaces of the component systems. The state transition function and output assignment function required to represent S also follows from the functions associated with the individual elements in the network. Each system in the network is governed by a transition function and any change in state is communicated to the appropriate systems via the communication links defined by the network. The system transition function then is really a fairly complex function which is in some sense the product of the appropriate individual function. But, the important point is that each individual element is quite simple and the rules for constructing the overall state transition function are straightforward. In other words, by knowing the properties of the component systems and the road map which defines their interconnection, the system properties easily follow.

One advantage of the network construction is complexity reduction. The number of transition probabilities that must be estimated is substantially smaller than would be the case without the network decomposition. The network representation also has a distinct substantive advantage. The primary reason for constructing the model in the first place is to help explain how operators perform discrete control tasks. The network is in essence a representation of the intelligence which might be brought to bear on the problem. It is one way of representing a complex problem in a manageable form. Each system in the network represents some important activity or subsystem which the operator must control. By identifying what these component systems are and how they interrelate, the discrete control model is constructed, and network representation follows as a natural by-product.

The key point is that the states of all systems at each level must be determined from simple logical operations on systems previously defined. Once the component systems have been defined the network must be defined. This is probably best accomplished in two steps because two kinds of information generally flow through the system network. In some cases the state of a given system is said to directly constrain the states which another system can occupy whereas in other situations the information that flows to a system only influences the decisions which specify the state. The two networks should be considered separately. The graph of the first type, the constraint network, should be constructed and the constraints themselves identified. After this has been accomplished the various systems which are decision loci are clearly identifiable and the decision influence network can be established.

In the next section these modelling ideas are applied to the problem of representing discrete control behavior in a simulated anti-aircraft artillery system.

3. DESCRIPTION OF THE AAA SYSTEM AND THE EXPERIMENT

The system which served as the focus of the study was a man-in-the-loop simulation of an anti-aircraft artillery (AAA) installation. This system consisted of a mock-up of the operators' consoles, including the major controls, switches, and displays; plus the computing equipment required to drive the displays, record data and generally simulate the AAA system and its environment. This particular simulator required a three person team consisting of a range tracker, an angle tracker and a commander. Basically the system consists of two optical sighting systems (left optics, right optics), a radar system with separate displays for the angle operator and range operator, a gun servo system which positions the guns as a function of tracking commands, and a lead angle computer. There are also a variety of switches and controls devices used to control and coordinate the activities of the system. The simulator is located at the Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base and the experiments which provided the data used in this analysis were performed by AMRL personnel.

The experiment was a highly stylized simulation of the AAA task. Subject teams were required to search for, acquire, and track simulated targets and to try to maximize the hit score attained on each target. The teams were free to select any system mode of operation at any point in time. Teams were asked to perform simulated "missions" which consisted of a sequence of 23 targets which they were required to acquire and track. Targets were presented one at a time and the target trajectory ran its full course before a new target was introduced. There was also a period of time between targets during which no target was present. The subject teams, however, remained actively involved in the search for the next target during this interval. Subject teams were given the approximate coordinates of the next target to maximize the

The main effort in discrete control modelling is spent in constructing the network. Once the network is obtained data analysis and the estimation of transition functions can proceed in a fairly mechanical way, but the analysis must start with the network and the success or failure of the modelling effort depends to some degree on the care which goes into the specification of the structure. A few comments about the overall process of discrete control modelling are provided below.

The first step is to determine all of the discrete outputs which the system is required to specify. These normally are the specific decision alternatives which the operators can select from and typically include items like switch settings and other discrete status indicators. Such items generally can be obtained from a detailed analysis of the system which the operators control. In some cases it may not be necessary or desirable to work at the level of individual switches in which case the analyst must define the proper level and specify in unambiguous terms exactly what the output primitives are to be. The individual items identified in this phase of the analysis determine the system output alphabet.

The second step is to identify the exogenous input variables which in some sense drive the system. These might include things like target trajectories, or command information from other systems. Some of this information will probably be in the form of continuous variables in which case rules for interpreting such data in events format must be defined. This step corresponds to some type of feature extraction through which the essential information classes are extracted from the data. For example, targets might be classed as maneuvering or non-maneuvering as a function of their time behavior. In essence the task is to abstract out a small number of information classes which can then be used for discrete control analysis.

At the same time other non-exogenous continuous information such as tracking errors must be represented in events format. There are no preset procedures for accomplishing this but rules specifically designed to match the problem context must be defined.

The next step requires that the elements to be used in the network be defined. It is important to note that, for purposes of data analysis, the state of any system defined must be computable from available information. That is, data analysis cannot proceed if the state of one or more systems in the network cannot be uniquely specified. With this constraint in mind, the process of defining the required or desired systems proceeds in several stages which are often patterned after a level of abstraction hierarchy. First level systems (components) are one level of abstraction away from the primitive data items and they consist of fairly independent subsystems. These can be established on functional grounds or for purposes of forming aggregate information about the primitives. Second level systems are formed in a similar manner from the primitives and the first level systems. These can be formed to provide coordination of the lower level activities and functions, or they can again simply be an aggregation. This procedure of subsystem definition continues until no further systems are needed.

probability of detection. This information simulated the role of an early warning system. Two of the three subject teams completed a total of 23 missions and the third team completed 22.

Four distinct trajectories were used. One was considered a simple "non-threat" type of trajectory, the other three were considered "threat" trajectories by the experimenters. Six disturbance pattern types were imposed on the three easiest trajectories and five were imposed on the most difficult one. The net result was 23 distinct experimental conditions made up of disturbance type, trajectory combinations. Each experimental condition was presented once during each mission (hence 23 targets/mission). Presentation of the conditions was randomized from session to session. Also, data collection was grouped into blocks of four missions for each of the three teams. After each block was completed, the parameters which controlled the onset and duration of the disturbance conditions were modified to prevent learning of the disturbance patterns.

Data were collected in time-series format for every simulated mission in the experiment. The time set of the data collection is the mission time set which means that a complete running record of all measured variables was collected from the beginning to the end of a mission. The data collected included discrete status indicator type information, certain continuous tracking information, and header information to indicate teams and trajectories. All data were collected at a 30 HZ sampling rate. The discrete data collected consisted of all switch settings, and status display states, plus certain variables intended to provide information about the activities and performance of the team.

4. DECOMPOSITION OF THE AAA SYSTEM

The operators of the AAA system together must make decisions which determine which activities are to be engaged in and which mode of operation is to be used at each point in time. Their individual activities and decisions must be coordinated if the system performance level is to be maximized. Such coordination is achieved only if certain information flows through the system and each operator performs his tasks accurately and in a timely manner. The model of the AAA system used for analysis of the experimental data was carefully structured to capture the coordination and communication requirements as well as quantify individual task performance.

The complexity of the total system makes it necessary to decompose the system into a number of smaller, less complex systems which are responsible for certain specific tasks or serve as the information transfer points necessary for coordination. These smaller systems are placed in a network in which the systems themselves are the nodes and the arcs the communication links. The network serves as the model of the team of operators working together on the discrete control problems imposed by the AAA system.

The network is probably best viewed as a related set of internal models. Certain parts of the network are best thought of as the internal models used by specific team members. But, other parts, particularly those which provide common communication points, are best thought of as models shared by two or more of the members. Through training and experience the team members learn the overall mission objectives and they learn what information must be shared, hence developing at some level a common representation of the problem and the system. Decisions made by the operators are presumed to be based on the state of these internal models. Furthermore, the decisions represent the desire to change the state of some component system thus enabling or disabling the occurrence of other events and decisions.

A brief overview of the specific systems used and some rationale for their construction is given below. More details are available in [1].

Several levels of analysis and types of decomposition are needed to construct a representation of a system as complex as this AAA system. The required decomposition takes place along three, not necessarily independent, dimensions:

1) Variable Type
   a) exogenous
   b) endogenous

2) Level of Analysis or Description
   a) primitive component
   b) major components
   c) functional systems
   d) coordination/communication systems
   e) management/command systems

3) System Type
   a) interface or information feedback
   b) decision controlled
   c) event controlled.

In terms of decomposition by variable type, the trajectory number, the target position system, and the disturbance system are the only exogenous variables. These systems are used to provide any target specific information used by other systems in the model and to explain any trajectory specific behavior. All other variables in the system are endogenous and characterize behaviors and decisions produced by the team members in response to the presented targets.

A hierarchical system involving the five levels of analysis listed above was used to establish the elements of the discrete control model. As one moves from the top of the list to the bottom, the view of the AAA system becomes more global and systemic. At the lowest level the perspective is that of individual controls and switches, at the top the perspective is that of overall mission objectives. Upper levels define abstract, less detailed views of the system; lower levels fill in the details. By moving from the top level to the bottom, any question about system performance and operation can be answered.

Level 1, the primitive component level consists of twenty simple systems which correspond to the basic switches and controls used during the experiments. The primitive components are listed in Table 1. These items are the decision or event controlled elements in the system. The state of each component listed in Table 1 must be established by some system higher in the system hierarchy.

Table 1

Primitive Components

Component Name	States
Antenna Horn	Search, Track
Automatic Circular Scan	Fast, Slow
Radar Mode	Circular Scan, Sector Scan, Manual, Automatic
Gun Servo Mode	Semi-Auto, Automatic
0^{o} Lead Enable	On, Off
Lead Enable	On, Off
Mode Switch	Mode I, Mode II
Computer Shunt	On, Off
Data Ready Indicator	On, Off
Coolant	On, Off
Trigger	On, Off
Display in Use	Optics, Not Optics
Azimuth Tracking Control	Rate, Position
Elevation Tracking Control	Rate, Position
Range Tracking Control	Fine, Coarse
Sight Selector	Left, Right
Sight Magnification	2x, 6x
Sight Filter	Clear, Neutral, Yellow
Lower Barrels	On, Off
Upper Barrels	On, Off

The system component level contains two types of systems; distinct system components such as the lead angle computer, and pseudo components defined by grouping certain primitive components which are manipulated together. The level two components are listed in Table 2. Complete state assignment rules are given in [1].

The sight system established the physical configuration of the optical sighting mechanism. The sight selector system is considered separately because of its importance in certain modes of operation (to be discussed when level four systems are discussed).

The range control control component is obvious. The gun configuration system defines precisely which barrels are enabled at any point in time and hence determines the maximum rate of fire possible. The angle track controls component establishes the exact configuration of azimuth and elevation controls.

The gun servo enabling network is a pseudo component. The state of this component establishes how tracking information flows through the system to the input of the gun drive mechanisms. Such flow can be disabled, fully enabled, or standby.

Table 2

List of Level 2 and Level 3 Systems

Level 2 System Components	Level 3 Functional Systems
Sight System	Fire Control Network
Sight Selector	Firing System
Range Control	Gun Directing System
Gun Configuration	Angle Track System
Angle Track Controls	Range Track System
Gun Servo Enabling Network	
Radar Antenna Drive	
Computer	

The radar antenna component defines the physical mode of operation of the mechanism which controls the motion of the antenna and the radar beam characteristics. State 1 is auto track which means that the track or narrow beam is in use and the range signals are under automatic control. The angle signals may also be under automatic control depending on the state of upper level systems. State 2 is manual tracking and states 3 through 6 are the various search modes which the team may use.

The computer is a physical component with three states of interest: standby, settling, operating. Settling refers to the period of time after which the computer is put in use but before a solution is reached. Operating refers to the period during which a lead angle solution is available.

The AAA system has other components, for example the A scope display used by the range operator, but none of these additional components have more than one mode of operation which is of operational significance. The components described above are precisely those which can potentially be used in multiple ways and which reflect the decision making activity of the team members.

Level 3 in the hierarchy is used to abstract out five major systems which perform the several functions which are prerequisite if the system is to meet engagement and mission objectives. These five are also listed in Table 2.

The states of the fire control network are defined to be locked, data enabled, fire enabled. When the fire control system is in the locked state the guns cannot be fired. When in the data enabled state, tracking data are available and with appropriate action by the angle operator the guns can be fired. When the fire enabled state is entered, the guns can be fired at any time. The firing system states are simply firing and not firing. The firing state is entered whenever the trigger is depressed, but firing actually occurs only if the fire control network is in the fire enabled state at the time. In other words, these systems must be coordinated for the overall system to function properly.

The gun directing system characterizes the status of the gun drive mechanism. The states are defined to be standby, 0° lead tracking, and lead tracking. When the gun directing system is in the standby state the guns are not in motion. When in the 0° lead state the input to the gun drive servomechanisms comes directly from the angle tracking system. In the lead tracking state the guns are driven by the output of the lead angle computer. The proper state for this system at any point in time depends on a number of factors which are established by the state of other higher and lower level systems.

The angle track and range track systems must provide the target state data which directly or indirectly drive the gun directing mechanism.

The angle track system states are defined to be optics auto, optics manual, radar auto, and radar manual. These define whether or not the angle operator is monitoring the PPI radar display or using one of the optical sighting systems and whether the angle track data available at the time is produced by automatic control or by the angle operator himself via manual control. The range tracking system, although logically a distinct system, has no autonomy in terms of discrete control. There is only one display and whether the mode is automatic or manual is completely controlled by other systems. The details of the range tracking system, therefore, need not be considered further for discrete control modelling.

Clearly, the systems which form the functional systems level of the hierarchy partition into three distinct groups: range and angle tracking; gun directing system; and fire control network and firing system. These groups define the three major functions of the AAA system: tracking targets, aiming the guns, and firing at the targets. Obviously, each of these systems must function properly for the mission objectives to be met.

Level four in the hierarchy is defined to be the communication/coordination level. The only system residing at this level is the engagement status system. This system is best thought of as a communication center through which information about the current activities of the system is passed. This information is then appropriately distributed and other system activities are enabled or disabled accordingly. The states of the engagement status system are: search, manual track, settling, and valid data. These states define the various conditions the system can be in from the beginning to the end of any single engagement. This system, together with other systems soon to be discussed, establishes whether or not things are progressing normally.

The fifth and highest level in the hierarchy is the management/command level. This level contains one system, the tactics system. This system is the locas of information and decisions concerning basic modes of operation. The tactics system, has five states: Normal Mode 1, Normal Mode 2, Mode 4, Emergency Mode 1, and Emergency Mode 2. Mode 1 refers to full automatic operation during tracking. That is, azimuth, elevation and range tracking data are all under full automatic control once the auto track mode (settling or valid data states of the engagement status system) is entered. The guns are directed by data from the lead angle computer in this mode. In Mode 2, only range data are placed under automatic control when the auto track mode is entered. Angle data are produced by manual tracking. The guns, however, are directed by the lead angle computer.

The emergency designation refers to fire control rather than tracking. In the emergency modes the computer shunt is turned on so that the guns can be fired whether or not the lead angle computer has reached a solution.

Mode 4 operation is a full manual mode in which the radar system is not used and the gun driver mechanism is slaved to the angle tracking output. This mode is functional only if the angle track controls are in the rate mode (State 4). Furthermore, the only display which produces meaningful data in this case is the right optical system.

The state of the tactics system determines how the activities of the major functional systems will be carried out once the engagement status reaches the settling and valid data states. Further, it determines whether the guns can be fired prior to the valid data state. Finally, if mode 4 is selected the normal constraints imposed by the engagement states system are overridden but additional constraints must be imposed on component level systems if the system is to function properly.

The only additional systems which must be defined under decomposition by system type are the interface or information feedback systems. There are four such systems: Tracking Performance, System Performance, Ammunition Balance, and Mission Status.

The tracking performance system provides feedback about the quality of tracking. It is hierarchically defined in the sense that angle tracking errors are deemed more important than range track errors. The states of the system are: no target on any display; angles locked; angles OK, range locked and track OK. No target on display can occur if the tracking error is very large, or if there is no target to track. Angles locked is any case in which azimuth or elevation error is sufficiently large that the automatic tracking system cannot function. Range locked is a similar condition for range tracking error.

The system performance feedback systems attempts to capture some information about overall system performance. The states of this system are: no data, off target, on target. As designed this measure is a very local measure of system performance. It would be desirable to have a more global measure, but implementation problems prevented the use of such variables during this analysis. This system does, however, provide information about time on target, time off target and similar data. Clearly, if one or more major functional systems is not performing adequately, system performance state 3 will not be occupied.

The ammunition balance system determines the relative number of rounds in the upper and lower magazines. In the absence of other information, these data can be used to manage the use of ammunition resources.

The mission status system is used to assess overall ammunition resources with respect to the requirements of the remaining portion of the mission. The state of this system establishes whether or not special ammunition control (i.e., special concern with firing control) is needed if the mission is to be completed without depletion of resources.

These four systems provide the several systems in the discrete control hierarchy with information about local and global performance. This information, particularly any change in state, is used in part to determine if control actions are required.

5. THE COORDINATION AND DECISION CONDITIONING NETWORKS

Thirty-nine simple systems were defined during the process of decomposing the AAA system. These systems are the nodes in the networks which are the discrete control model. These structures organize the available knowledge about the AAA system and the discrete control tasks required for its operation. The objective in this section is to display these networks and examine some of their properties.

Although thirty-nine is a fairly large number of system elements to consider in a model of this type, each system is quite simple. No system has more than seven states and most have only two or three. Furthermore, it will be argued that the states occupied by these systems at any point in time are controlled by a fairly small number of decisions.

Two types of information must flow through the network. First, constraint information or direct control information and second, conditioning or influence information. The first type is said to actually cause specific state transitions to occur in other systems. Such transitions can be deterministic in that a specific state is occupied after the transition, or they can be non-deterministic in which case the new state is required only to be a member of a specific set. Conditioning information on the other hand does not directly constrain behaviors. Rather, it provides information to a given system about the state of other systems and this information may influence state transitions in the system receiving the information. Any state transitions which take place in this case are the result of a discrete control decision and this decision is based in part on the conditioning information in force at the time.

Several systems are controlled both by external systems and by internal decisions, depending on the situation. In specific situations this type of system's actions may be constrained or controlled by some other system in which case it is directly controlled. But, in other cases such constraints are relaxed and the behaviors of the system under question are decision controlled. This is one of the mechanisms by which overall coordination of the system is achieved and it is also a reason for structuring the system in a hierarchical fashion.

Figure 2 is a network diagram which shows the information sources and the receiving systems in the control/coordination network. The arcs (links) in this network should be thought of as communication channels through which the state of the originating node is made known to the receiving node. The state transitions which occur, or which are enabled to occur, in the receiving system are functions of the state of the originating system.

Space limitations prevent displaying the transition graphs of each system in Figure 2. The interested reader is referred to [1]. The general form of these diagrams is shown in Figure 3, which is the graph for the gun directing system. The elliptical

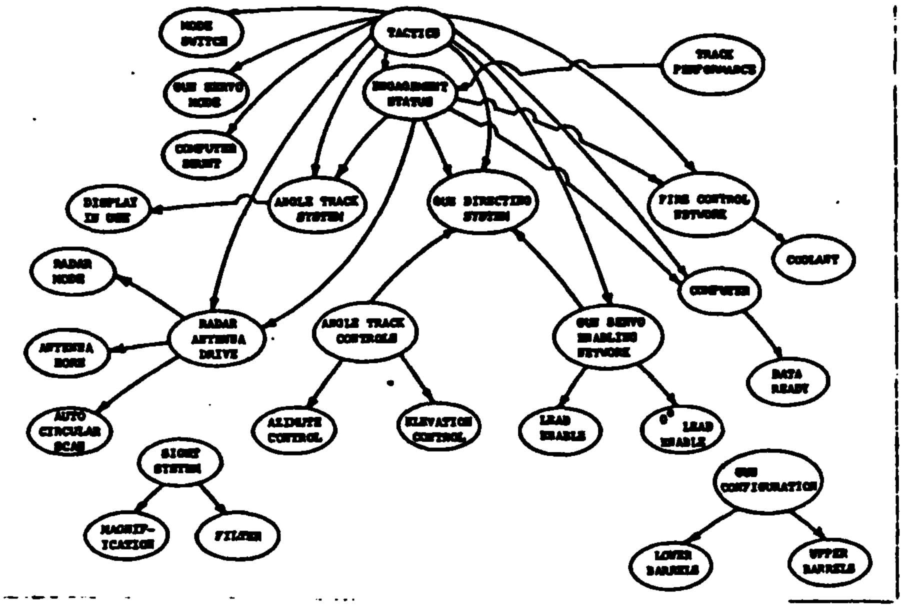


Figure 2.--Control/Coordination Network

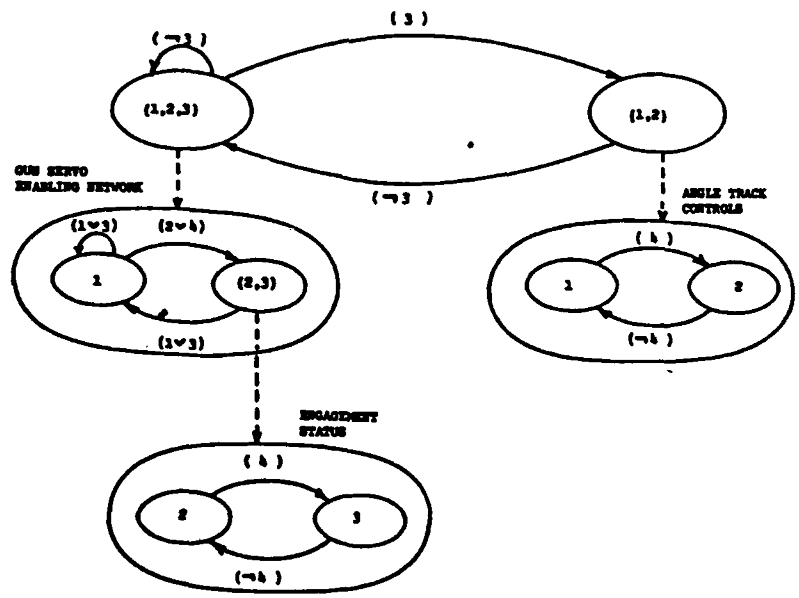


Figure 3.--Gun Directing System State Transition Diagram

figures represent states or sets of states. The description consists of the numerical value or set of values assigned to the node. State transitions are represented by solid lines connecting states and the input conditions which cause the transition to take place are defined by the bracketed symbols displayed on each transition arc. These symbols define logical expressions formed from the possible input values and a transition occurs when the appropriate logical expression is "true." To explain this more completely, any time the value of an input variable changes an event is said to occur and this event leads to a state transition in the system receiving the input. The transition which takes place depends on which logical expression is true at the time. Some nodes do not display transitions from themselves to themselves. These correspond to cases in which any input event will produce a transition out of the node in question.

The primary input variable, in this case tactics, explains transitions on the upper parts of the graph. The effect of secondary and other inputs is shown in those parts of the diagram connected to the primary via dashed lines. Secondary inputs are always used to provide a more detailed explanation of information in the primary diagram. They explain which state or states of the many allowed under the primary condition are actually occupied. A node connected to a primary node with a dashed line (a secondary conditioning variable) should be viewed as a more detailed representation of the primary node.

Several other important properties of the discret control model can be inferred from the coordination/control network and the corresponding transition graphs. First of all, any system for which all maximum resolution nodes contain single states (e.g., the gun directing system) is completely controlled by external sources. These systems for the most part are the lower level primitive components. The second class of systems is that for which one or more maximum resolution nodes is a set of states. Such systems are, at least under some circumstances, partly decision controlled. The third major class of systems, which with one exception do not appear on Figure 2, is event controlled systems. These are the information and feedback systems which interface the finite state systems with the various sources of continuous data. Lists of all three system classes are given in Table 3.

Of the systems controlled by decisions, seven are effectively controlled by external decisions in the sense that only in specific cases, usually indpendent on the tactics state, are they decision controlled. Of these, mode switch, automatic circular scan, antenna horn switch, 0^{o} lead enable, and lead enable, come under decision control only in cases in which their state is of no consequence. Generally, no state change would be made and these systems would remain in the state occupied prior to the occurrence of the event which placed them under decision control. The computer shunt is under decision control only when the tactics system is in state number three. In this situation the firing system cannot be operated unless the operator places the shunt in the on state. Hence, the computer shunt in actuality is constrained to be in state one if the system is to operate in these circumstances. The coolant system is under decision control only when the fire control network is in the locked state. The coolant

Table 3

Breakdown of Systems By Type of Control

Controlled By Internal Decisions	Controlled By External Decisions	Controlled By Exogenous Events
Tactics	Gun Directing System	Tracking Performance
Engagement Status	Computer	System Performance
Angle Track System	Gun Servo Mode	Ammunition Balance
Fire Control Network	Data Ready Indicator	Mission Status
Gun Servo Enabling Net	Display in Use	Target Position
Radar Antenna Drive	Radar Mode	Disturbance
Mode Switch*	Azimuth Control	
Computer Shunt*	Elevation Control	
Coolant*	Sight Filter	
Automatic Circular Scan*	Magnification	
Antenna Horn Switch*	Lower Barrel	
0^o Lead Enable*	Upper Barrel	
Lead Enable*		
Sight System		
Gun Configuration		
Sight Selected		
Range Control		
Firing System		
Angle Track Controls		

*Denotes systems which are effectively external decision controlled. See discussion in text.

state will not influence system performance until the fire control network is unlocked, and in that case coolant is controlled from the fire control network.

If the above systems are removed from the list, twelve systems remain in the decision controlled column. These can easily be partitioned in terms of importance.

The tactics system is obviously a key element. It interacts with ten other systems and it is the key element in establishing the system configuration. Engagement status is also a key element and it provides information to five systems. The angle track system, fire control network, gun servo enabling network and firing system follow in terms of impact on system configuration and overall performance. The remaining systems, although important, provide alternative means of accomplishing the same tasks and they probably have a lesser impact on total system performance. The decision flexibility in the radar antenna drive, for example, is in establishing the specific mode of automatic search. The major activity would be defined at a higher level.

Some general observations about system coordination can be made at this point. The coordination problem faced by the AAA team members might be defined roughly as follows: to direct each major subsystem into the proper state for each phase of an engagement. What is deemed the proper state will depend on the mission status, resources available and the characteristics of the target.

The network described above clearly illustrates a number of coordination activities. Specifically, the selection of the tactics state defines some major parameters which determine the configuration of the system and also the way in which the tracking phase of an engagement is to be carried out. For example, if the tactics system is placed in state three, the system is greatly simplified and the angle operator is responsible for manually finding and tracking any targets. The radar system, range tracking system, computer and most of the displays are of no interest. The communication of tactics information to the appropriate system elements then defines the set of states which those elements can use and thereby constrains behaviors to be consistent with the objective as defined by the tactics system.

The second point which should be made about coordination concerns the engagement status system. Whereas tactics determines the basic system structure and establishes what activities take place, enagement status provides the vehicle for coordinating the time phasing of these activities. In rough terms, the enagement states define what each system should be engaged in at a given time. Engagement status is the system through which the focus of control changes as the engagement evolves. During the search phase the focus in in the angle track system and associated subsystems. The status of all other systems is of very little concern during this time. During manual track the focus includes the angle and range tracking systems. During the settling phase, the focus of control is switching from the tracking systems to the gun directing systems and fire control networks. Once the valid data state is reached the focus is in the fire control network and firing systems. During this phase the other system components are involved primarily in monitoring activities, trying to determine whether or not performance is satisfactory.

The heterarchical nature of the system is quite clear given the above perspective. Tactics sets some major constraints and unless a change is needed control flows to engagement status which in some sense directs control at the appropriate time to the tracking systems, gun directing, and fire control systems. A given system retains control until its task is complete or a lower or high level system intervenes and takes control for some reason. When a given system is the focus of control, the various subsystem which define it are active. The states of these subsystems are manipulated to accomplish the task. When a system is not the focus of control, its subsystems are much less active and generally exhibit no state change behaviors.

Errors and mistakes can also be described in terms of this network. The above discussion is based on the assumption that the operator or operators responsible for a specific activity were in fact prepared to carry it out. If control is given to a particular system and the operator whom this system represents in the specific situation fails to perform, he in essence has failed to accept control. This presumably would be detected and corrected at some point, but it certainly represents a deviation from the design condition and from standard procedure.

A second possible source of error exists in the class of systems which were called effectively decision controlled. Most of these systems have a nominal or preferred state and if for any reason the system state is changed during a period in which it is inactive, this might not be immediately detected when the system next becomes active. The operators would have to detect a problem and diagnose the source before making corrections and if the system causing the problem happens to be one whose state is seldom changed, this could take some time.

In summary then, the coordination/control network shown in Figure 2 defines the architecture of the discrete control system. It defines what information flows through the system, how activities and behaviors are enabled and disabled and they show how the focus of control is passed from one major system element to another. Furthermore, possible sources of error can be identified. These include the failure of an operator to accept control when it is passed to him and failure to detect an improper system state.

From Table 3 it was determined that the important decision controlled systems are the following: tactics, engagement status, angle track system, fire control network, firing system, gun servo enabling network, sight system, sight selected, range control, angle track controls, and gun configuration. Of these several can be removed from consideration because data analysis showed that the subject teams did not use the available alternatives. Tactics, for example, showed almost no decision activity and was set at emergency mode 1 by all teams. Gun configuration can be eliminated because all decision activity was routine and not a major factor in resource management. Of the remaining systems, the overall behavior of the teams can be described in terms of four systems: engagement status, angle track system, fire control network and firing system.

After considerable testing, it was determined that the information needed to condition engagement status decisions is provided by track performance. Transition matrices and time summaries for teams one and two are given in Tables 4 and 5. As would be expected, transitions during the track OK situation are basically the same for both teams. Team two, however, generally shows a longer time in state than team one in this situation. They managed to stay in the valid data state for about 800 time frames which is roughly twice as long as team one.

Transition patterns during the no target phase are also different. Team two preferred to go into manual track from search if a change was made. Team one, on the other hand, actually started the lead angle computer a fair number of times in the no target situation. These transitions, however, took place after about 100 seconds without a target (3000 frames) and, therefore, they may correspond to cases in which the target was not detected.

The matrices for the angles locked case also show team one's reluctance to use manual track and a preference for transitioning from search directly to the settling state, state three. In other words, they preferred to try the computer even though tracking errors were large. The entries in row three show the same tendencies.

Angle track system activity as a function of tracking performance is given in Tables 6 and 7. Both teams in the no target situation show a preference for the radar-auto state, state one. Apparently search was accomplished with the radar system in an automatic mode (sector search). Most transitions from state one were to state two, the radar-manual state. This transition signals the start of manual search. Transitions from state two in the no target situation were most often to the optics-manual state for team one which means they were almost always making a display change, probably in an attempt to find the target. It is interesting to note that both teams almost always (probability .979) transitioned from state two to state 4, optics manual, in both the angles locked and the range locked cases. They both also showed a high probability of exiting the radar-auto state. These imply a very strong preference to complete the target acquisition phase using the optical sighting system. Furthermore, almost all transitions from state four were to state two in the cases where angle error and/or range error were locked. This means that the activity during manual acquisition of the target consisted of display changes. The track OK matrices show the very definite tendencies to get into auto track and use the optical display system.

The conditioning variables for the fire control network are the angle track system and the engagement status. Nearly all team two activity took place in engagement status states one or four (which are search and valid data) and angle track system

Table 4A

Team 1
Engagement Status Transition Matrices

Track Performance	Transition Matrix				Transition Count
	0.000	0.407	0.584	0.009	113
1	0.924	0.000	0.076	0.000	131
(No Target)	0.615	0.038	0.000	0.346	78
	0.907	0.093	0.000	0.000	97
2	0.000	0.295	0.705	0.000	88
(Angle Error	0.000	0.000	0.000	0.000	0
Locked)	0.900	0.000	0.000	0.100	40
	0.800	0.700	0.000	0.000	5
3	0.000	0.530	0.470	0.000	215
(Angles OK,	0.722	0.000	0.278	0.000	72
Range Locked)	0.000	0.000	0.000	0.000	0
	0.000	0.000	0.000	0.000	0
	0.000	0.000	1.000	0.000	128
4	0.000	0.000	1.000	0.000	25
(Track OK)	0.081	0.003	0.000	0.916	298
	0.873	0.127	0.000	0.000	197

Table 4B

Team 1
Engagement Status Time In State Data

Track Performance	Transition	Avg. Condition Time (Frames)	Standard Dev.	Count
1	1-2	816	1181	46
1	1-3	3090	3091	66
1	2-1	1016	1166	121
1	3-1	103	184	48
1	4-1	535	298	88
2	1-2	2060	2020	26
2	1-3	3712	4491	62
2	3-1	50	19	36
3	1-2	2212	2534	114
3	1-3	4402	3557	101
3	2-1	1088	809	52
3	2-3	1950	940	20
4	1-3	3872	3726	128
4	2-3	2580	904	25
4	3-1	94	133	24
4	3-4	101	0	273
4	4-1	378	222	172
4	4-2	546	209	25

Table 5A

Team 2
Engagement Status Transition Matrices

Track Performance	Transition Matrix				Transition Count
1	0.000	0.892	0.108	0.000	93
	0.958	0.000	0.042	0.000	261
	0.455	0.364	0.000	0.182	11
	0.667	0.167	0.166	0.000	6
2	0.000	0.955	0.027	0.018	111
	0.000	0.000	0.000	0.000	0
	0.250	0.750	0.000	0.000	4
	1.000	0.000	0.000	0.000	3
3	0.000	0.805	0.177	0.018	169
	0.738	0.000	0.250	0.012	84
	0.000	0.000	0.000	0.000	0
	1.000	0.000	0.000	0.000	3
4	0.000	0.000	1.000	0.000	152
	0.045	0.000	0.955	0.000	22
	0.033	0.000	0.000	0.967	241
	0.830	0.157	0.013	0.000	223

Table 5B

Team 2
Engagement Status Time In State Data

Track Performance	Transitions	Avg. Condition Time (Frames)	Standard Dev.	Count
1	1-2	804	1132	83
1	1-3	3978	2151	10
1	2-1	518	625	250
2	1-2	2567	3245	106
3	1-2	2680	3035	136
3	1-3	5076	4854	30
3	2-1	425	373	62
3	2-3	835	457	21
4	1-3	5121	4604	152
4	2-3	779	644	21
4	3-4	101	0	233
4	4-1	809	296	185
4	4-1	807	267	35

Table 6

Team 1
Angle Track System Transition Matrices

Tracking Performance	Transition Matrix				Transition Count
1	0.0000	0.716	0.2840	0.0000	455
	0.1361	0.000	0.0005	0.8634	2035
	0.6310	0.000	0.0000	0.3690	274
	0.0000	0.945	0.0550	0.0000	1574
2	0.0000	0.812	0.1780	0.0000	16
	0.0210	0.000	0.0000	0.9790	373
	0.0530	0.000	0.0000	0.9470	38
	0.0000	0.839	0.1610	0.0000	336
3	0.0000	1.000	0.0000	0.0000	2
	0.0160	0.000	0.0000	0.9840	1134
	0.0000	0.000	0.0000	0.0000	0
	0.0000	0.927	0.0730	0.0000	1427
4	0.0000	0.261	0.7370	0.0020	414
	0.8460	0.000	0.0000	0.1540	26
	0.7730	0.014	0.0000	0.2130	497
	0.0000	0.064	0.9360	0.0000	140

Table 7

Team 2
Angle Track System Transition Matrices

Tracking Performance	Transition Matrix				Transition Count
1	0.000	0.627	0.372	0.001	825
	0.594	0.000	0.000	0.406	1093
	0.744	0.000	0.000	0.256	391
	0.000	0.957	0.043	0.000	507
2	0.000	0.912	0.088	0.000	34
	0.021	0.000	0.000	0.979	188
	0.333	0.000	0.000	0.667	9
	0.000	0.973	0.027	0.000	111
3	0.000	1.000	0.000	0.000	17
	0.033	0.000	0.000	0.967	548
	0.000	0.000	0.000	0.000	0
	0.000	0.936	0.064	0.000	516
4	0.000	0.437	0.560	0.003	359
	0.833	0.000	0.000	0.167	18
	0.784	0.114	0.000	0.102	324
	0.000	0.031	0.969	0.000	163

states three or four (optics-auto and optics-manual). That is, team two almost always used the optics when firing. Team one showed an unexpected amount of activity with the angle track system in the radar-auto state. This suggests that they may have on occasion prepared to fire before switching to the optical sighting system.

The condition variables for the firing system are the fire control network and engagement status. One interesting item was observed from the firing system analysis. Team one had a tendency when manually *tracking (engagement status one)* to try to fire the guns without first putting the fire control network into the fire enabled state. It appears that when involved with the tracking activities they sometimes forgot how the system worked and deviated from the standard procedures. Team two did not have this problem.

The decision conditioning network is the scheme by which the conditioning which was used in the above analysis is best described. This network which was empirically derived is shown in Figure 4. Essentially, tracking performance influences decisions made in the angle track and engagement status systems. Engagement status then influences the activities of the firing system and the fire control network as well as some of the lesser systems. Those systems shown in Figure 4 which are not connected to any other system are those systems which were either routinized or showed little activity. This simple network localizes essentially all of the decision making activity which was shown in the data. The conjunction of this network and the coordination/control network shown in Figure 2, together with the appropriate state transition graphs, are the discrete control model obtained from an analysis of the system and a comprehensive analysis of the data.

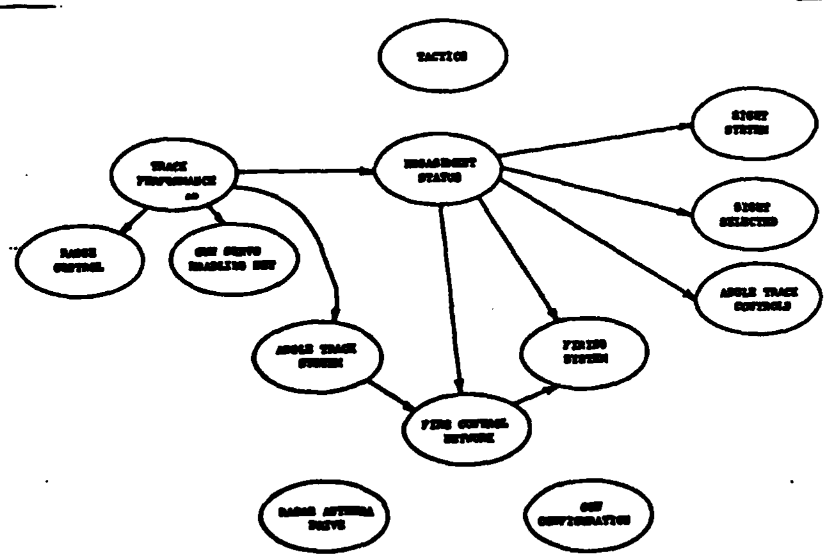


Figure 4.--Decision Conditioning Network.

6. CONCLUSIONS

Based on the analysis it is fairly easy to construct a scenario of the way in which the subject teams performed their tasks. First of all, search for targets was generally accomplished using the automatic sector search mode and the primary display was the PPI radar screen. In cases where the target was slow to appear some teams might occasionally switch to the left optics sighting system and less often to the right optical system. As soon as the target appeared the angle operator would switch to the left optics system and switch from the radar driven search mode to a manual mode for acquisition. During this period the range operator was switching his control back and forth between the coarse and fine setting. The commander was resetting the gun servo enabling system to state three and thereby moving the gun directing system from standby to 0^{o} lead tracking. Also during the acquisition phase and the early phases of tracking the angle operator determined which of the four trajectories the current target was following. This determined in part the strategy that he then followed.

Once the target was acquired the system was either put into auto track or, if the trajectory was particularly easy, tracking often continued in a manual mode. Team two used manual tracking for the most difficult highly maneuvering target as well. If manual tracking was used, firing started within about 20 seconds of acquisition (i.e., when the target came within range). In cases where auto track was used, firing started shortly after the fire control solution was achieved. Easy targets were disengaged after a few hits. The gun directing system was then generally disabled for a few seconds after the target disappeared from the screen. The angle track system was then reconfigured for search and the cycle started again.

Several interesting observations can be made about the performance of the teams. Team one had some problems with the fire control network interlocks when they used manual tracking. They forgot to enable firing before trying to fire. The other teams had no problem and team one had no problem in the automatic tracking cases. Team one may have become so involved in performing the tracking task that they forgot how the system worked. There were other pieces of evidence which showed that the teams infrequently made incorrect switch settings or failed to reconfigure the system quickly enough.

It was a surprise that no team used the six power optical system very much for tracking. Apparently the feedback provided by tracers reduced the need for precise visual information. It definitely seems that a style of tracking was used in this experiment which differed qualitatively from that used in simple tracking studies.

In general terms the commander's tasks were very trivial. Teams obviously learned the limited number of trajectories which were used and they keyed their actions to the trajectory. The attempt to introduce uncertainty via the disturbances or simulated countermeasures did not seem to have much impact. They may have delayed the start of auto track for example, but they did not alter the basic patterns of behavior as represented by the various transition matrices of the discrete control model. The fact that every mission contained exactly the same number of trajectories greatly simplified ammunition management. The subject teams knew that they could and should go after all targets. They did not have to be selective or evaluate the threat potential of any target. There was certainly no risk associated with missing one and there was no significant scoring penalty.

The model which seems to best capture the various teams' performance is really a set of finite state systems organized into two networks, the coordination/control network (Figure 2) and the decision conditioning network (Figure 4). The systems included in these networks were established through a detailed analysis of the AAA system, its functions, and the tasks of the operators. By decomposing along several structural dimensions, and particularly by analyzing at several levels of abstraction, an effective and useful representation of the discrete control system was obtained.

In general terms this representation is a model of an organizational structure which the operators might use to reduce the apparent complexity of their task and generally achieve coordinated actions and acceptable performance. It is really just a structured representation of the available knowledge of the system and its functions.

The coordination/control network is basically hierarchical and reflects the constraints on lower level decision making activity imposed by upper level decisions. In the terms of the finite state systems representation, state transitions in lower level systems are disabled, enabled or constrained as a function of the state of upper level systems. The decision conditioning network establishes the information flow patterns which are needed to explain, at least in part, the decisions which are made (i.e., the state transitions which take place). The systems in the decision network are represented by generalized stochastic automata in which state transitions are conditioned by the information flowing into the system from other nodes in the network. The two networks in conjunction form a heterarchical system description in which decision making activity flows from one functional area to another as a function of the established constraints and the environmental situation.

In general terms then, the discrete control methods so far developed seem to have potential. They can be used to make sense out of complex systems and identify the key decision points. They can describe quite complex behaviors in terms of a relatively small number of decisions. The structure of the model is quite easy to understand and the individual finite state systems are all simple and intuitive. Grasping the overall view; i.e., all levels simultaneously, is more difficult and amount of statistical information which can be produced is overwhelming. These problems are minimized, however, if one restricts attention to only the one or two levels which are most important for a given question.

REFERENCES

1. R. A. Miller, Ohio State University Research Foundation, Ohio State University, "Identification of Finite State Models of a Human Operator," Final Report, AFOSR Grant No. 77-3152, March 1979.

ACKNOWLEDGEMENTS

This research was supported in part by the Air Force Office of Scientific Research, Grant #77-3152 and this support is gratefully acknowledged.

APPENDIX B

This paper was presented at the 1979 International Conference on Cybernetics and Society and appears in the proceedings. (pp. 210-216).

A FINITE STATE SYSTEMS MODEL
OF COORDINATION IN MULTI-PERSON TEAMS

R. A. Miller
Department of Industrial and Systems Engineering

The Ohio State University
Columbus, Ohio 43210

Abstract

A review of several diverse concepts of coordination drawn from the motor performance, man-machine systems, and organizational behavior literatures is presented. Several key properties of coordination are then abstracted and used to synthesize a structure which can be used to formally address problems of coordination within small teams. The structure is a network of nondeterministic finite state systems and it is argued that this structure corresponds to an internal model of the task/environment/team system which is shared by the team members.

I. Introduction

Questions of crew or team design and the corresponding questions of performance have long been within the purview of man-machine systems researchers. Much knowledge, particularly behavioral knowledge, has been accumulated via the many studies which have been performed (see, for example, [1] for an excellent review). At this point general guidelines are available for use by system designers (e.g. "Visual methods of communicating information within teams are superior to verbal methods," [1] pg. 287) but methods for predicting team performance are much less readily available. [1,2,3]. Furthermore, conceptual or analytic structures which can serve as the foundation for theory construction and systems oriented empirical research do not seem to be currently available. The purpose of this paper is to present a fairly general mathematical structure based on the theory of finite state systems which has proved useful in modelling certain aspects of small team performance. This structure seems to be particularly useful for defining, clarifying and analyzing team coordination problems and it can be used in conjunction with simulation for performance prediction.

In the usual manner, teams are clearly distinguished from more general small groups by the constraints under which they operate. Teams are assumed to be well organized, highly structured, and they are assumed to utilize relatively formal operating procedures [1]. Typical examples include tank crews, train crews and the cockpit crews of large commercial and military aircraft. The structured nature of teams is heavily used in the modelling scheme which is developed.

The paper is organized in the following way. The concept of coordination is discussed in the next section to provide the required background for modelling. In section 3 the finite state modelling structure is presented. Some general observations about applicability are made in the final section.

II. The Concept of Coordination

The concept of coordination is one which seems to be intuitively simple and clear, but it remains doggedly hard to formally define. The concept has certainly been recognized and used in a variety of disciplines ranging from psychology, economics, management science and systems engineering (see for example [4], [5], [6], [7]). Several of these views of coordination are briefly outlined in this section to provide a reasonable foundation for the formal development which follows in the next section.

In general terms, a system is thought to be coordinated when its component parts are adjusted in such a way that the overall system performance or behavior is in some sense harmonious. There are obviously a number of imprecise terms used in the above statement, but it clearly contains the gist of most types of coordination ranging from eye-hand coordination and coordinated movement of limbs of the body to the coordination of decentralized divisions of an organization.

There are two key attributes which follow from the above general definition. First, the system of interest must consist of more than one subsystem and second, the behaviors of the subsystems must be constrained and synchronized in some manner. Furthermore, the various subsystems must be at least partially autonomous and must be capable of producing more than one behavior. Otherwise the issue of coordination becomes synonomous with centralized control.

The nature of the sybsystems is one of the things that differentiates uses of the concept of

This research was supported in part by the Air Force Office of Scientific Research under grant number AFOSR-79-0015.

coordination across disciplines. For example, muscle groups might compose the subsystems when motor coordination is studied [4,8] whereas divisions or departments might be the basic objects in the study of organizational behavior [6,9]. Clearly, there is no particular reason to constrain attention to physical or organizational subsystems. Tasks and activities are also legitimate objects of coordination and, precisely speaking, it is usually the activities of subsystems which one wants coordinated rather than the subsystems themselves. The concept as it is traditionally used in the man-machine systems area supports this view (see for example Roby [10] Chapter 13).

It is generally agreed that coordination requires communication and many studies, particularly organizational behavior studies, have explored the relationship between the degree of interdependence of subsystems and communications requirements [6, 11]. Similarly in the man-machine systems area, communications are the subject of a substantial proportion of all team performance work. [1, 2, 10]. Murphy [2] for example reviews a number of suggestions for improving air crew performance and display and/or information systems modifications constitute the majority of them. Meister [1] argues that the system developer's interest is designing and training teams to work efficiently and that the distinctive element of teams is "teamwork," (pg. 231). But, he further points out that researchers who have attempted to isolate the teamwork element usually resort to "communication" as the operational definition.

Coordination then is a concept which is potentially relevant whenever a system is composed of a number of subsystems with communication paths linking at least some of them together. Most researchers would probably argue that this is a necessary but not sufficient condition because the subsystems must exhibit some degree of autonomy before questions of coordination arise. That is, if there is no autonomy, the system functions in a centrally controlled manner. For example, one does not usually speak of a coordination problem when analyzing an electrical circuit constructed of several discrete components. Here the operation of each component is simply determined by its state and the signals it receives. It has no decision-making capability and no choice in the generation of a response. It does not possess the flexibility to operate independently.

Constrained independence is a view of coordination which is quite common, particularly in motor performance. Turvey et. al. [4] use this concept heavily when developing the idea of a coordinative structure (a group of muscles controlled as a unit). Ashby [12] argues that coordination is identified with "deviations from statistical independence in an n-dimensional frequency table." This somewhat overly simplified view does emphasize two important facts: 1) potential for a variety of behavior on the part of the components and 2) simultaneous constraints on the various "degrees of freedom" imposed by coordination. In motor performance then, coordinated movements are marked by constraints on the many degrees of freedom of the limbs which produce the motion. The view expressed by Turvey et. al. [4] is that this constraining is not accomplished through central control of each muscle, but by activating a relatively small number of coordinative structures which are tuned to the specific task.

One important attribute of coordination apparent in the constrained autonomous system view presented above is that coordination in general is accomplished not through preprogrammed or controlled activities but rather through the definition of strategies for such activities. Specific actions or activities are contigent upon or conditioned by events which may be either internal or external to the overall system. In other words, coordination tends to be a structural property of the system more than a behavioral one. However, specific behaviors will clearly depend on the underlying structure and reflect the coordinative strategy.

This perspective is a common one in the organization theory literature. March and Simon [6] go so far as to say that "The problem of arranging the signalling system for interdependent conditional activities is the coordination problem." (pg. 28). Clearly, they view coordination as a problem of organizational design, a problem of architecture. This does not mean that agents or systems whose primary function is coordination cannot be included. Obviously such systems exist and in fact they are the focus of a tremendous amount of research [9]. But, the important point is that it is not the activities of these systems which constitute coordination. Rather, it is the relationship of such agencies with other components which accomplishes coordinated activities. The function of coordinating agencies is to serve as common communication points and provide signalling and sequencing information to other subsystems.

The structural view of coordination is common in man-machine research. Murphy [2] and Meister [1] both address questions of task organization, crew organization, the design of operating procedures, communications procedures and their impact on crew performance. In other words, the design questions have for the most part guided man-machine and human factors work on crew performance. Most of this work, however, simply assigns to individual team members the role played by departments or divisions in organizational behavior studies. Communications then generally are reduced to questions of who talks to whom and when. Some attention has been paid to the relationship between overall team performance and team structure but it is generally agreed that these issues remain quite poorly understood (Meister [1], pg. 236).

It would be a mistake to assume that man-machine research has produced the concepts required to talk sensibly about team performance and coordination. As pointed out by Meister [1], most formulations consider teams to be described by the same processes used to describe individuals. This clearly is not adequate but better representations have not been forthcoming.

To complete this brief review of notions of coordination, a few observations about what might be called a systems engineering perspective are made. The work by Mesarovic and his colleagues [7] is without doubt the most cited work on coordination in the engineering disciplines. For our porposes only two features of their work need to be discussed.

First, they do not utilize a structural perspective in the sense discussed above. Rather, the problem of coordination as defined by Mesarovic is a "second level" optimization problem through which coordinating information is determined and then transmitted to lower level systems. In other words, the emphasis is on the determination of the information which is to be transmitted rather than on the architecture of the system itself. It is not too much of an overstatement to say that the emphasis is placed on the abstract language used for communication rather than on coordination and communication per se.

The most interesting aspect of the Mesarovic construction is the set of coordination principles which he presents. Specifically, three principles (interaction prediction, interaction decoupling, interaction estimation) are cited as the means available to a supremal coordinating unit to influence the decision processes of the infimal units which brings us to the second major point. This perspective assumes that all systems exist only to solve specified decision or control problems and that the so called infimal units are totally non-interacting. Hence, coordination is required to achieve overall optimal performance. The point is, that this is a very limited class of problems, particularly in the small team situation. At best it corresponds only to centralized coordination [10] utilizing a feedback process [6]. Neglected are other schemes such as chained coordination [10] in which control shifts from one agent to another, and parallel [10] coordination which deals with within subsystem sequencing and timing. Furthermore, it does not allow for coordination via less explicit forms of coordination, e.g. coordination by program or plan ([6] pg. 160).

It is believed by at least some researchers that not all interactions that take place within teams are explicit communications [1]. In fact, key issues in small team research are identifying the processes by which teams perform their tasks and making more explicit the nebulous interactions which occur. The systems engineering results provide little help with these problems.

In summary then, the concept of coordination applies to systems which have the following properties:

- a set of partly autonomous subsystems
- some means of communicating between subsystems
- performance requirements that depend on more than individual subsystem performance.

Further, the organizational behavior literature provides some clues and ideas about coordination, but it porvides no explicit formal models. The optimization oriented systems engineering literature, and the math programming literature focuses on problems which are much too narrow in scope to be of very much use in man-machine systems work.

Generally speaking, the man-machine systems literature provides a number of useful guidelines and a substantial amount of data, but it doesn't provide much of an integrated conceptual structure. Murphy [2] states "Crew performance may be conceptualized as an interplay of two complex systems: the crew and the task environment." Unfortunately, the technical means to operationalize this concept in an integrated model do not yet fully exist. The construction presented in the next section has proved to be of some use in formally representing the concept of coordination within small teams. This construct is systemic in scope and it may ultimately provide one method of relating structural, design oriented information and team performance.

III. A Network Representation of the Team/Task System

The general class of systems of interest consists of that in which a team of operators have available a finite number of available alternatives with which to either directly or indirectly control the behavior of some vehicle, process or plant over time. In most applications control will be indirect in the sense that alternatives will refer not to specific control devices but rather to more general control related tasks or to modes of operation.

The modelling questions focus on the problems of capturing in some mathematical representation the way in which team members (or the system designer) might decompose a complex problem into simpler parts and how they then manage to coordinate their individual activities and configure the system so that acceptable overall system performance is achieved. The basic questions therefore are questions of knowledge representation and dynamic communication in a complex system. A general hierarchical/heterarchical structure which allows for structural coordination of subsystems by upper level components and which utilizes a heterarchical control structure to shift the focus of control to the proper subsystem at the proper time has been developed.

For purposes of this paper a modelling perspective, rather than a design perspective, is taken. A given team, considered as a unit, performs the required tasks and behavioral and performance data are obtained form the experiment.

The modelling task is to provide some mathematical structure which can to some degree interpret, organize and explain the observed behaviors.

The modelling process involves several steps starting with a thorough analysis of the overall task faced by the team and of the environment in which they perform. This analysis must identify candidate functions and procedures and it usually involves decomposition of the system/environment along several dimensions. A structure must then be synthesized to meet each function in a coordinated way.

For this work the synthesis involves constructing a set of finite state systems each of which represents procedural or factual information about some function. These systems are then formed into a network, somewhat like a cellular automata [13, 14], which is a dynamic representation of the knowledge structure required to perform the task. This structure is probably best thought of as a team "internal model" in that it contains the factual and procedural knowledge about the task and environment shared at least in part by members of the team. This representation then is an extension of the internal model concept used so much in single operator performance modelling (see for example [15]). Some of the technical aspects of the network modelling procedure are now presented.

It is assumed throughout this development that information about the controlled system and the environment which is displayed to the operator, or otherwise provided him, is discrete. In other words, communication is accomplished through a language with a finite number of symbols. This is not a restriction in the type of systems of interest given that highly stylized and formalized means of communication are typically used. It is further assumed that time is recorded only at discrete points, i.e. the time set is discrete. This assumption is made only for clarity of presentation and can be easily relaxed. The reader is referred to [17] for details.

Now, let T denote the time set of interest (interval of data collection, or mission time set, etc.); let A denote the input set (set of possible input symbols to the team, assumed finite); and let B denote the output set (set of explicit alternatives available to the team members). A behavioral representation of team performance is then a relation S,

$$S \subseteq A^T \times B^T \qquad (1)$$

where

$$A^T \triangleq \{x/x: T \rightarrow A\}$$

That is, A^T is the set of time functions with domain T, codomain A or in otherwords it is the set of possible input sequences. B^T is similarly defined. The system S then is the relation consisting of the possible input-output behaviors which can be constructed from input symbols in A, and output symbols in B. This system can be thought of as the sample space from which all behaviors in an experiment involving the team must be drawn. The explicit modelling task is to provide a constructive specification of *S*. This is accomplished with the network of finite state systems.

A nondeterministic finite state system is specified by a quintuple (A, B, C, λ, Φ) where A and B are as defined above, input and output sets respectively, C is a set of states (state space), λ is an output assignment function, and Φ a nondeterministic next state function. Specifically,

$$\lambda: C \rightarrow B \qquad (2)$$

$$\Phi: A \times C \rightarrow \Pi(C) \qquad (3)$$

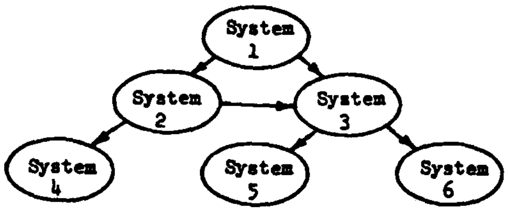
where $\Pi(C)$ denotes the power set of C (i.e. set of all subsets of C). Clearly, λ assigns an output symbol to each state and Φ assigns a set of states to each input, state pair.

A deterministic finite state system differs from the above only in terms of the state transition map. In the deterministic case the codomain is the state space C rather than the power set $\Pi(C)$. The interpretation is that in the nondeterministic case the current input and current state are not sufficient information to specify precisely which state will next be occuppied, but they are sufficient to constrain the next state to be within a given set.

Probabilistic finite state systems differ from nondeterministic ones in that a probability distribution is defined on the state space. In this case not only is the set of next states known, but the probability that a particular state will be occuppied is also known.

Nondeterministic, and to some degree probabilistic, finite state systems form the basic components in the team model.

The network shown in figure 1 is a simple illustration of the concept. Each system or node



Network of Systems
Figure 1

in the network is a nondeterministic finite state system represented by the sets and functions of the type mentioned above. The arcs of the network represent the paths by which output information from a given system is communicated to other systems in the network. The overall network then forms a constructive specification of the system S (see equation (1)). A few additional technical details are required before the representation is complete.

At one level of abstraction the network is simply a directted graph with nodes consisting of the systems and arcs consisting of communication links. Formally, let

$$N = \{i \mid i \text{ a system name}\}$$

and let

$$G \subseteq N \times N \ \epsilon\ (i, j) \ni G \quad (4)$$

if and only if system j is connected to system i via a link from i to j. In the example of Figure 1.

$$G = \{(1,2), (1,3), (2,3), (2,4), (3,5), (3,6)\}$$

The graph G captures the essence of the communication paths, but there is still some room for ambiguity. For example, system 1 in Figure 1 communicates with both system 2 and system 3. Clearly, different information could be sent to each of these. The ambiguity is best resolved by allowing multiple input-output paths, or in other words allowing for multidimensional input and output sets.

Suppose that system i is defined in terms of Ai, Bi, Ci and

$$\Phi i : Ai \times Ci \to \Pi(Ci) \quad (5)$$

$$\lambda i : Ci \to Bi \quad (6)$$

If there are multiple input and output paths the sets Ai and Bi must be cartesian products of the form:

$$Ai = Ai_1 \times Ai_2 \times \ldots Ai_k$$

$$Bi = Bi_1 \times Bi_2 \times \ldots Bi_e$$

with each set in the product representing the set of symbols which can be passed over the specified communication channel.

The last remaining technical issue concerns the compatibility of interconnections. Consider a labelling function L,

$$L : G \to I \times I \quad (7)$$

defined so that $((i, j), (k, m)) \in L$ if and only if output k of system i is connected to input m of system j. The labelling function then defines the detailed signal path from system to system and it can be used to identify the above mentioned consistency requirement. If $(k, m) = L(i, j)$ then $Bik = Ajm$. That is, the output sets and input sets used at opposite ends of a channel must be the same.

There are a few more conditions dealing with systems that communicate in only one direction which must be addressed. First some more notation. Let

$$G^1 = \{i \mid \exists\ j \ni (i, j) \in G\}$$

$$G^2 = \ j \mid \exists\ i \ni (i, j) \in G\}$$

These are the source and receiving nodes, respectively. Now, let

$$N_I = \{i \mid i \in N,\ i \notin G^2\}$$

$$N_O = \{i \mid i \in N,\ i \notin G^1\}$$

N_I consists of those nodes which do not receive information from any system and N_O is those which do not send to any system. All systems in N_I are constrained to have no input alphabet, i.e.,

$$j \in N_I \Rightarrow \Phi j : Cj \to \Pi(Cj).$$

The systems in N_I then in some sense represent the basic information input to the system. The outputs of systems in N_O on the other hand define the overall system outputs.

Suppose that

$$N_I = \{i_1, i_2, \ldots i_k\}$$

and

$$N_O = \{j_1, j_2, \ldots j_e\}$$

Then, from the argument above

$$S \subseteq (Bi_1 \times Bi_2 \times \ldots Bi_k)^T \times (Bj_1 \times Bj_2 \times \ldots Bj_e)^T$$

forms the behavioral representation of the overall system. In abstract terms this relation is a more detailed construction of the behavioral representation (1) with

$$A = Bi_1 \times Bi_2 \times \ldots \times Bi_k$$

$$B = Bj_1 \times Bj_2 \times \ldots \times Bj_e$$

Systems constructed in the above fashion have a few additional properties which should be pointed out. First of all, systems in the set N_I are completely autonomous and do not receive inputs from any other system. Such systems are used to represent exogenous information sources such as systems or actors in the task environment which provide information to the team, but are not significantly influenced by the actions of the team over the time interval of interest.

A second key point concerns the interpretation of the nodes. Most network models which have been used in the past to model teams place specific team members at the nodes. This is generally not the case here. The overall structure, as mentioned previously, is best thought of as an internal model of task/team environment system which is shared by the team. Nodes then can correspond to systems for performing specific tasks, to knowledge about specific pieces of equipment, or to any other relevantinformation. In some cases a subset of the nodes would correspond to the knowledge structure held by an individual operator but this need not be the case. More generally, the overall structure is shared by all team members, but specific members are responsible for the decision-making or other activity associated with specific state determination within a given subsystem. That is, each member could have the same general knowledge of system organization plus any specialized knowledge or skill required to perform those tasks for which he is responsible. Shared responsibility is not precluded.

Space limitations do not permit the presentation of a detailed example of such modelling. Such an example is available in [16] and [17] and the interested reader is referred to those sources. Suffice it to say that with a careful and detailed analysis of system requirements, a network realization of the knowledge structure, which must be developed by a team in order to successfully meet these requirements, can be constructed. Such structures have potential normative and training applications as well as the descriptive applications discussed here and in [16].

IV. Conclusions

In summary, the representation consists of a set of nondeterministic finite state systems of the form (5) and (6), together with a directed graph (4) and a labelling function (7). Together these structures establish a network of communicating systems in which behaviors are interdependent and yet some autonomy is maintained by individual subsystems (nodes in the network). Specifically, the use of nondeterministic finite state systems means that information received by a given subsystem does not completely specify the state of that system. Rather, it only constrains the state to be in a specified set. The interpretation is that subsystems (or the operators responsible for the subsystem) determine the specific state within the set established by the overall constraints and thereby show some autonomy. If desired, these decision processes can be described with probabilistic finite state systems (see [17] for details) or they can be described via other decision theoretic models.

The key substantive point is that the systems and the network which organizes them constitutes an abstract knowledge representation, or internal model, shared by the team members. This structure establishes interdependencies and constraints via the nondeterministic systems which form the network. For example, activities in some subsystems can be conditionally disabled by passing information to that system which results in a transition into a standby state. Similarly, some task or activity can be enabled by the same type of process.

Overall coordination is achieved in precisely the same manner. The structure of the system, which by the way includes specification of the transition functions of the subsystems, determines how information is shared, used and stored. The structure defines the overall schema, and given that it is composed of dynamic systems, specific behaviors (i.e. specific time histories) will show event and context dependencies.

In essense, this model represents teams as a parallel or distributed information processing systems. It presumes a well defined, and to some degree formalized, communication system and it general requires highly structured tasks. As was pointed out in section 2, these are precisely the conditions which distinguish teams from small groups.

A very importnat property is that no special type of coordination is presumed. Central, chained, parallel or any other type can be realized through definition of the subsystems and the network. In fact, coordination is achieved through special organization of the component systems, and specific types of coordination represent special, constrained realizations of the overall system.

The level of abstraction of this type of modelling is such that the emphasis is clearly placed on structural issues and not on specific task performance or on the procedures and algorithms required to accomplish a given task. The model deals with how the overall task/environment/team system is interconnected and how constraints are conditionally defined over time, but it does not deal with <u>how</u> a given subsystem state is determined from the set allowed by the nondeterministic next state transition function. This is as it should be. The selection or performance of a specific task is a much more restricted and localized problem better studied with existing man-machine system research methods.

The ideas so briefly described here have been used to model one fairly complex system [16, 17]. The techniques continue to be refined and developed with much current effort being expended to clarify and identify the detailed mathematical structure of the network models. The objective is to identify major properties and to make some progress toward using such constructs in the design process. The methods are also being used for analyzing certain coordination problems in a simple motor performance task.

References

1. Meister, D., <u>Behavioral Foundations of System Development</u>, John Wiley and Sons, New York, 1976.

2. Murphy, M. R., "Coordinated Crew Performance in Commercial Aircraft Operations," <u>Proceedings of the 21st Human Factors Society Annual Meeting</u>, San Francisco, Calif.,

Oct. 1977.

3. Pew, R. W., S. Baron, C. E. Feehrer, and D. C. Miller, "Critical Review and Analysis of Performance Models Applicable to Man-Machine Systems Evaluation," B. B. N. Report #3446, March 1977.

4. Turvey, M. T., R. Shaw, and W. Mace, "Issues in the Theory of Action: Degrees of Freedom, Coordination Structures and Coalitions," in J. Requin (Ed.), Attention and Performance, vol. 7, Erlbaum, Hillsdale, N.J., in press.

5. Arrow, K. J., and L. Hurwicz, "Decentralization and Computation in Resource Allocation," in R. W. Pfouts (Ed.), Essays in Economics and Econometrics, Univ. of North Carolina Press, Chapel Hill, N.C., 1963.

6. March, J. G. and H. A. Simon, Organizations, John Wiley and Sons, New York, 1958.

7. Mesarovic, M. D., D. Macko, and Y. Takahara, Theory of Hierarchical, Multilevel Systems, Academic Press, New York, 1970.

8. Greene, P. H., "Seeking Mathematical Models for Skilled Actions," in D. Bootjin and H. C. Muffley (Ed.), Biomechanics, Plenum Press, New York, 1969.

9. Johnson, R. A., F. E. Kast, and J. E. Rosenweig, The Theory and Management of Systems, 3rd Edition, McGraw Hill, New York, 1973.

10. Roby, R. T., Small Group Performance, Rand McNally, Chicago, 1968.

11. Baker, F., Organizational Systems: General Systems Approaches to Complex Organizations, Irwin, Homewood, Ill., 1973.

12. Ashby, W. R., "The Set Theory of Mechanism and Homeostaesis," in D. J. Stewart (Ed.) Automation Theory and Learning Systems, Academic Press, New York, 1967.

13. Codd, E. F., Cellular Automata, Academic Press, New York, 1968.

14. Jackson, P. C., Introduction to Artificial Intelligence, Petrocelli/Charter, New York, 1974.

15. Jagacinski, R. J. and R. A. Miller, "Describing the Human Operator's Internal Model of a Dynamic System," Human Factors, Vol. 20, Number 4, 1978.

16. Miller, R. A., "A Discrete Control Analysis of Coordination Activities in a Simulated AAA System," 14th Annual Conference on Manual Control, Wright State University, 1979.

17. Miller, R. A., "Identification of Finite State Models of Human Operators," Ohio State University, Systems Research Group, March 1979.