1976

Sampled data adaptive digital computer control of surface ship maneuvers.

Uhrin, John Joseph

Monterey, California. Naval Postgraduate School

SAMPLED DATA ADAPTIVE DIGITAL COMPUTER CONTROL OF SURFACE SHIP MANEUVERS

John Joseph Uhrin

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SAMPLED DATA ADAPTIVE DIGITAL COMPUTER
CONTROL OF SURFACE SHIP MANEUVERS

by

John Joseph Uhrin III

June 1976

Thesis Advisor: George J. Thaler

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Sampled Data Adaptive Digital Computer Control of Surface Ship Maneuvers		5. TYPE OF REPORT & PERIOD COVERED Engineer's Thesis; June 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John Joseph Uhrin III		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1976
		13. NUMBER OF PAGES 371
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Underway Replenishment, Ship Dynamics, Hydrodynamic Interaction, Destroyer Maneuvering, Simulation, Ship Propulsion, Hydrodynamics, Sea State, Low Order Models, Ship Speed Control, Optimal Control, Ship System Modeling, Replenishment At Sea, Nonlinear Control,

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

The replenishment at sea (RAS) maneuver is studied in detail for heading and speed control. design of purposefully nonlinear control laws is accomplished for the Mariner hull using the linearized equations of motion in three degrees of freedom. Extensive use of low order modeling and optimal control theory was made. Procedure steps are presented in detail to facilitate redesign for other ship types. The results are verified using

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20.

DSL simulation for a number of possible RAS scenarios. The control systems are also tested in a sea state to insure proper operation in the presence of external perturbations.

19.

Automatic Control, Control System Design, Adaptive Control, Automated Ship Control, Digital Computer Control

Sampled Data Adaptive Digital Computer Control of Surface Ship Maneuvers

by

John Joseph Uhrin III
Lieutenant, United States Navy
B.E.E., Villanova University, 1967
M.S., Naval Postgraduate School, 1975

Submitted in partial fulfillment of the
requirements for the degree of

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1976

ABSTRACT

The replenishment at sea (RAS) maneuver is studied in detail for heading and speed control. Design of purposefully nonlinear control laws is accomplished for the Mariner hull using the linearized equations of motion in three degrees of freedom. Extensive use of low order modeling and optimal control theory was made. Procedure steps are presented in detail to facilitate redesign for other ship types. The results are verified using DSL simulation for a number of possible RAS scenarios. The control systems are also tested in a sea state to insure proper operation in the presence of external perturbations.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (cont.)

LIST OF FIGURES (cont.)

LIST OF FIGURES (cont.)

LIST OF FIGURES (cont.)

LIST OF FIGURES (cont.)

TABLE OF TERMS AND ABBREVIATIONS

ADX - DX referenced to control ship's head

ADY - DY referenced to control ship's head

Alongside - position at which longitudinal position ADX is 0.0

Approach Phase - phase in RAS scenario at which the control ship comes alongside the reference ship

Approach Speed - speed at which the control ship will commence approach to come alongside the reference ship

AT - real time as referenced to the full size Mariner hull

Control Ship - ship making the RAS approach

Desired Distance - lateral distance at which RAS desired

DSL - Digital Simulation Language (IBM develcped)

DX - center of ship's geographic separation along X axis

DY - center of ship's geographic separation along Y axis

Geographic Coordinates - earths coordinate system

JCL - Job Control Language for IBM 360/67 computer

Kt., kts. - knot, knots - 1 nautical mile/hour or 2000 yards/hour

L, Ship length - length of one mariner hull used in this thesis (527.8 feet)

Lateral Distance - equal to ADY

Longitudinal Distance/Position - equal to ADX

LUC - nondimensionalizing scaling factor

Port Side To - approach (control) ship replenishes with its port side toward supply (reference) ship

RAS - Replenishment At Sea

Receiving Ship - control ship or ship B

Reference Ship - ship that maintains course and speed

Reference Speed - speed of reference ship

TABLE OF TERMS AND ABBREVIATIONS (cont.)

Replenishment Speed - signaled intended speed at which RAS will be conducted

Stbd Side To - approach (control) ship replenishes with its stbd side toward supply (reference) ship

Supply Ship - reference ship or ship B

T - nondimensionalized time used in DSL runs

Turn Phase - phase in RAS scenario at which the ships are in their desired positions and the reference ship is turned

Yaw - ship's heading in relation to true north

Y Coordinate - geographic reference system E is +, W is -

X Coordinate - geographic reference system N is +, S is -

ACKNOWLEDGEMENTS

In the course of thesis research many people contribute to the final product. It is not possible to afford individual credit to all that have given of themselves for this particular research. Some, however, have rendered assistance that has proven invaluable.

Professor G. J. Thaler has been, without a doubt, the prime guiding force that has kept this study within a sound perspective.

Three members of the computer center staff, M. Anderson, Kris Butler, and Ed Donnellan have tolerated numerous intimidations with good humor which made many otherwise arduous hours bearable. Without their professional assistance and personal contributions to the computer center operations, this thesis would have fallen far short of its goal.

And finally, my dear wife Mary who has endured a husband that has spent endless hours married to the IBM 360 computer. Her undying support and valuable encouragement is held in the highest esteem.

I. INTRODUCTION

The advent of the digital computer as standard equipment on board virtually all modern Naval ships has opened the field of Digital Computer Control in almost all aspects of ship life. The computer has been a viable asset in fire control systems for years and has been used extensively for aids to ship maneuvering control in the form of NTDS (Naval Tactical Data System) readouts. The declining costs of general and special purpose computers has made their inception as a manpower replacement or augmentation a reality. Their high speed and accuracy can make them perform functions with much greater safety than previously attainable with time proven (and sometimes time weary) "seaman's eye."

This then is the basis for this thesis; a study of the types of maneuvers that can be handled more accurately and safely than presently being accomplished.

One area of study is the total Replenishment At Sea (RAS) problem including both course and speed control for the approach and alongside phases. This situation has always been one of extreme danger due to the collision potentials involved. However, other dangers are involved in the on deck evolutions when the ships are not kept at a fairly constant distance. Sudden violent maneuvers may cause the stress on the lines between ships to increase enough to cause the lines to part. The reality of this danger is readily apparent to anyone who has ever seen a manila line or steel cable part or a kingpost shackle break or a kingpost suddenly bend under these extreme stresses. A system which will minimize these potential dangers is well worth investigation.

Of course with a digital computer, the algorithm for RAS can be modified or replaced to enable its use as a maneuvering control device for other situations such as formation steaming or single ship navigation transit control.

II. MODELING

A. MARINER DYNAMICS

In the conception of this thesis, realistic models of modern destroyer hull configurations were sought. This search proved fruitless. The hydrodynamic coefficients for present day destroyers are not currently available. However, some naval and civilian research is presently being conducted to obtain these coefficients.

A complete set of these coefficients is necessary for any maneuvering control system design. A hull configuration which has been under continual study with well defined and verified hydrodynamic coefficients was chosen[1]. This hull is commonly referred to as the "Mariner" hull.

The development of the equations of motion in six and three degrees of freedom have been well documented[2]. The model used for this thesis is the equations of motion in three degrees of freedom linearized with second order and higher terms eliminated. These equations are characterized by dependency on small perturbations about a specific operating point. The maneuvers experienced in the following chapters do not entirely meet this criterion. The inadequacy and shortcomings of this model are of little consequence because relevant hydrodynamic coefficients are not available, and the methods presented can be applied to any ship type.

The development of the model is readily available to the interested reader [3]; only a summary of the equations and their corresponding hydrodynamic coefficients are presented here.

The equations of motion used are as follows:

$$(X_{\dot{u}}-m)\dot{U}+X_u(U-u_0)+X_d d=0$$

$$(Y_{\dot{v}}-m)\dot{V}+Y_v V+(Y_r-m)R+Y_{\dot{r}}\dot{R}+Y_d d=0$$

$$(N_{\dot{r}}-I_z)\dot{R}+N_r R+N_{\dot{v}}\dot{V}+N_v V+N_d d=0$$

The direction and sense of the terms in the above equations are shown in figure II-1. Letting:

$a_{11}=m-Y_{\dot{v}}$

$b_{11}=-Y_v$

$c_{11}=0$

$a_{21}=-Y_{\dot{r}}$

$b_{21}=m-Y_r$

$c_{21}=0$

$a_{12}=-N_{\dot{v}}$

$b_{12}=-N_v$

$c_{12}=0$

$a_{22}=I_z-N_{\dot{r}}$

$b_{22}=-N_r$

$c_{22}=0$

$a_{33}=m-X_{\dot{u}}$

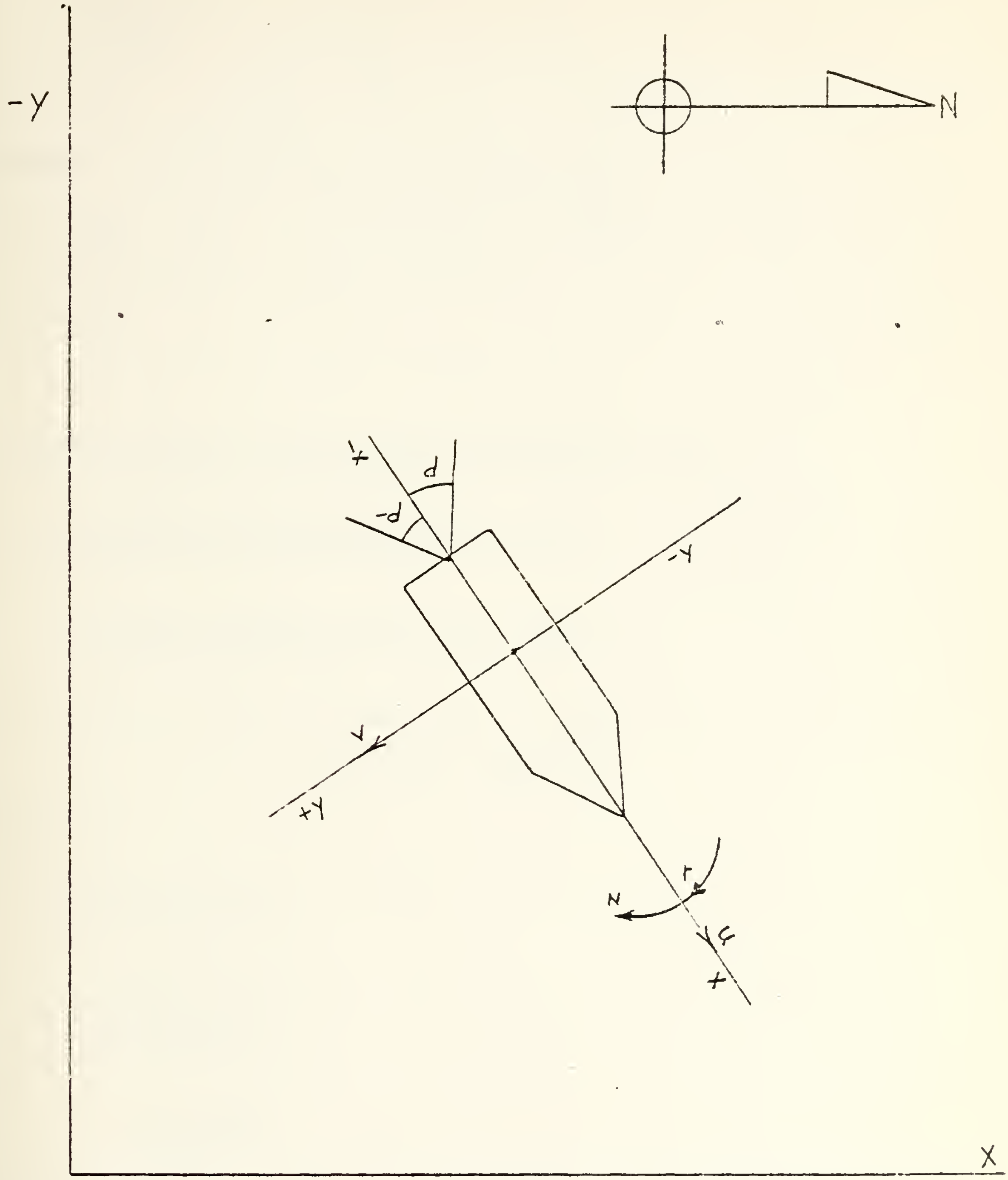


Figure II-1

Direction and Sense of Hydrodynamic Terms

$b_{33}=-X_u$

$c_{33}=0$

Setting

$V=\dot{A}$

$\dot{V}=B$

$U=\dot{C}$

The equations of motion can be written as:

$$a_{11}\ddot{A}+b_{11}\dot{A}+c_{11}A+a_{21}\ddot{B}+b_{21}\dot{B}+c_{21}B=IF1$$

$$a_{12}\ddot{A}+b_{12}\dot{A}+c_{12}A+a_{22}\ddot{B}+b_{22}\dot{B}+c_{22}B=IF2$$

$$a_{33}\ddot{C}+b_{33}\dot{C}+c_{33}C=IF3$$

or:

$$a_{11}\ddot{A}+a_{21}\ddot{B}=I1$$

$$a_{12}\ddot{A}+a_{22}\ddot{B}=I2$$

$$a_{33}\ddot{C}=I3$$

where:

$IF1=-Y_d \cdot d$

$IF2= N_d \cdot d$

$$IF3=-\int X_u \, dt$$

and:

$$I1=-b_{11}\dot{A}-c_{11}A-b_{21}\dot{B}-c_{21}B+IF1$$

$$I2=-b_{12}\dot{A}-c_{12}A-b_{22}\dot{B}-c_{22}B+IF2$$

$$I3=-b_{33}\dot{C}-c_{33}C+IF3$$

By solving this system of equations, the following relationships are established:

$$\ddot{A}=(a_{22}a_{33}I1-a_{33}a_{21}I2)/DEL$$

$$\ddot{B}=(a_{11}a_{33}I2-a_{33}a_{12}I1)/DEL$$

$$\ddot{C}=I3/a_{33}$$

where:

$$DEL=a_{33}(a_{11}a_{22}-a_{12}a_{21})$$

which yields the solution:

$$V=\dot{A}=V_0+\int_{t_0}^{t} \ddot{A} \, dt$$

$$\psi=B=\psi_0+\int_{t_0}^{t} \dot{B} \, dt=\psi_0+\int_{t_0}^{t} [\dot{B}(0)+\int_{t_0}^{t} \ddot{B}dt]dt$$

$$U=\dot{C}=u_0+\int_{t_0}^{t} \ddot{C} \, dt$$

The space coordinate system is defined as follows:

$$\dot{Y}=U\cdot SIN\psi+V\cdot COS\psi$$

$$\dot{X}=U\cdot COS\psi-V\cdot SIN\psi$$

where:

$$X=X_0+\int_{t_0}^{t}\dot{X}\,dt$$

$$Y=Y_0+\int_{t_0}^{t}\dot{Y}\,dt$$

Table II-1 summarizes the symbols and nomenclature used in the foregoing abbreviated solution of motion in three degrees of freedom. The applicable characteristics of the Mariner hull are presented in table II-2 with the corresponding nondimensionalized hydrodynamic coefficients and the DSL computer program variable names delineated in table II-3.

Computer Program #1 is the basic DSL program that was developed from these equations of motion. This program uses two ships to illustrate the turning characteristics of the Mariner hull for various rudder commands. Figure II-2 shows the difference between a step model and a ramp model rudder in a geographic plot. Figure II-3 shows the corresponding difference in yaw.

TABLE II-1

SYMBOLS AND NOMENCLATURE

Symbol	Definition
$X_{\dot{u}}$	derivative of longitudinal force with respect to longitudinal acceleration $\dot{U}$
U_u	derivative of longitudinal force with respect to longitudinal velocity U
$Y_{\dot{v}}$	derivative of lateral force with respect to transverse acceleration $\dot{V}$
Y_v	derivative of lateral force with respect to transverse velocity V
$Y_{\dot{r}}$	derivative of lateral force with respect to angular acceleration $\dot{R}$
Y_r	derivative of lateral force with respect to angular velocity R
Y_d	derivative of lateral force with respect to rudder angle d
$N_{\dot{v}}$	derivative of yaw moment with respect to transverse acceleration $\dot{V}$
N_v	derivative of yaw moment with respect to transverse velocity V

TABLE II-1 (cont.)

SYMBOLS AND NOMENCLATURE

Symbol	Definition
$N_{\dot{r}}$	derivative of yaw moment with respect to angular acceleration $\dot{R}$
N_r	derivative of yaw moment with respect to angular velocity R
N_d	derivative of yaw moment with respect to rudder angle d
$\dot{R}$	yaw angle acceleration
R	yaw angle velocity
u_0	initial velocity of origin of body axes relative to fluid
$\dot{V}$	transverse acceleration of ship axes relative to fluid
V	transverse velocity of orign of ship axes relative to fluid
X	hydrodynamic longitudinal force
Y	hydrodynamic lateral force
$\dot{U}$	longitudinal acceleration of ship axes relative to fluid

TABLE II-1 (cont.)

SYMBOLS AND NOMENCLATURE

Symbol	Definition
U	longitudinal velocity of ship axes relative to fluid
Ψ	yaw angle
At	actual time
T	nondimensionalized time
x_g	longitudinal distance that the ship center of gravity is forward of the ships axes
u_1	longitudinal velocity of ship axes relative to fluid (operating point)

TABLE II-2

CHARACTERISTICS OF MARINER-TYPE STUDY SHIP

Length, ft.	527.8
Beam, ft	76.0
Draft, ft	29.75
Displacement, tons	16,800.
Block coefficient, C_b	0.6

TABLE II-3

NONDIMENSIONAL HYDRODYNAMIC COEFFICIENTS

Coeficient	Computer Program Variable Name	Nondimensional Value
$(X_{\dot{u}}-m)$	MXUD	-0.0085
X_u	XU	-0.0012
Y_v	YV	-0.01243
$(Y_{\dot{v}}-m)$	MYVD	-0.015
(Y_r-mu_1)	MYR	-0.0051
$(Y_{\dot{r}}-mx_g)$	YRD	-0.00027
Y_d	YDELR	+0.0027
N_v	NV	-0.00351
$N_{\dot{v}}$	NVD	-0.000197
$(N_r-mx_g u_1)$	NR	-0.00227
$(N_{\dot{r}}-I_z)$	IZNRD	-0.00068
N_d	NDELR	-0.00126
X_n	---	-0.0000462
Y_n	---	-0.0000052
N_n	---	+0.0000026
X_d	XDELR	0.0

NOTE: $x_g=0.0$

TABLE II-3 (cont.)

NONDIMENSIONAL HYDRODYNAMIC COEFFICIENTS

Values based on the following operating point:

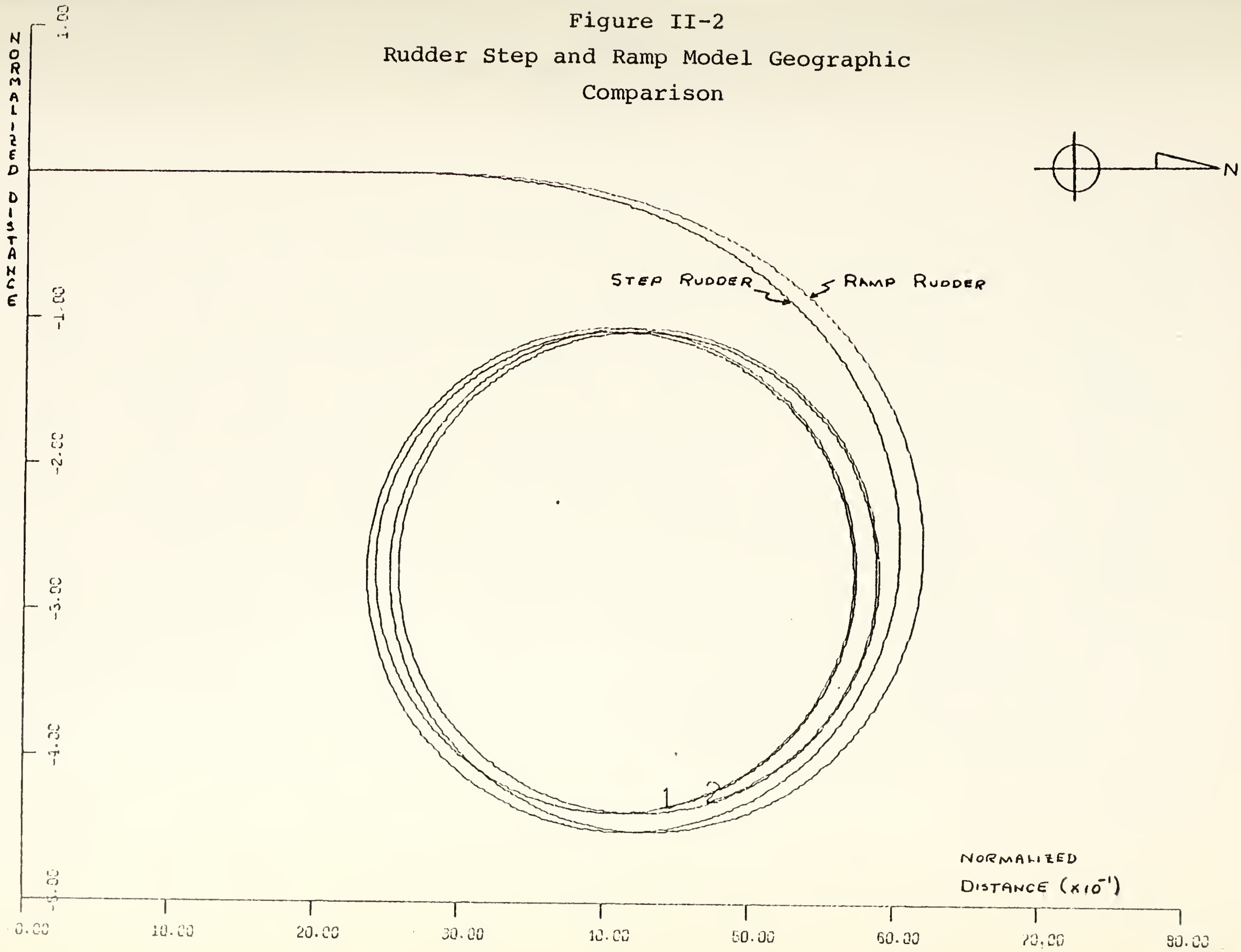
$u_1=1.0$ (15 Kts)

$\dot{\psi}=0.0$

$\dot{V}=0.0$

Figure II-2

Rudder Step and Ramp Model Geographic Comparison



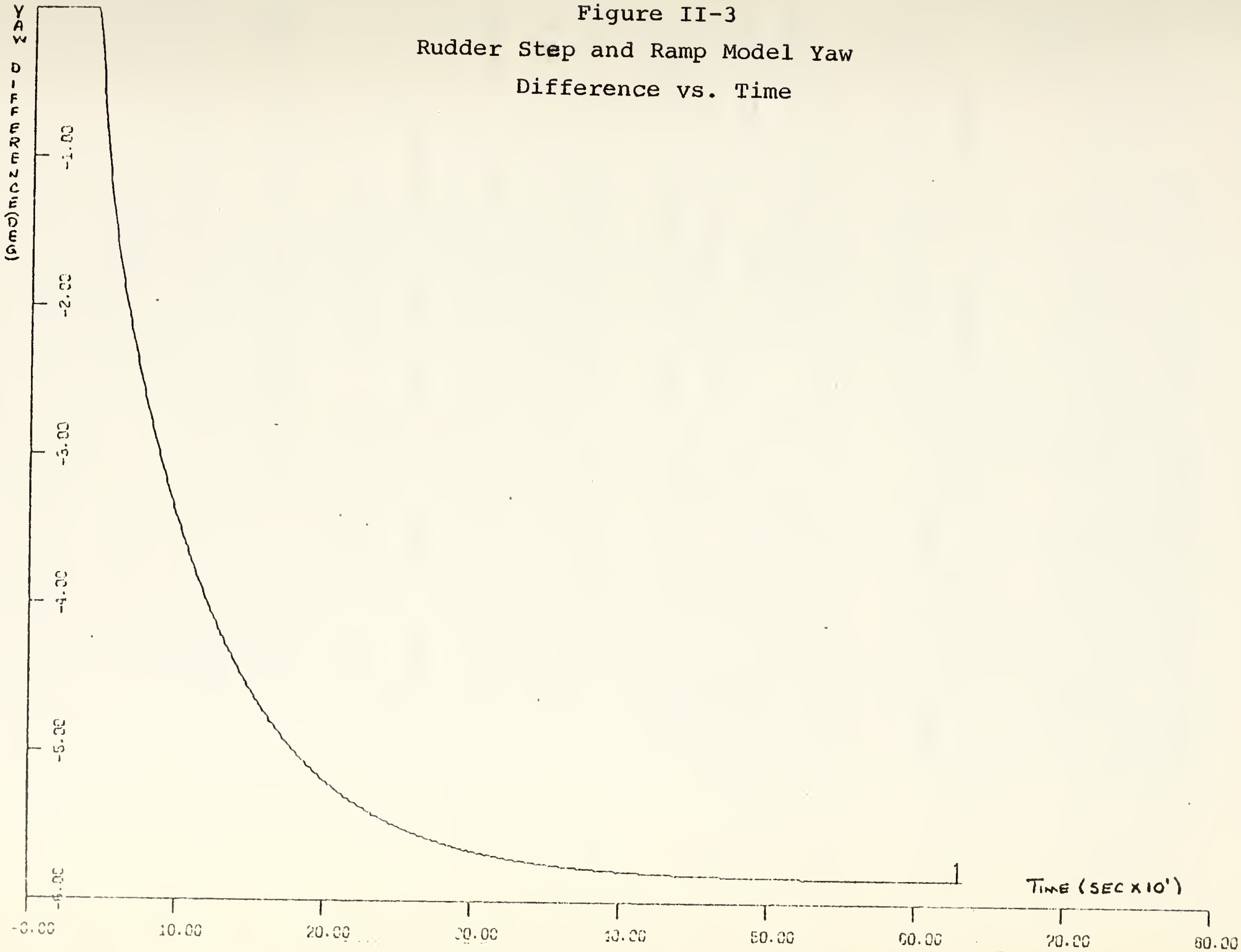


Figure II-3
Rudder Step and Ramp Model Yaw
Difference vs. Time

B. RUDDER RESPONSE

The previous section indicates a marked difference in behavior between step and ramp rudder models. This prompted an investigation into realistic rudder modeling which would fulfill the requirements of limit stops and maximum rudder rate.

NSRDC[*] has modeled the rudder of the DD-931 Class Destroyer. The basics of this model are presented in the block diagram of figure II-4.

The first limiter models the rudder stops which for the Mariner are ±30 degrees. The second limiter models the proportional band of a variable-displacement pump by limiting its maximum percent stroke. The limits for this nonlinear element have been found to be ±7 degrees.

The transfer function (K_g/s) accepts an input error signal of up to 7 degrees, converts it to a rudder rate, and integrates the rudder rate to obtain rudder angle. Letting:

$\dot{d}_m$ = Maximum rudder rate (2.0 degrees/sec)

d_{emax} = Maximum error input (7.0 degrees)

The system gain can be defined as:

$$
\begin{aligned}
K_g &= \dot{d}_m / d_{emax} \\
&= 2.0/7.0 \\
&= 0.285714 \text{ /sec}
\end{aligned}
$$

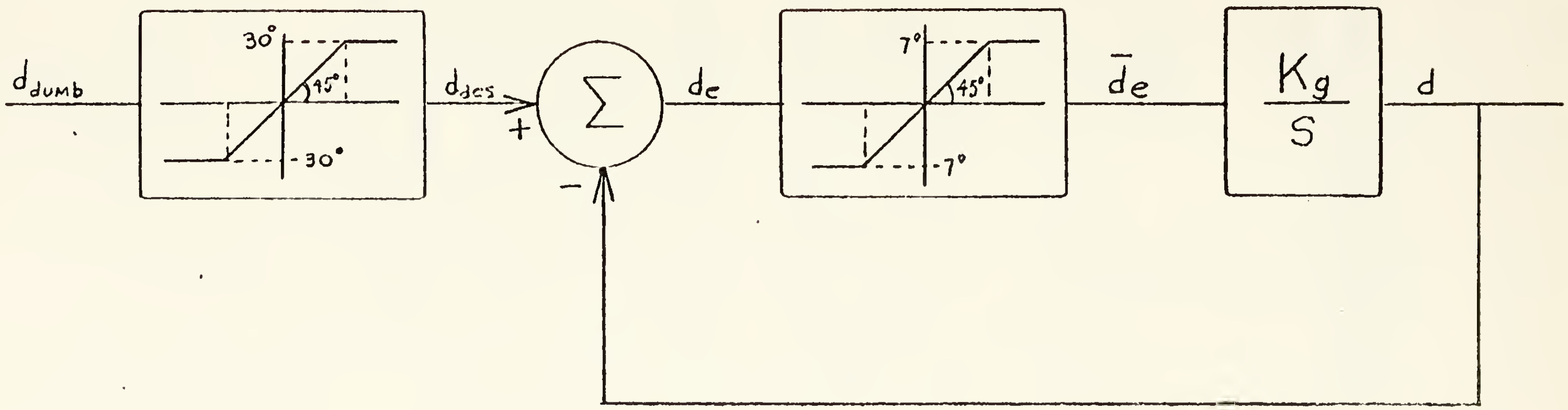


Figure II-4
Rudder Block Diagram

To convert this model to the required nondimensionalized form, the following manipulation is required:

$$K_g' = K_g \bullet L/u_1$$

$$= 0.285714 \bullet 527.8/(15 \bullet 1.689)$$

$$= 5.95224$$

where:

L = ship length

u_1 = operating point speed (15 Kts•1.689 ft/sec/Kt)

Computer Program #2 is the DSL program which models this system. The curves of figure II-5 and II-6 exhibit the responses of various step rudder commands. These are tabulated and cross referenced in table II-4.

These responses show the characteristics of a realistic rudder in that the rudder is never allowed to slam into the stops. They exhibit the time delay between command and response which is a function of the rate of response (2.0 degrees/sec). A control system design with this scheme is a much more difficult problem than one with an idealized rudder (step response) because the entire rudder control system becomes quite nonlinear.

TABLE II-4

RUDDER COMMAND AND RESPONSE

Figure	Curve	Rudder Command(deg)	Initial Condition(deg)
II-5	1	+30.0	-30.0
II-5	2	+25.0	-25.0
II-5	3	+20.0	-20.0
II-5	4	+15.0	-15.0
II-5	5	+10.0	-10.0
II-5	6	+ 5.0	- 5.0
II-6	1	+30.0	0.0
II-6	2	+25.0	0.0
II-6	3	+20.0	0.0
II-6	4	+15.0	0.0
II-6	5	+10.0	0.0
II-6	6	+ 5.0	0.0

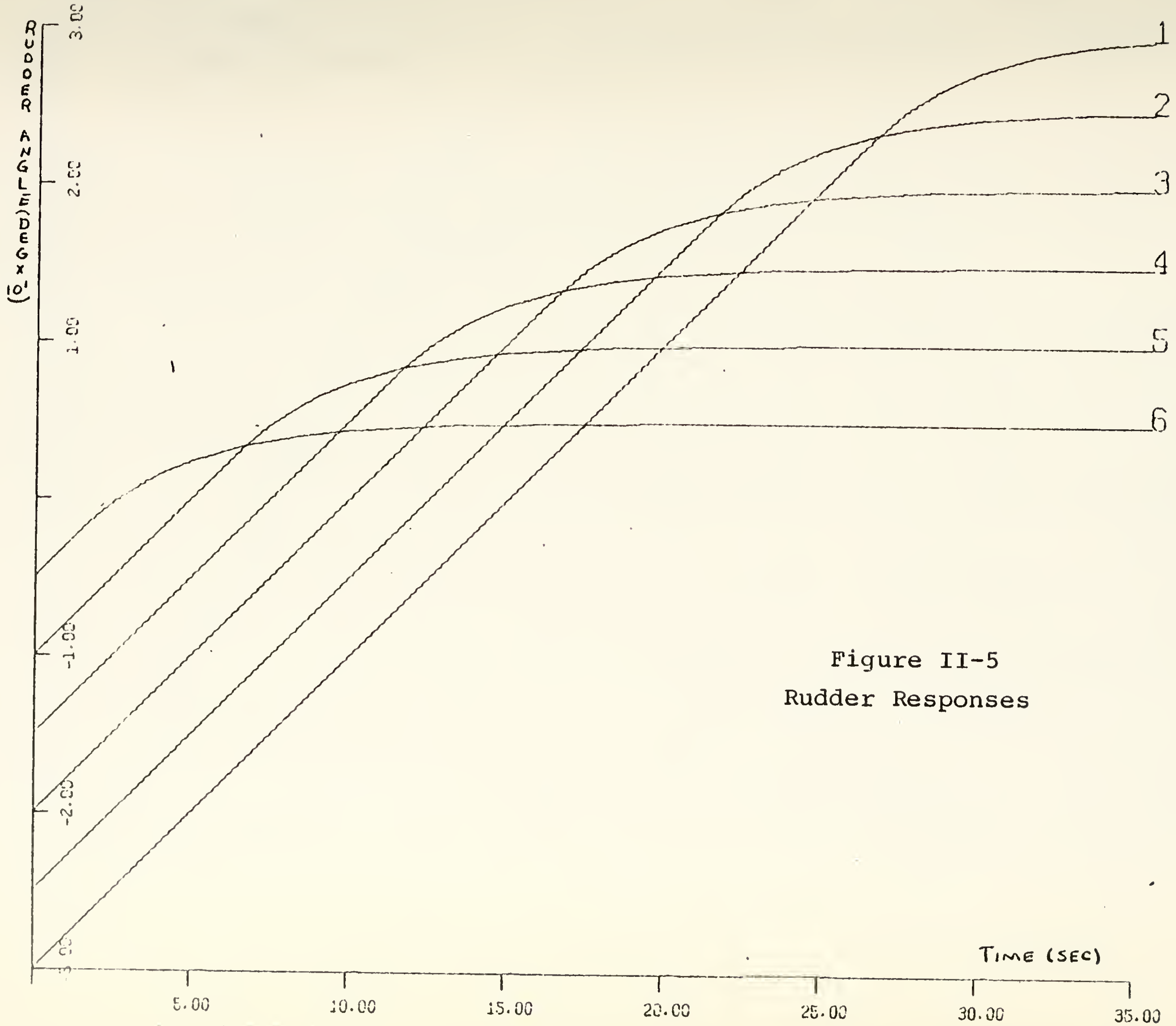
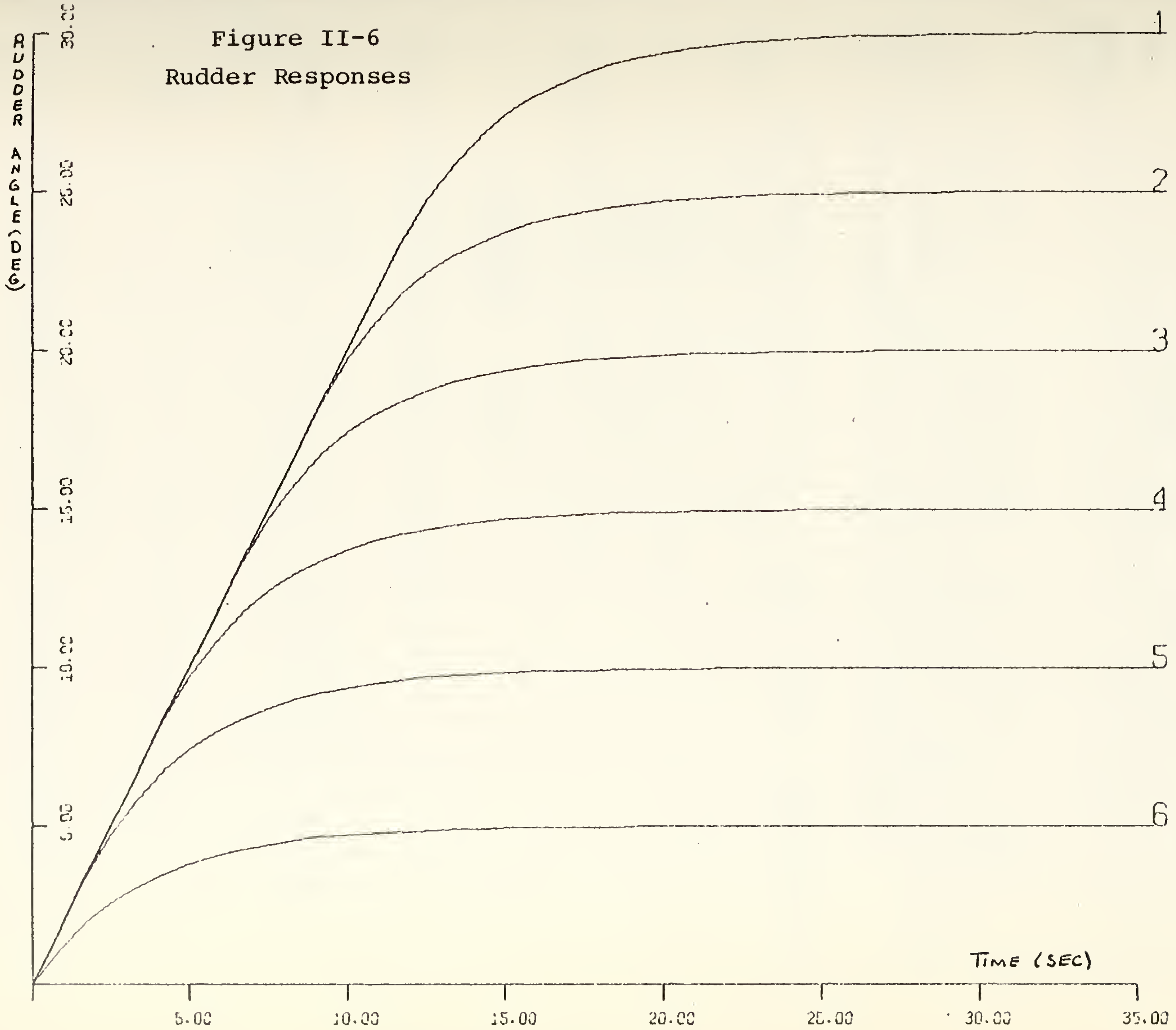


Figure II-5
Rudder Responses

Figure II-6

Rudder Responses



C. ENGINE RESPONSE

Figure II-7 portrays a complex model of a gas turbine propulsion plant[5][6]. This model contains the elements required for a complete dynamic study of the system. For the purpose of this thesis, such a complicated model is not required if the overall input-output relationship can be established.

Reference 5 establishes an output speed (U) relationship for a step input of desired speed (Ud) and is redrawn as figure II-8. The relationship appears to be that of a first or second order system with a time delay.

The system equations for a first order approximation with a time delay may be written as:

$$SPDIN = K*SPDDES*e^{-Ts}$$
$$SPDERR = (SPDIN-SPDOUT)*G$$
$$SPDOUT = \int SPDERR \, dt$$

Which yeilds the transfer function:

$$\frac{SPDOUT(s)}{SPDIN(s)} = \frac{G*K*e^{-Ts}}{s+G}$$

Which is block diagrammed in figure II-9.

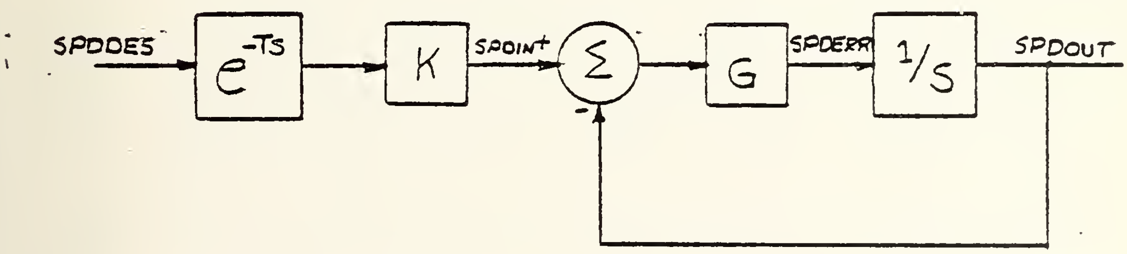
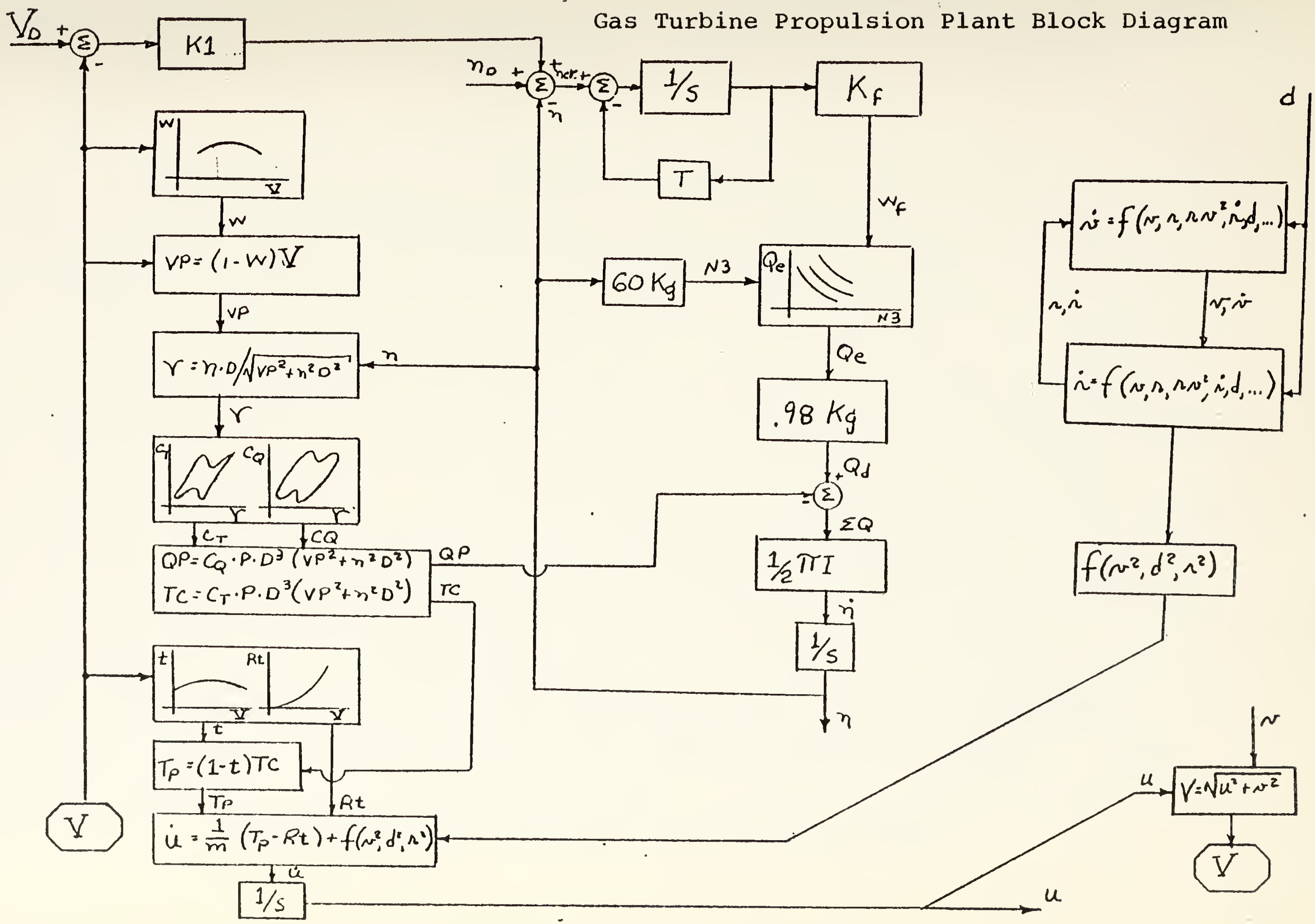


Figure II-9
Propulsion Plant Low Order Model Block Diagram

Figure II-7

Gas Turbine Propulsion Plant Block Diagram



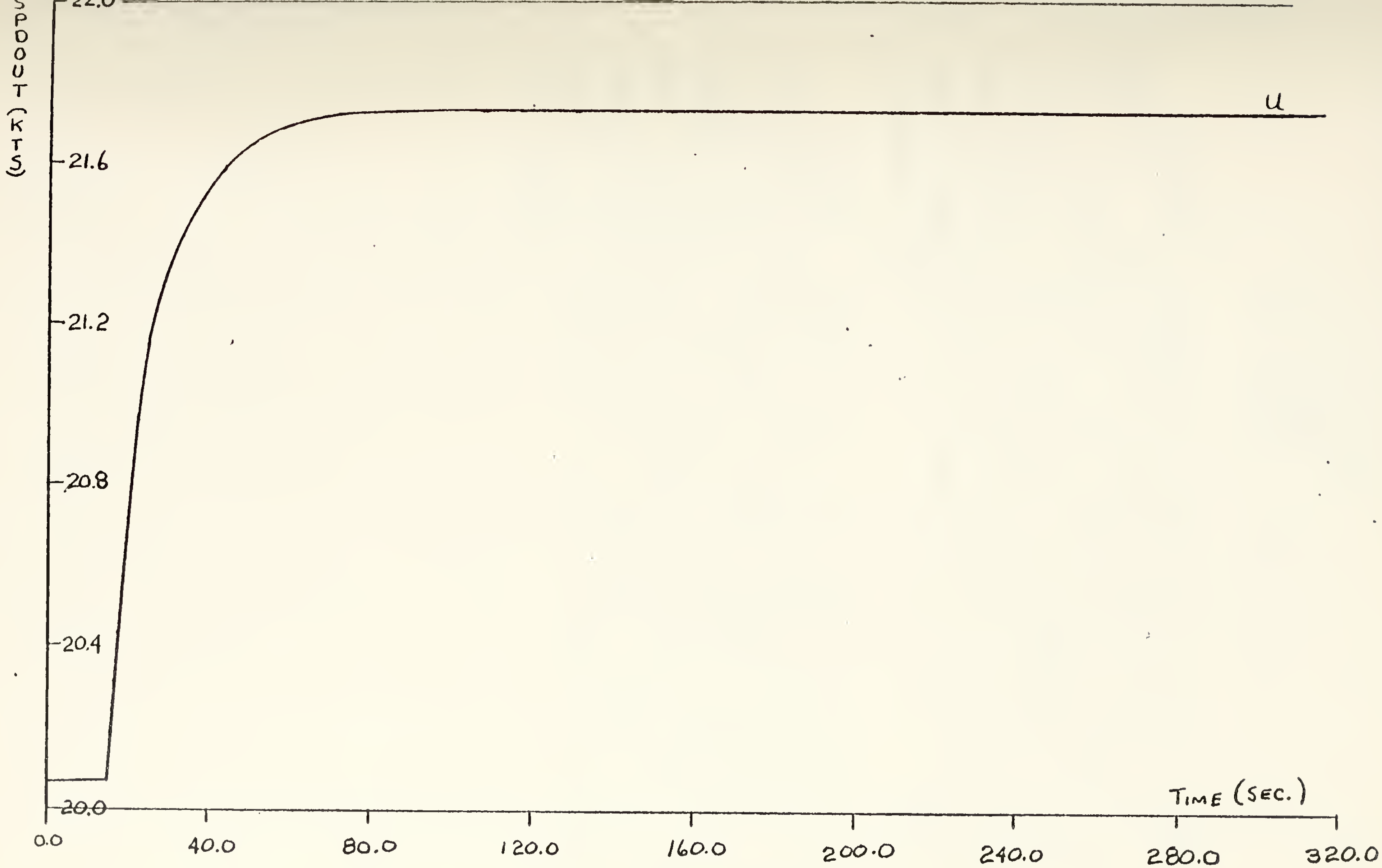


Figure II-8

Propulsion Plant High Order Model Step Response

From figure II-8, the time delay, system gain, and time constant can be estimated as:

$$T = 4.88 \text{ sec}$$

$$K = 0.9877$$

$$G = 0.092$$

Computer Program #3 was used to obtain the step input response. The original complex system output and the low order approximation are compared in figure II-10. As indicated in this figure, the two responses are very close. Considering the linearized approximations made in the equations of motion, this response is accurate enough for system study use and is used as the model for speed control in chapter III.

Similar methods may be used to obtain simplified low-order models for other high-order propulsion systems now in use (e.g. pressure fired boiler systems, 1200 lb. systems, etc.). They may not, however, simplify to a first order approximation suitable for system study. A method of computer determination of low-order models of high-order systems is contained in ref. 7 and may be mechanized for this purpose.

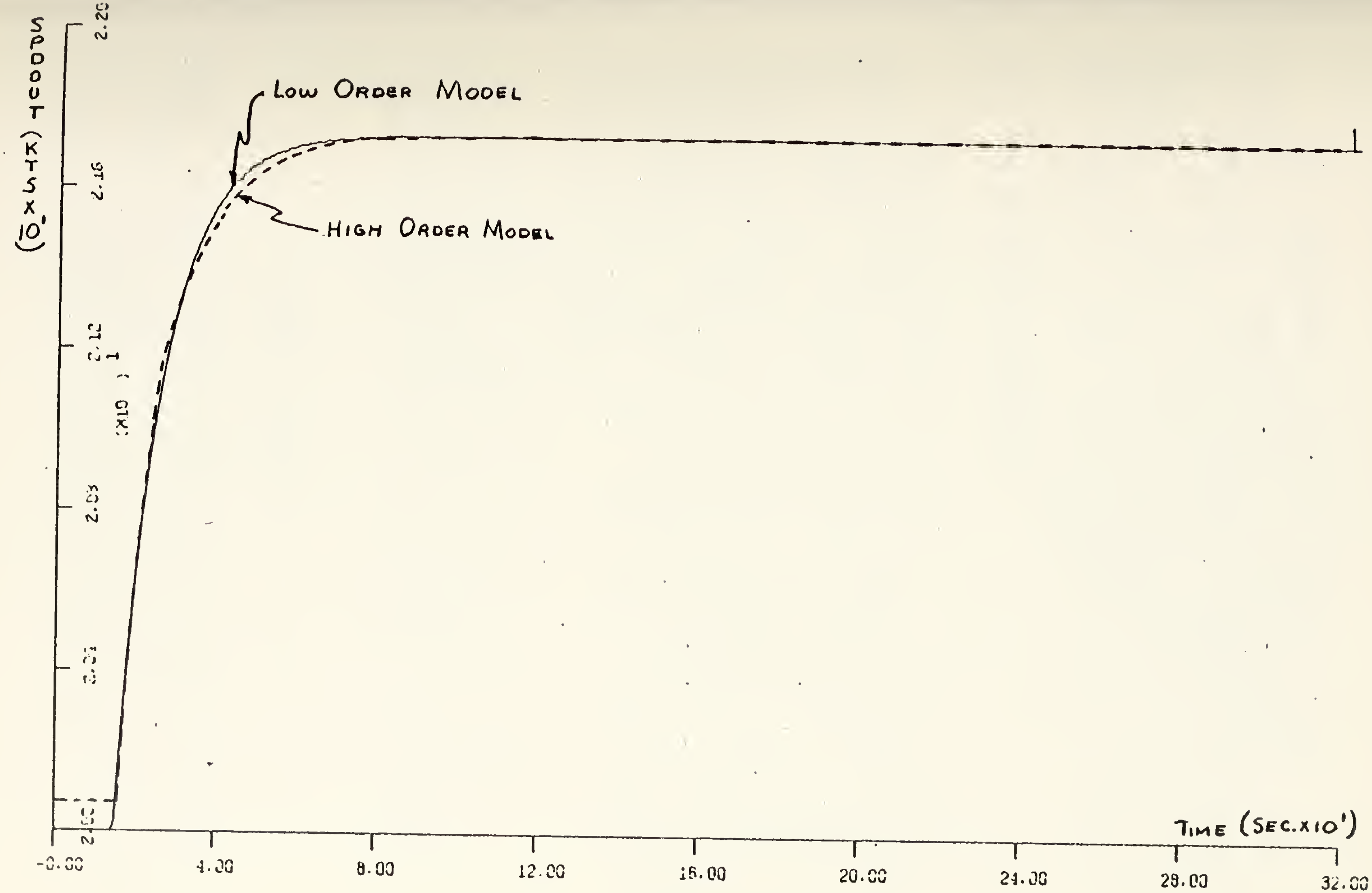


Figure II-10
Step Response Comparison of Low and High Order Propulsion
, Plant Models

D. EXTERNAL FORCES

The modeling of ship dynamics cannot be complete without the introduction of external forces which perturb its responses. These forces are caused by many factors and some are more relevant than others in the scope of this thesis. The two that are considered can cause substantial perturbations that must be modeled and eventually accounted for in the control system design.

1. Two Ships in Proximity

Whenever two ships operate in close proximity (less than 250 feet), suction and pressure forces between hulls are present. Studies have been conducted on the Mariner hull[1] which have produced data for construction of a family of curves for two ships passing on the same heading. No data has been found for the cases of two ships not on the same heading. Other restrictions on the work presented in ref. 1 are that the ships are of the same type and of similar hull ratios.

Interactive effects vary as the square of speed. However, this is only true if both ships are at the same speeds. The interaction modification factor is based on the normalized speed of 15 kts. This factor can thus be written as:

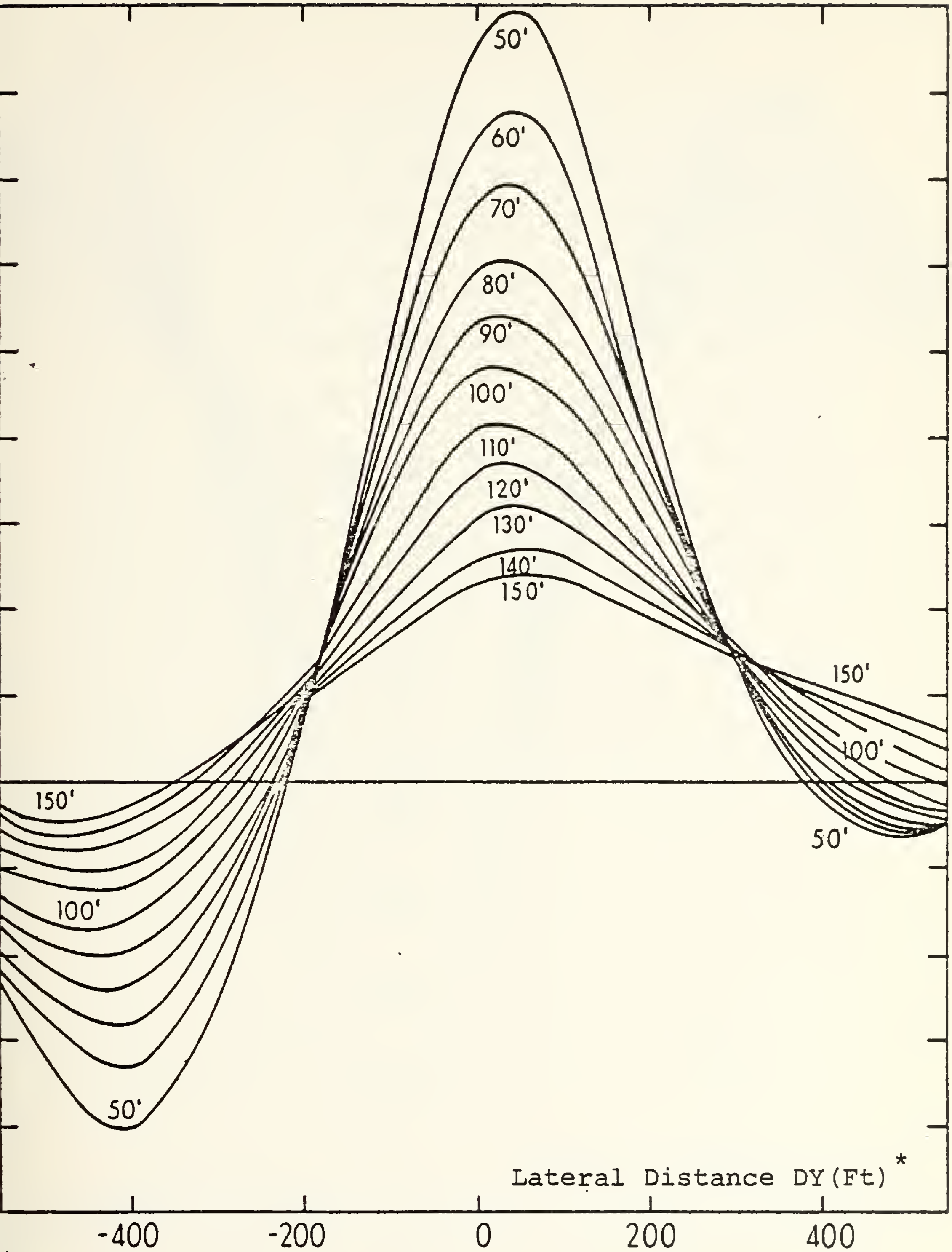
$$SPDF = CDOT1^2$$

Exact effects on the interaction forces and moments in the situation where the ship's speeds differ are not available. This is inconvenient since during the approach phase, the normalized speed of the approach ship (ship B) can be as high as 1.5. If the effect on ship B is as stated above but with its own speed causing the interaction modification, the interactive forces and moments can be 2.25 times greater than without speed considerations.

Without the ability to pin down this relationship, it was ignored in the development of the control laws presented in this thesis. Appendix C was written with the expressed intent of illustrating the effect of modifying the interactive forces and moments to the extremum case mentioned above. It must be realized that this case is not considered likely in that it is felt that the interactive forces and moments modification on ship B are more apt to be caused by the speed of ship A. If this is so, since ship A is kept at a constant 15 kts., the interactions need no modification for speed consideration in this thesis.

Reference 1 also gives a method of modifying the interactive forces and moments based on different ship lengths. For ease of computation and graphical presentation, the two ships were considered of equal lengths. To modify this to ships of dissimilar lengths, the resulting hydrodynamic derivatives must be modified as shown in ref. 1 (also shown in appendix C).

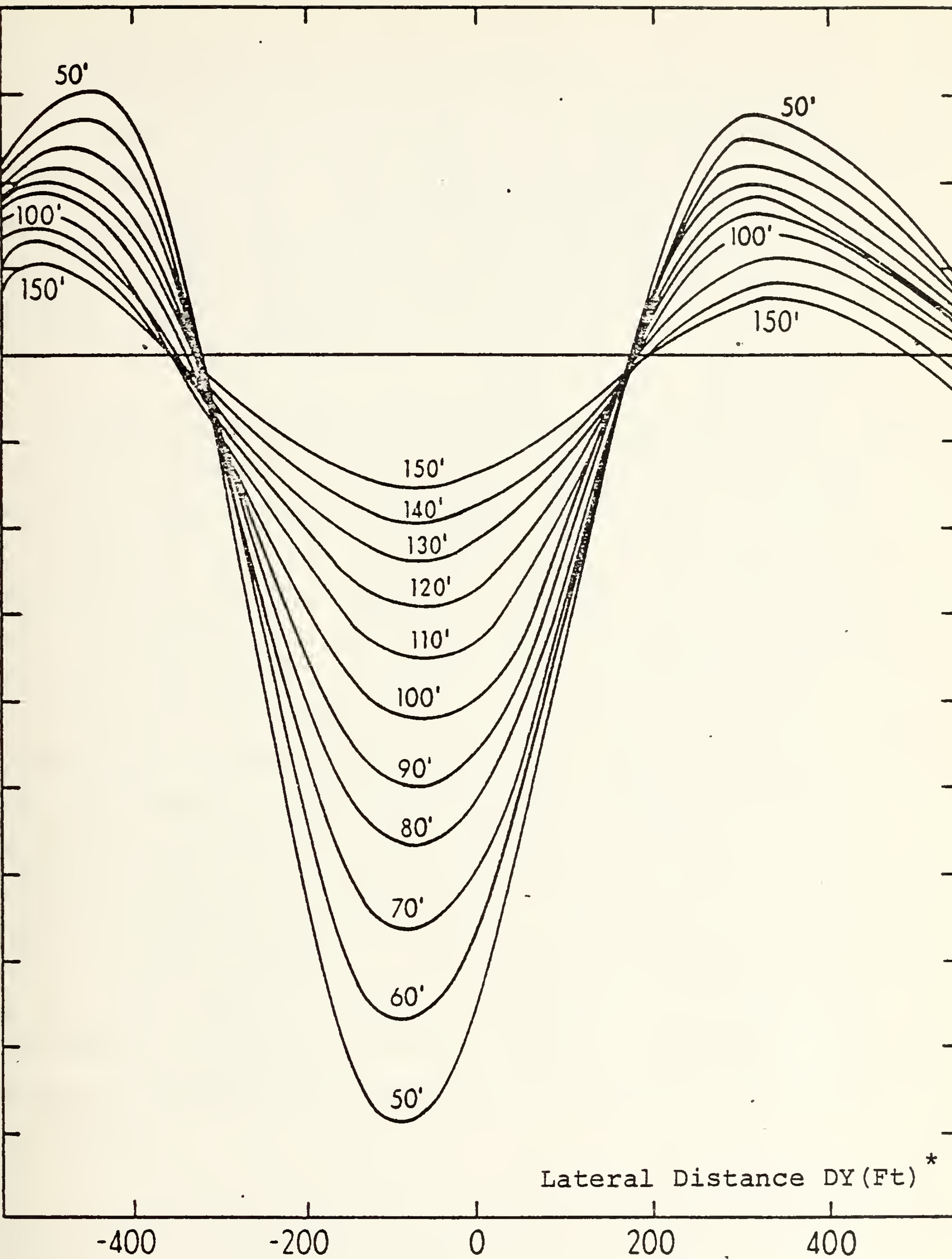
Since no closed form expression existed for these forces, the family of curves reproduced as figures II-11 and II-12 were quantized in subroutine SLOPES (an adaptation of the subroutine of the same name from ref. 11). (Appendix C contains a curve fitted subroutine that was compiled after the completion of the research on this thesis. It was not used for any design or simulation runs except for those presented in that same appendix.) An interpolation algorithm is used to approximate the intermediate values between quantized values and between the curves of the family.



*Note: To convert to normalized lateral distance DY - Divide by the ship length L.

Figure II-11

Family of Interactive Y force Curves



*Note: To convert to normalized lateral distance DY - Divide by the ship length L.

Figure II-12

Family of Interactive N Moment Curves

The main purpose of this subroutine is to compute the interactive forces between ships in the replenishment at sea situation and output the values for perturbation of the control ship only. Even though both ships are affected by these perturbations, a one ship control system which is effective regardless of the other ships motion is considered. Consequently, the interactive forces on the second ship are ignored.

Subroutine SLCPES is contained in appendix A. Figure II-13 is a geographic plot of the two ships passing at 105.6 feet with their rudders amidships (0 degrees). Figure II-14 and figure II-15 show the magnitude of the lateral force (Y force) and rotational moment (N moment) of the reference ship on the ship making its approach (control ship). The reference ship is at 15 kts. and the approach ship is at 22.5 kts. The control ship starts its approach 5 ship lengths (2639.0 feet) astern and 0.4 ship lengths (211.12 feet) laterally displaced. The most graphic portrait of the effects of these forces and moments appears in the yaw changes which are presented in figure II-16. From these figures it becomes readily apparent that these perturbations cause violent motions of the ship which must be accounted for in any control system development. Throughout the development of such a control system in chapter III, these forces and moments are considered inherent in the model for RAS control.

2. Waves

The modeling of sea state in the form of waves and wave interactions has occupied the time of many researchers [8][9]. The exact formulation of waves will not be accomplished in this thesis. Since the main concern here will be to test the control scheme developed in chapter III, a much simplified wave generator can be used. To introduce the required experimental perturbations on the designed control system a periodic wave system with a fundamental frequency and its second harmonic is used. Some small random wave properties are introduced that ride on these two sinusoids. A simple expression of this combined wave can be written as:

$$W = WF \bullet \sin(WE) + (PI \bullet WF^{2}/WL) \bullet \sin(2 \bullet WE) + WF \bullet WRV \bullet \sin(WE)$$

where:

- W denotes the Wave
- WF is the Wave Force
- WE is the Wave Encounter radian Frequency
- WL is the Wave Length
- WRV is the Wave Random Variable
- PI is 3.1415926

With this wave as a basis, a method of modeling this in the dynamic environment of the total ship simulation was defined. The modeling includes the introduction of this wave into the three degree of freedom equations of motion. To accomplish this a set of defining relationships were developed. First the general wave direction is input to establish the direction of the wave encounter on the ship. If the ship direction is YAWDP2 and the wave direction is WD, the expected wave direction is defined as:

$$EWD = WD - YAWDP2$$

Next the wave encounter frequency (radian frequency) can be established with knowledge of the ships normalized true speed (CDOT2), wave length (WL) and normalized wave velocity (WV). The wave encounter frequency (WEF) is then:

WEF = 2•PI•(CDOT2+WV•cos(EWD))/WL

The total wave encounter (WE) is nothing more than the wave encounter frequency (WEF) times time. This gives the wave encounter radian frequency required in the simple expression for the combined wave previously shown.

This does not complete the task, since the individual wave forces of each degree of freedom must be derived for this general wave expression, namely the components of WF. Again a much simplified version of the more complex real life wave forces were used. The X and Y forces are considered first. These can be modeled as cosine and sine functions of the expected wave direction (EWD) such that:

WFX = WF•cos(EWD)
WFY = WF•sin(EWD)

where WF is the total wave force of the encountered wave.

The rotational N forces are a little more difficult to establish. By considering that no rotational forces are created by a wave directly on the bow or stern, or directly off the beam, and that it is maximum when the wave is at 45 degrees off the bow or stern, a much simplified approximation is developed. Realizing that this method is very crude, the N force can be written as:

WFN = WF•sin(2•EWD)

To add more creditability to the wave model, a random variable is added to the wave force at the waves fundamental frequency. A gaussian (normal) distribution was chosen with a zero mean and a standard deviation of one-tenth the maximum allowable force of WF. A zero mean signifies that the expected amplitude of the random wave is 0.0, while the standard deviation signifies that 68% of the random waves will have amplitudes less than one-tenth of the maximum allowable force of WF. Also, 94% will have amplitudes less than one-half the maximum allowable force of WF. This small added perturbation allows for verification of the model simulation with a stochastic force, which in turn adds creditability to the developed control systems.

What remains is to define the total wave force (WF). It is important not to fall into a common simulation pitfall which inevitably causes unneeded design changes. A sea state does not increase at an infinite rate. It therefore is incorrect to start a simulation with initial conditions set for calm sea and immediately introduce a high sea state perturbation. The initial large perturbation transient can give results that are not only unrealistic, but can cause the model and control system to produ3e unstable results. This is especially true in this case since the linearized (small perturbations about an operating point) equations of motion are used.

With this in mind, a ramp feed in of the wave force with a limiter at the desired maximum wave force (WFMA) was used. The slope of the ramp was established to impart minimum initial transients, yet increase the wave force to an acceptable testing level within the time frames of the simulations of chapter III. The slope is designed such that the maximum wave force is reached in 94.815 seconds actual simulation time (4.548 seconds problem time).

Computer program #4 was used to verify the wave action model. Table II-5 on page 56 indicates the figures produced and changes in input wave length (WL) and wave direction (WD) for each run. The input parameters that were constant for all runs are tabulated below:

YAWDE2 = 0.0

CDCT2 = 1.5

WS = 5.0*

WFMA = 0.1137

*NOTE: WS is the unnormalized wave speed. Conversion to normalized wave velocity is:

WV = WS/15.0

Introduction of the wave forces is accomplished by multiplying the established wave forces by the rudder hydrodynamic coefficients for the individual reference directions. This effectively scales the wave forces to the ship model being used. The wave force result is coded in the ship simulation program as follows:

```
IF12 = KA1•D2+YY2+KA1•WY
IF22 = KB1•D2+YN2+KB1•WN
IF32 = KC1•D2+NC2+KC1•WX
```

Detailed results of the wave force effects are given in chapter III and will not be dealt with here.

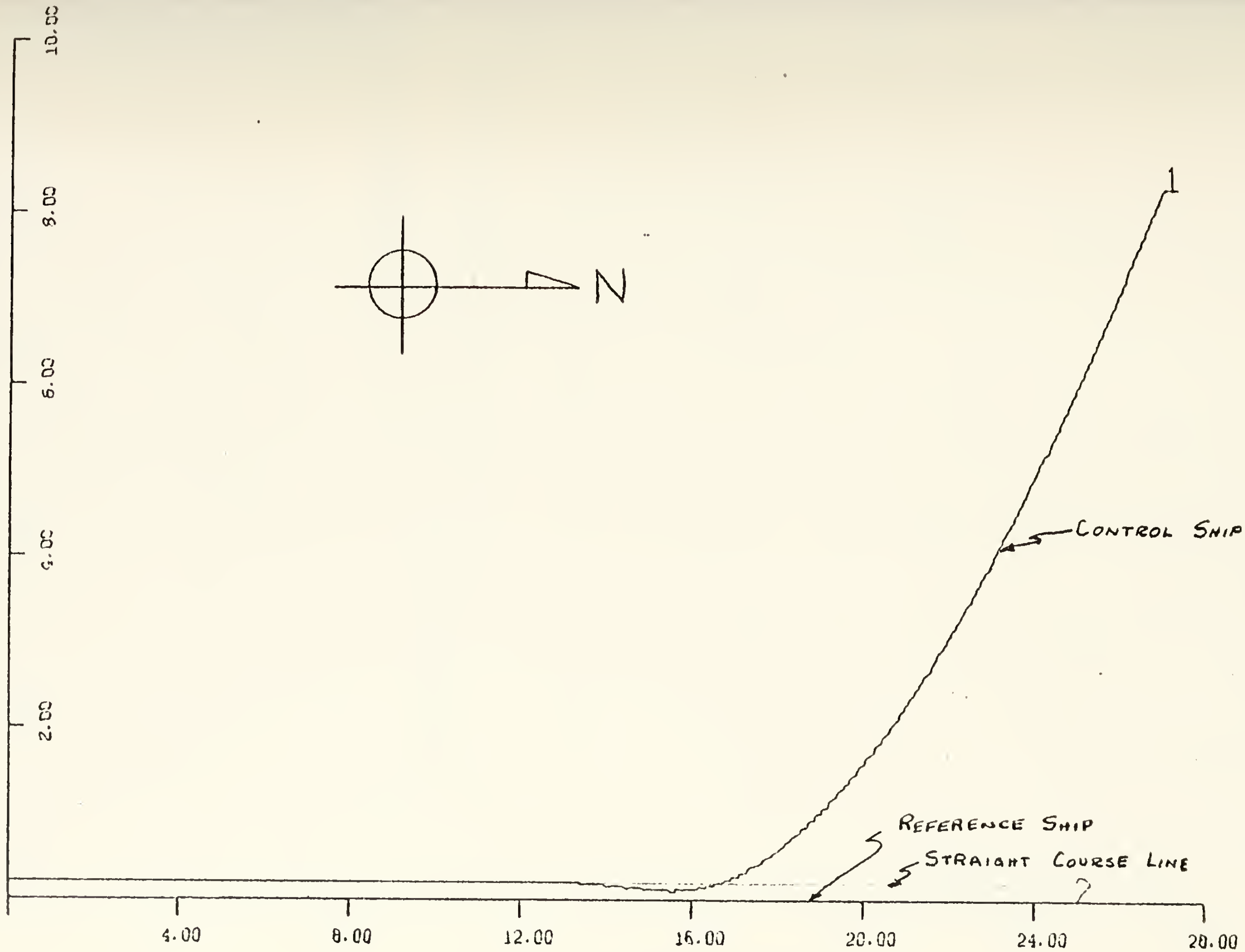


Figure II-13

Interactive Forces Effect on the Geographic Plot

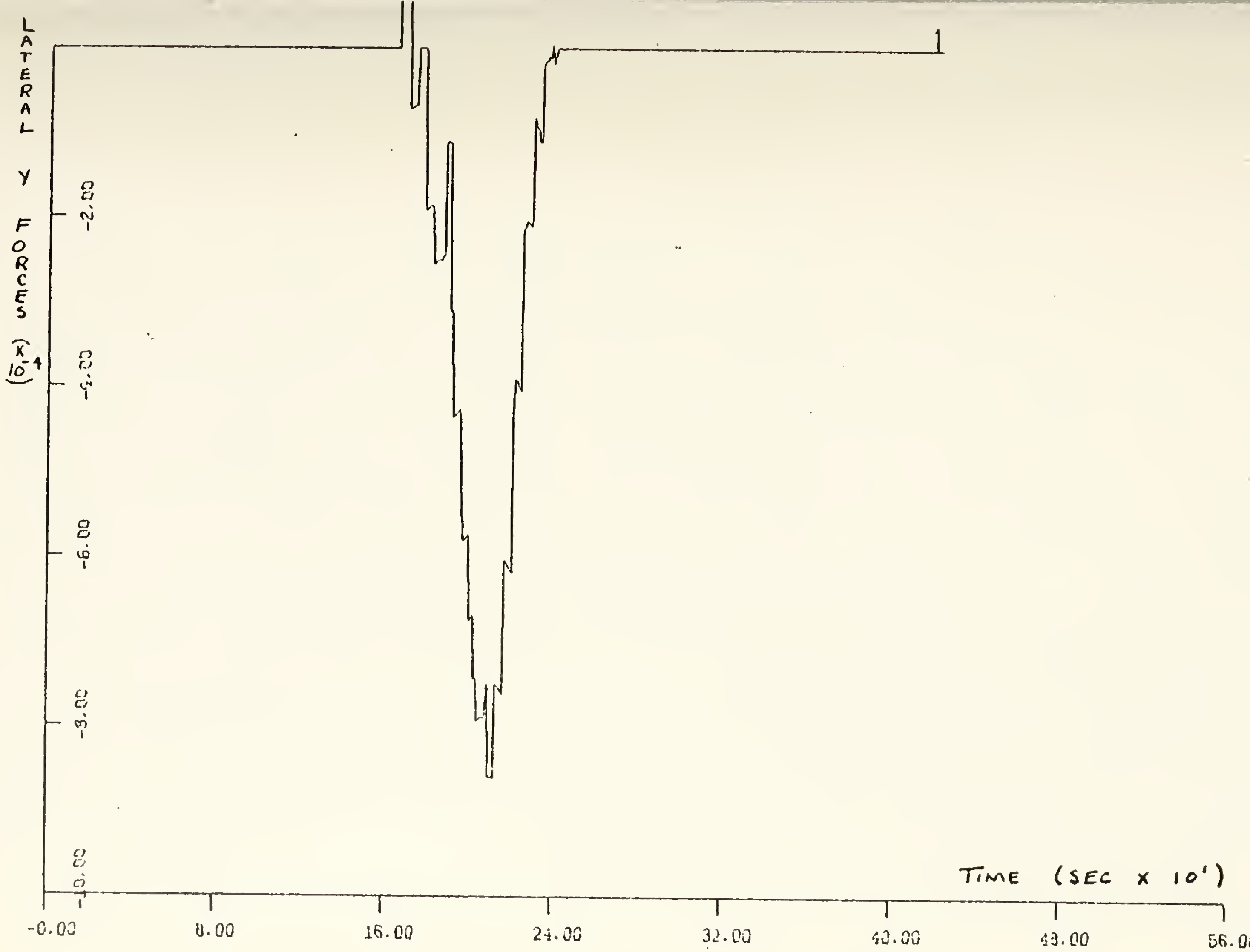


Figure II-14
Lateral (Y) Forces

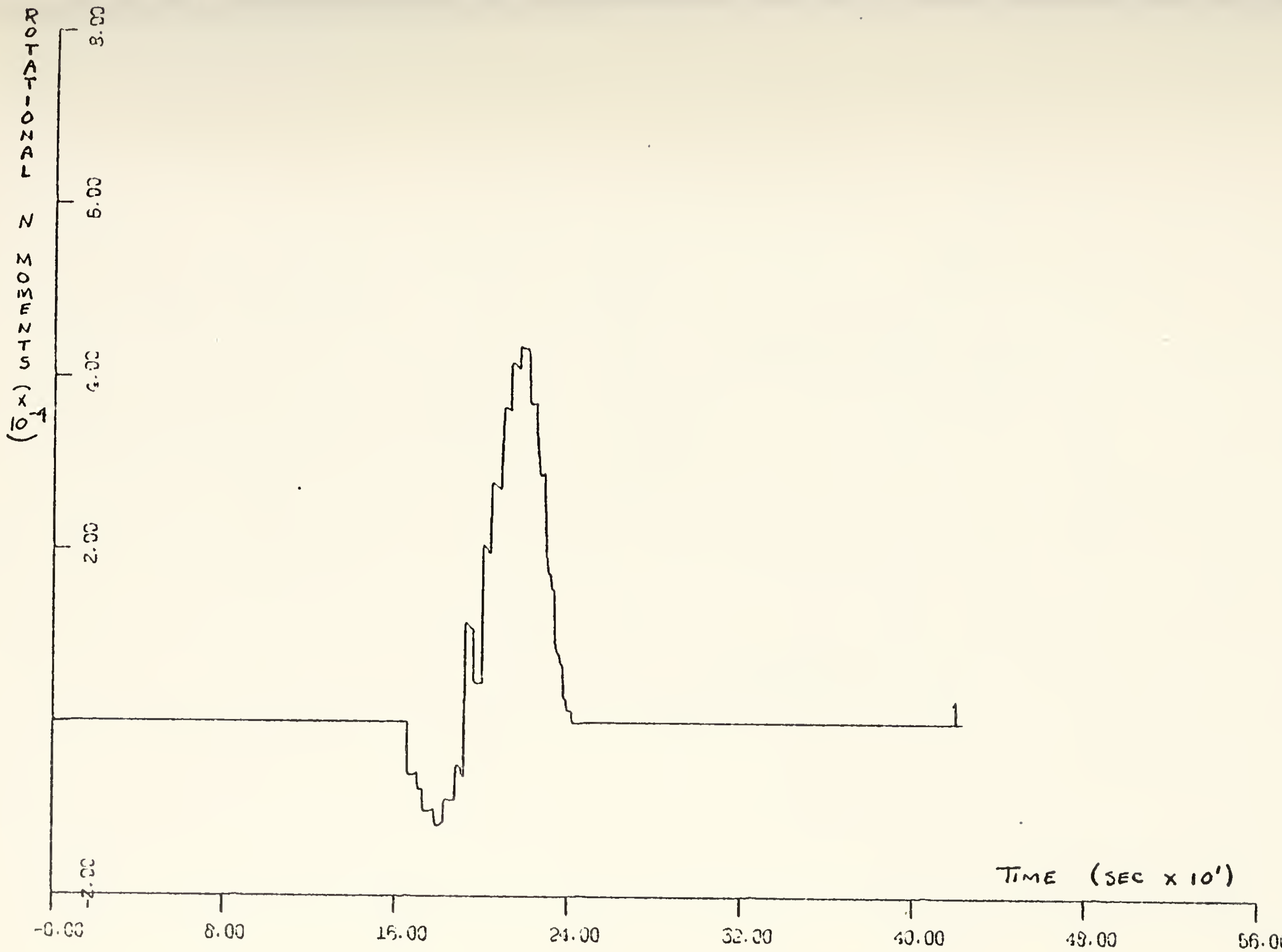


Figure II-15
Rotational (N) Moments

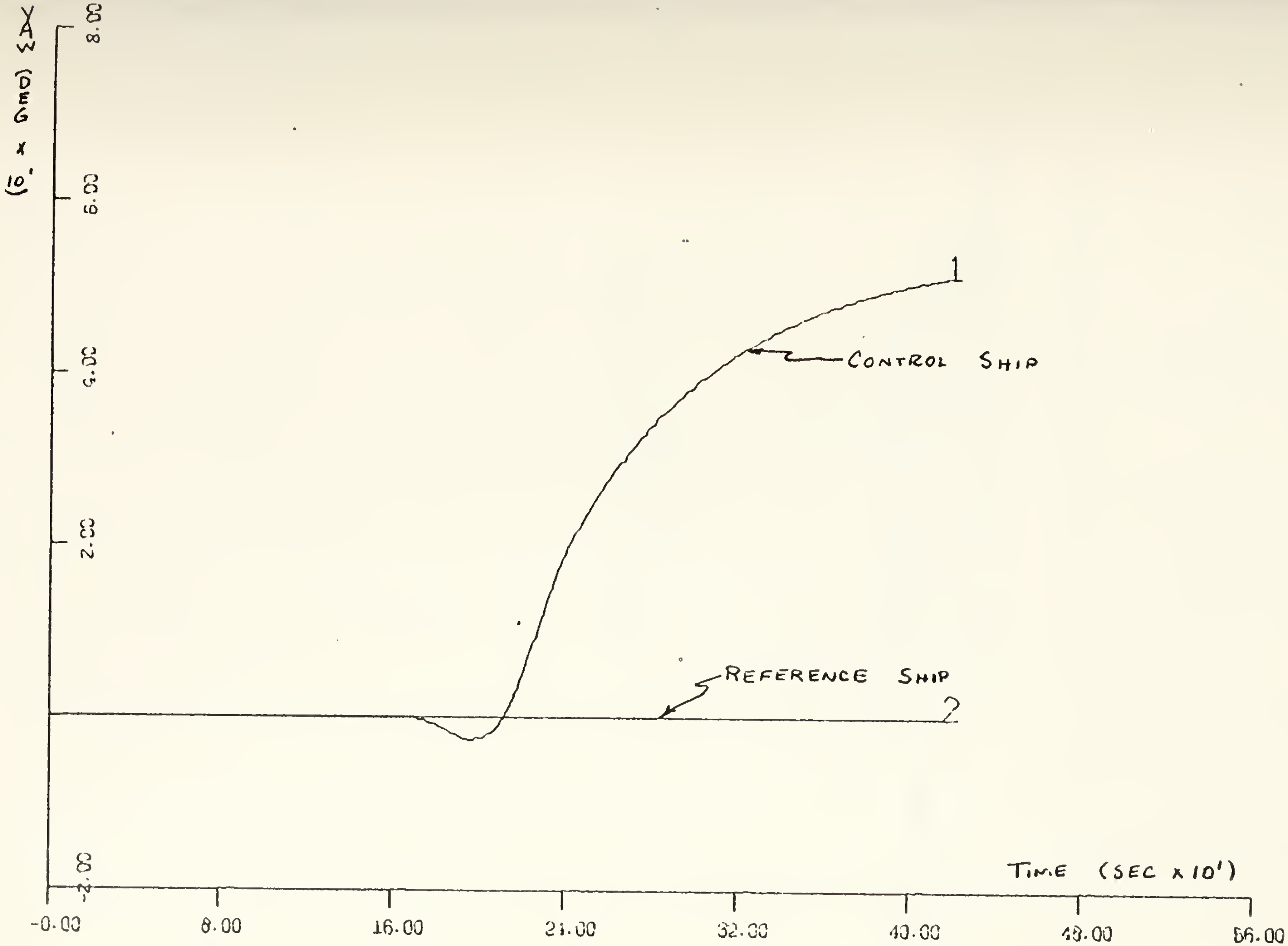


Figure II-16
Interactive Forces Effect on Yaw of the Control Ship

Run	Figure*	Input Parameters WL*	WD*
1	II-17	0.5	015
2	II-18	1.0	015
3	II-19	1.5	015
4	II-20	0.5	030
5	II-21	1.0	030
6	II-22	1.5	030

*NOTE: WL is given in ship lengths
WD is given in degrees
Curve numbers of all runs corresponding to wave force components are:

Force	Curve
WX	1
WY	2
WN	3

Table II-5

Wave Simulation Listing

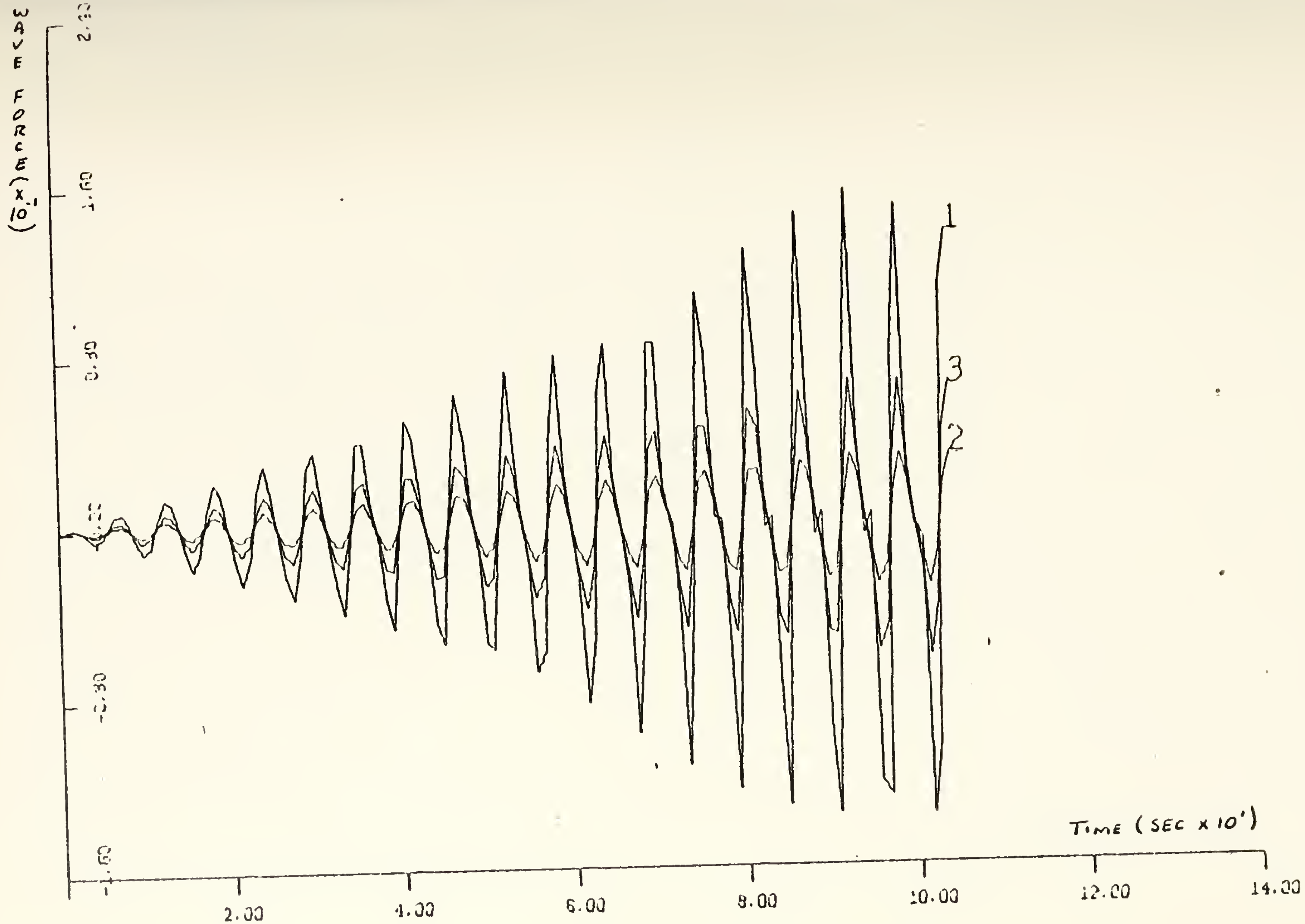


Figure II-17
Wave Simulation Run #1

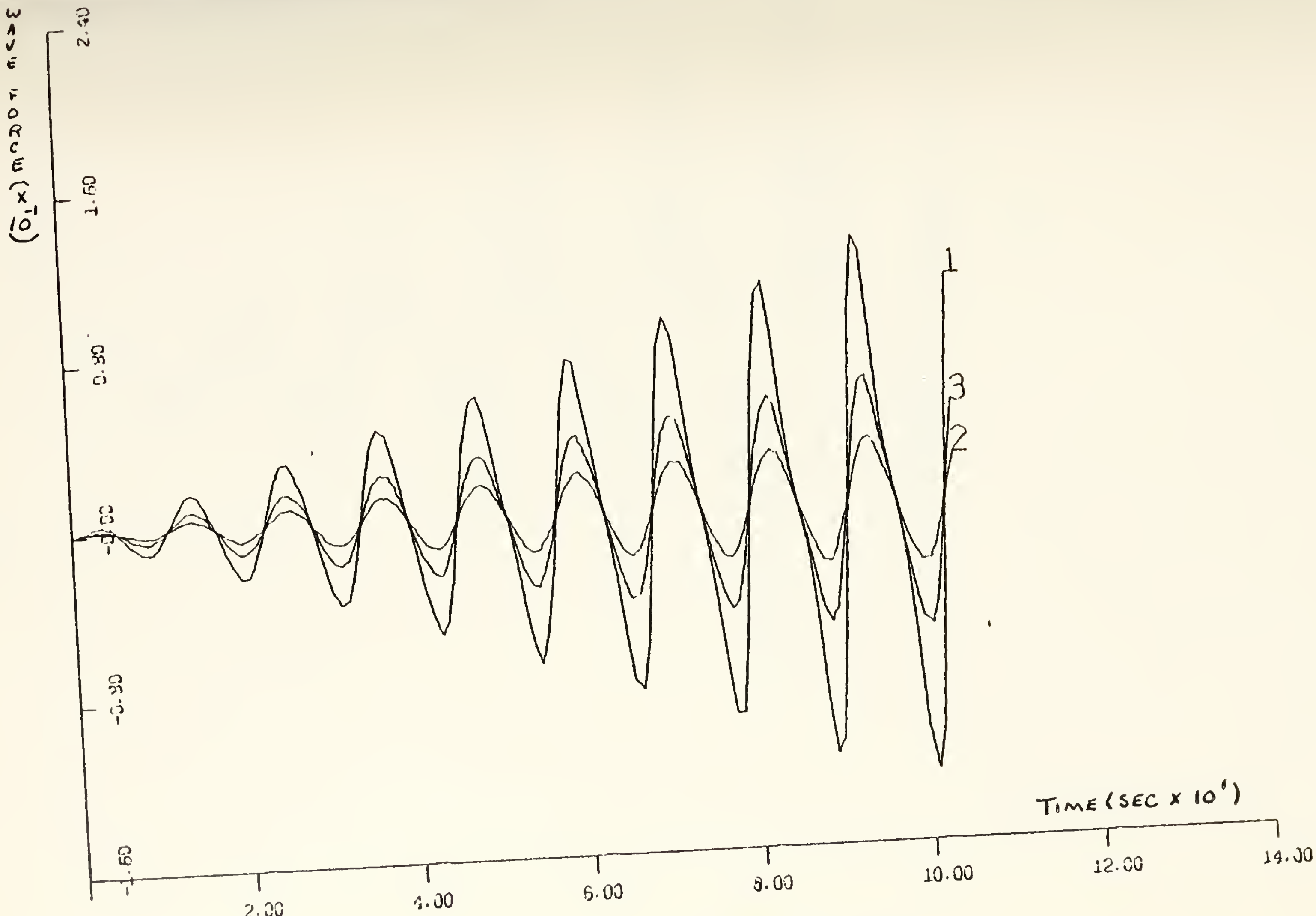


Figure II-18
Wave Simulation Run #2

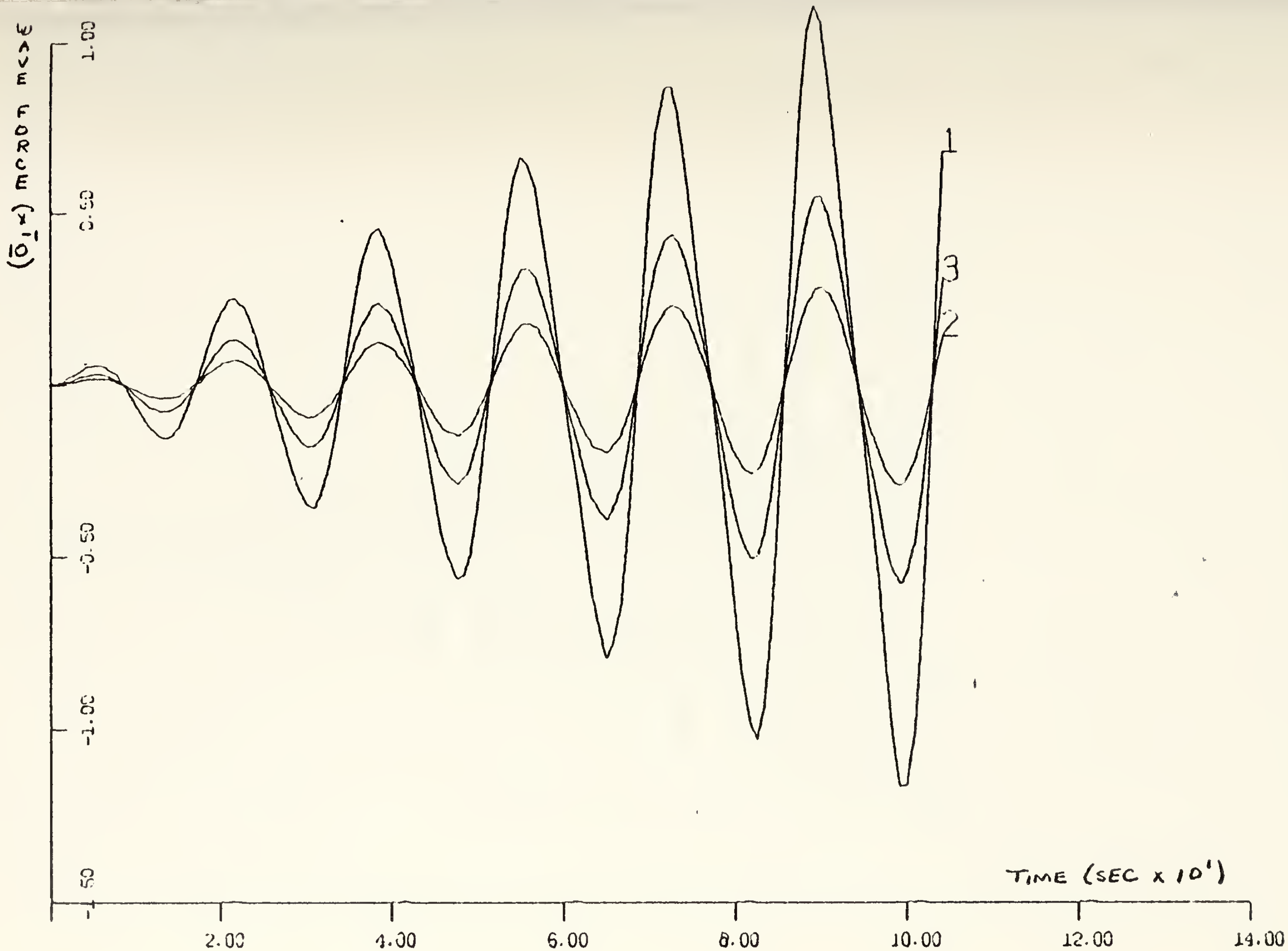


Figure II-19
Wave Simulation Run #3

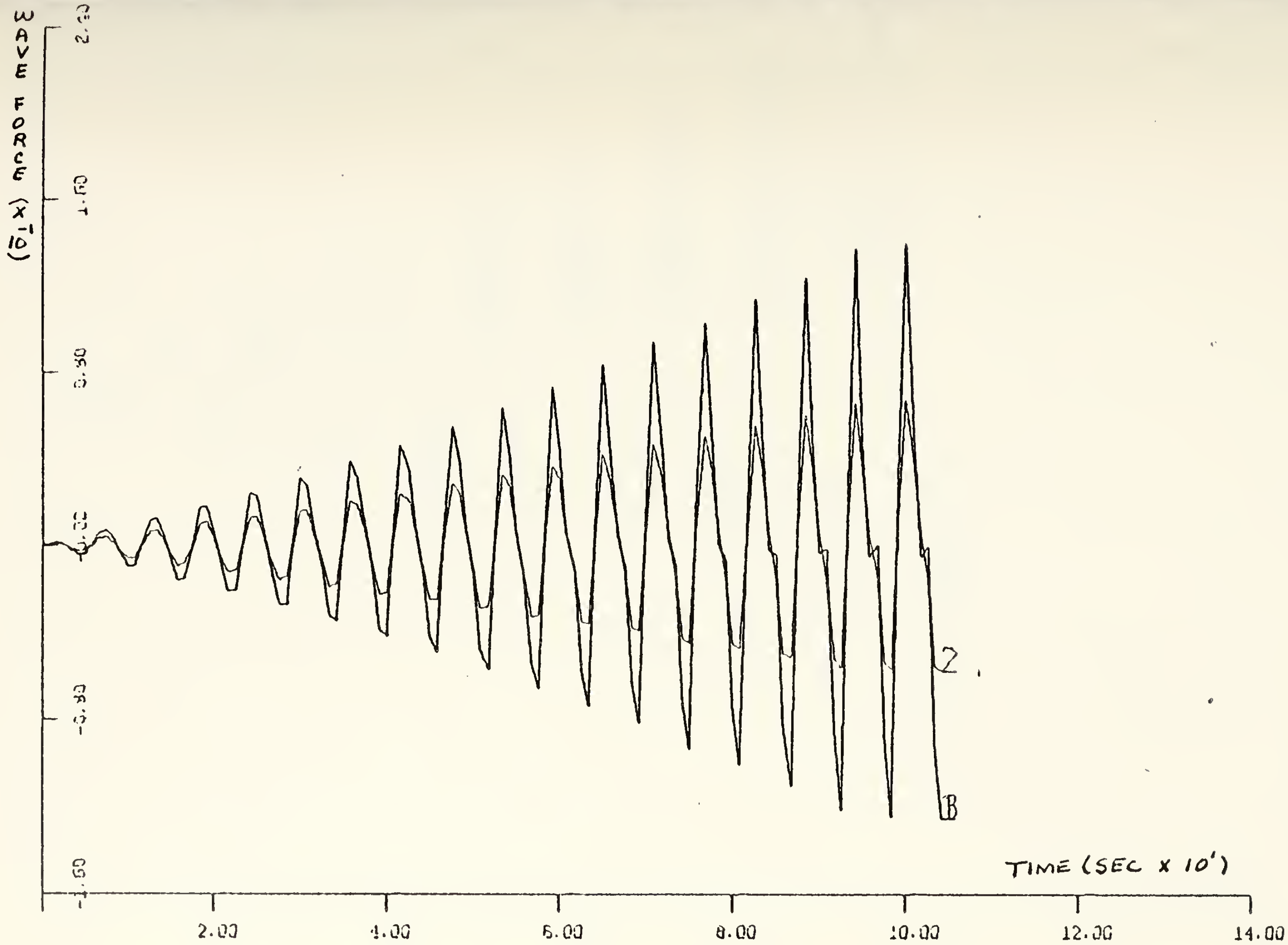


Figure II-20

Wave Simulation Run #4

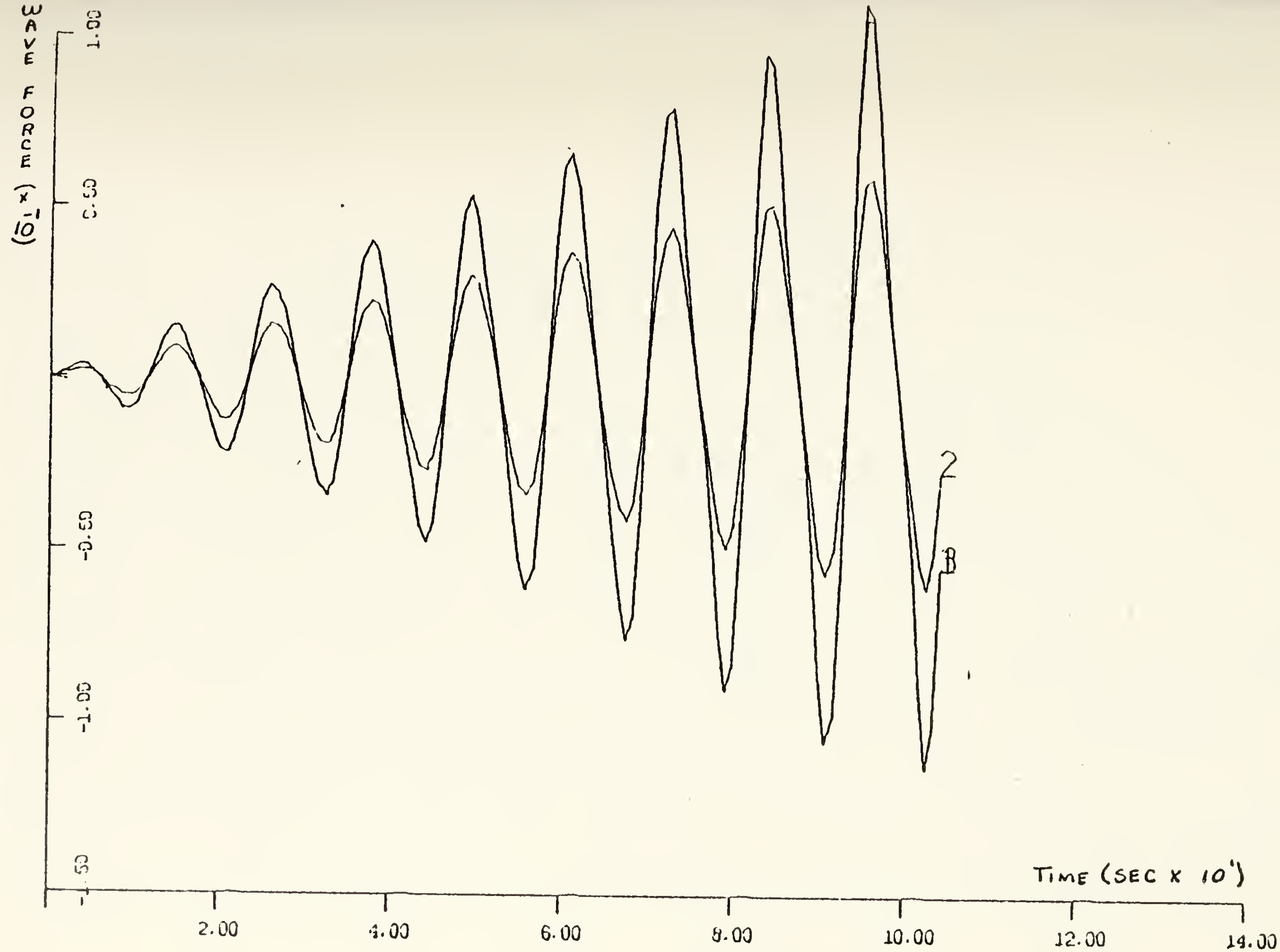


Figure II-21
Wave Simulation Run #5

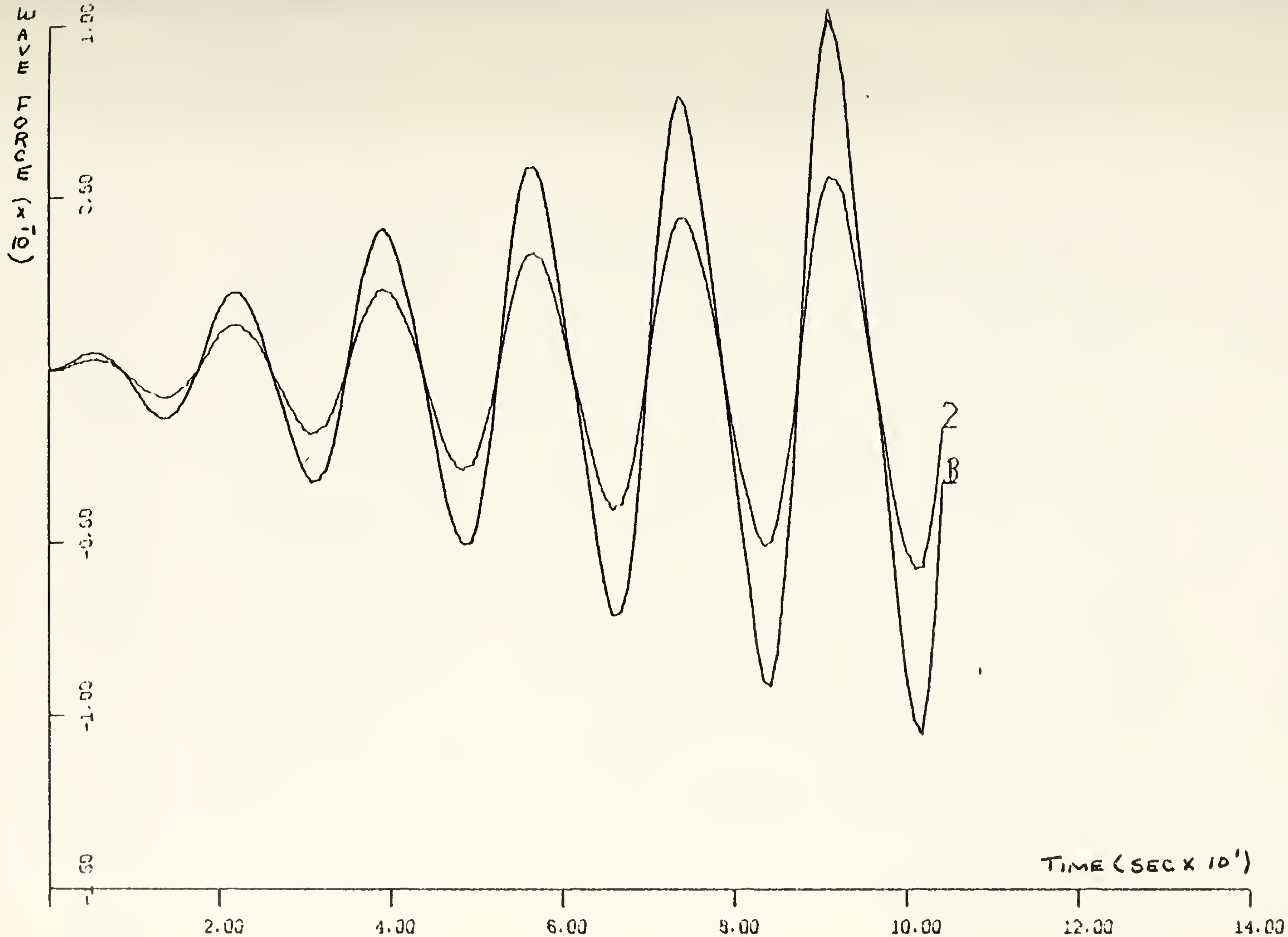


Figure II-22
Wave Simulation Run #6

III. REPLENISHMENT AT SEA

A. HEADING CONTROL

1. Control Choice

Many studies involving replenishment at sea (RAS) have treated the problem as a multivariable system[10][11][12][13]. Academically, there is nothing wrong with this approach. However, as a practical system it leaves much to be desired. The key drawback in the multivariable system is the inescapable dependency on a command and control link between the replenishment ships. The unreliability of UHF communications at these close distances is a much experienced phenomenom. It is felt that any knowledgeable commanding officer would not entrust the safety of his ship to such a questionable link. An alternate method which is described here is a modern extension of the long trusted "seaman's eye" concept, where the sensors and control devices must be self contained on the ship making the approach (hereafter referred to as the receiving ship or ship B).

In all present day RAS operations, the ship on which the approach is being made (hereafter referred to as the supplying ship or ship A), must maintain the replenishment course and speed. The receiving ship accomplishes the maneuvers to maintain station relative to the supplying ship.

The parameters which are presently measured "visually" are relative position (in both the X and the Y directions), relative head (usually in reference to ordered replenishment course), and relative motion in the X direction (for speed matching). These parameters are usually visualized by the conning officer who in turn gives corrective orders to the helmsman. The helmsman must then translate these verbal orders into rudder and speed commands through the helm and lee helm consoles. The accuracy of the execution of the conning officer's orders is extremely dependent on the ability of the helmsman and throttleman. This system can be quite effective, and it can also be quite disastrous. This fast reacting and constantly changing environment lends itself to breakdown in communications and manifests the inability of some individuals to cope with the required critical man-machine interfaces.

To eliminate these problems, present state of the art digital computers and sensors are available for immediate implementation of a completely automatic ship control system. Such a control system may be installed on individual ships and be used for RAS without the requirement of having the matching installation on the other ship of the hookup (another drawback of the multivariable approach).

2. Control Method

One of the many pitfalls that may be encountered in digital simulation is the reality of the parameters that are measurable in the real world situation as opposed to those that are incidently available in the simulation. With this fact as a keynote, Subroutine RBMEAS (Range and Bearing MEASurement) was developed. This subroutine, as listed in appendix A, defines the forward (FWD) and after (AFT) relative and true bearings, and ranges from the receiving ship to similar points on the supplying ship. Figure III-1 delineates the terms used in the subroutine. The SDFn terms

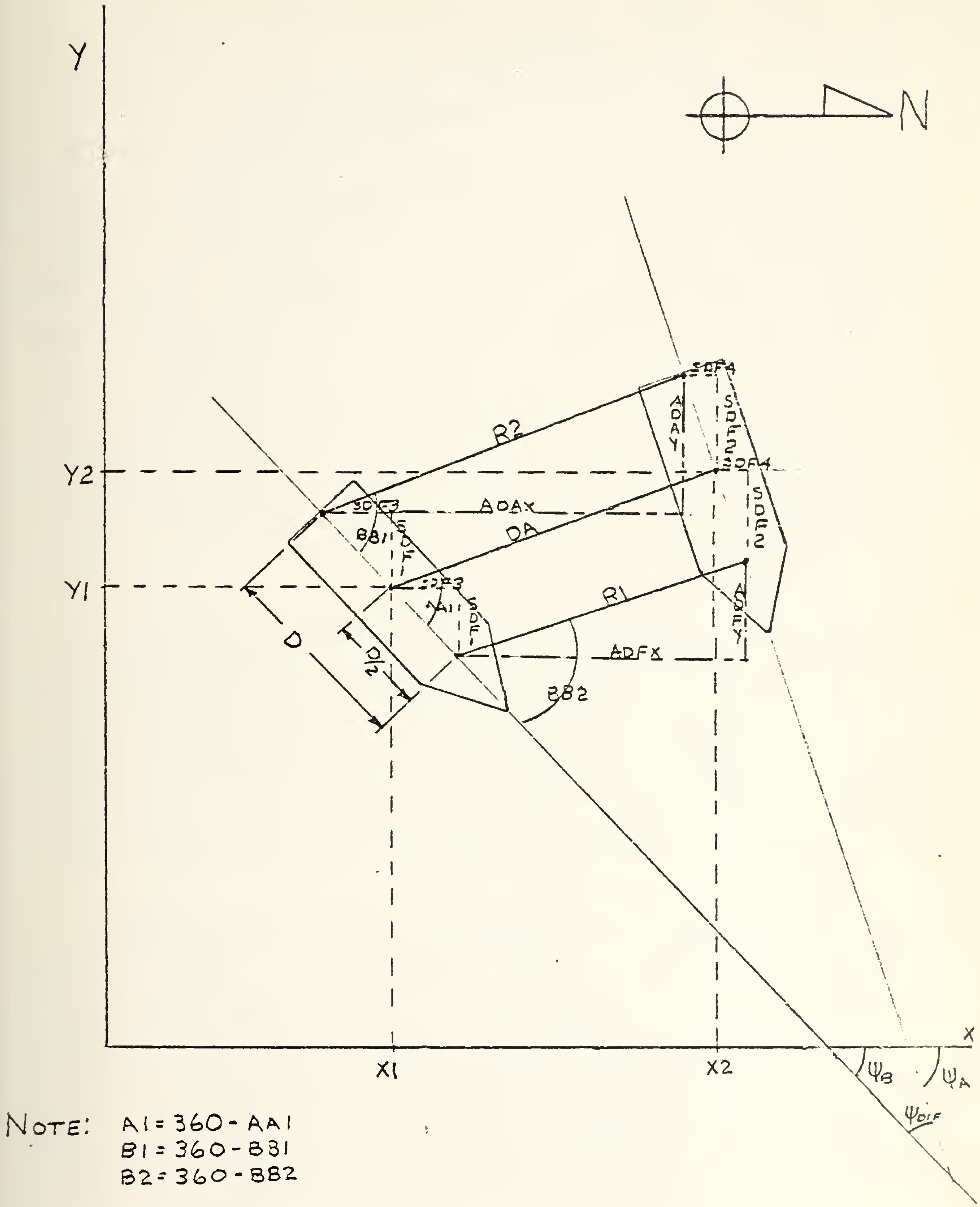


Figure III-1
Measurement Techniques

are used to position the bow and stern sensors and reflectors on ship B and A, respectively, in geographic coordinates as a function of ships head. The FWD distance on the X coordinate is ADFX and the Y coordinate is ADFY. Similarly, the AFT distances are ADAX and ADAY. R1 and R2 are the FWD and AFT ranges measured by a highly accurate ranging device installed on ship B. This same ranging device, if properly provided with a pinpoint reflector on the supply ship(ship A), will give accurate relative bearings FWD and AFT., B1 and B2 respectively. The distance between sensors may be varied, but as a rule should be kept as far apart as possible to allow maximum sensitivity. The distance used in this thesis is 1.0 (one ship length), and the distances were considered the same for both ships. This is not a necessary condition and may be changed to suit the situation.

Subroutine RBMEAS assumes highly accurate sensors in both range and bearing measuring ability. Such sensors are presently available in the form of Radar altimeters[14] and Laser ranging devices. Another possibility for a measuring method is a single sensor time sharing to obtain range and bearing to both reflectors from a single device. Such a single sensor scheme is sketched in figure III-2.

Once the FWD and AFT parameters are available, they may then be used to determine other desired quantities. Subroutine HDGRAS (HeaDinG control for RAS) was developed to output the desired heading corrected for heading difference of ship A and B and the projected correction for distance error. This subroutine is listed in appendix A. The center range and bearing are the average of the FWD and AFT range and bearing output from Subroutine RBMEAS.

The additional heading due to distance is projected as if ship B maintained its present course until it was

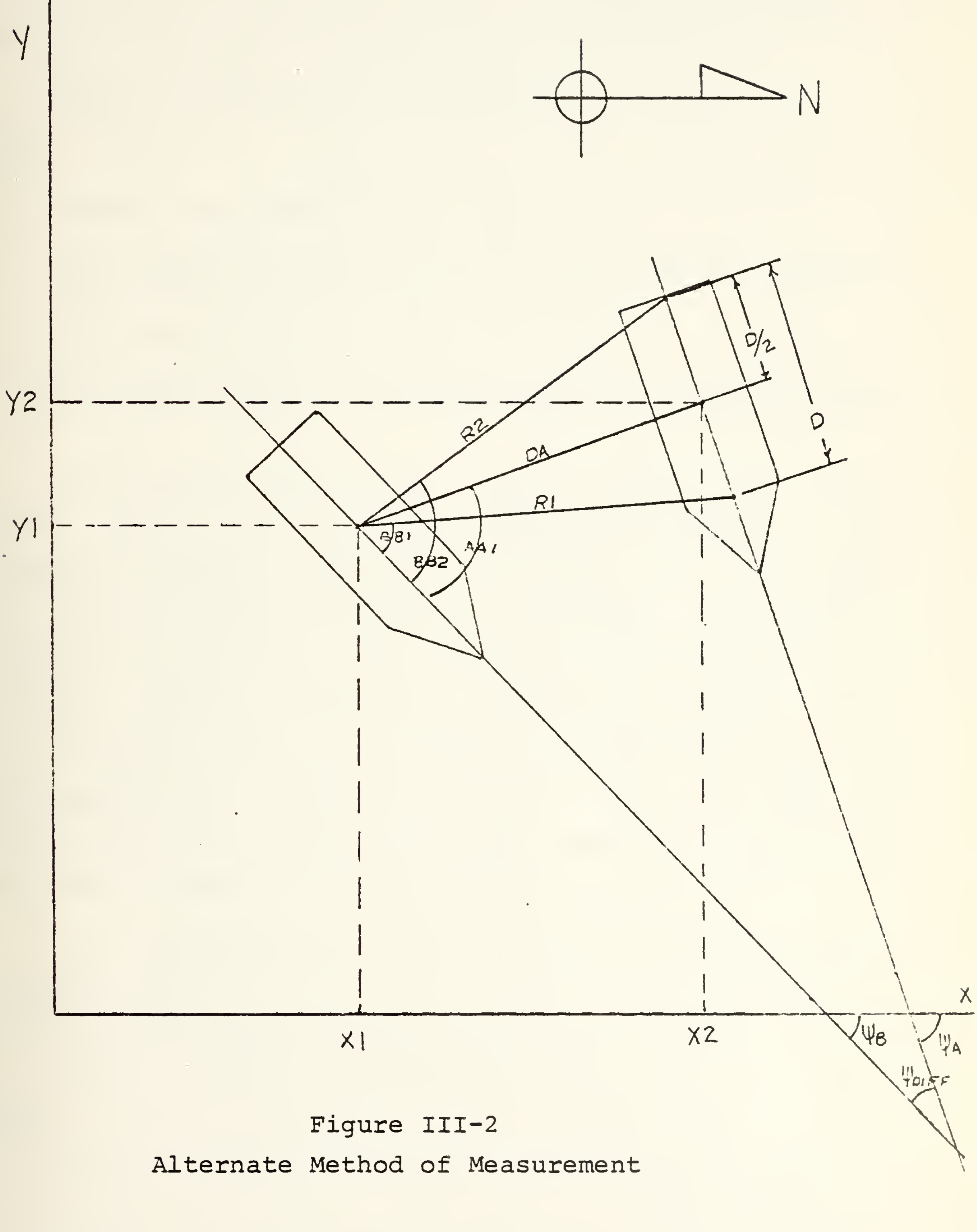


Figure III-2
Alternate Method of Measurement

perpendicular to the center of ship A. The reasoning behind this is illustrated in figure III-3. If the present course will cause ship B to arrive on the station desired (DS), no heading change is required. The expression for PSIADC (Ψ Additional heading due to Distance Correction) is:

PSIADC = RSENS•(DDC+DA•SIN(AA1))

where:

RSENS = Range SENSitivity gain

DDC = Distance Desired Corrected for side of approach

DA = center Distance Absolute (range)

AA1 = 360 degrees - relative bearing of center position

The heading difference of ship A and B is desired since, even if the range when alongside is correct, a large disparity in heading cannot be tolerated. It is realized that some heading difference (crabbing) is necessary to maintain the distance. This crabbing is due entirely to the pressure forces modeled in chapter II. This heading difference is found by computing the difference in the perpendicular projection between the FWD and AFT measurements and finding the arcsin of this difference divided by the distance between sensors. Figure III-4 indicates a sample of this procedure.

The expression for total desired heading is given as follows:

PSIDES = PSIADC+WTSENS•PSIDIF+PSIB

where:

PSIDIF = Ψ additional heading due to heading DIFference

WTSENS = WeighTed heading difference SENSitivity gain

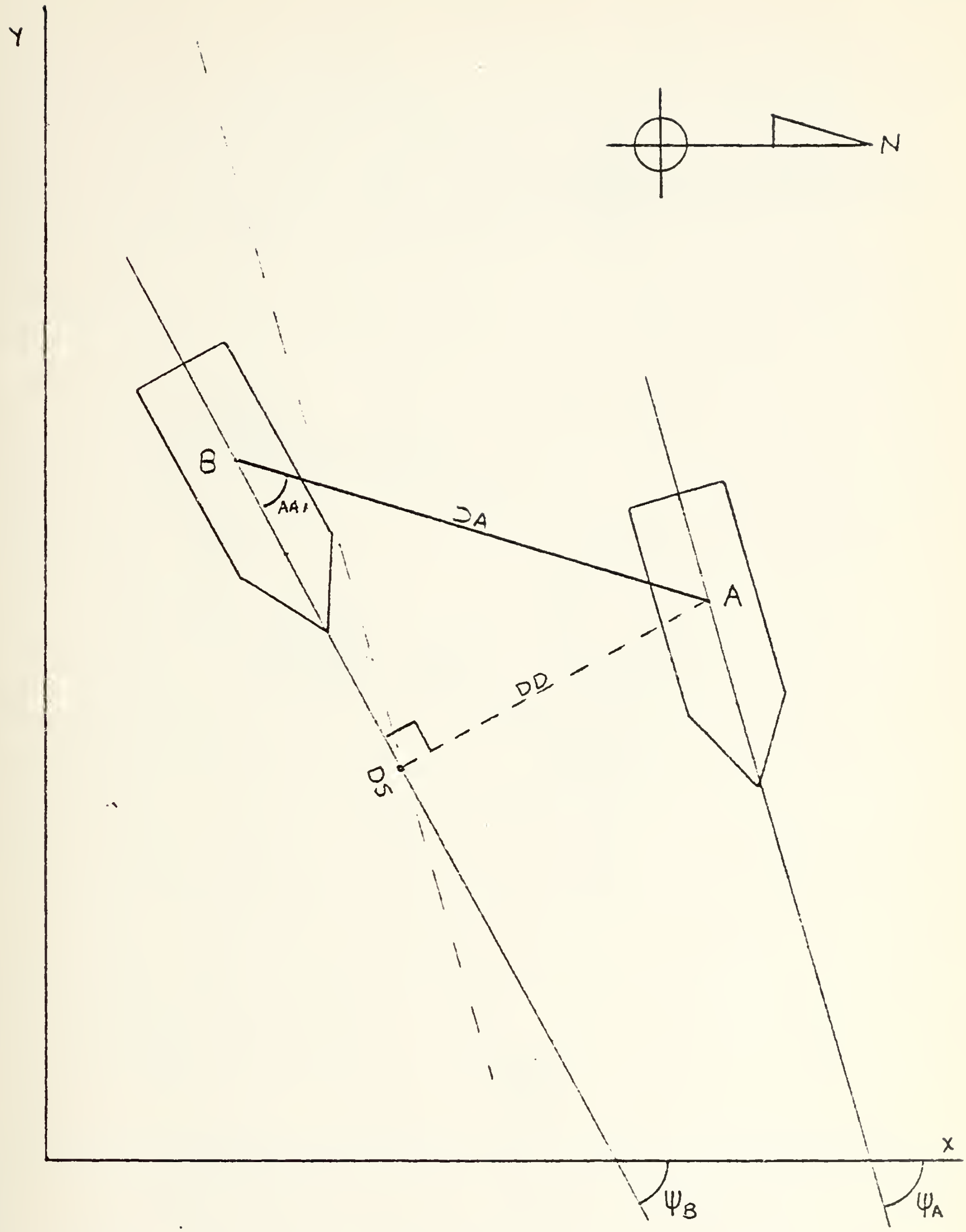


Figure III-3
Distance Logic

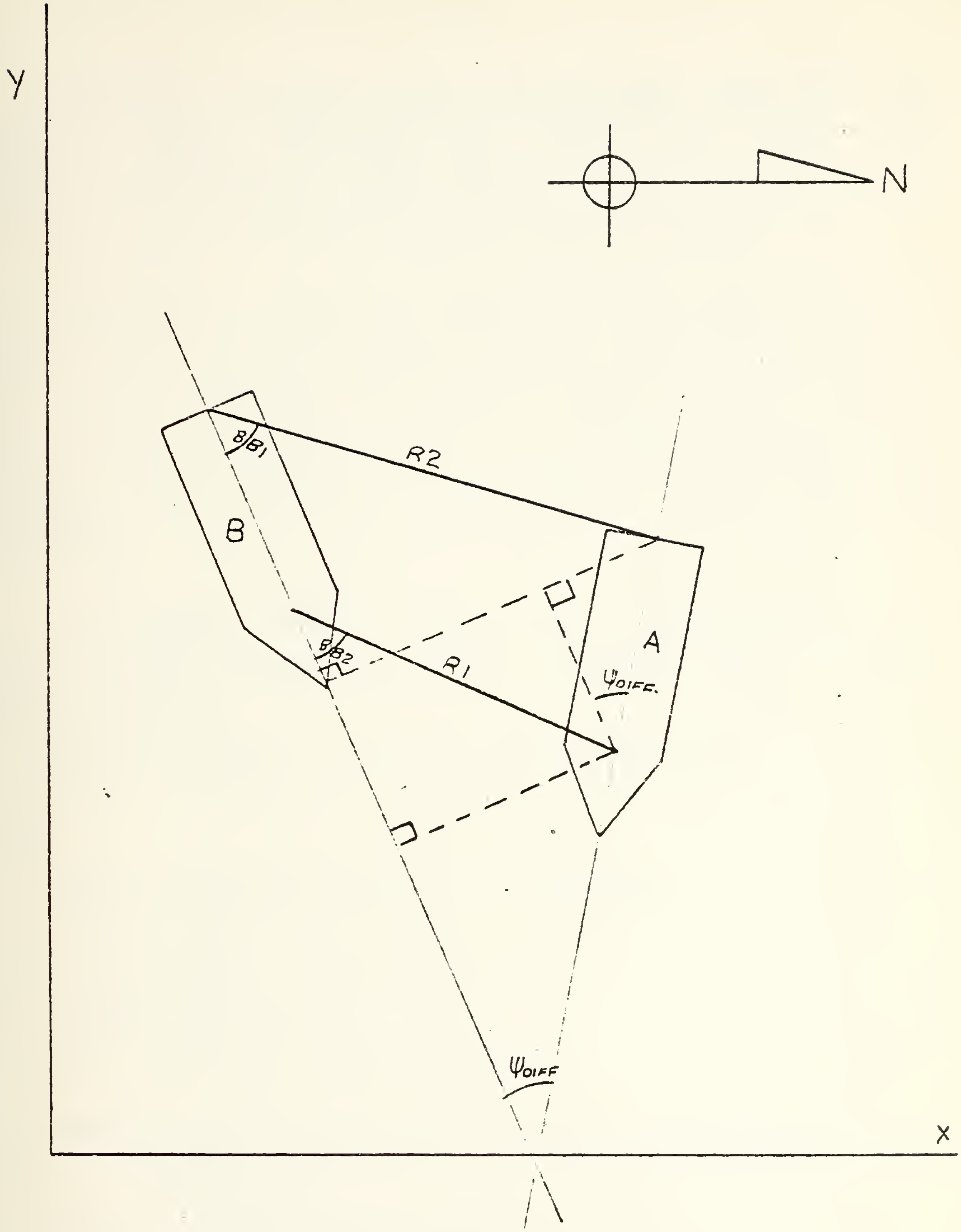


Figure III-4
Heading Difference Calculation

PSIDES = ψ (heading) DESired

Throughout the subroutines and main DSL programs, the Function DEGRAD (conversion of degrees to radians and radians to degrees) is used freely. An explanation and listing of this function are presented in appendix A.

The angular velocity of the receiving ship's head is also of concern in the RAS situation. This quantity may be thought of as similar to tachometer feedback in a simple servo control system; and is necessary to damp out the response (the responses of this control system without this feedback is presented in the latter section of this chapter).

The desired rudder command is a combination of the desired heading, angular velocity feedback, and a rudder gain as follows:

Desired Rudder = (YAWD2-PSIDED+BDOTFB)*RGN

where:

YAWD2 = heading of ship B (in degrees)

PSIDED =PSIDES (in degrees)

BDOTFB = VFBG*BDOT2D

VFG = Velocity FeedBack Gain

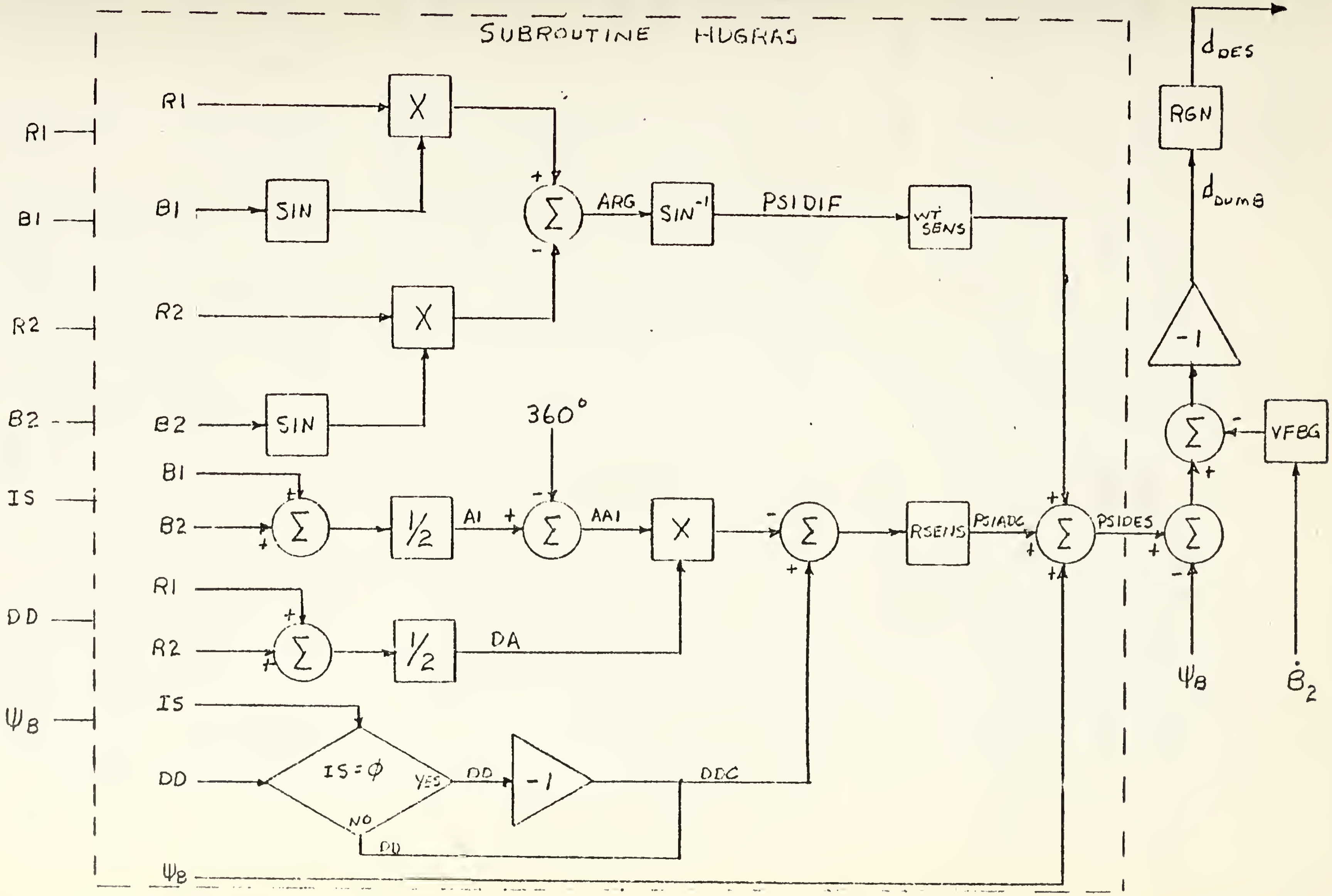
BDOT2D = angular velocity of ship B heading angle (in degrees/sec)

BDOTFB = angular velocity FeedBack

RGN = Rudder GaiN

The convention for rudder response dictates negative rudder as being right rudder, which causes positive yaw. This necessitates making the desired rudder the negative of the forcing function and feedback quantities. The block diagram of figure III-5 presents the control loop from measurement inputs to desired rudder command.

Figure III-5
Rudder Control Block Diagram



3. Optimization

Thus far the control choice has identified four gains (RSENS, RISENS, RGN, VFBG) that must be set for proper position attainment. The nonlinear nature of the system which appears in the form of distance measurements, interactive forces and rudder modeling do not allow for straight forward determination of these gains with normal optimal control theory.

a. Technique

Grossly nonlinear systems require special handling to determine proper gain settings. The method chosen for this purpose was an optimization algorithm developed by M. J. Box (programmed locally as subroutine BOXPLX). This subroutine, listed in appendix A, was used to locate the cost function saddle point in four dimensional space (the dimensions being the previously mentioned gains). The drawback associated with this method is the necessity of iterating the complete nonlinear simulation within function FE for every evaluation of the cost function. The gains sought were found, but unfortunately only after 2 1/2 to 3 hours of CPU time with every 400 iterations allowed.

The mechanics involved in optimizing the chosen cost function include required sub-calculations in many functions and subroutines. Figure III-6 is a flow chart which demonstrates the steps, subroutines and functions required.

Initial optimization was accomplished for one set of initial conditions. By looking at the RAS situation, a probable set of circumstances were envisioned. The scenario setting is the approach phase where the replenishing ship

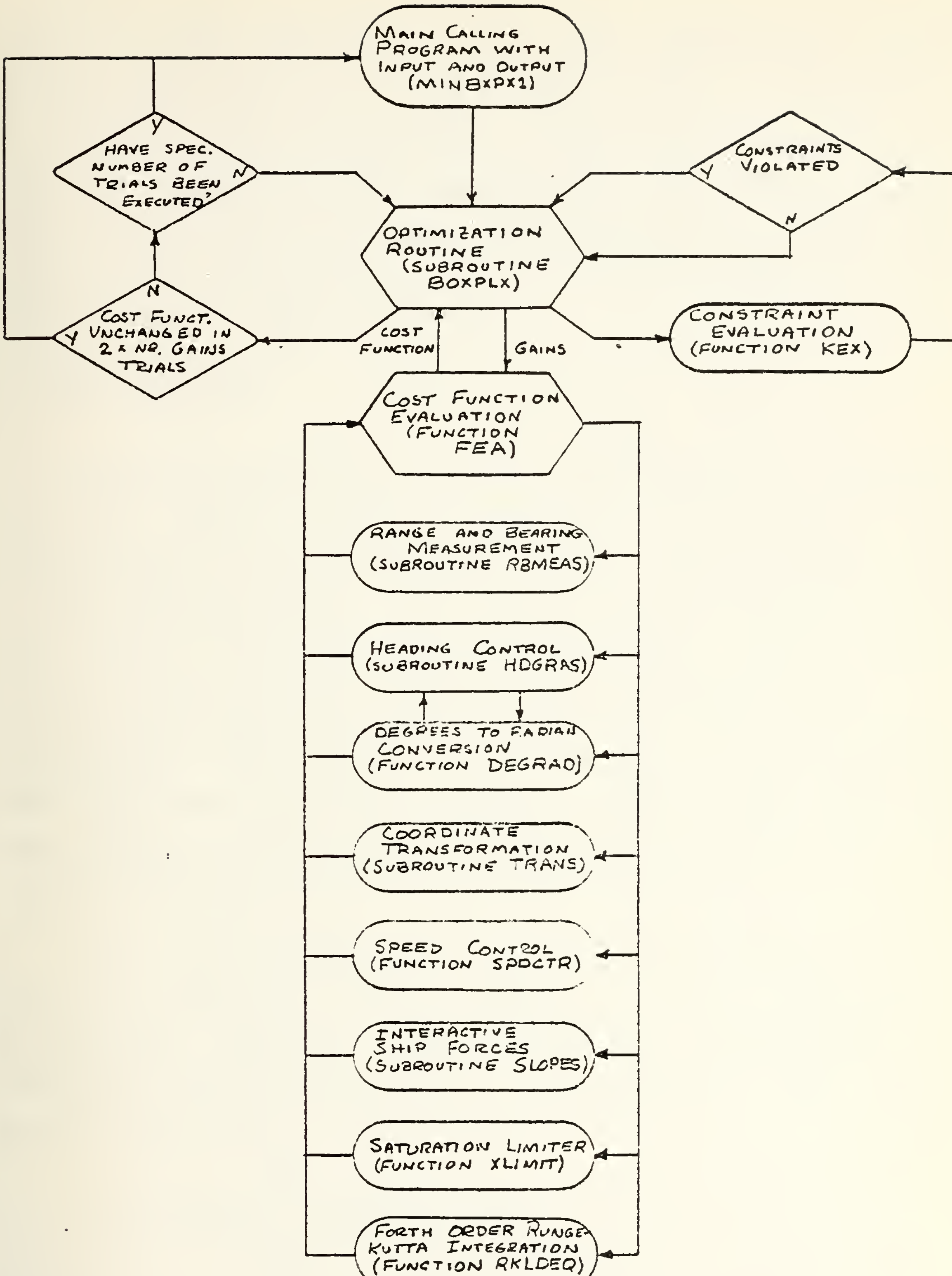


Figure III-6
Optimization Flow Chart

starts a wide approach at 0.4 ship lengths (211 feet) lateral displacement and 5 ship lengths (2639.0 feet) astern of the supply ship. The desired final position is alongside at 0.2 ship lengths (105.6 feet) lateral separation. Both ships have the same initial heading (YAW angle). The supply ship is at 15 kts. (1.0 normalized speed) and the receiving ship makes its approach at 22.5 kts (speed control will be covered later in this chapter).

b. Cost Function

Normal costing of displacement error with the integral of the squared error (ISE) was considered as the optimization tool in subroutine BOXPLX. However, this type of performance measure would weigh the initial displacement error equally with the final position error. This problem can be circumvented by comparing the displacement error to a pre-computed reference track instead of to the desired displacement. For the envisioned scenario, it was conceived that the cost function should weight the distance displacement heavier when the ships are alongside than when the approach is started 5 ship lengths astern.

This was accomplished by using the integral of time times the absolute error (ITAE) as the optimization performance measure. The reference displacement was considered the desired position displacement. The object function can then be written as:

$$CBJ = \int_{t_o}^{t_f} t \bullet |DD-ADY| \, dt$$

where:

DD = Desired Distance

ADY = Actual Displacement in the Y direction

t = time

A performance measure that is designed to obtain good performance must also take into account other factors besides just position accuracy. Consequently, another cost criterion was decided upon which would also set the gains to reduce the amount of rudder control required when alongside. This particular feature is derived from the desire not to over control with the rudder in such close proximity to another vessel. The inclusion of this term in the performance measure is weighted by unity while the distance accuracy is weighted by a factor of 10.0. This will tend to allow rudder action if the desired position is not maintained. The final approach phase cost function for obtaining optimum gains has the form:

$$CBJ = \int_{t_0}^{t_f} t \bullet (10.0 \bullet |DD-ADY| + 1.0 \bullet |D2|)\ dt$$

where the additional term is:

D2 = rudder response of the replenishing ship

c. Results

In the process of deciding on the best gain definitions previously mentioned, many optimization runs were made. Each set of gains were then simulated in a corresponding DSL program to obtain performance confirmation. Many of these runs did not live up to expectations; causing re-evaluation of the control scheme until the one presented in this thesis was formulated.

Table III-1 shows the input upper and lower limits of search (EU, EL), starting value guess (XS), optimum gain settings (Output) and associated object function value (CBJ) for 20.0 second normalized time simulation run in function FE. These values were then introduced into the DSL simulation program listed as program #5. The results of this simulation are shown in figures III-7 thru III-12. The

Gain	RSENS	WTSENS	RGN	VFBG
BU	2.0	20.0	50.0	10.0
BL	0.1	0.1	1.0	0.01
XS	1.0	1.0	10.0	1.0
OUTPUT	1.86642	2.3869	23.4185	4.35162
OBJ	60.7103			

Table III-1

Approach Phase Optimization Results

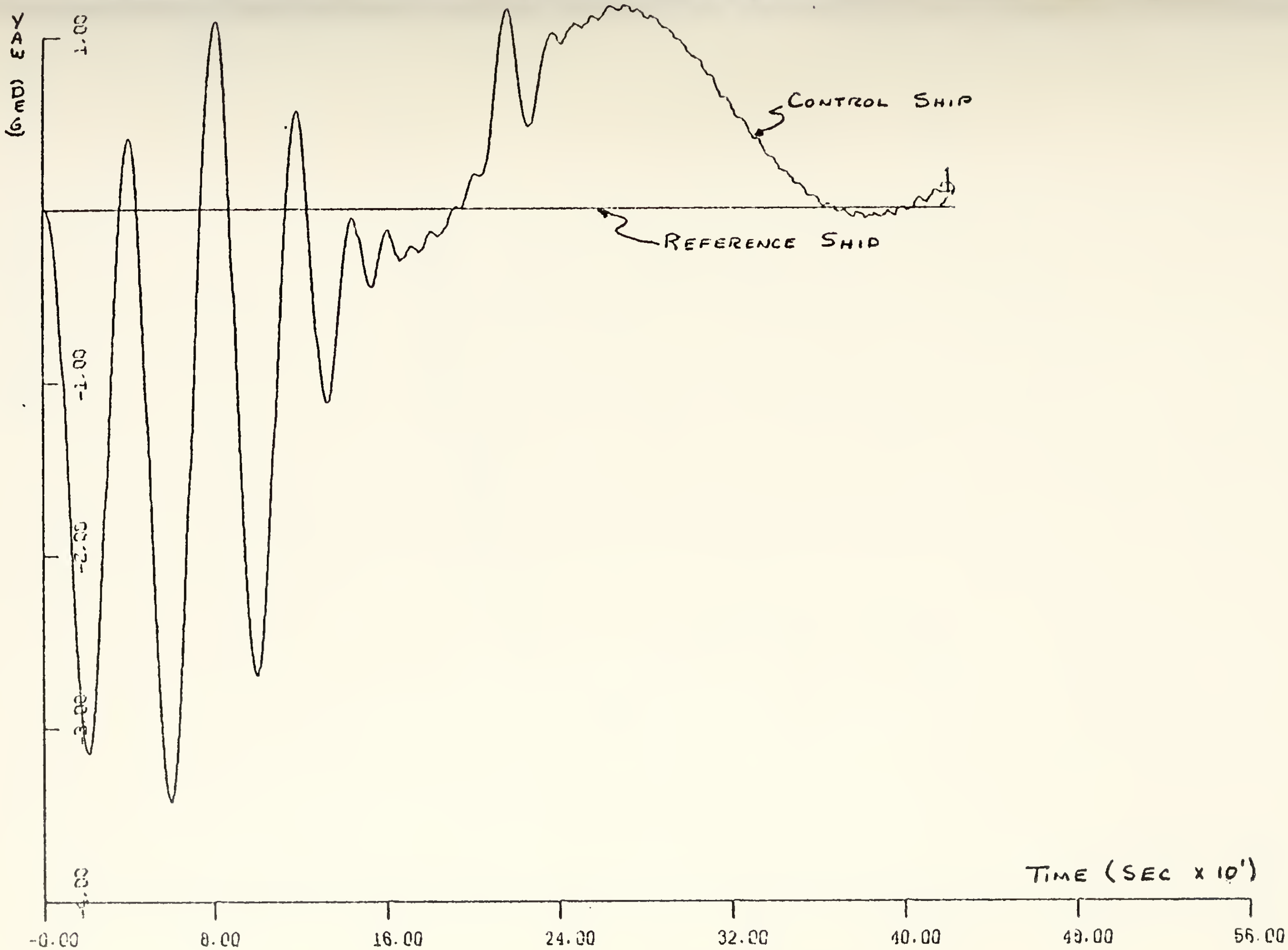


Figure III-7
Approach Phase Yaw Result

Figure III-8

Approach Phase Y Forces

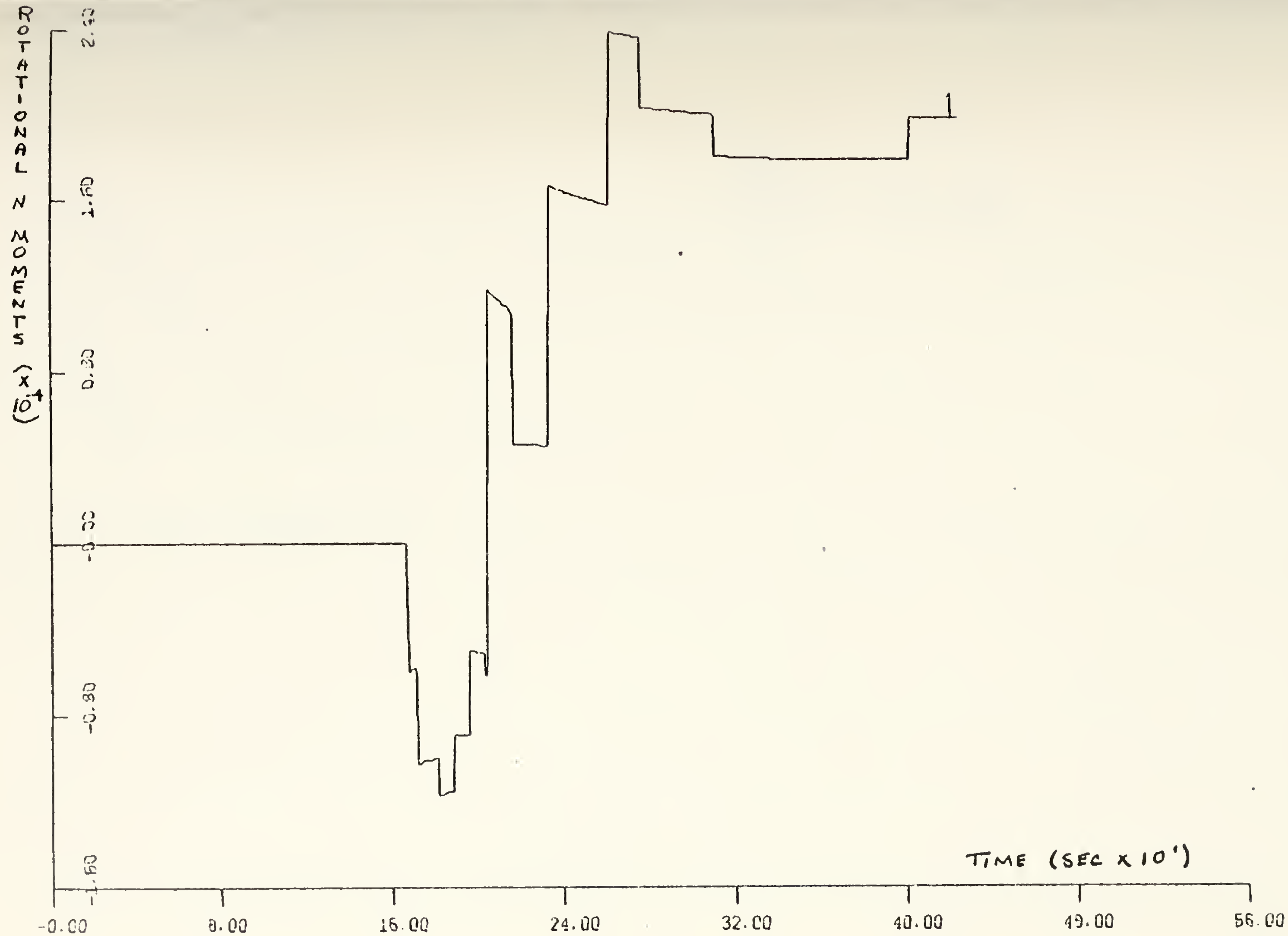


Figure III-9
Approach Phase N Moments

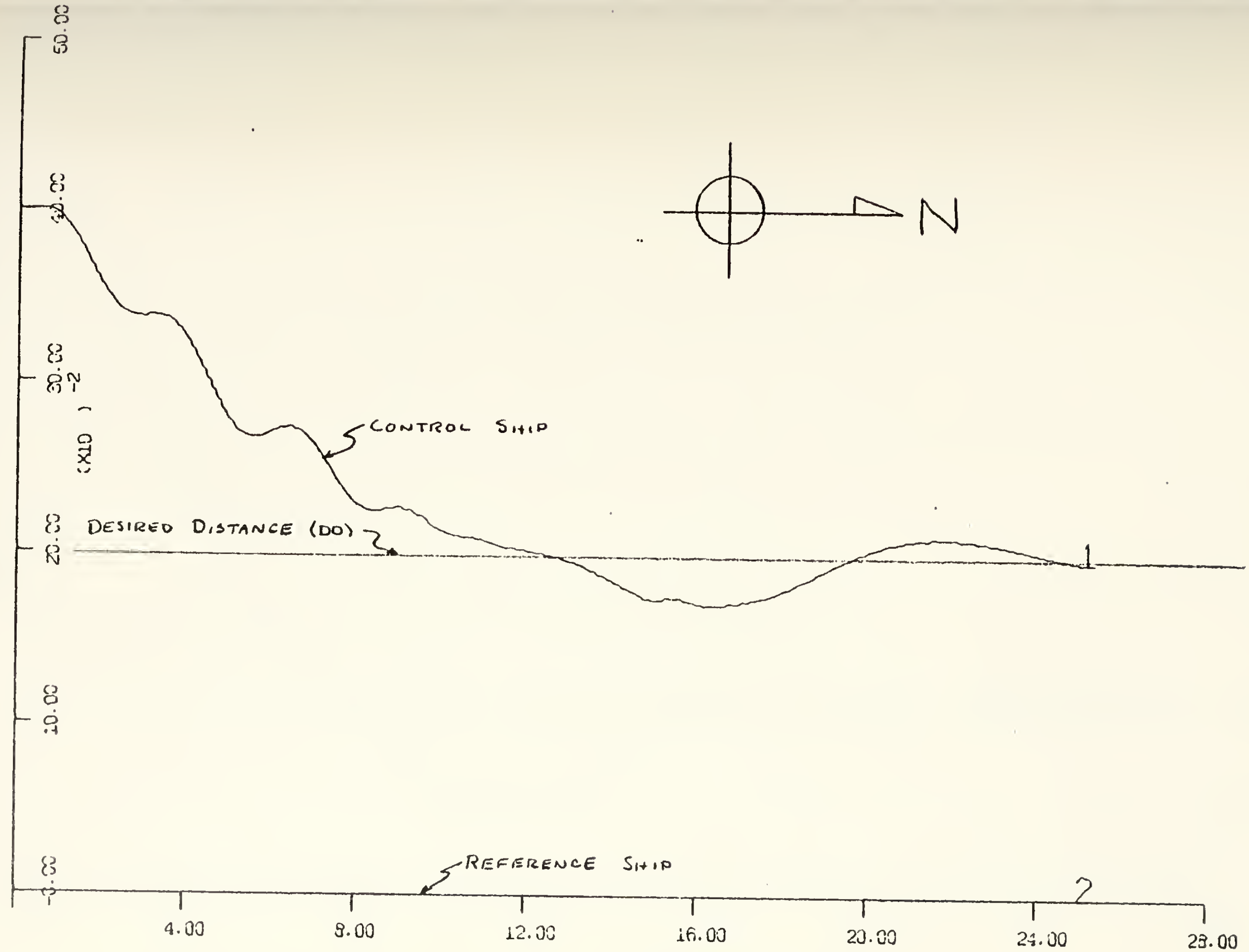


Figure III-10
Approach Phase Geographic Plot

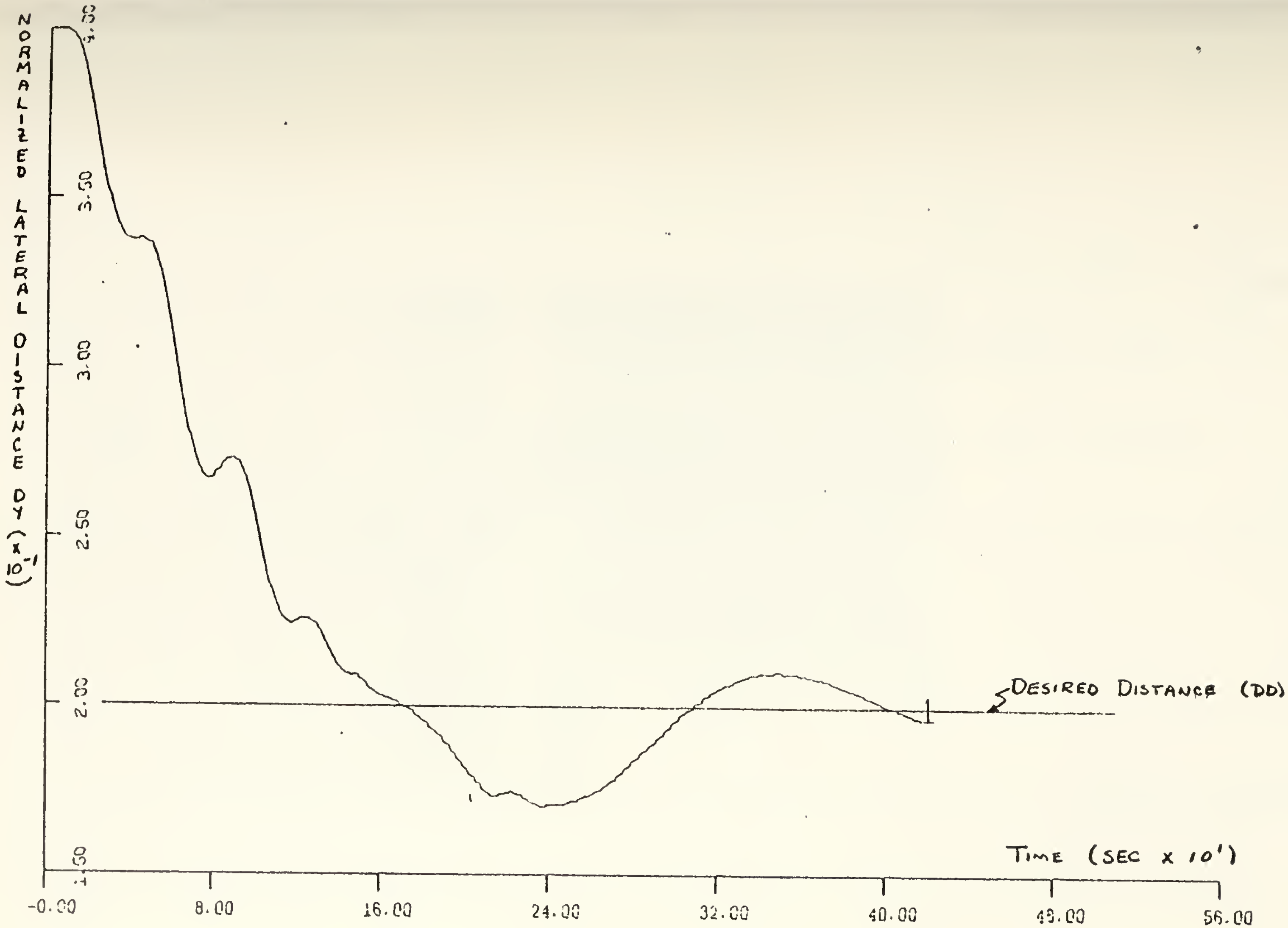


Figure III-11

Approach Phase Lateral Distance DY

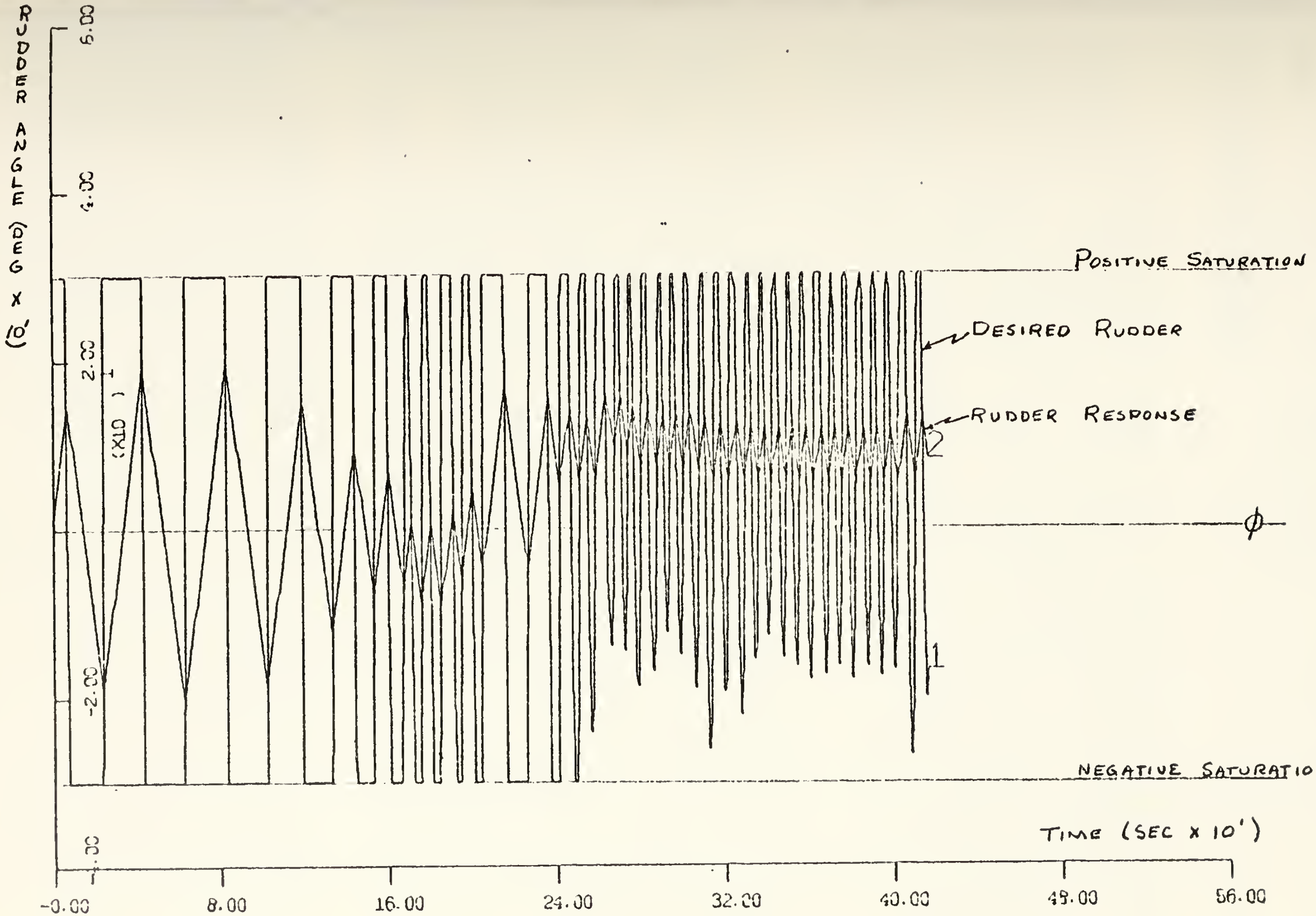


Figure III-12
Approach Phase Rudder Response

geographic plot of figure III-10 indicates excellent positioning in the lateral direction while the rudder response of figure III-12 shows that it settles out to a fairly constant steady state value as the ship settles into its desired position. The time coordinates in all plots are shown in actual full scale time.

d. Control Testing

Now that the "proper" gain settings were obtained, more extensive testing of the control system was required. Three different tests were contemplated: (1) allow a large perturbation turn of the reference ship (supply ship), (2) start approach of the receiving ship (control ship) from different initial conditions of lateral and horizontal displacements, and (3) induce external perturbations in the form of wave forces.

The first test was simulated by turning the reference ship by normal rudder action of figure III-13. This turn with 5 degrees rudder accounted for a total reference yaw change of 15 degrees. The rudder action of the controlled ship shown in figure III-14 was as expected. However, the distance maintainment portrayed in figure III-15 was totally unacceptable. The maximum excursion from the desired distance of 105.56 feet (0.2 normalizes distance) was 55.419 feet (0.105 normalized distance). Variances of this magnitude cannot be tolerated in the RAS environment.

Faced with this situation, the tact chosen was to re-evaluate the gains for the new scenario which is called the turn phase. In this phase the initial conditions assume steady state positioning alongside such that the lateral position displacement (DY) is equal to the desired distance [105.56 feet (0.2 normalized)] and that the horizontal position displacement (DX) is 0.0 (alongside).

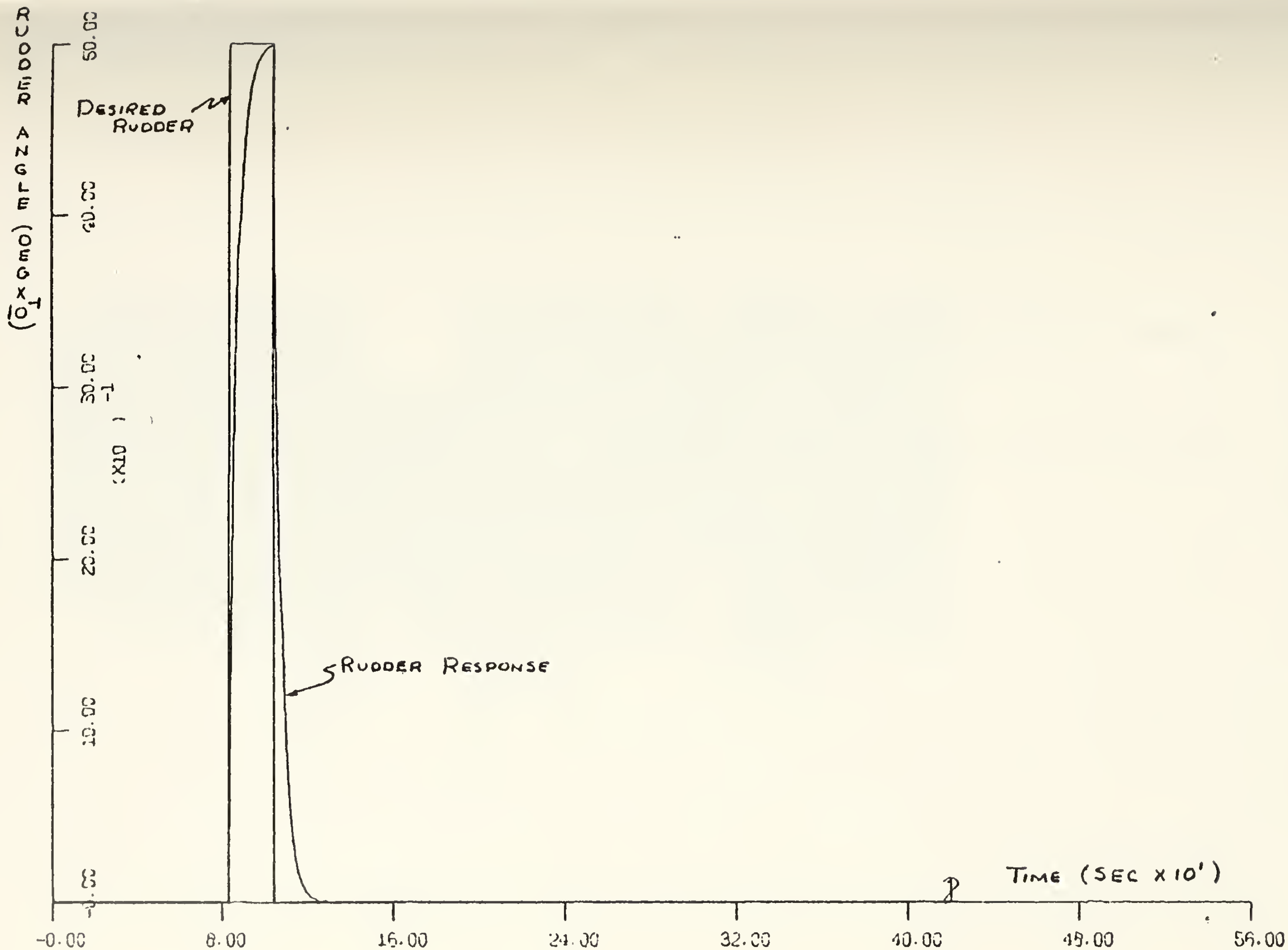


Figure III-13

Turn Phase Rudder Action of Reference Ship

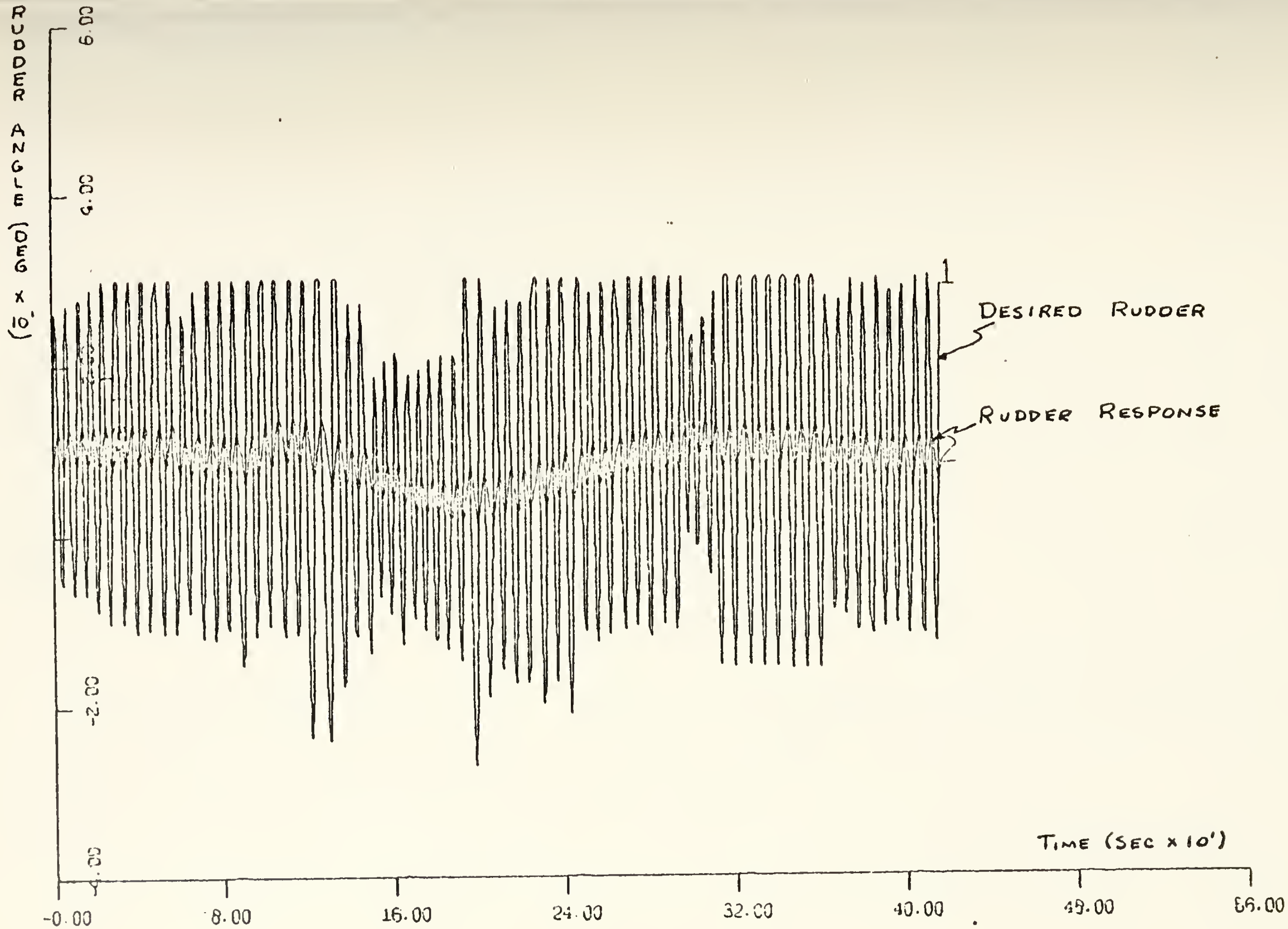


Figure III-14
Turn Phase Rudder Response

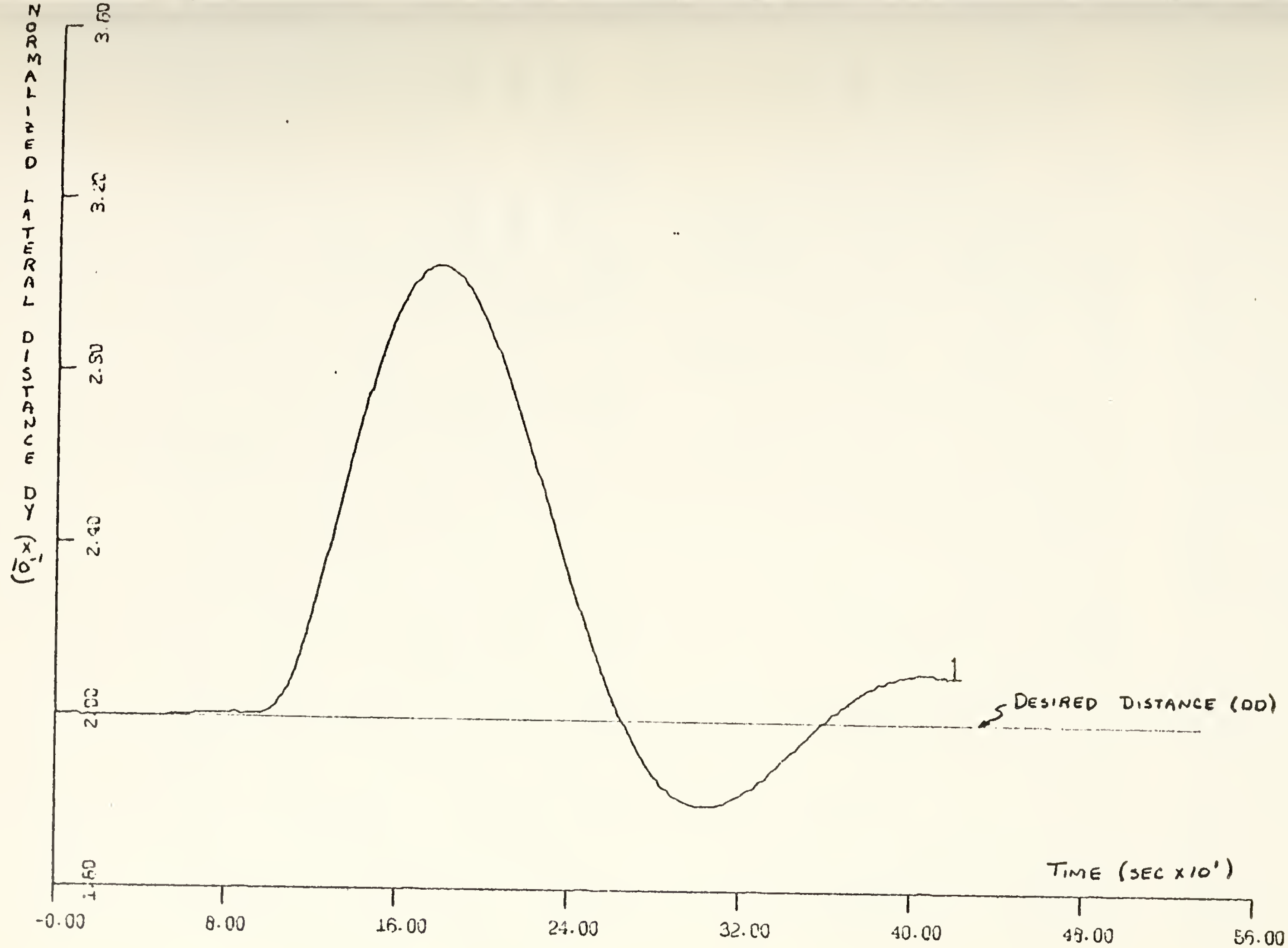


Figure III-15

Turn Phase Lateral Distance DY

Some initial perturbation is introduced by assuming the relative yaw angle when alongside is negligible.

e. second optimization

The same procedure was followed in obtaining gains that would optimize a chosen cost function. Figure III-6 still applies except that function FEA is replaced by function FEB (listed in appendix A) to simulate the new conditions.

Cost function criteria change in this instance since the ships start at the desired position and optimally stay at the same relative positions. Also, the rudder response to such a large turning perturbation must be free to cause achievement of the desired position. Due to these considerations, the integral of the absolute error (IAE) performance measure was chosen for the optimization criterion and can be written as:

$$OBJ = \int_{t_o}^{t_f} |ADY| \, dt$$

Table III-2 shows the results of the turn phase optimization and the comparison with the approach phase gains. Again DSL simulation was performed using the turn phase scenario. Figures III-16 thru III-21 portray the graphical results. The rudder response of figure III-21 indicates very sensitive response to the interactive forces shown in figures III-17 and III-18. The lateral distance separation of figure III-20 indicates excellent position maintainment with maximum excursion error of only 2 feet (0.0038 normalized). This minimal variation is well within that which can be tolerated in the RAS environment.

Gain	RSENS	WTSENS	RGN	VFBG
BU	2.0	20.0	50.0	10.0
BL	0.1	0.1	1.0	0.01
XS	1.0	1.0	10.0	1.0
OUTPUT	1.99765	0.7357	49.9776	0.084028
OBJ	0.009145			
Approach Phase Output	1.86642	2.3869	23.4185	4.35162

Table III-2

Turn Phase Optimization Results

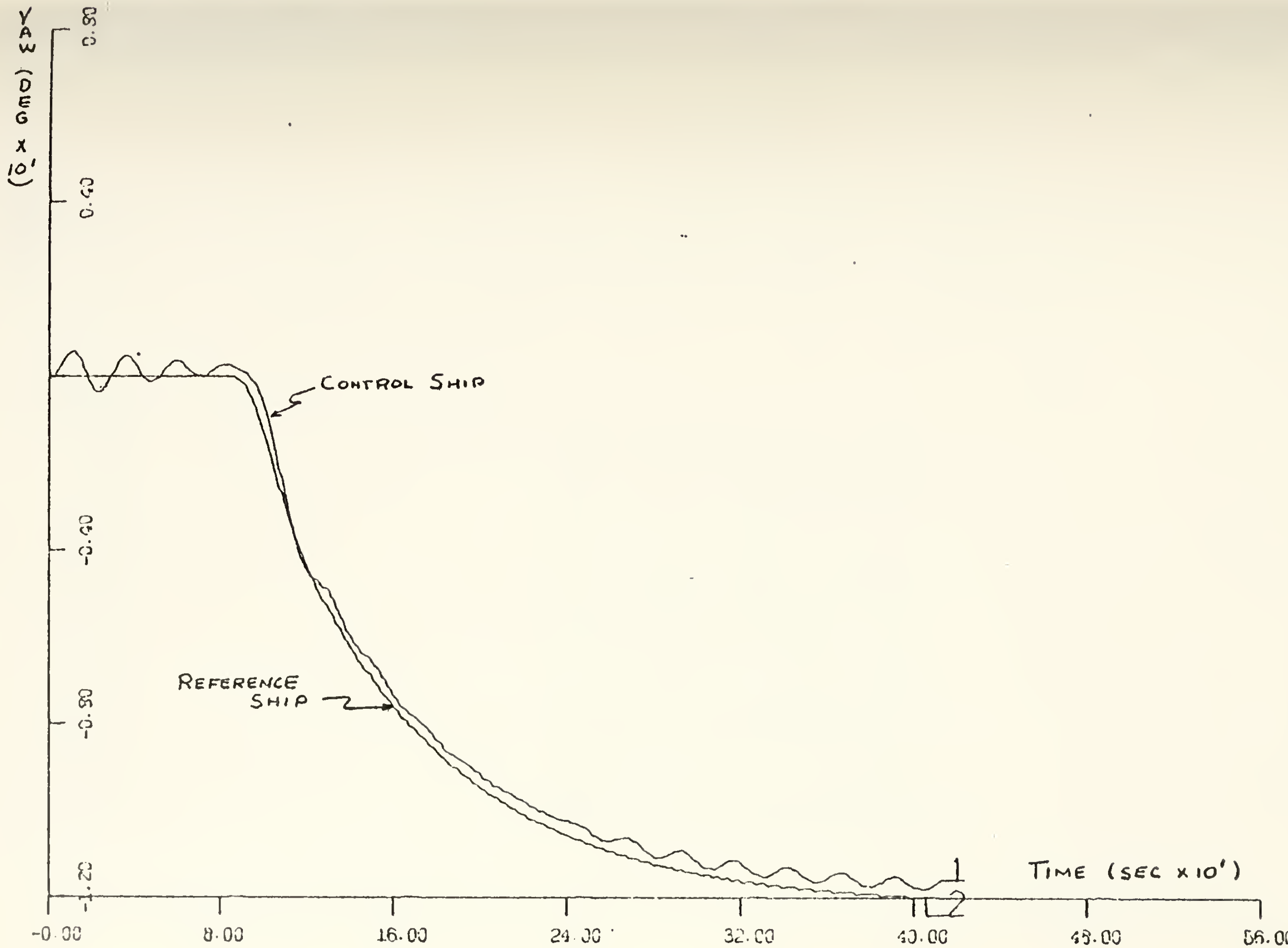


Figure III-16
Turn Phase Yaw Response

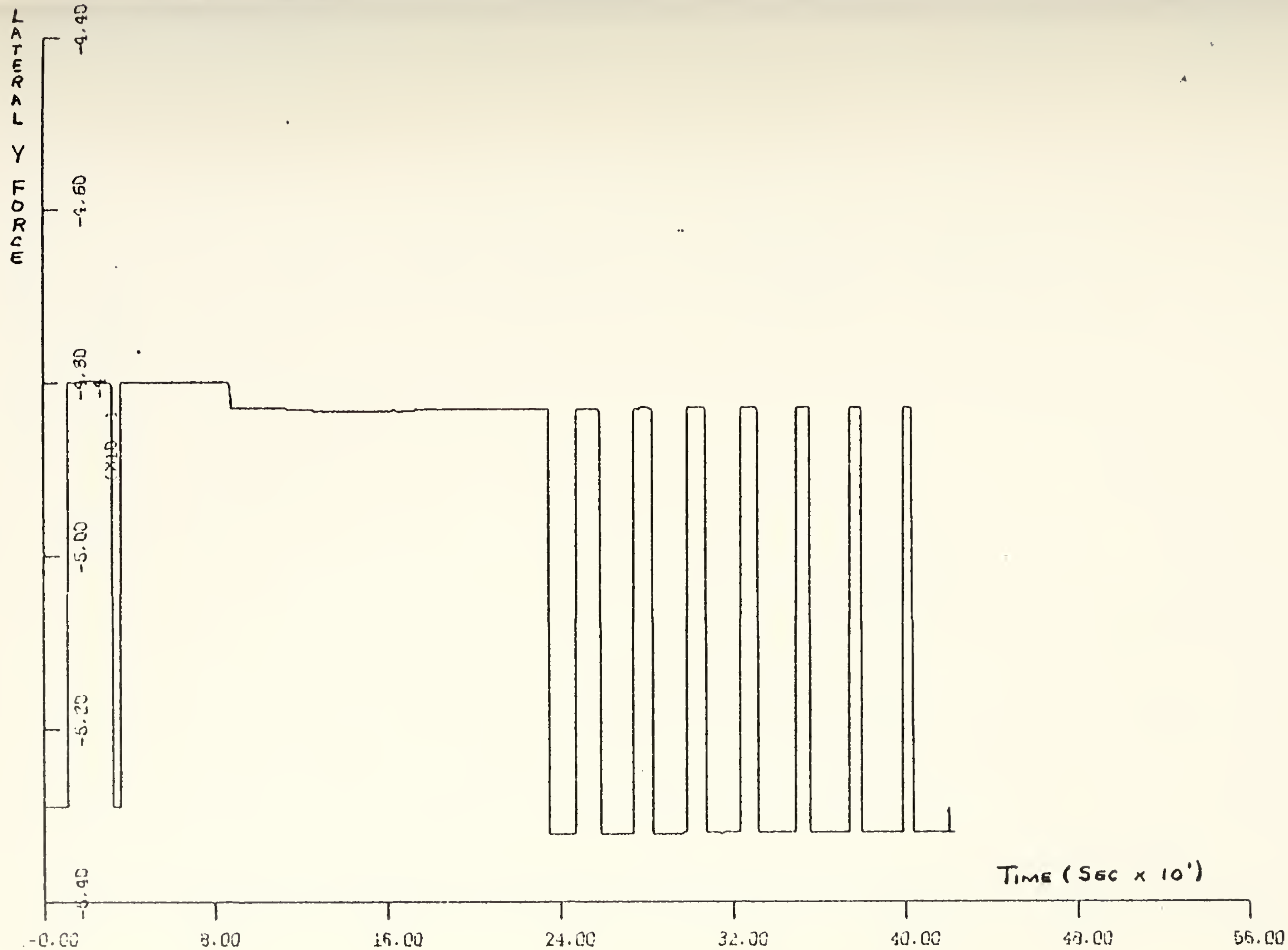


Figure III-17
Turn Phase Y Forces

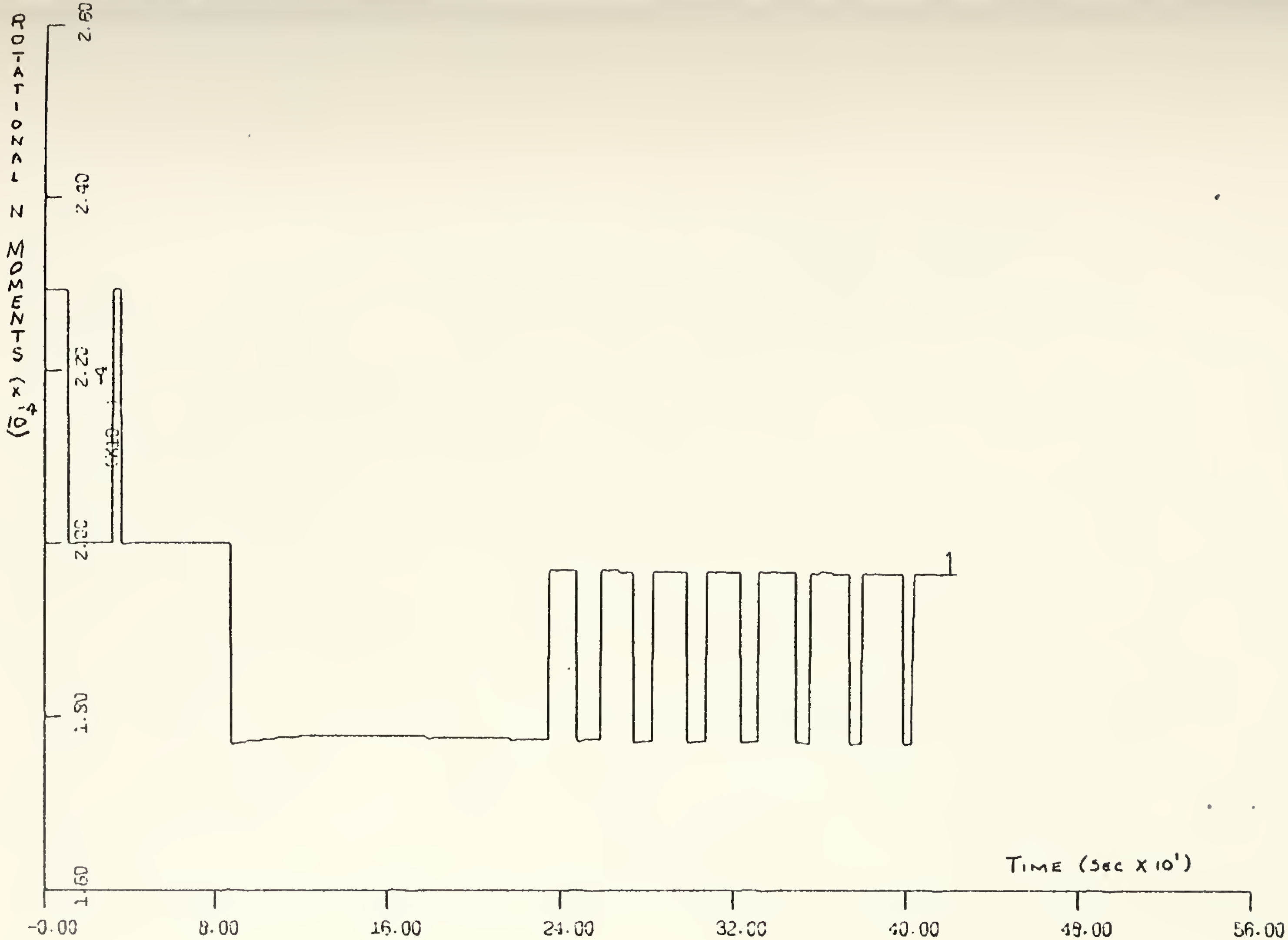


Figure III-18
Turn Phase N Moments

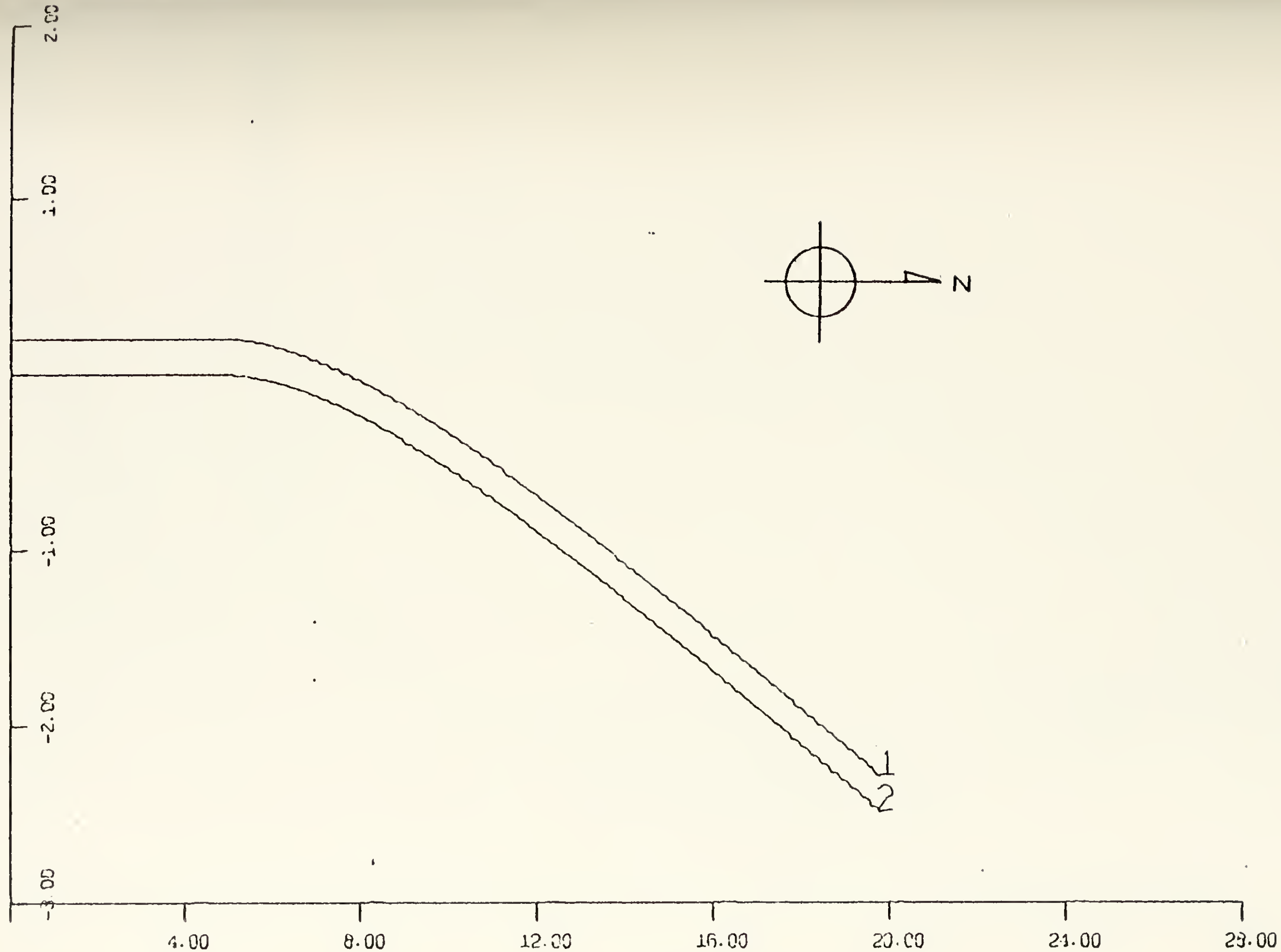


Figure III-19
Turn Phase Geographic Plot

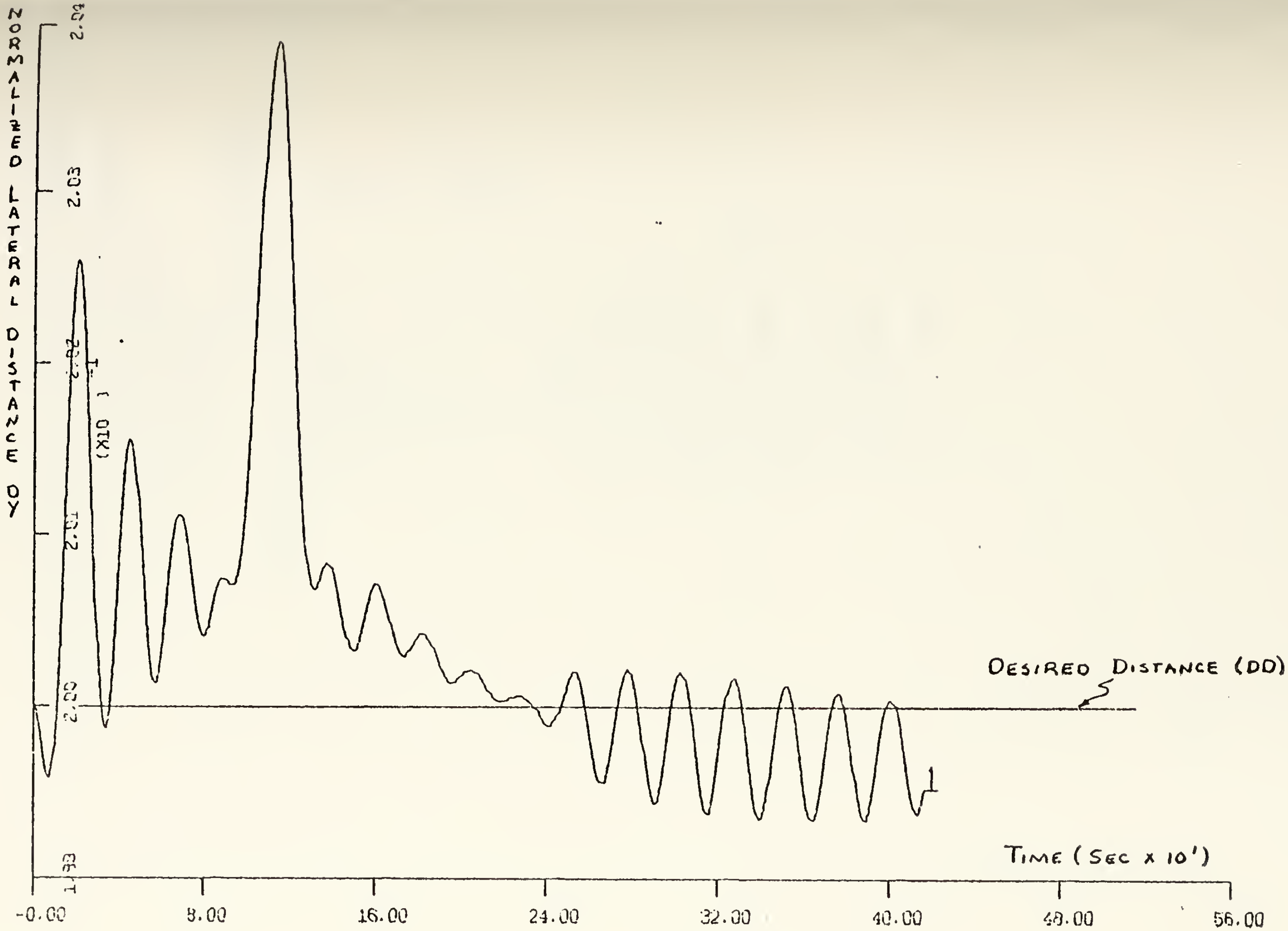


Figure III-20

Turn Phase Lateral Distance DY

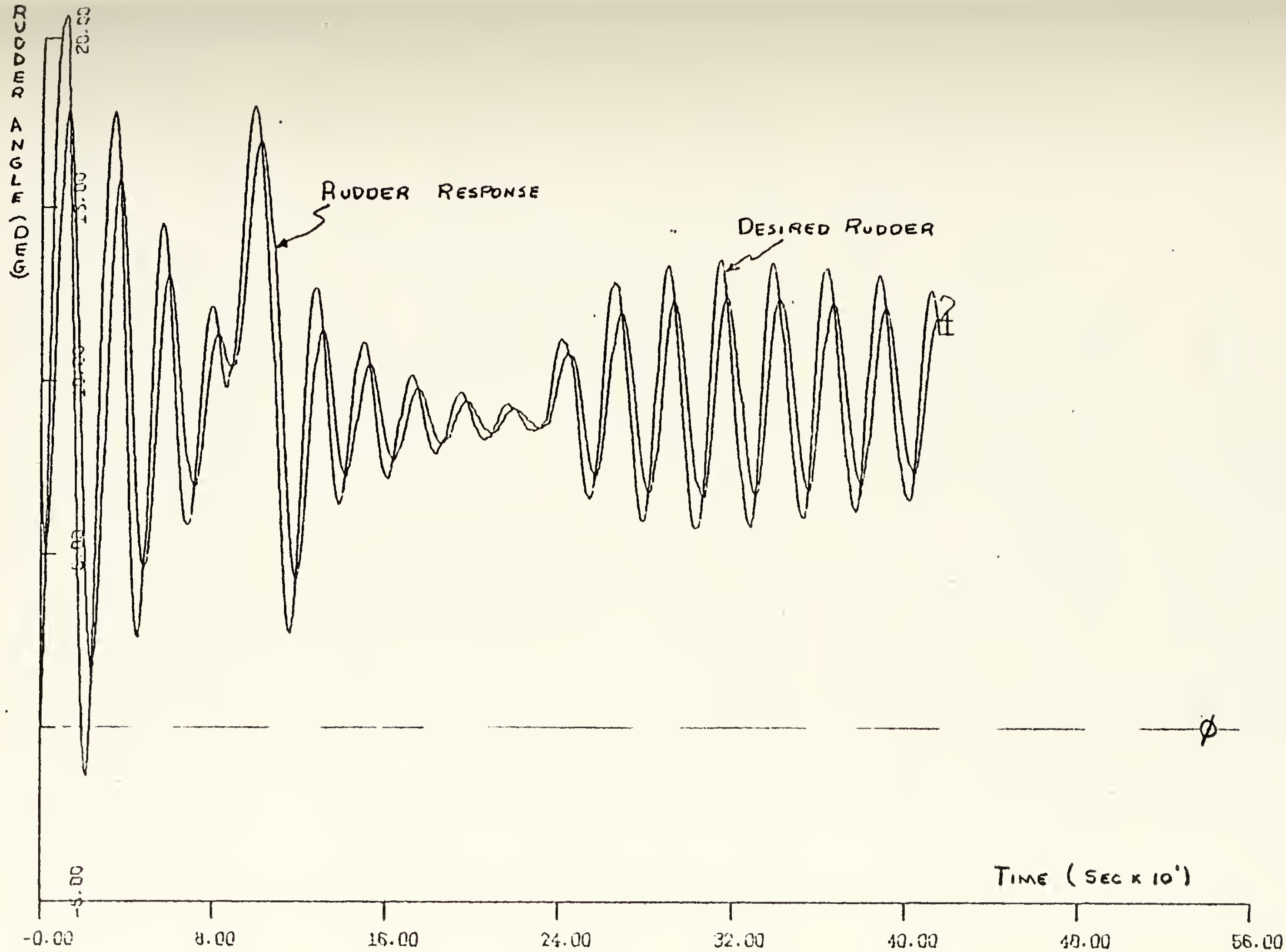


Figure III-21

Turn Phase Rudder Response

f. Continued Control Testing

To alleviate suspicions that the response from the gains obtained in the approach phase could be improved by the gains obtained in the turn phase, a simulation of the approach phase was accomplished with the new gains. Figure III-22 is the graphic display of the effect of these gains on the approach phase lateral distance positioning.

Careful analysis of the results thus far clearly indicate the need for an adaptive control scheme to allow gain adaptation to meet the design specifications. A full adaptive control scheme for systems of this type is outside the scope of this thesis. References 15 thru 23 are indications of some of the literature available for pursuit of a completely adaptive control system.

What was done here is development of a simple algorithm to sense when the conditions were adequate to switch from one set of gains to another. This may be done with the two sets of gains developed thus far. However, for the sake of simulation efficiency, a third set of gains was introduced. This third set amounts to a change of one approach gain (RSENS) which has previously been defined as the range sensitivity gain. The simulation efficiency is increased by decreasing the time required for the approach phase to reach steady state. A consequence of this procedure is a reenforcement of the need for a completely adaptive control scheme.

Repeated simulation revealed that commencing the turn (in effect switching gains), before a reasonable steady state was reached caused results similar to those shown in figure III-22. An increase in RSENS to a value of 4.0 when the lateral separation error is less than 0.05 (normalized)

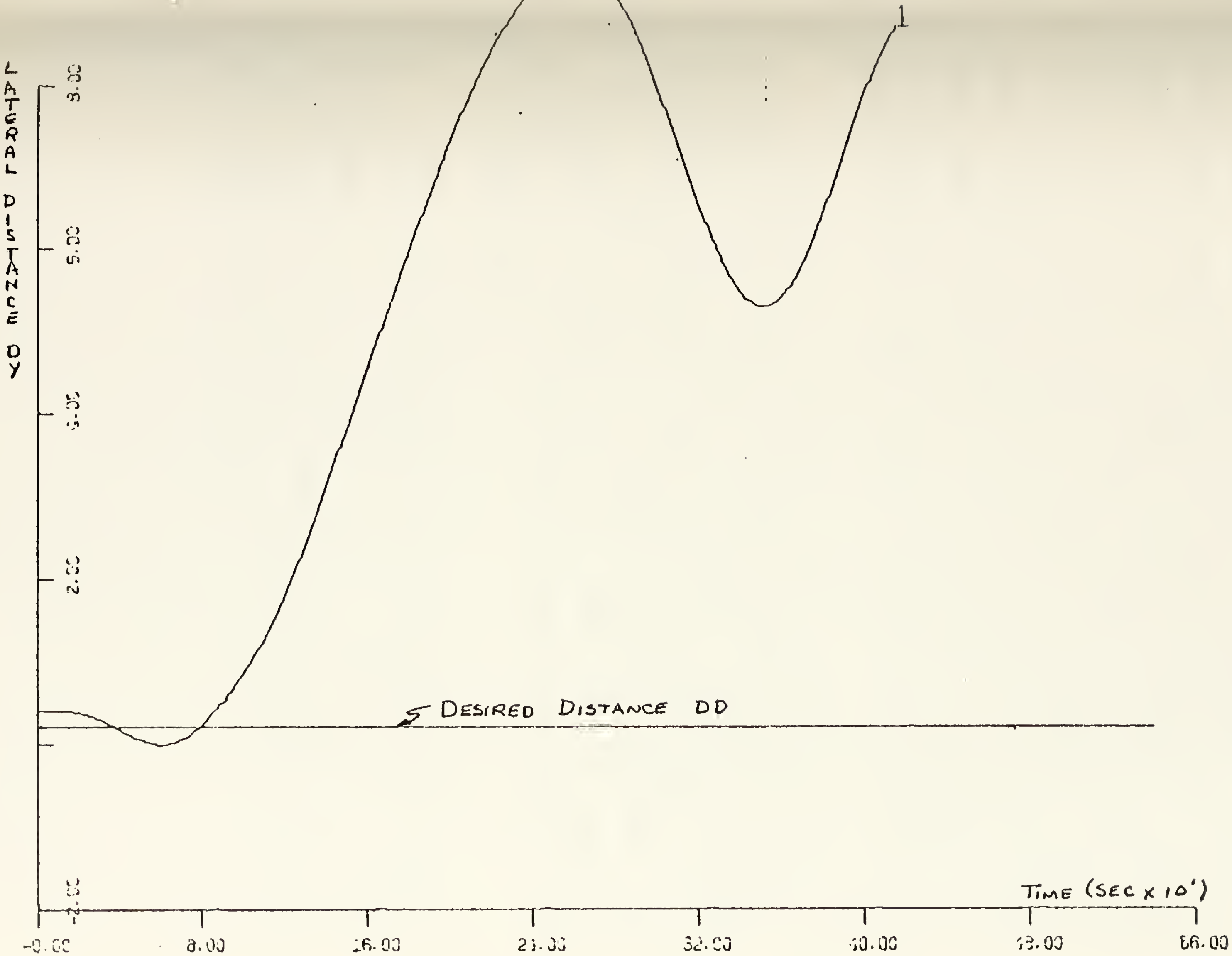


Figure III-22
Approach Phase Lateral Distance DY

and greater than 0.005 (normalized) forces acceptable steady state in approximately 1/2 the time previously required using a single set of approach gains.

Subroutine SWTCH (listed in appendix A) incorporates this simple adaptive gain schedule with a counter mechanism to sense when steady state is reached. Further study indicated a need to damp the yaw oscillations to a greater extent if the yaw velocity (BDOT2D) exceeded 2.0 degrees/sec when the gains are initially switched to the more sensitive ones of the turn phase. This is an artifical adaptive gain for VFBG caused by computer time restrictions prevalent in a full scale computer simulation where both the approach and turn phases are desired. If the gain switching point is moved up in time, as would normally be the case in a real life situation, this damping increase would not be required.

The results of the full scale simulation using computer program #6 are shown in figures III-23 thru III-34. The approach phase plots of figures III-23 thru III-28 show definite improvement over that previously shown in figures III-7 thru III-12. Figure III-27 indicates that the overshoot is reduced to 10.6 feet (0.02 normalized) as opposed to 17.9 feet (0.034 normalized) that was prevalent in figure III-11.

The turn phase plots are shown in figures III-29 thru III-34 and show responses very similar to those shown previously in figures III-16 thru III-21. The only significant differences occur in the initial responses which are due to the incorrect initialization when the turn phase was simulated individually.

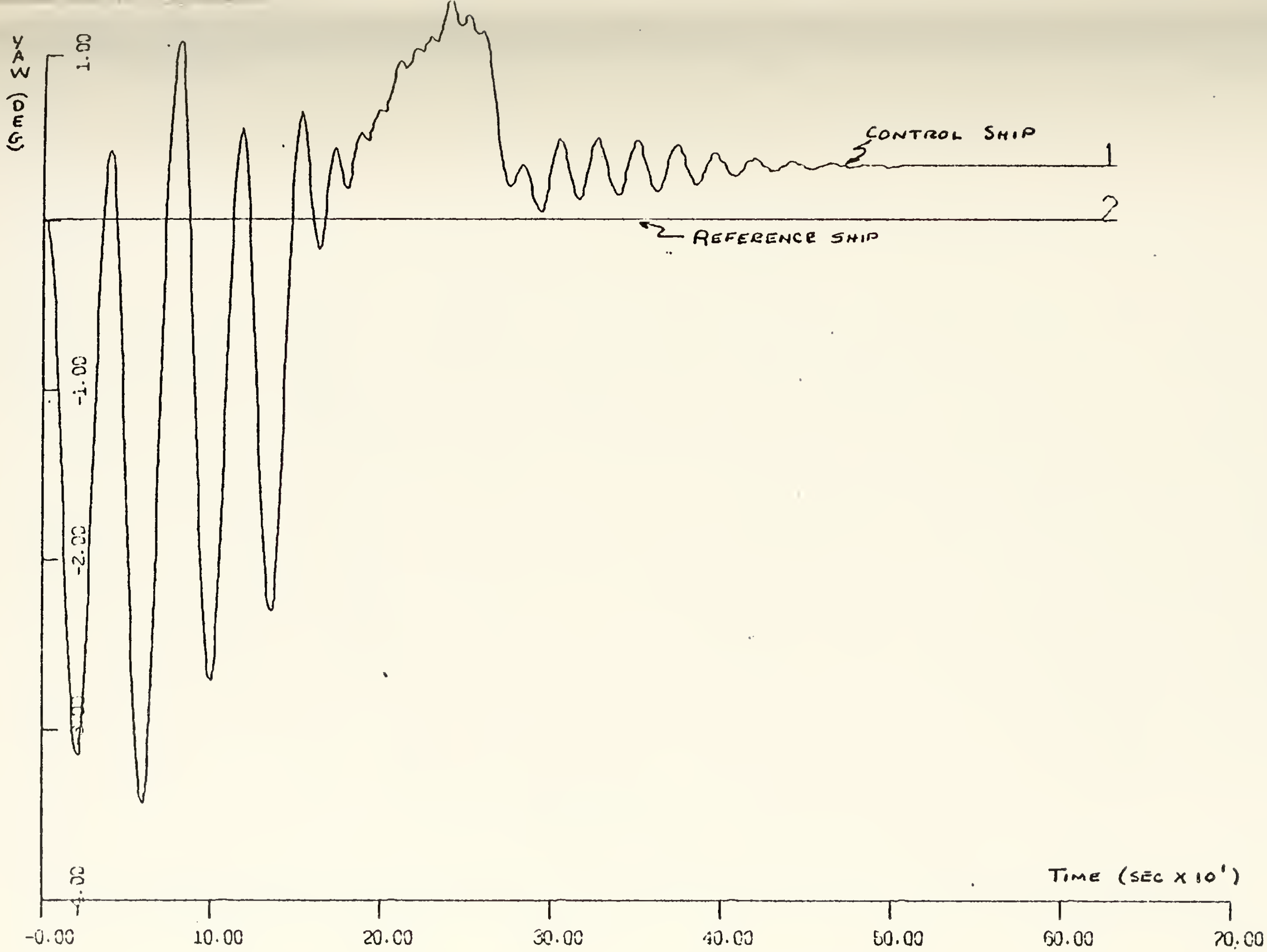


Figure III-23

Approach Phase Yaw Response

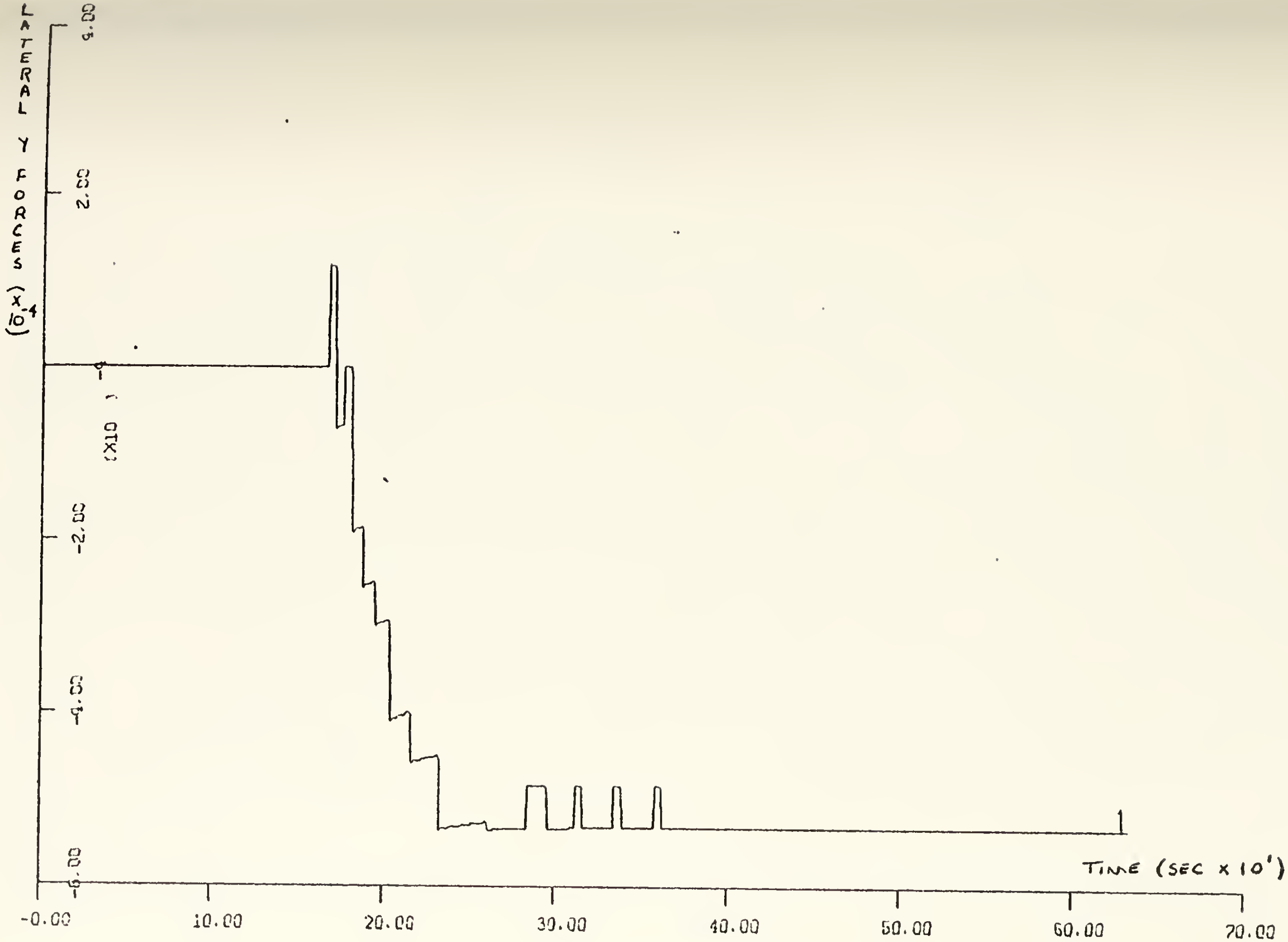


Figure III-24

Approach Phase Y Forces

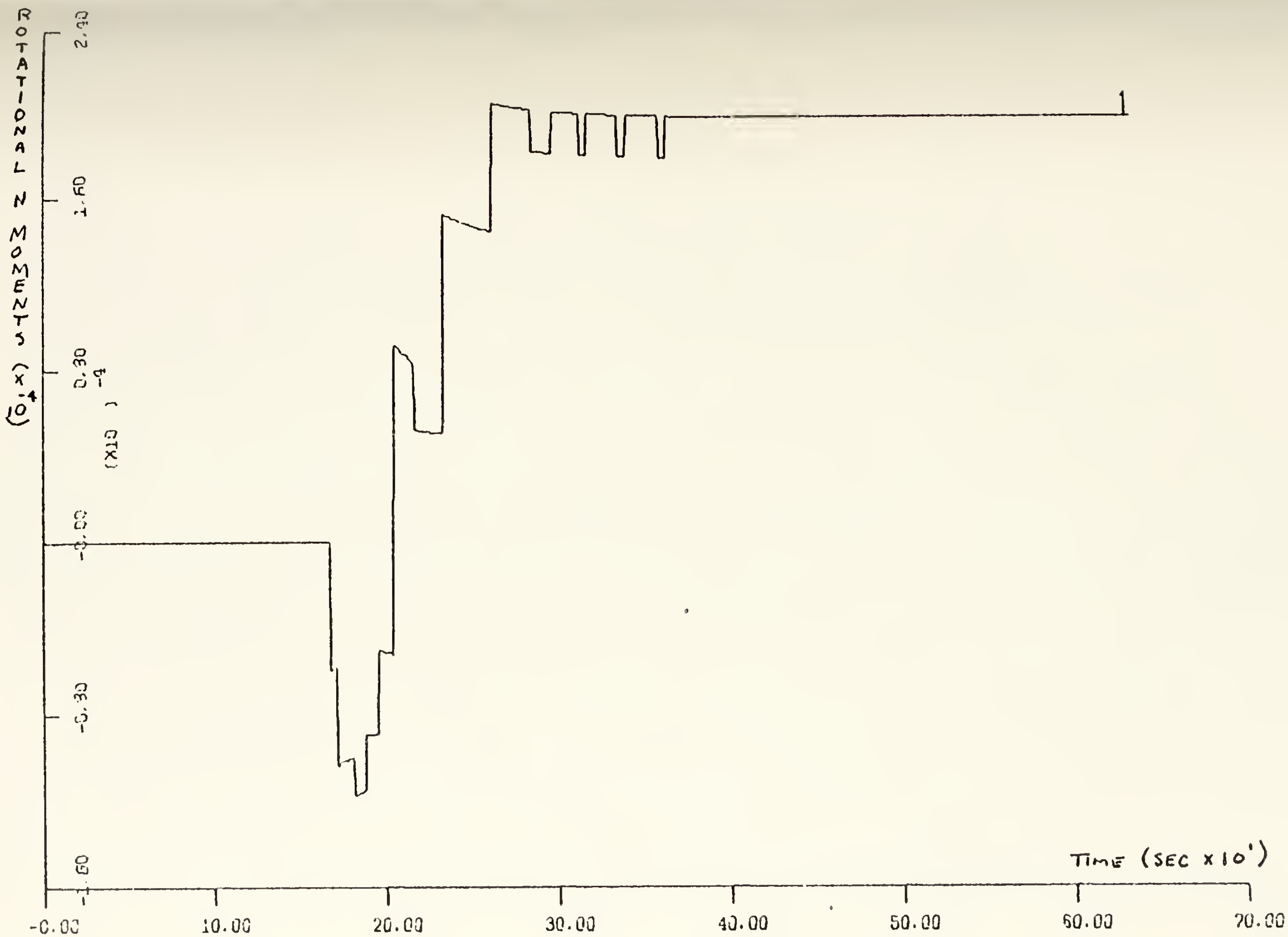


Figure III-25

Approach Phase N Moments

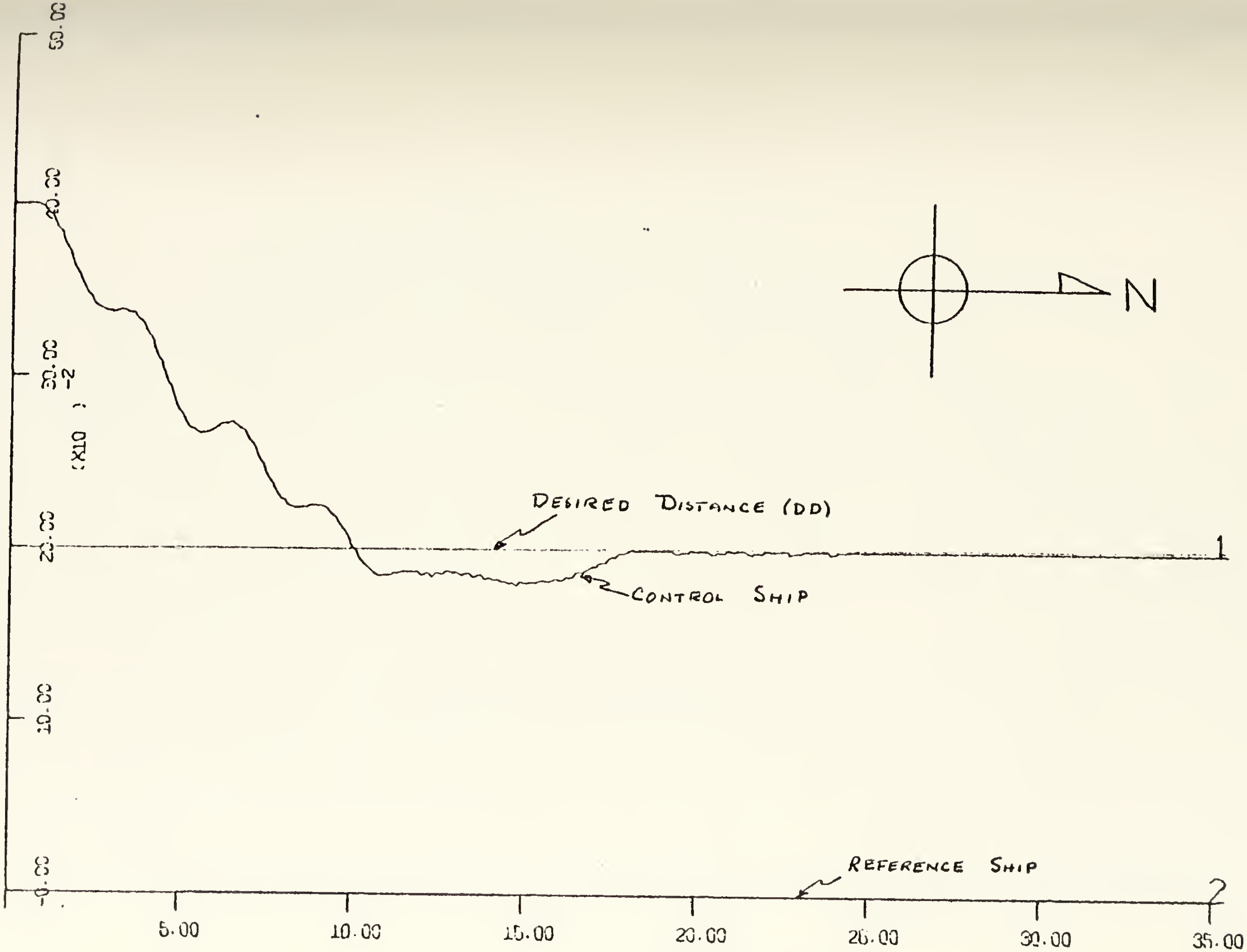


Figure III-26
Approach Phase Geographic Plot

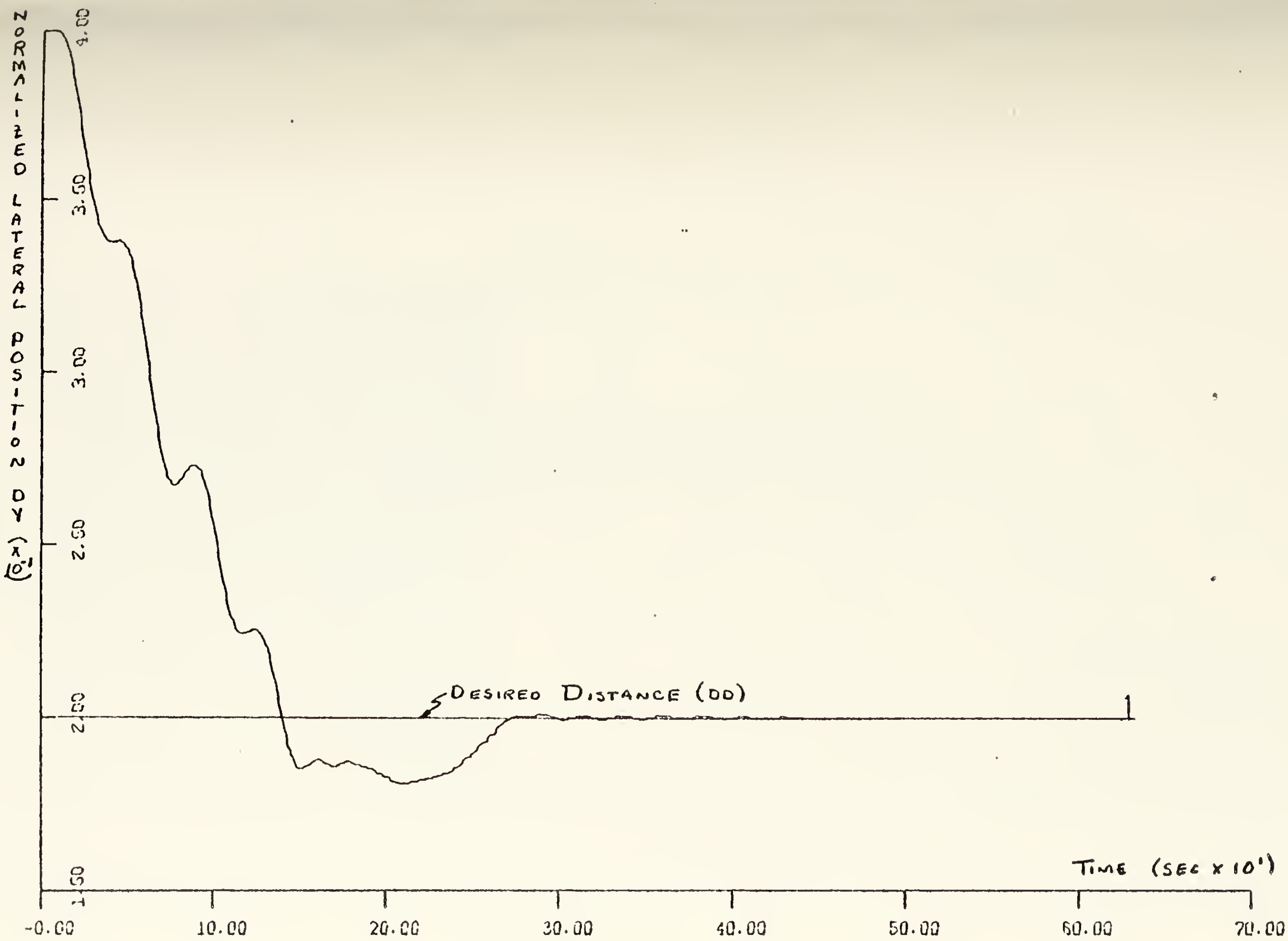


Figure III-27

Approach Phase Lateral Distance DY

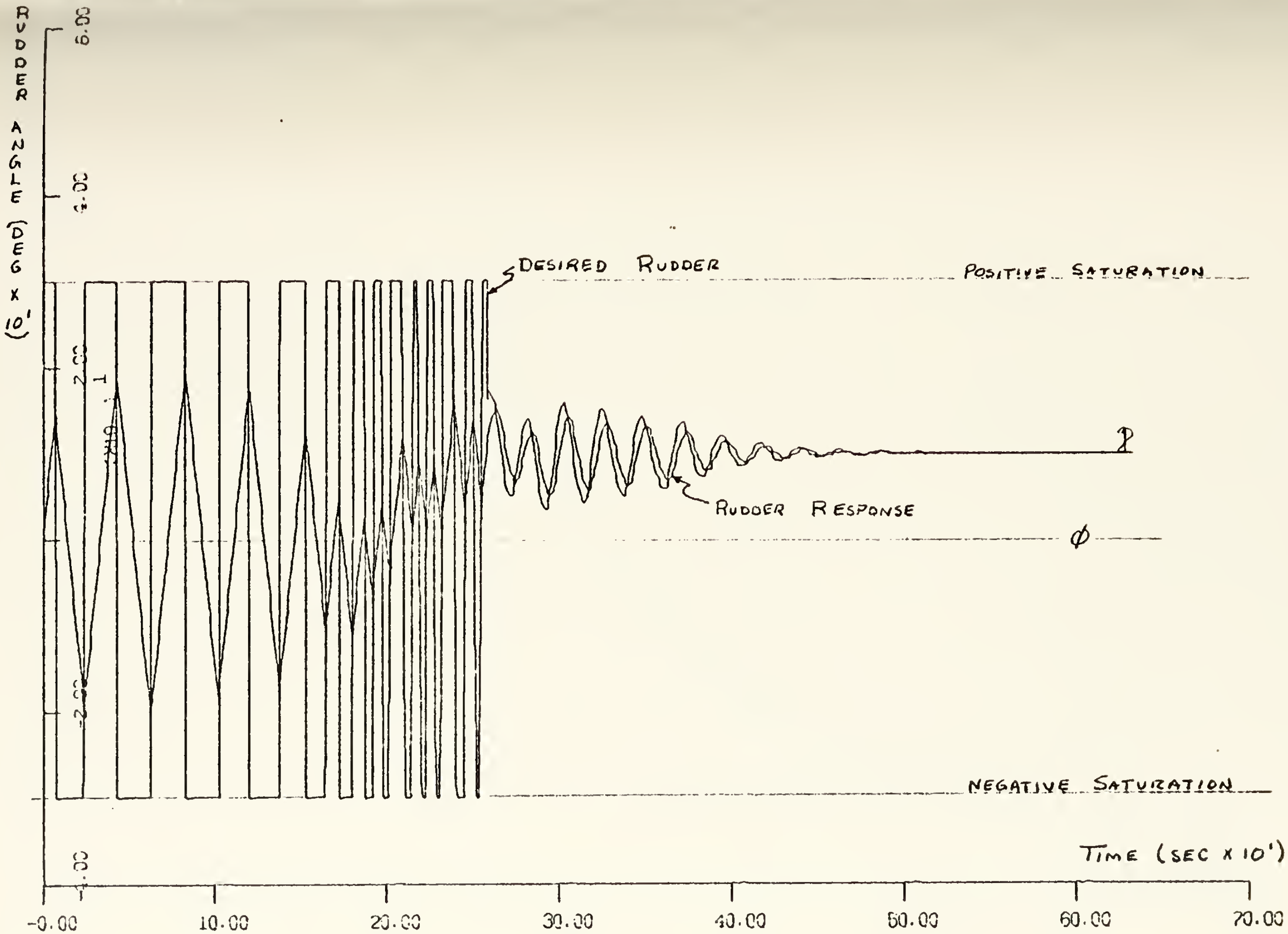


Figure III-28
Approach Phase Rudder Response

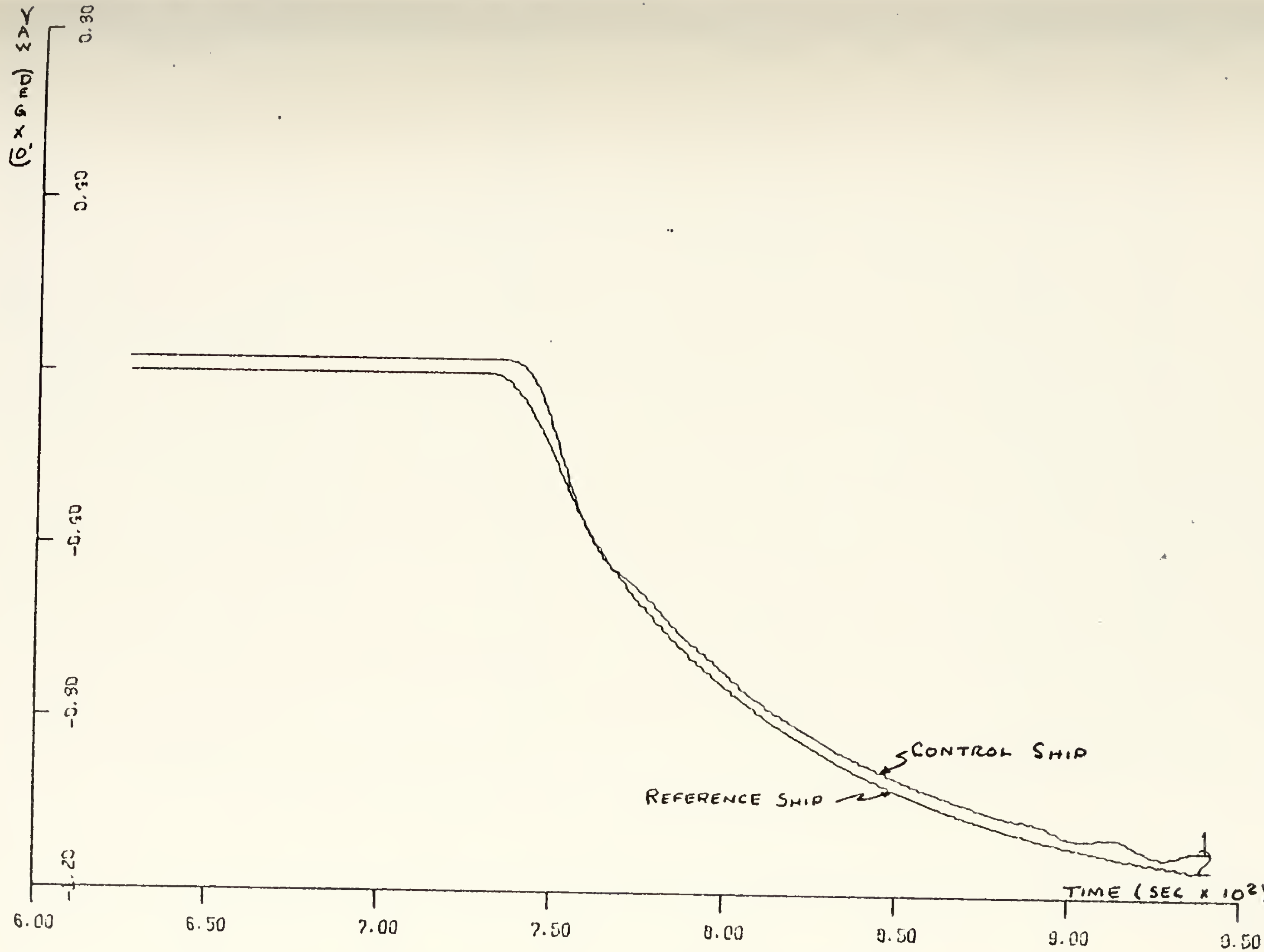


Figure III-29

Turn Phase Yaw Response

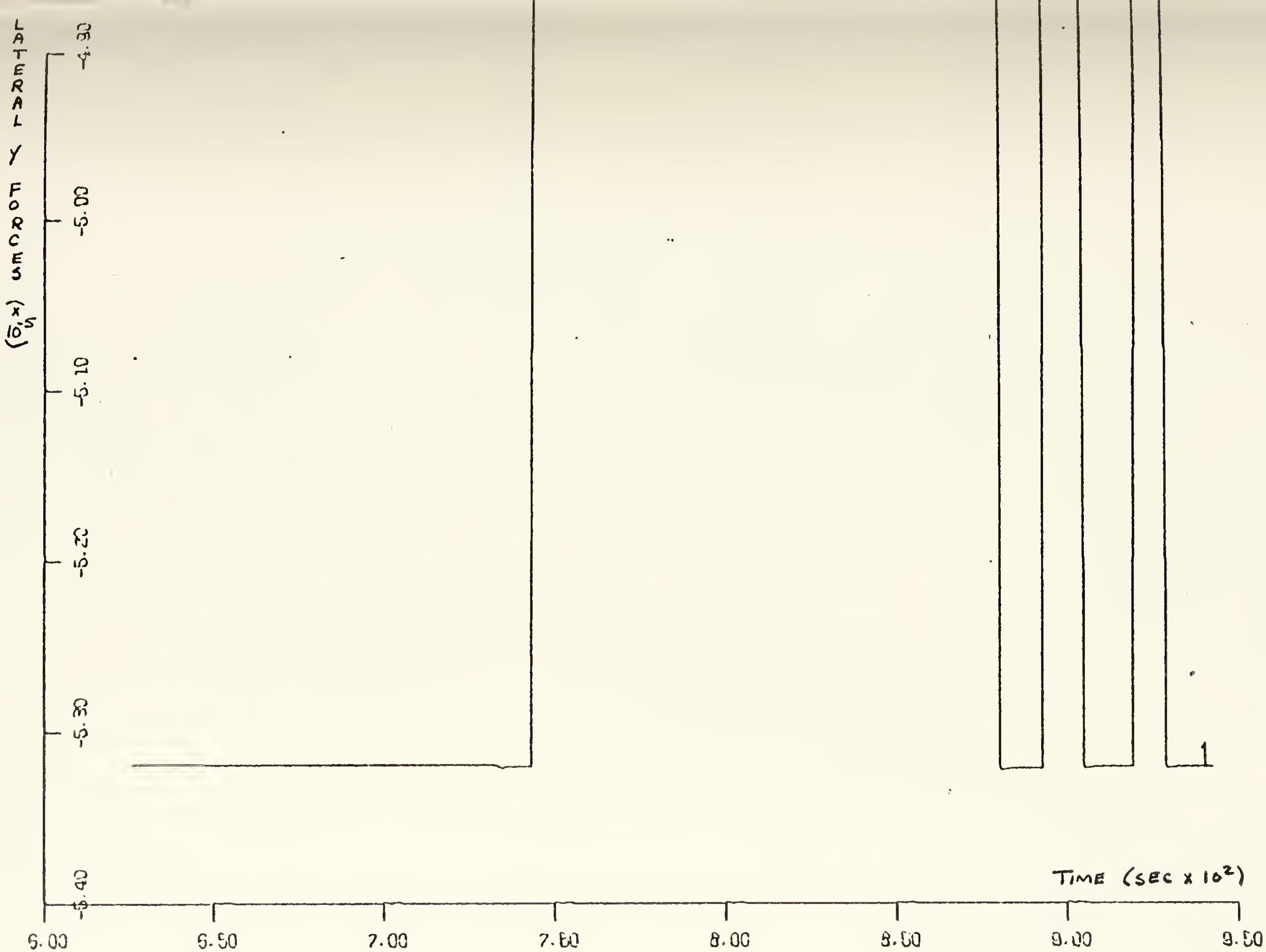


Figure III-30

Turn Phase Y Forces

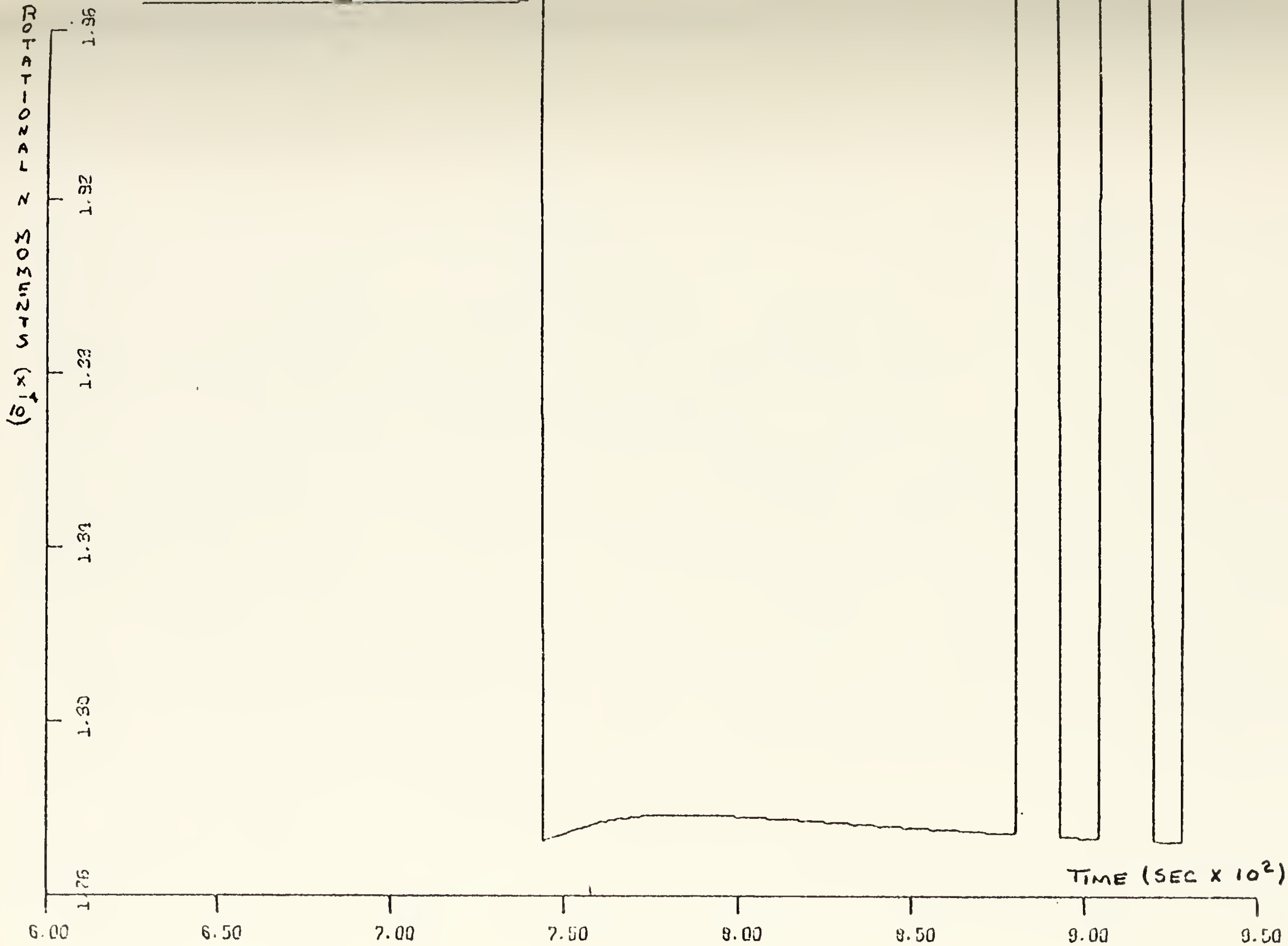


Figure III-31
Turn Phase N Moments

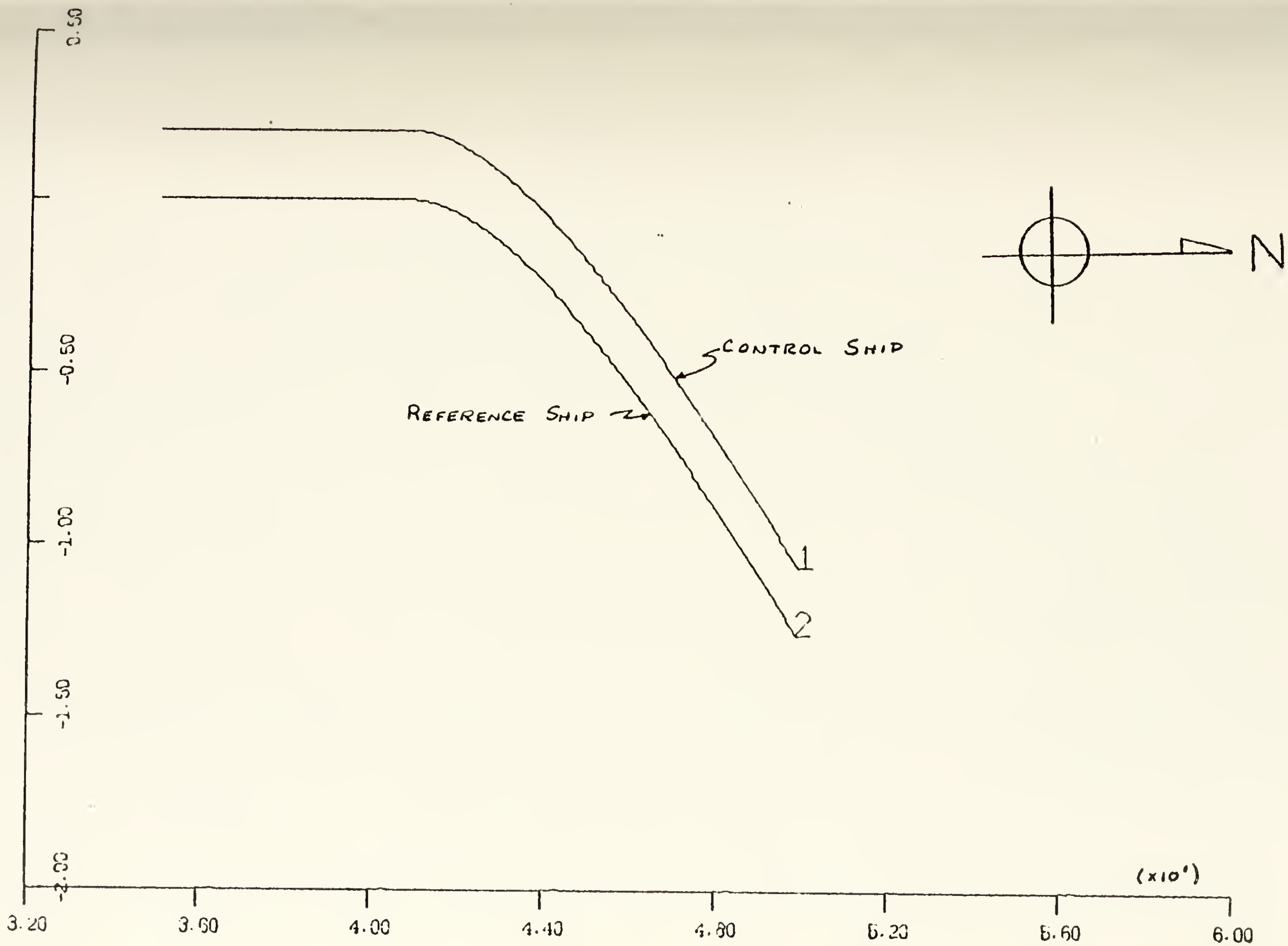


Figure III-32
Turn Phase Geographic Plot

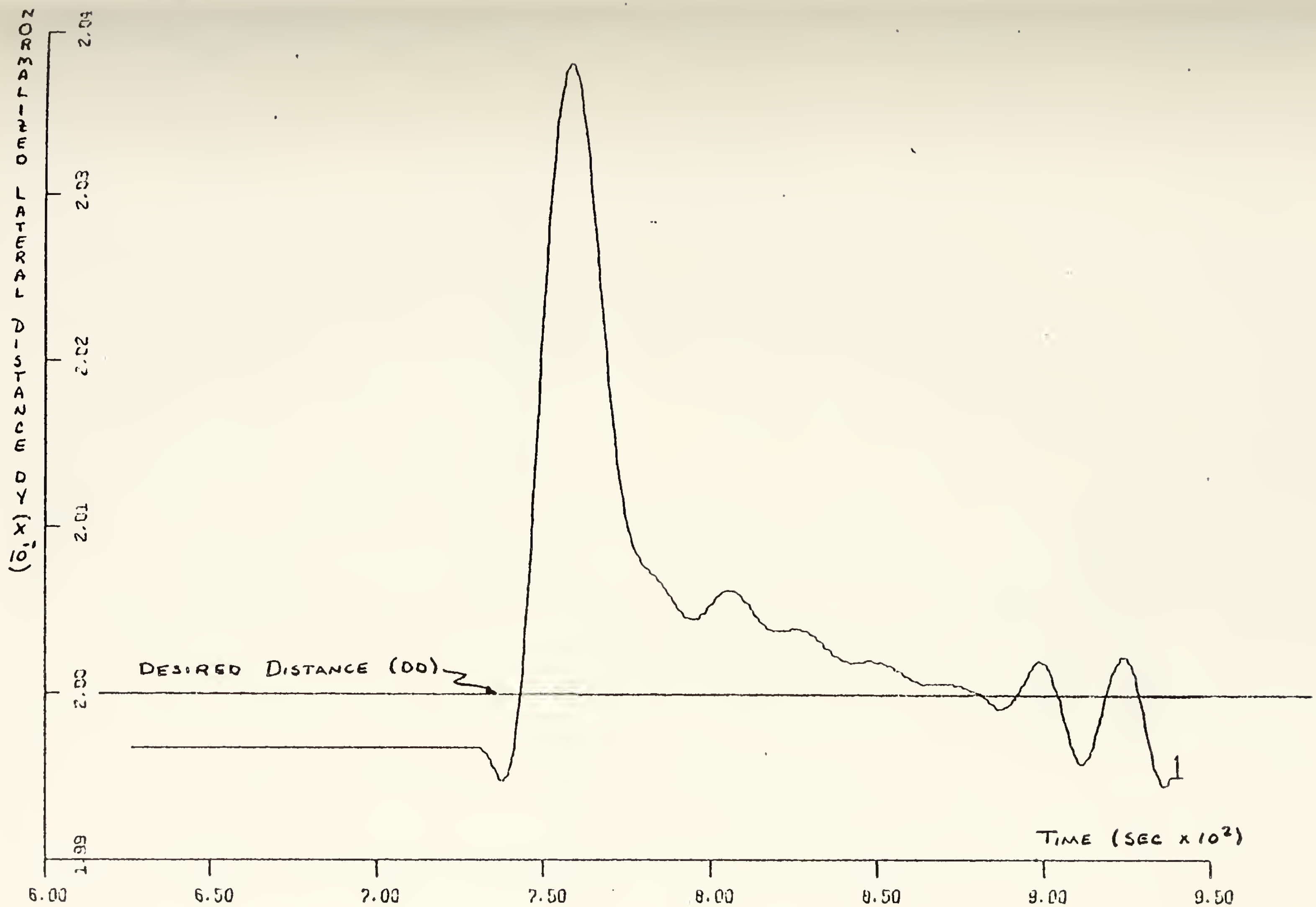


Figure III-33

Turn Phase Lateral Distance DY

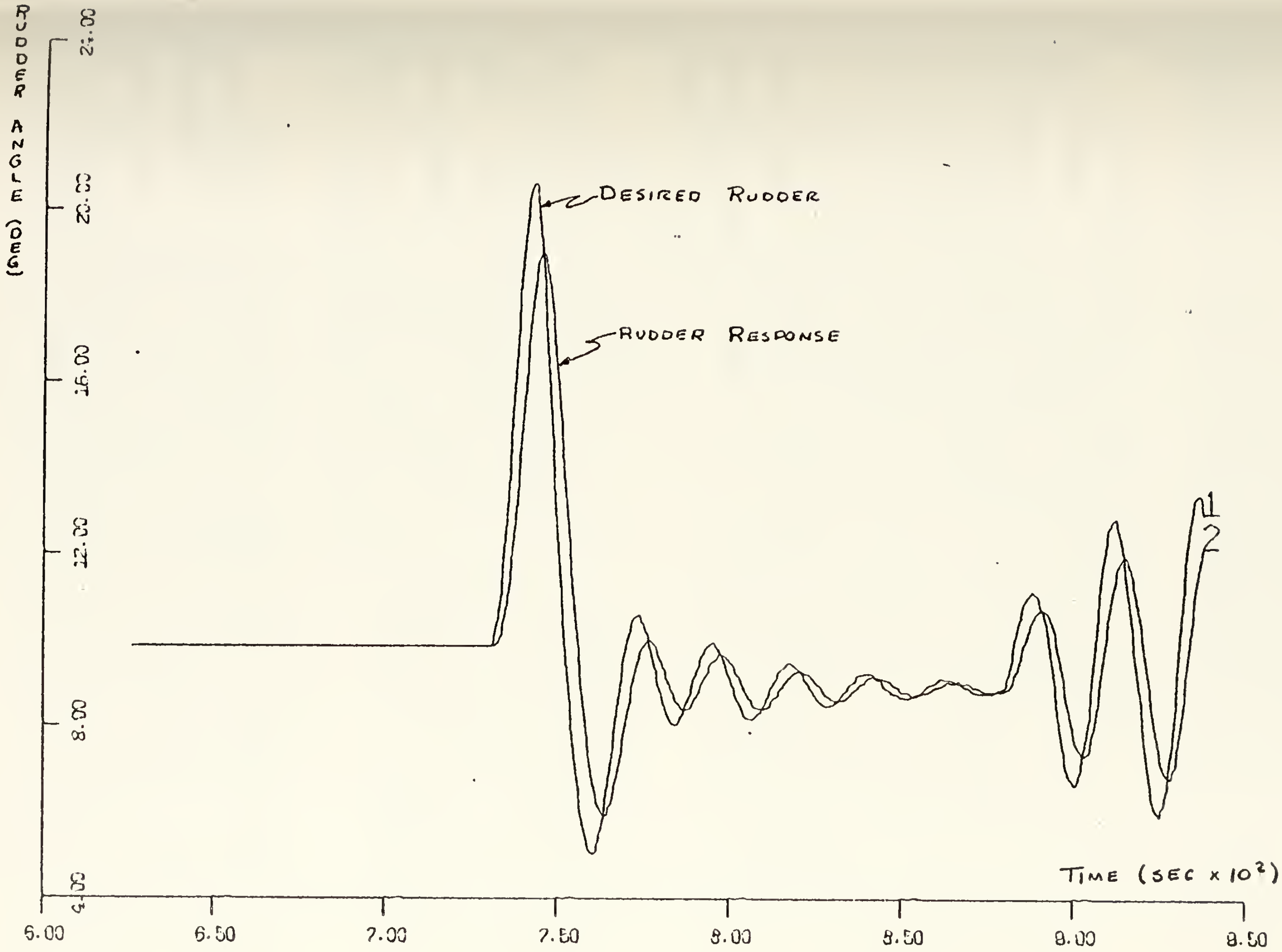


Figure III-34
Turn Phase Rudder Response

g. Varying Initial Conditions

The results obtained in the previous section are most gratifying but actually incomplete. This system must work for other initial conditions quite different from those envisioned in the optimization scenario. The initial approach can realistically commence at points other than 5 ship lengths astern and displaced by 0.4 ship lengths.

By simulating this system with varying initial positions, the relative efficiency and worth of the control system can be observed. This was done in successive test runs whose initial conditions and corresponding plot figures are tabulated in table III-3. For the sake of brevity only those figures required to illustrate the relative efficiency of the control system are included. The corresponding initial optimization simulation figures are listed for cross reference. The turn phase plots for all runs except 4 and 6 exactly match that of the initial simulation and are not repeated here.

Runs 3, 5 and 6 were accomplished to show that no ambiguities exist in the control scheme to prohibit adequate real life initial conditions. Run 3 simulates the situation most often encountered by this author in the RAS environment. This scenario starts the control ship dead astern at 5.0 ship lengths and brings it alongside at 0.2 ship lengths lateral separation.

Run 5 is a situation where the approaching ship is purposely placed out of position on the wrong side for approach. The control scheme adequately corrects the placement error and will do so for all cases of this type, provided that there is adequate maneuvering room astern of the reference vessel (in this case 2.6 ship lengths was

	RUN	Initial Development	1	2	3	4	5	6
Initial Condition	X01	5.0	4.0	3.0	5.0	5.0	5.0	5.0
	Y02	0.4	0.3	0.25	0.0	0.2	-0.4	-0.4
Desired Distance	DD	0.2	0.2	0.2	0.2	0.15	0.2	0.2
Approach Side	IS	STBD	STBD	STBD	STBD	STBD	STBD	PORT
Approach Phase Figures (III-)								
YAW		23	35	38	41	44	47	50
Geographic Plot		26	36	39	42	45	48	51
Rudder Response		28	37	40	43	46	49	52
Turn Phase Figures (III-)								
YAW		29				53	29	56
Geographic Plot		32				54	32	57
Rudder Response		34				55	34	58

Table III-3

Initial Condition Simulation Cross Reference

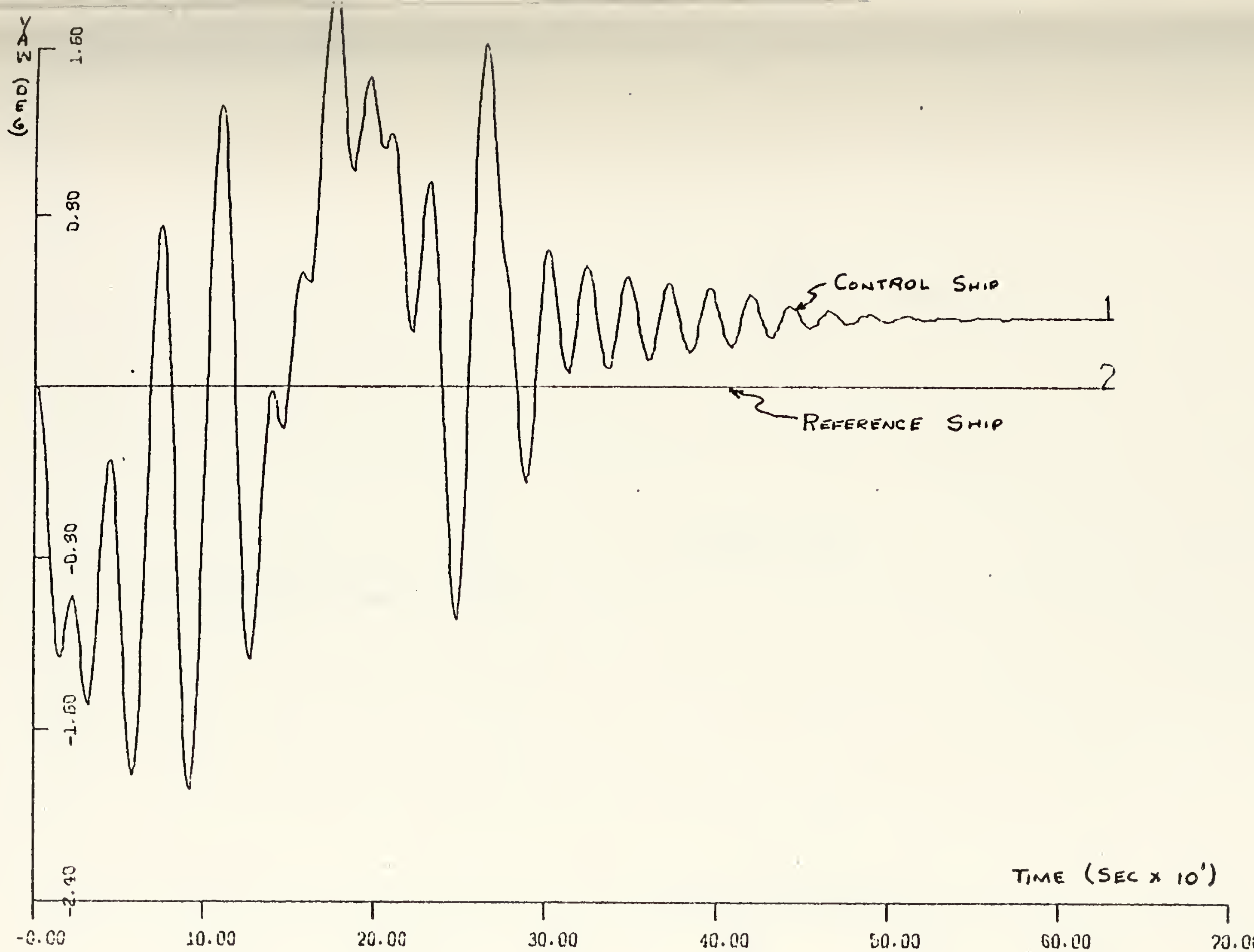


Figure III-35

Approach Phase Run #1 Yaw Response

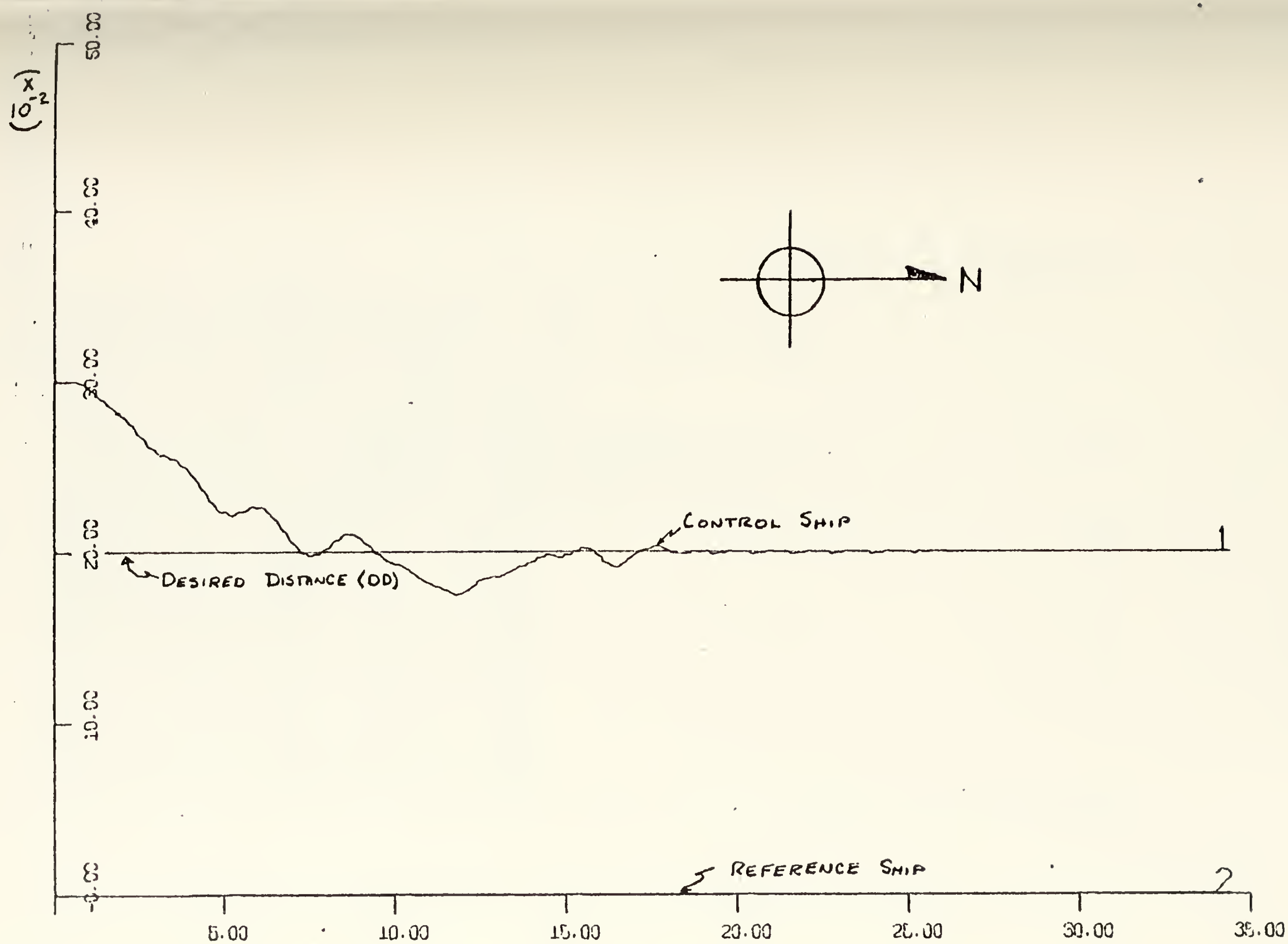


Figure III-36

Approach Phase Run #1 Geographic Plot

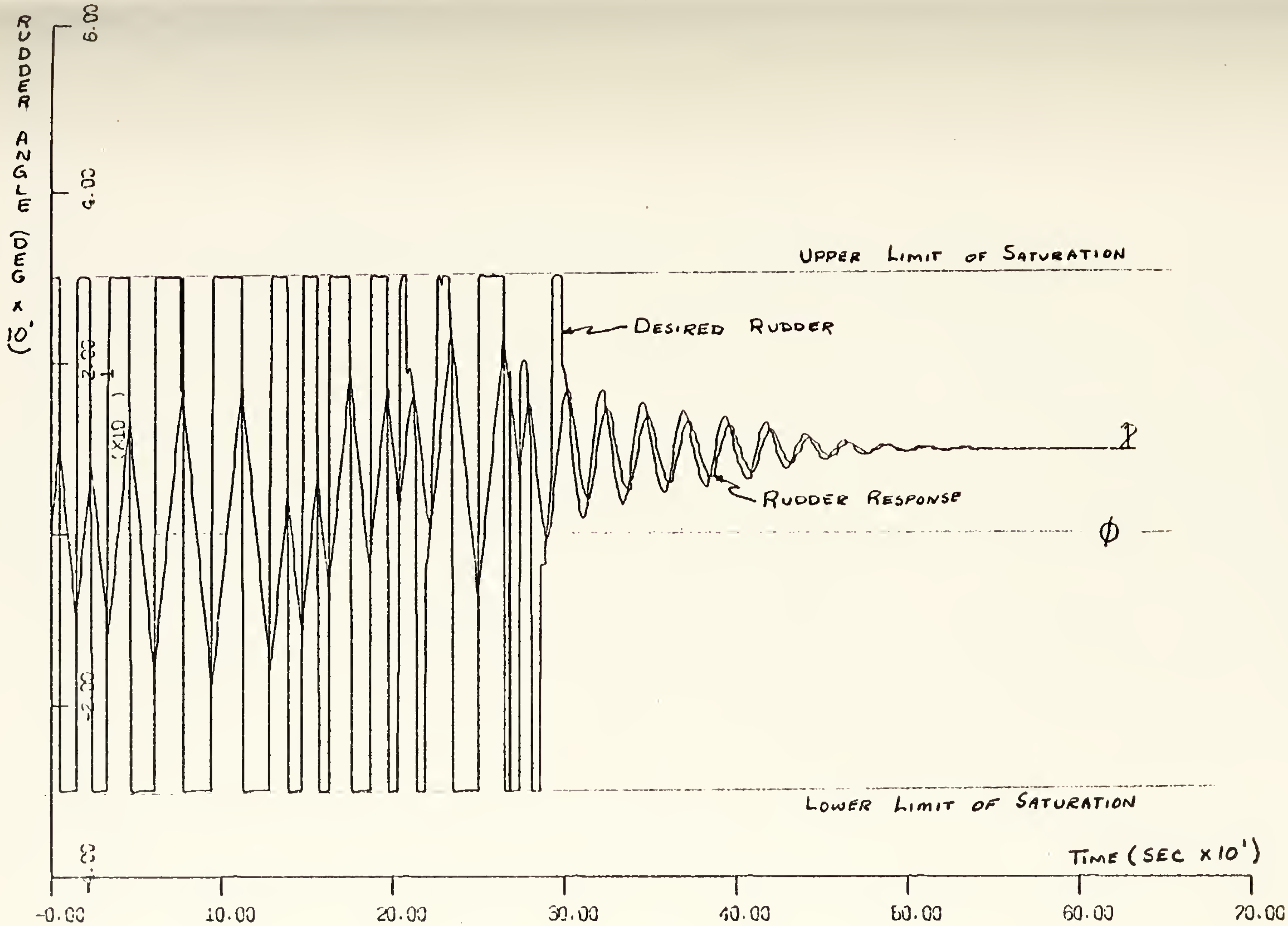


Figure III-37

Approach Phase Run #1 Rudder Response

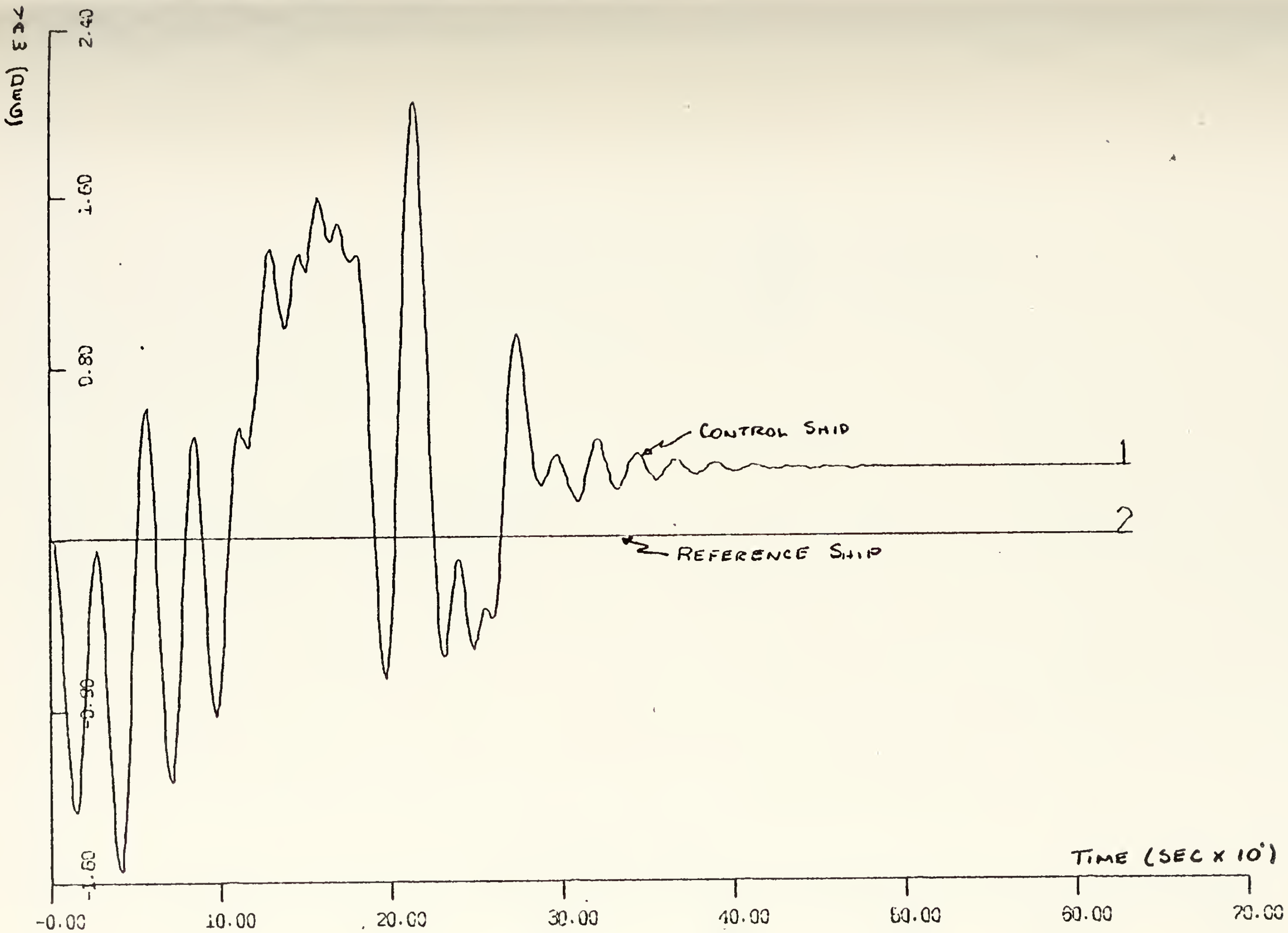


Figure III-38

Approach Phase Run #2 Yaw Response

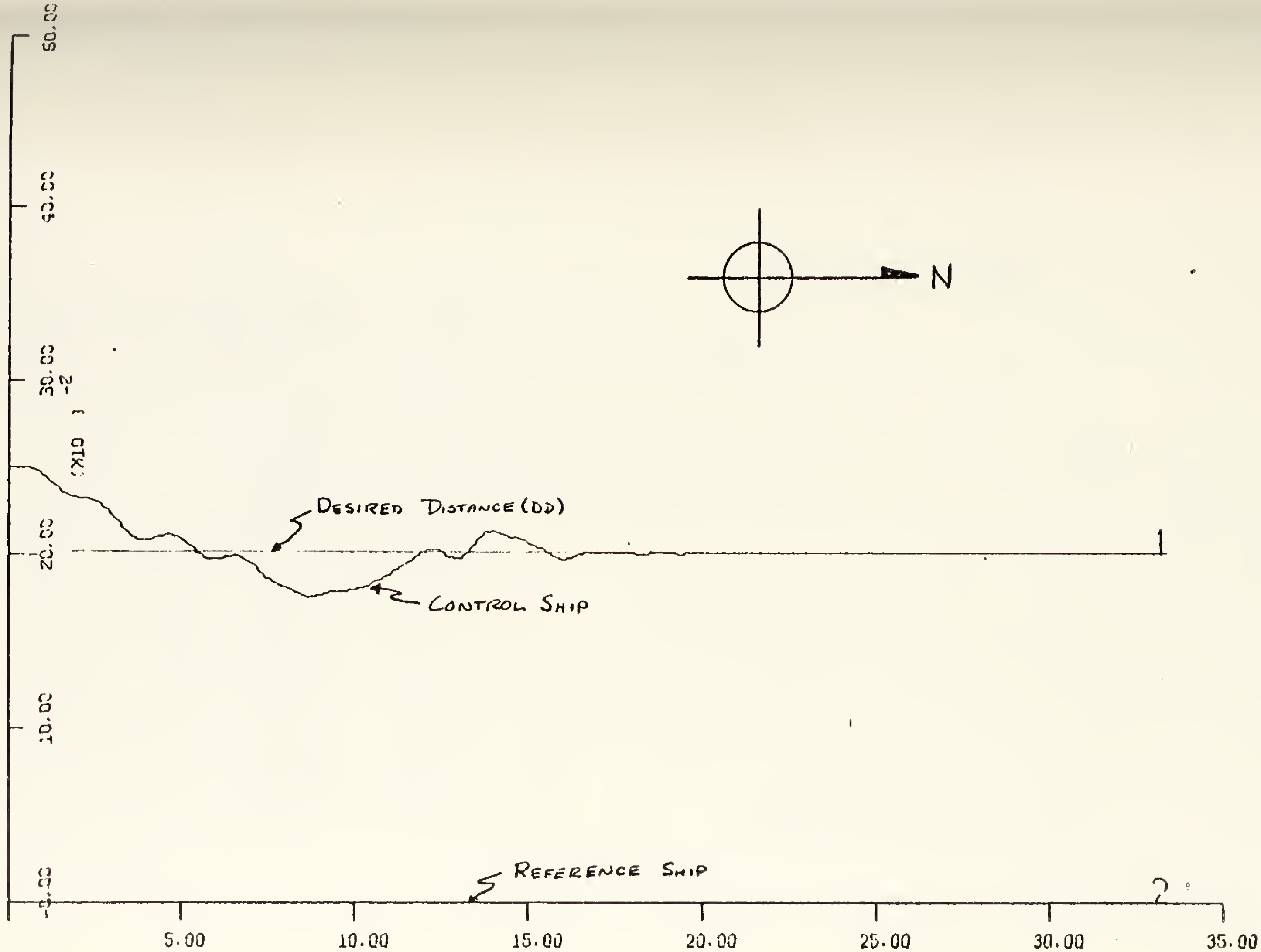


Figure III-39
Approach Phase Run #2 Geographic Plot

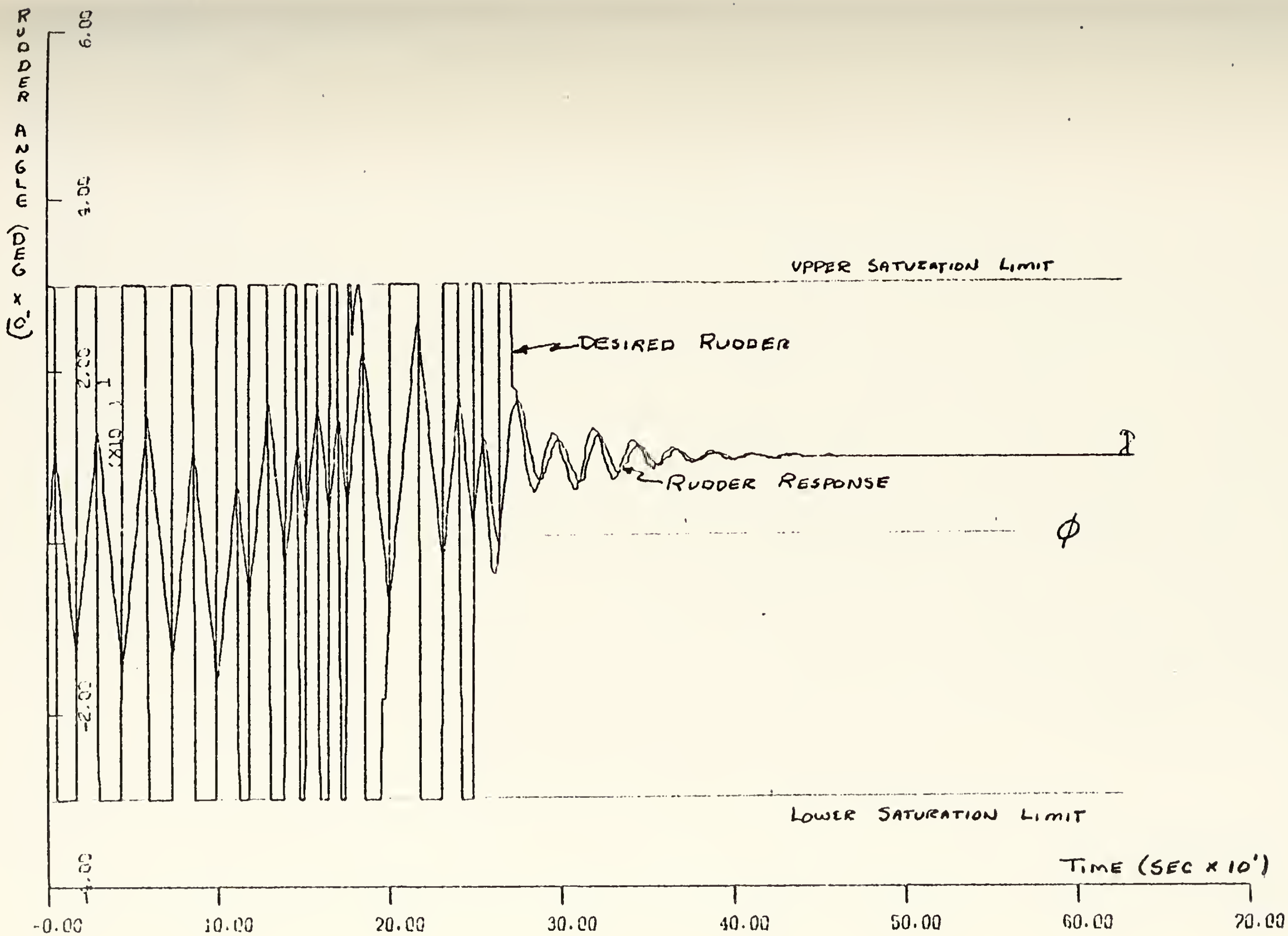


Figure III-40

Approach Phase Run #2 Rudder Response

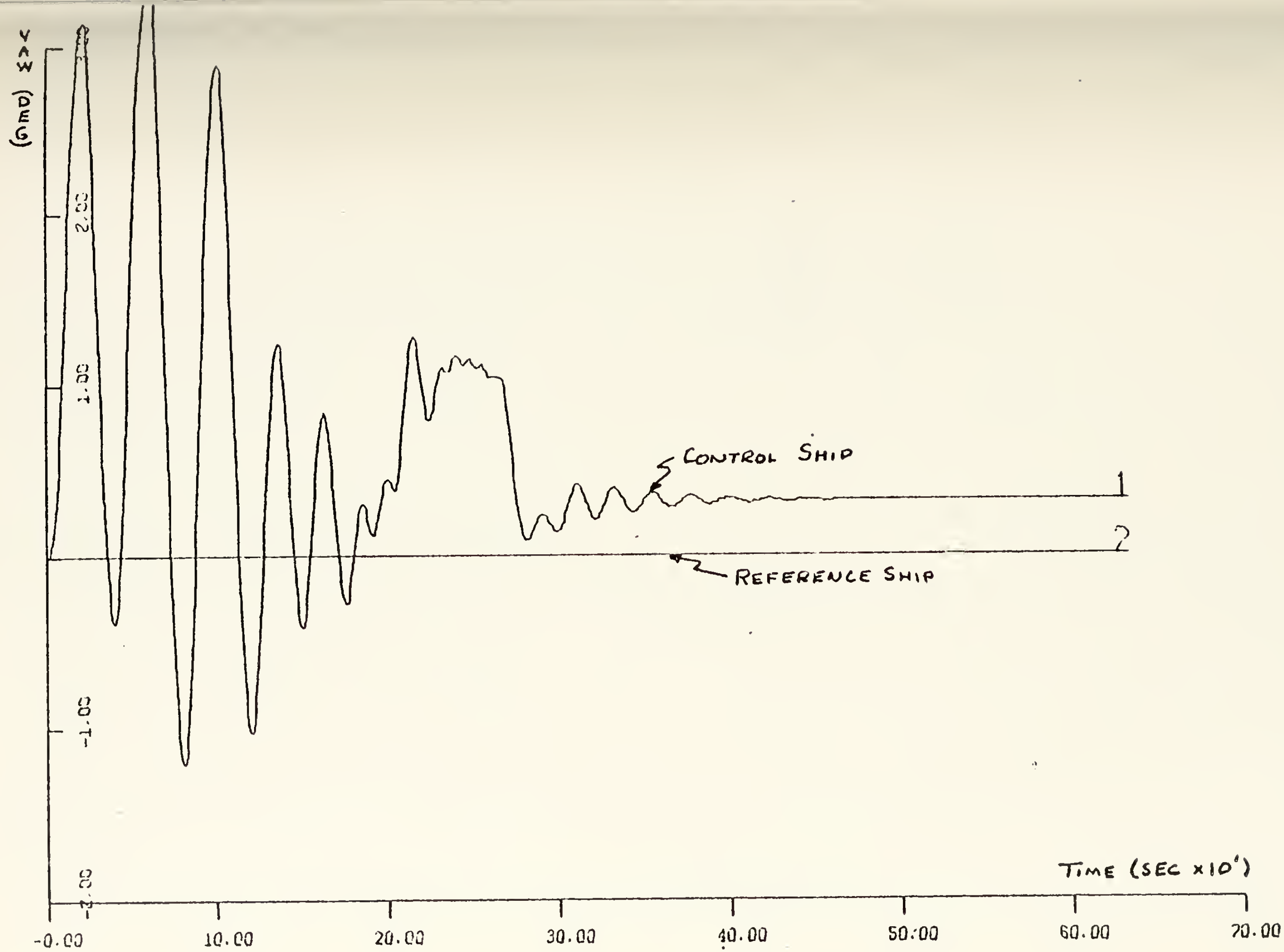


Figure III-41

Approach Phase Run #3 Yaw Response

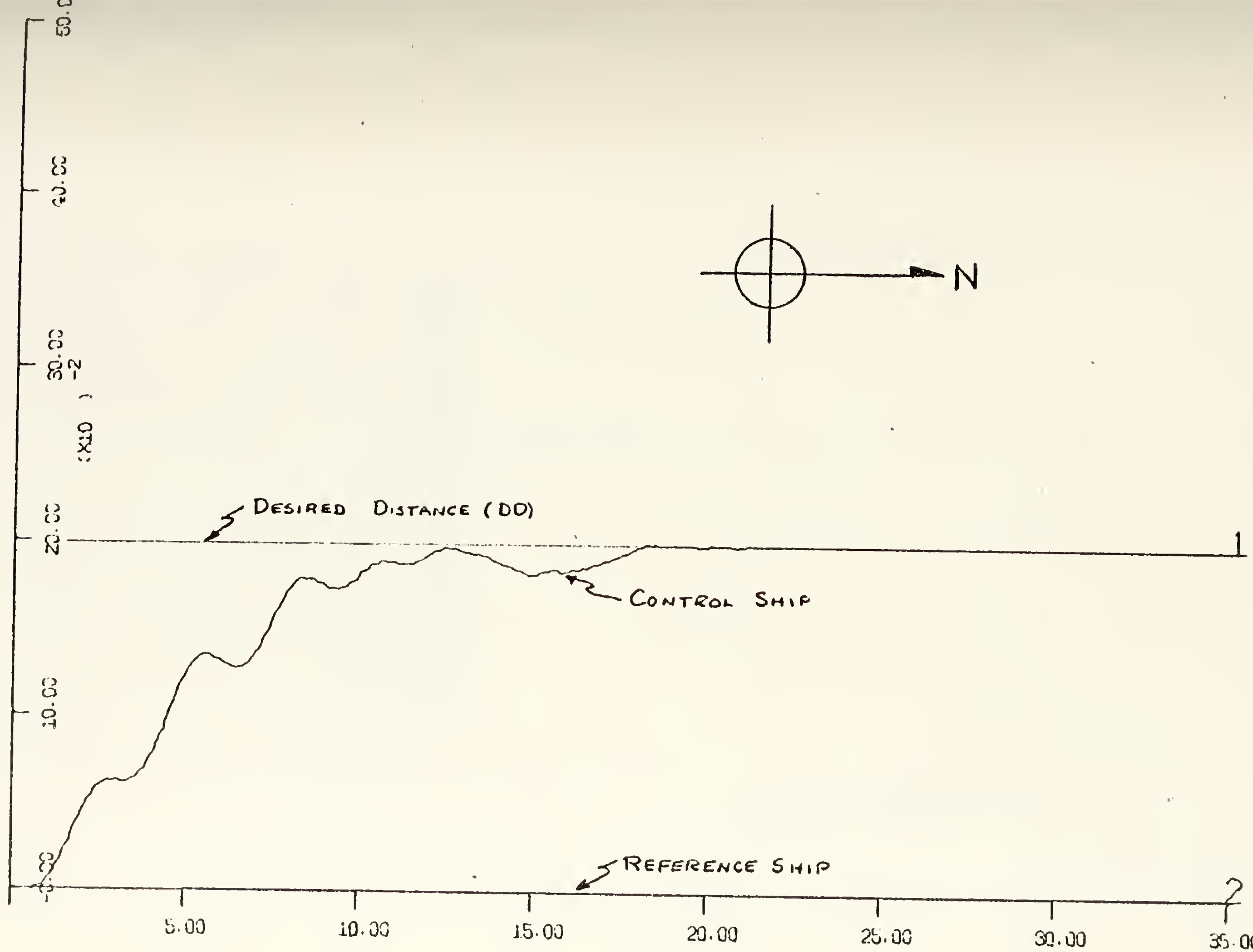


Figure III-42
Approach Phase Run #3 Geographic Plot

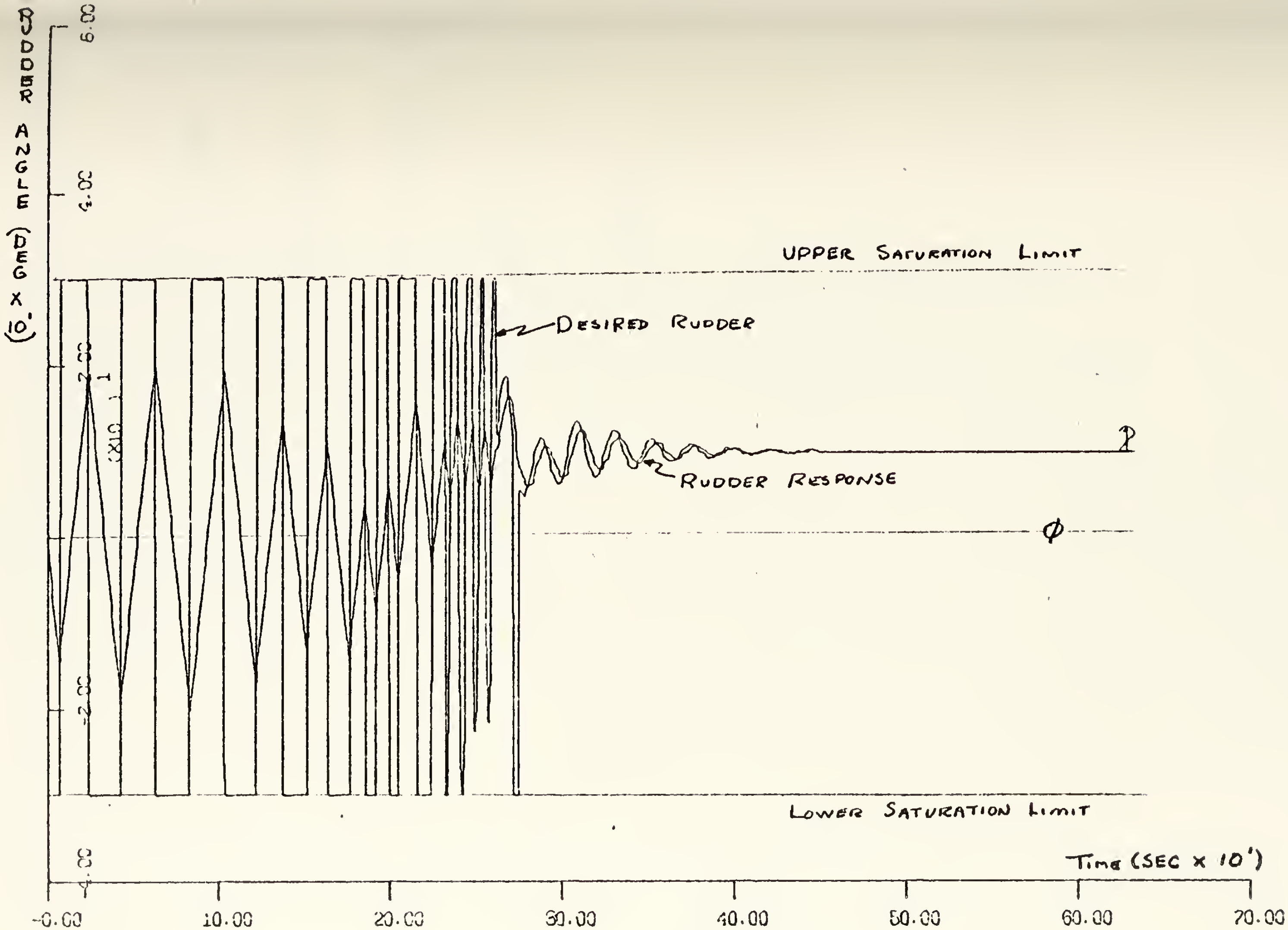


Figure III-43

Approach Phase Run #3 Rudder Response

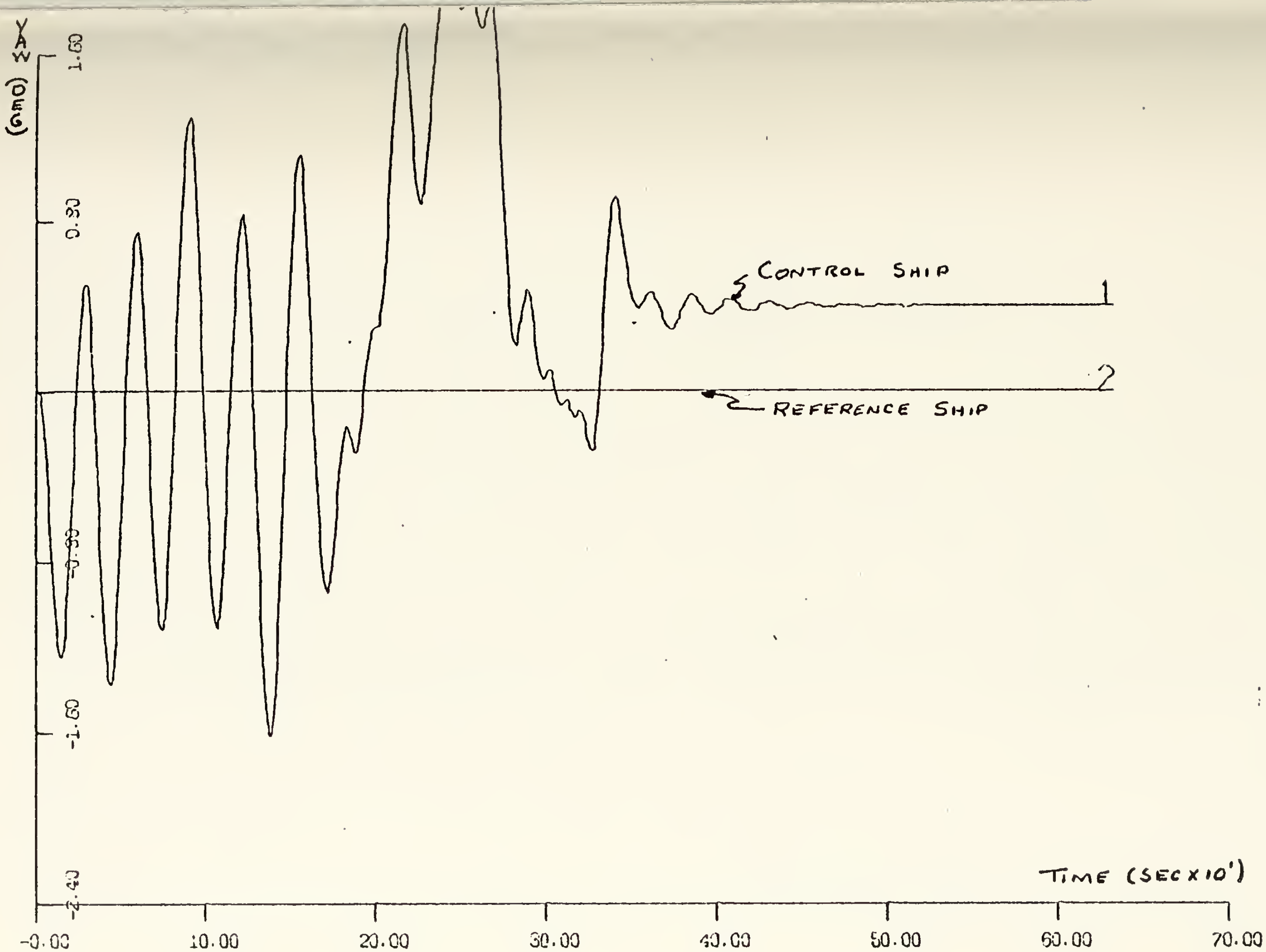


Figure III-44

Approach Phase Run #4 Yaw Response

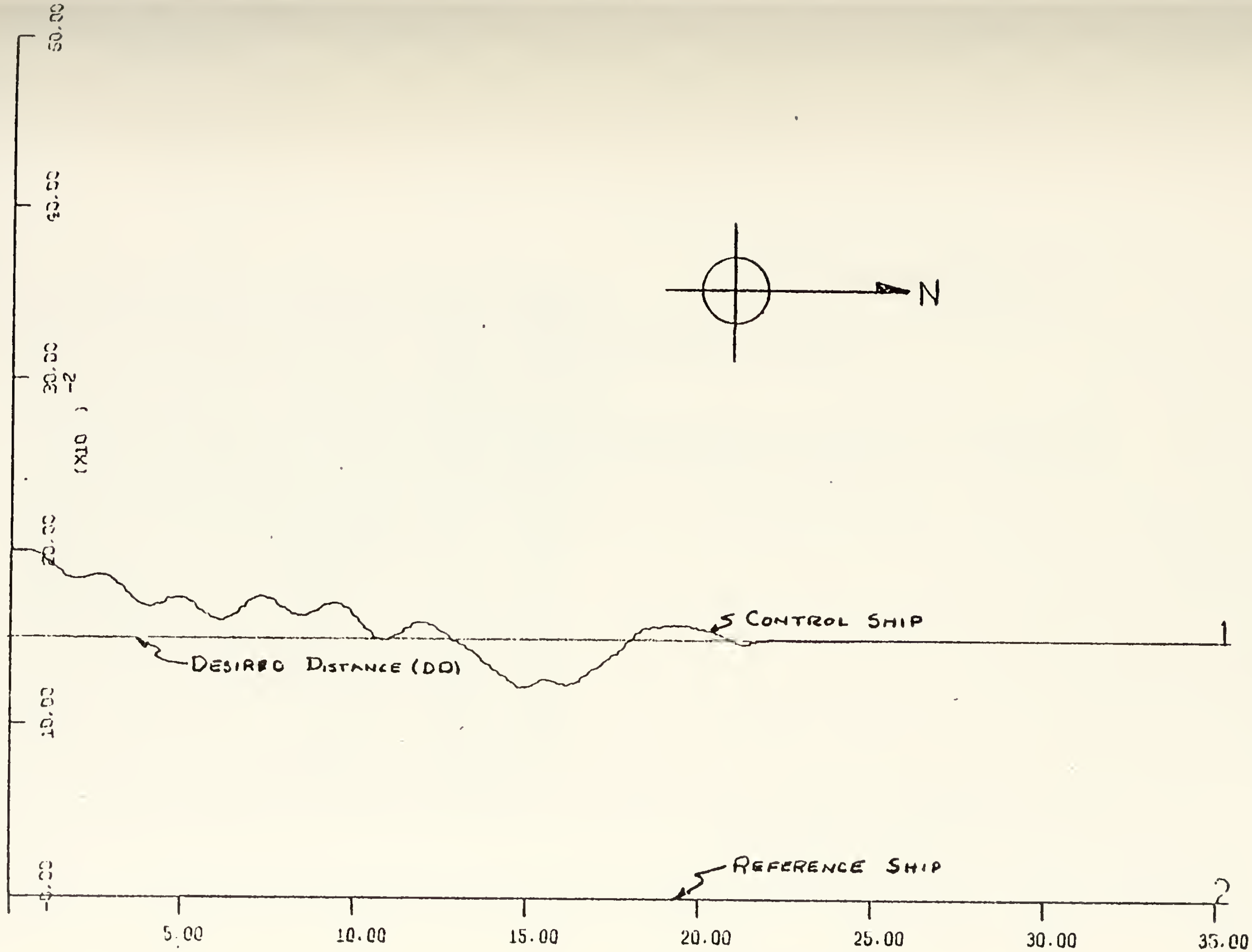


Figure III-45

Approach Phase Run #4 Geographic Plot

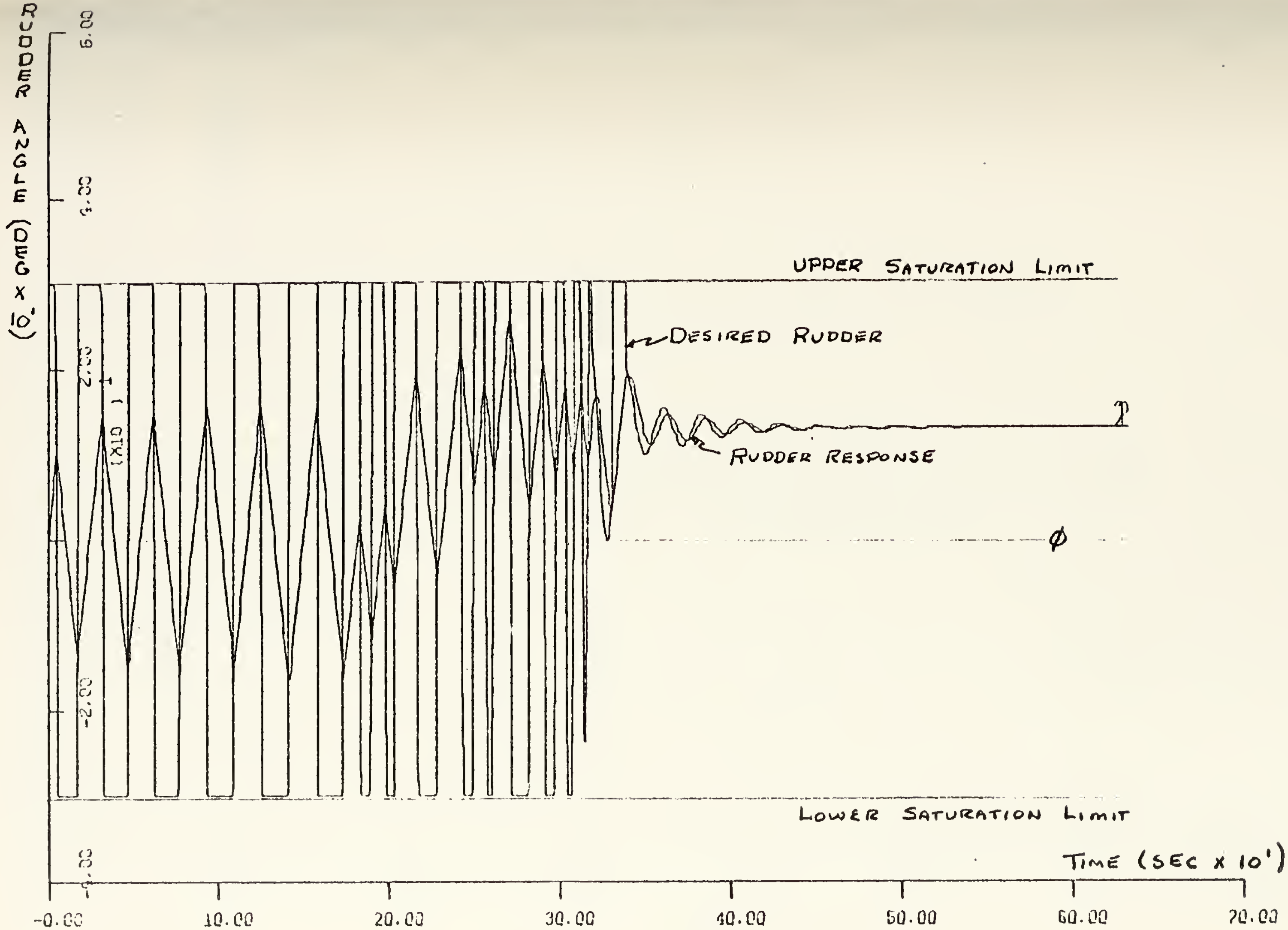


Figure III-46

Approach Phase Run #4 Rudder Response

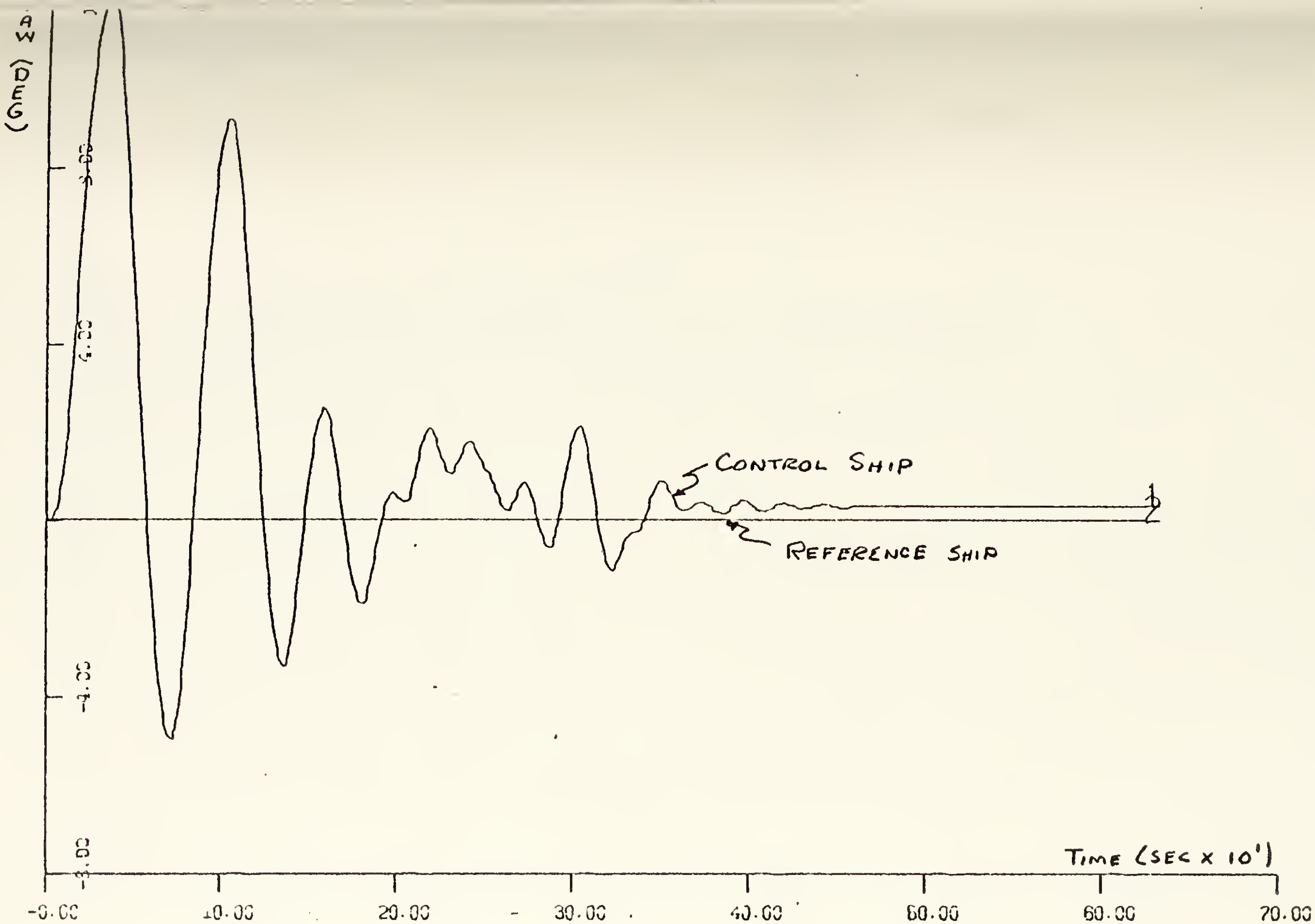


Figure III-47
Approach Phase Run #5 Yaw Response

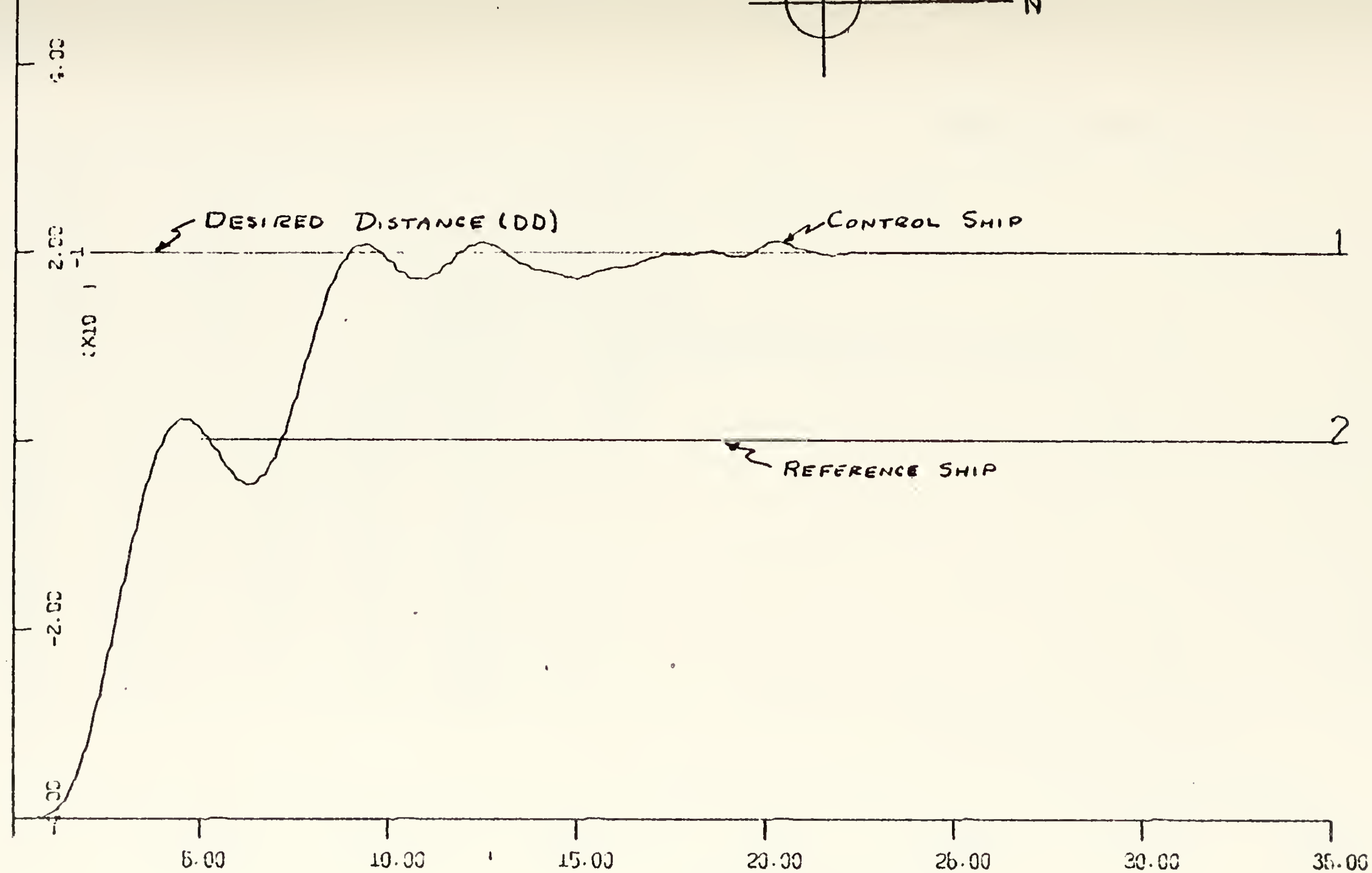


Figure III-48

Approach Phase Run #5 Geographic Plot

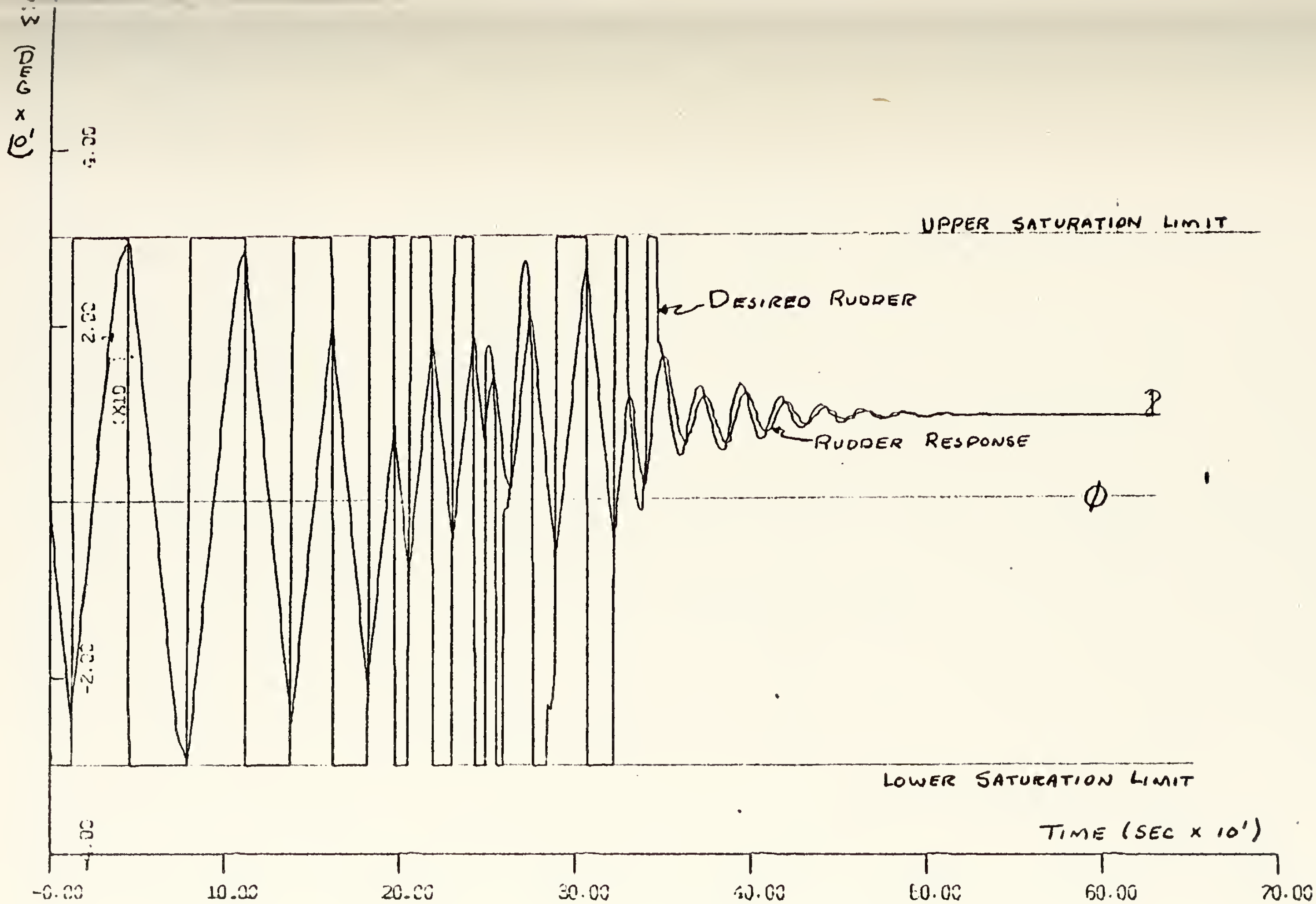


Figure III-49

Approach Phase Run #5 Rudder Response

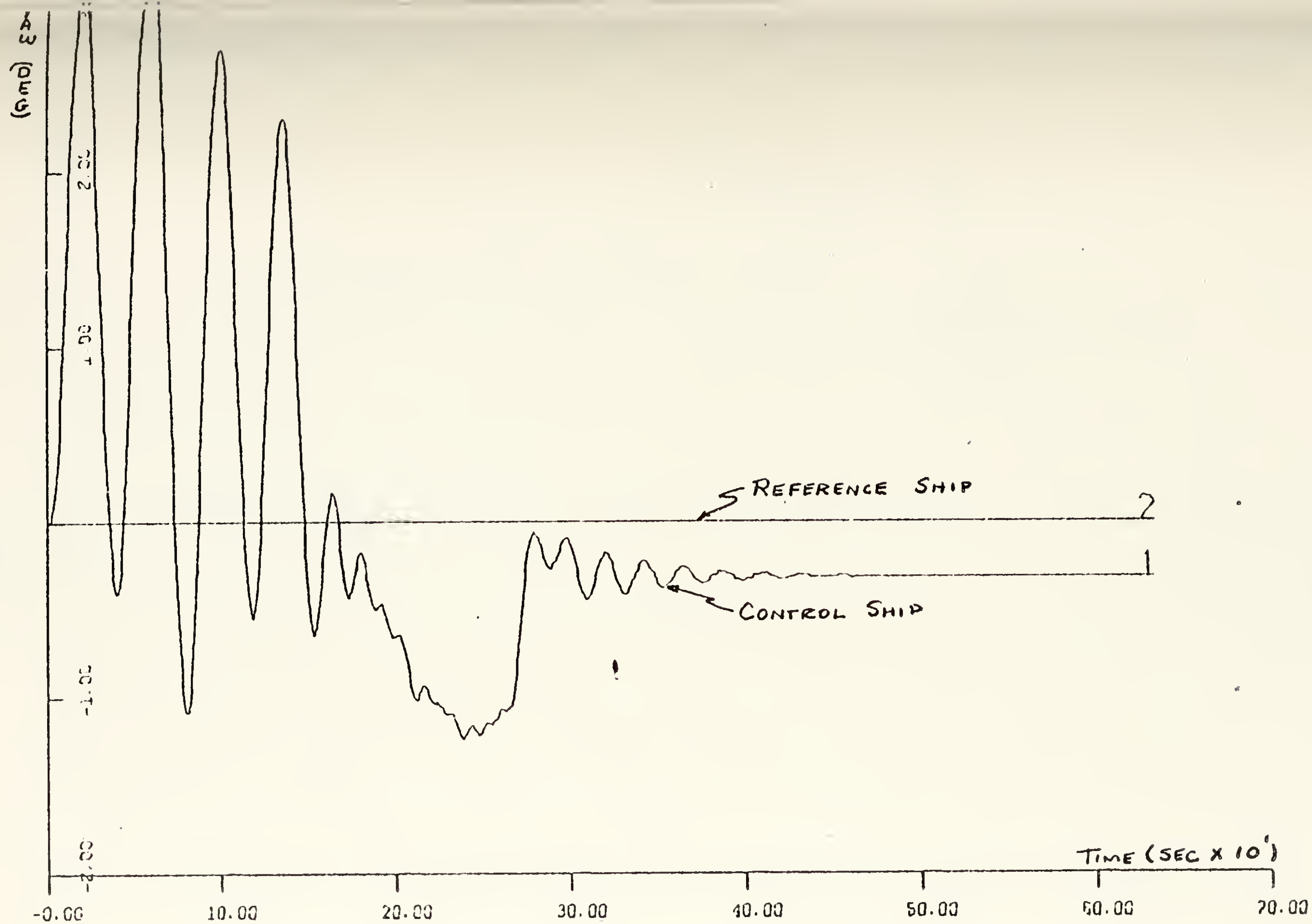


Figure III-50

Approach Phase Run #6 Yaw Response

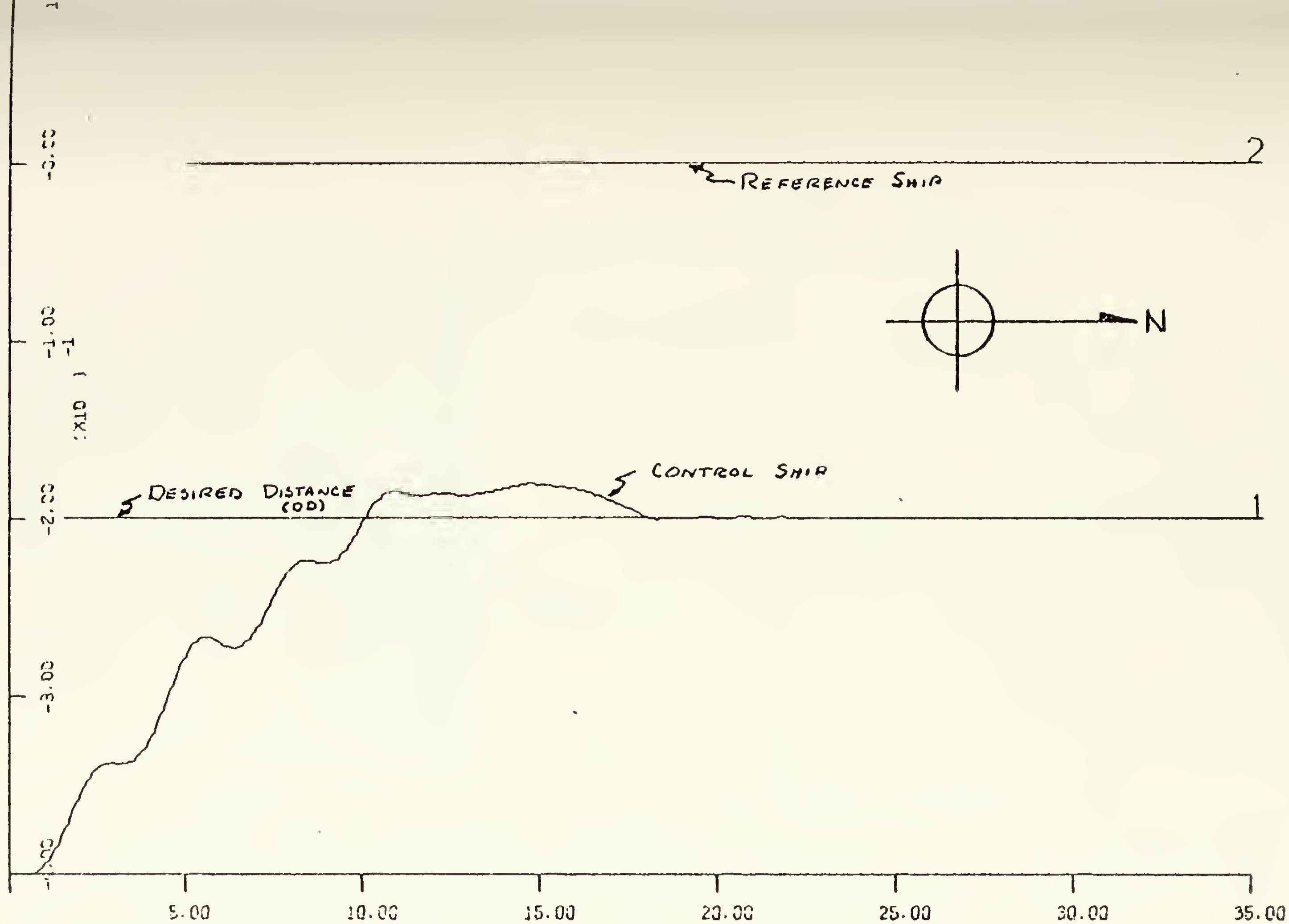


Figure III-51

Approach Phase Run #6 Geographic Plot

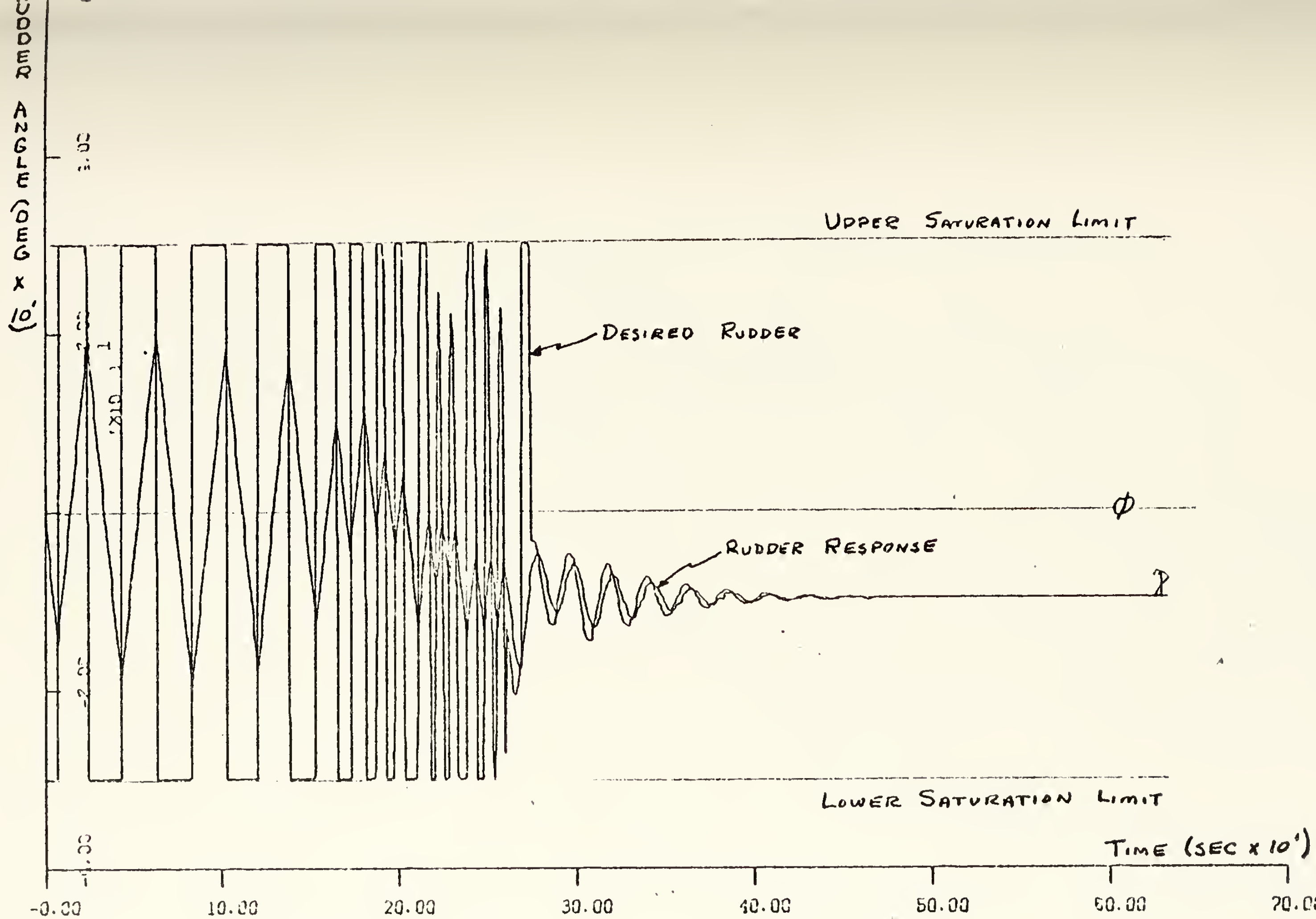


Figure III-52

Approach Phase Run #6 Rudder Response

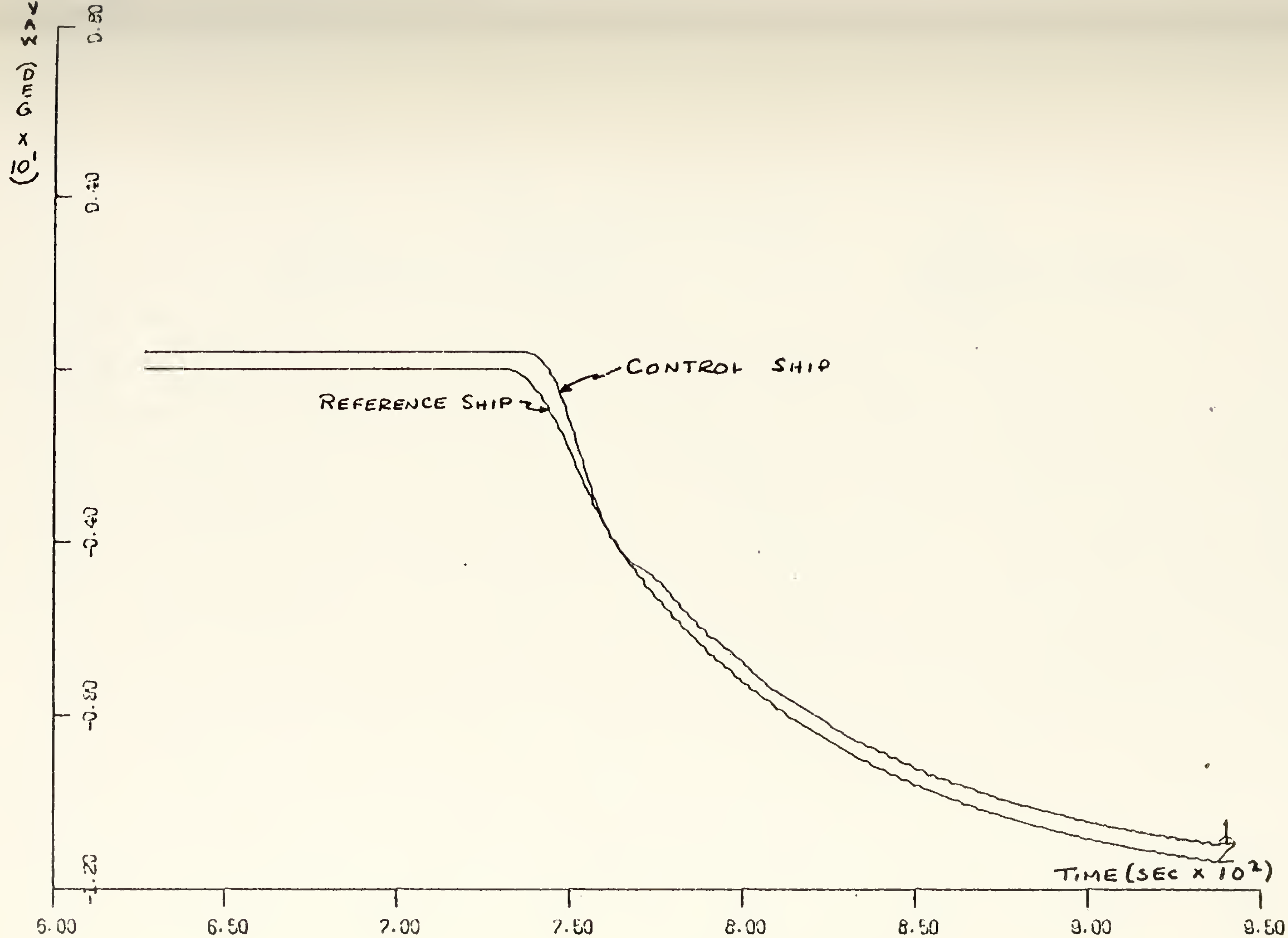


Figure III-53

Turn Phase Run #4 Yaw Response

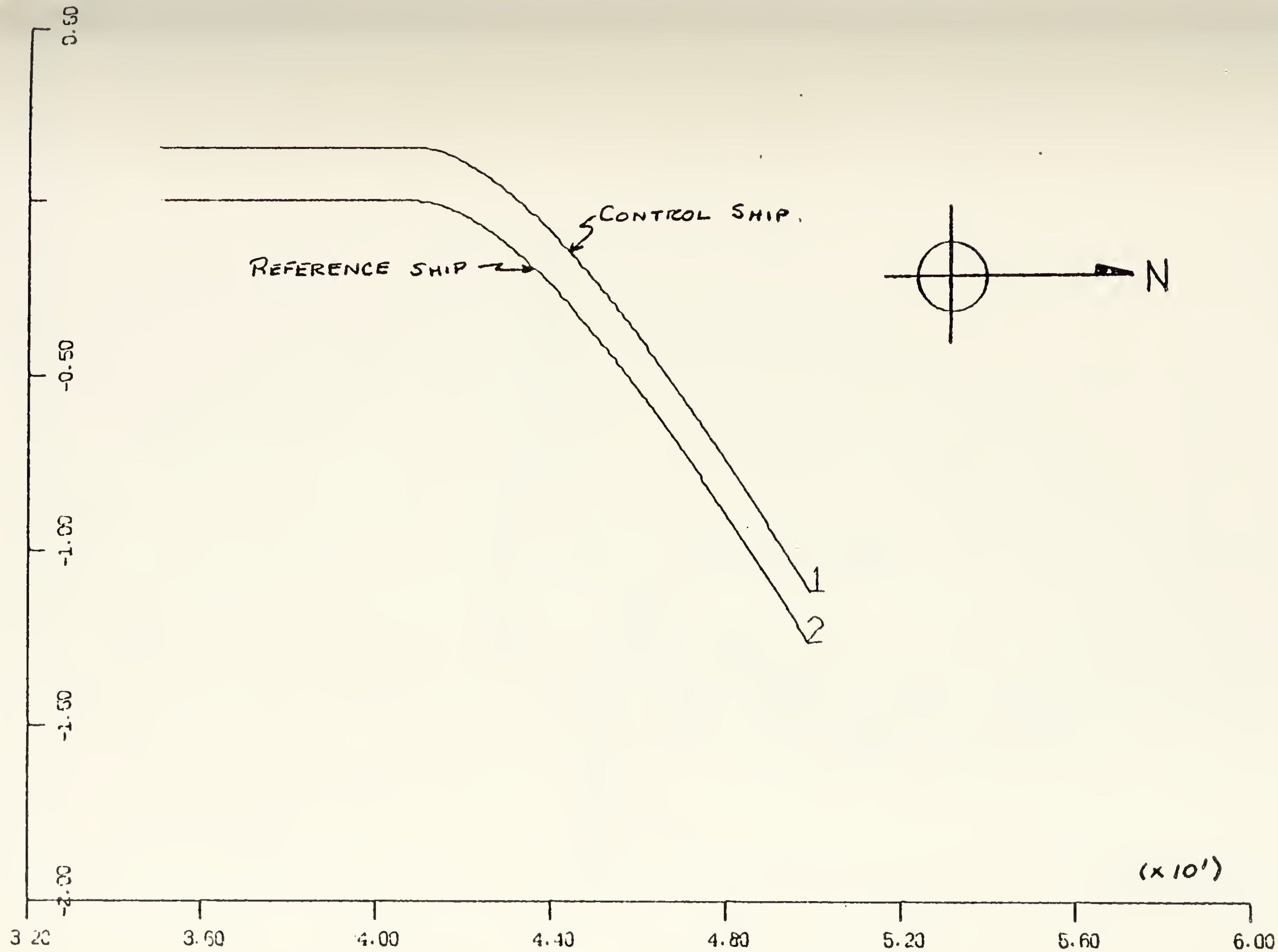


Figure III-54

Turn Phase Run #4 Geographic Plot

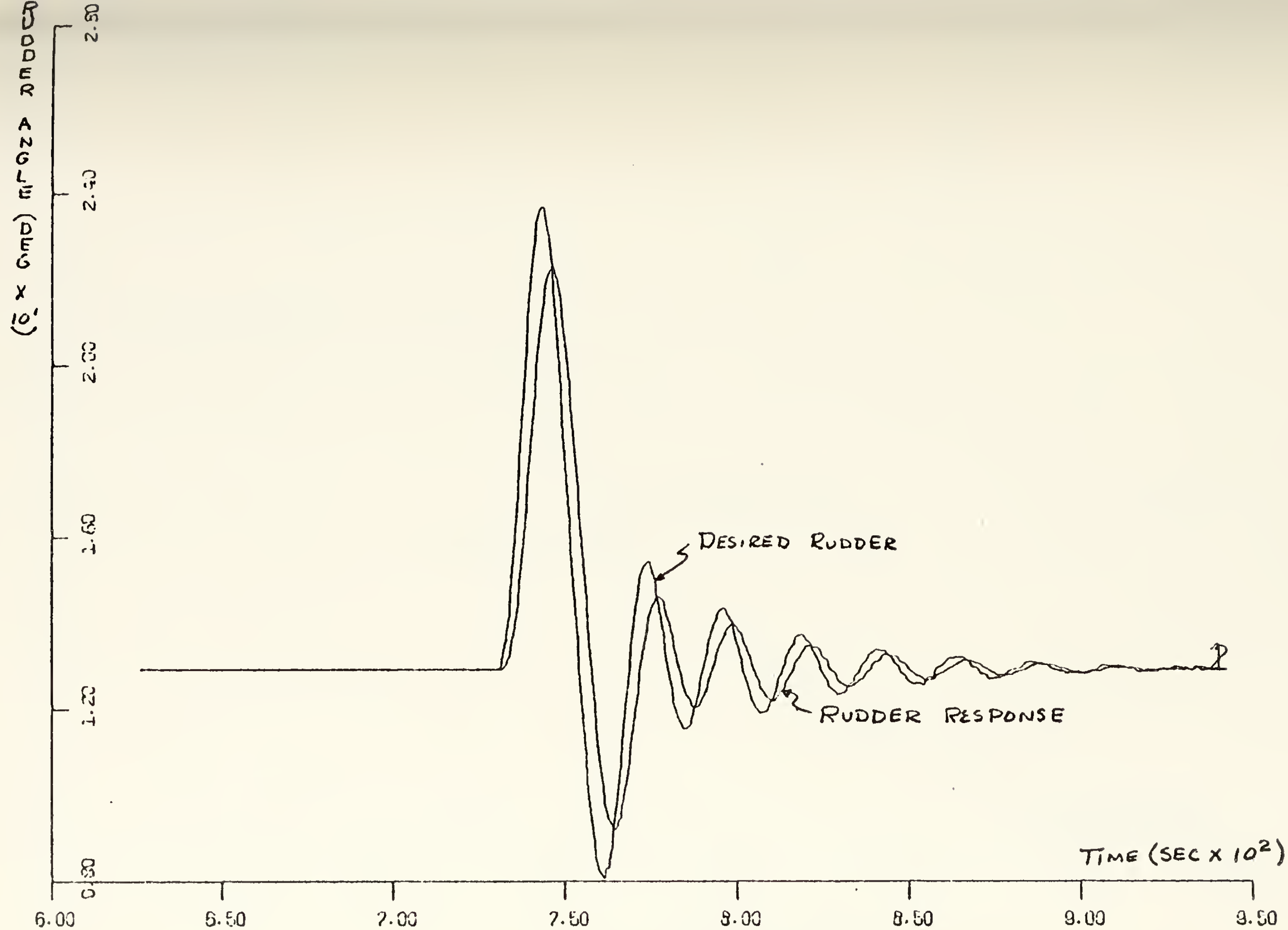


Figure III-55

Turn Phase Run #4 Rudder Response

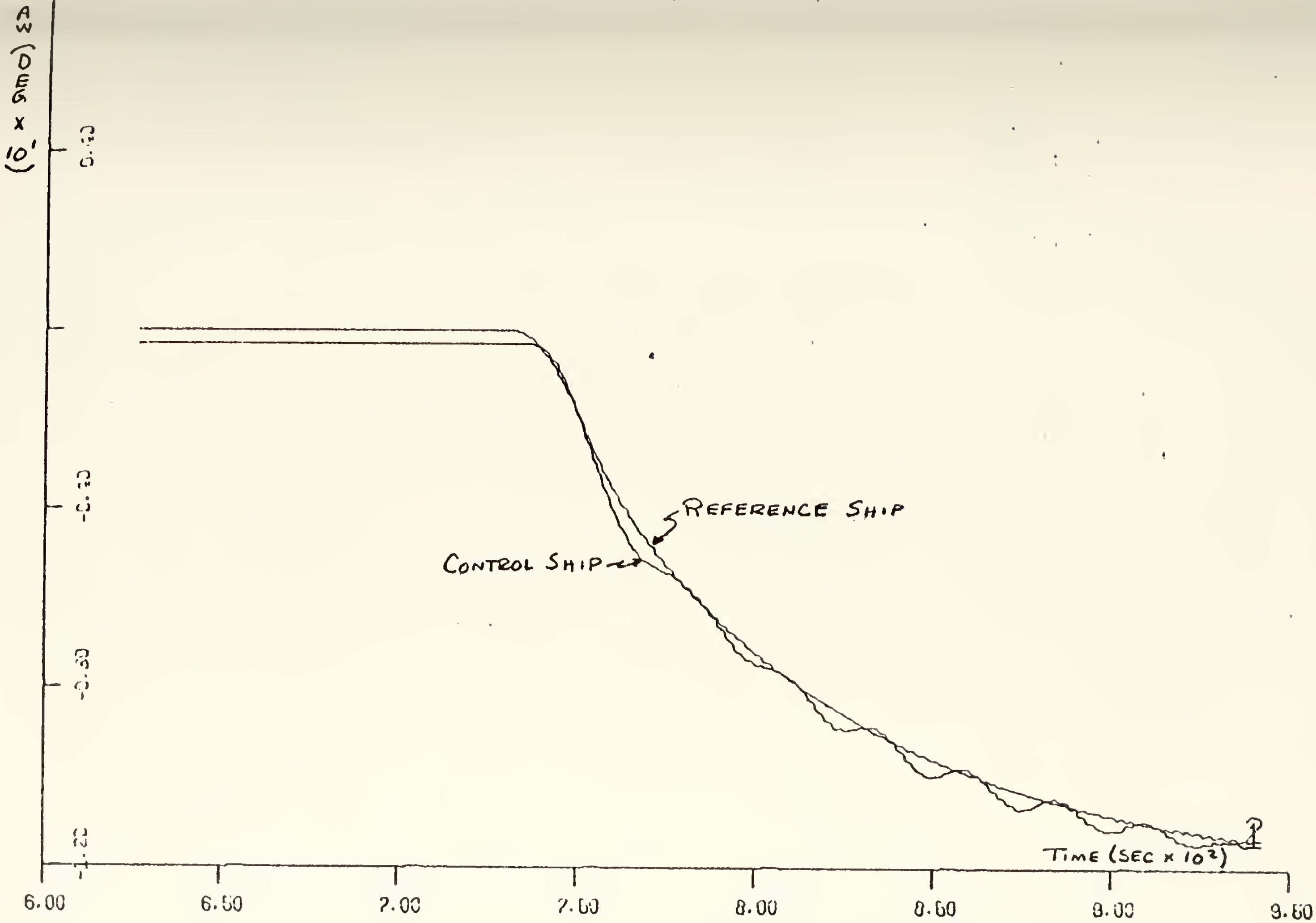


Figure III-56

Turn Phase Run #6 Yaw Response

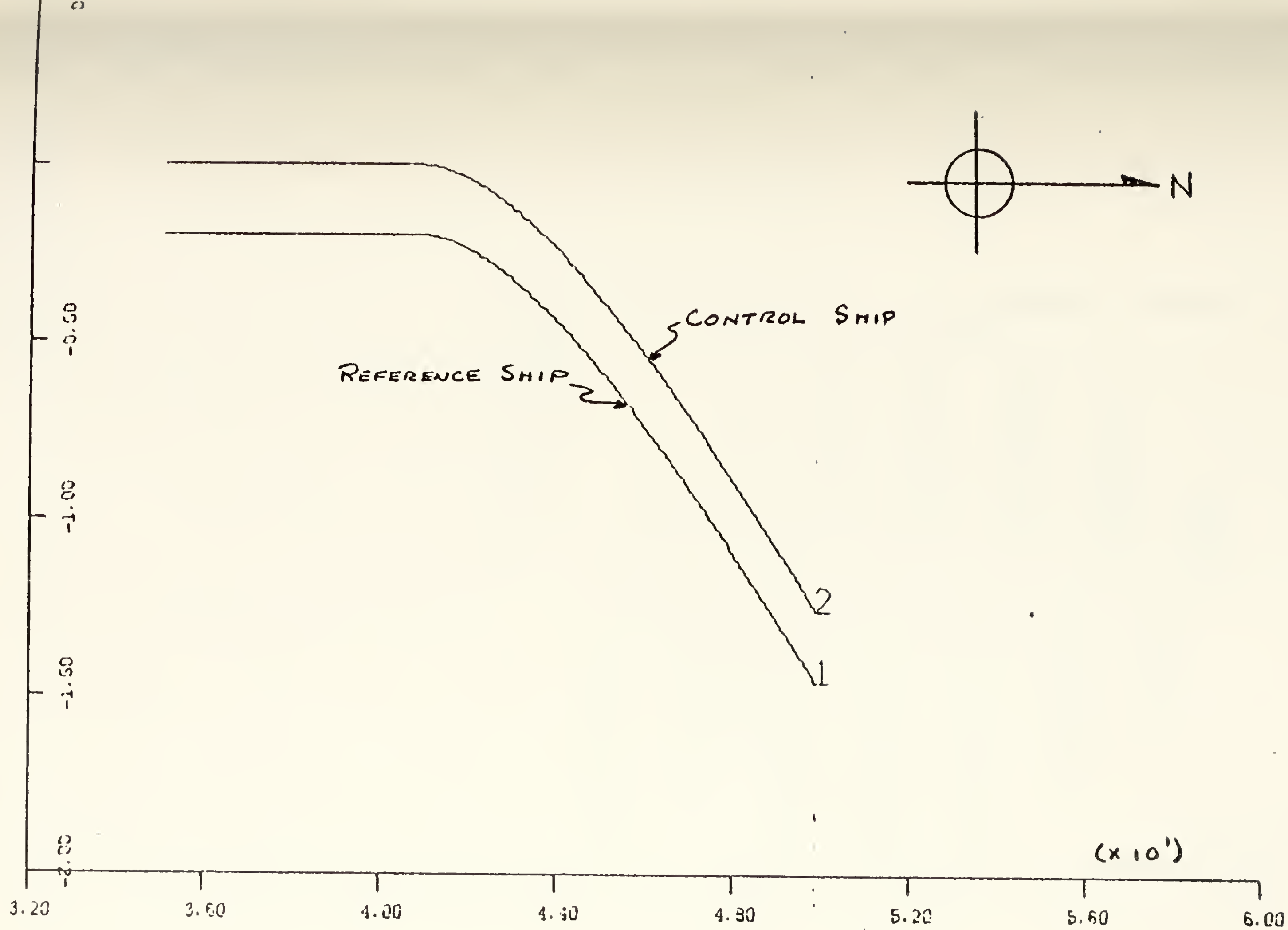


Figure III-57

Turn Phase Run #6 Geographic Plot

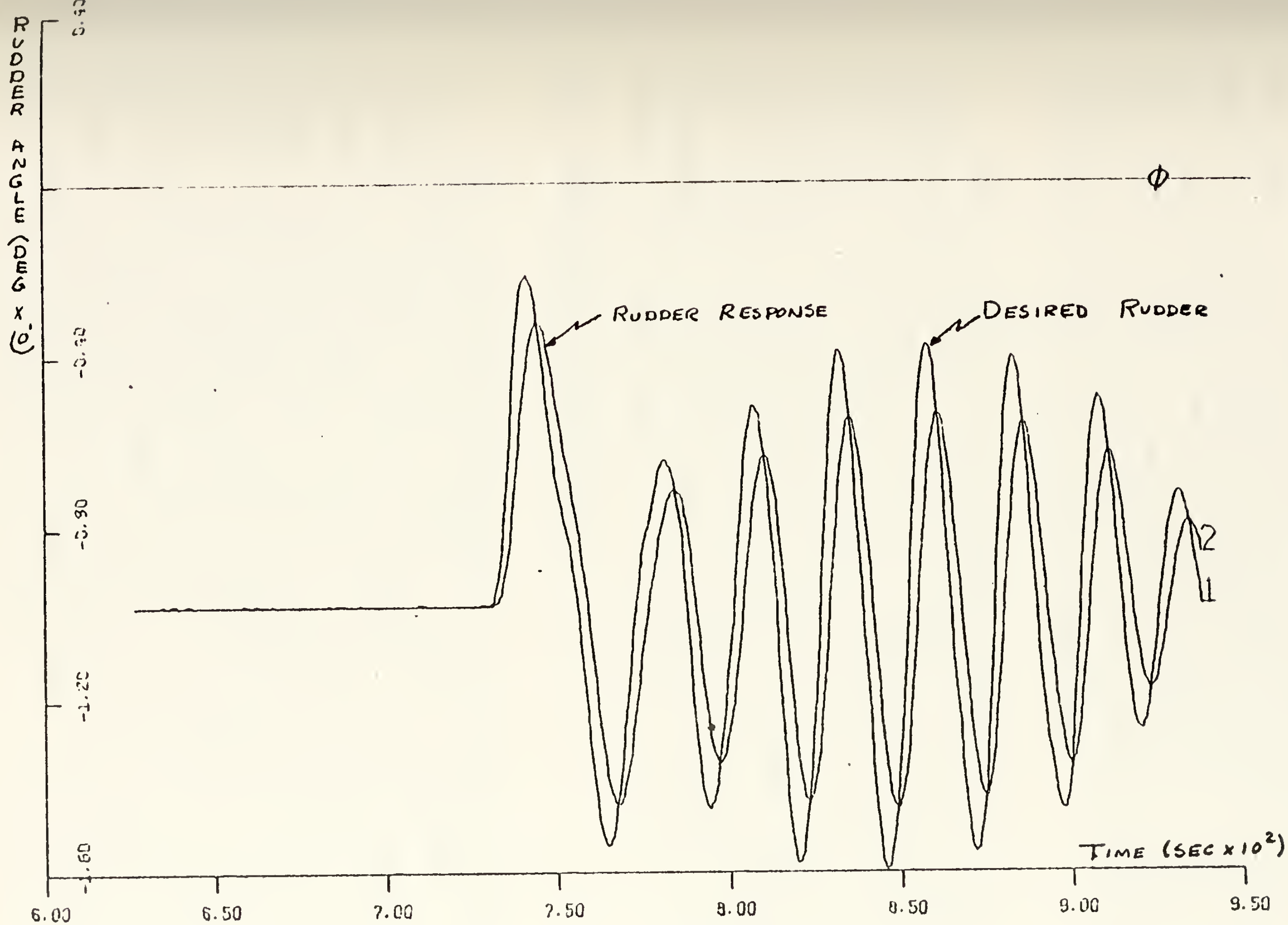


Figure III-58

Turn Phase Run #6 Rudder Response

experienced which gives 1.6 ship lengths bow to stern clearance).

The purpose of run 6 is to provide simulation for an approach from the opposite side again disproving any concern for ambiguity in the trignometric measurement scheme utilized. In all runs it must be emphasized that DD is the positive absolute distance desired and that IS provides the code flag for the desired side of approach. The system will work with DD set to some negative quantity; but the side of approach will reverse itself and the position placement will be correct, but on the side not desired.

Run 4 takes the desired distance in to 0.15 ship lengths (80.0 feet). This distance is usually the minimum desired by a prudent seaman. Again, even with this minimum distance, the control system performs up to desired standards. The importance of this run cannot be overlooked. Performance of the system at this extremum indicates that the gains utilized are correct for all expected conditions encountered in calm seas. Figures III-59 thru III-64 portray the remaining plots obtained in run 4.

h. Performance in Sea State

The calm sea performance of the heading control system is only part of the system testing required. Of even greater concern is the adequacy of the control when sea state is introduced. Section D.2. of chapter II models the three components of waves with two sinusoids and a small random impulse wave. These forces were introduced into the total RAS simulation as shown in computer program #7. In this program the wave length (WL) is set to one ship length and the wave direction (WD) is -015 degrees true. This scenario allows for a port turn into the prevailing sea as is common practice in experienced RAS evolutions. By

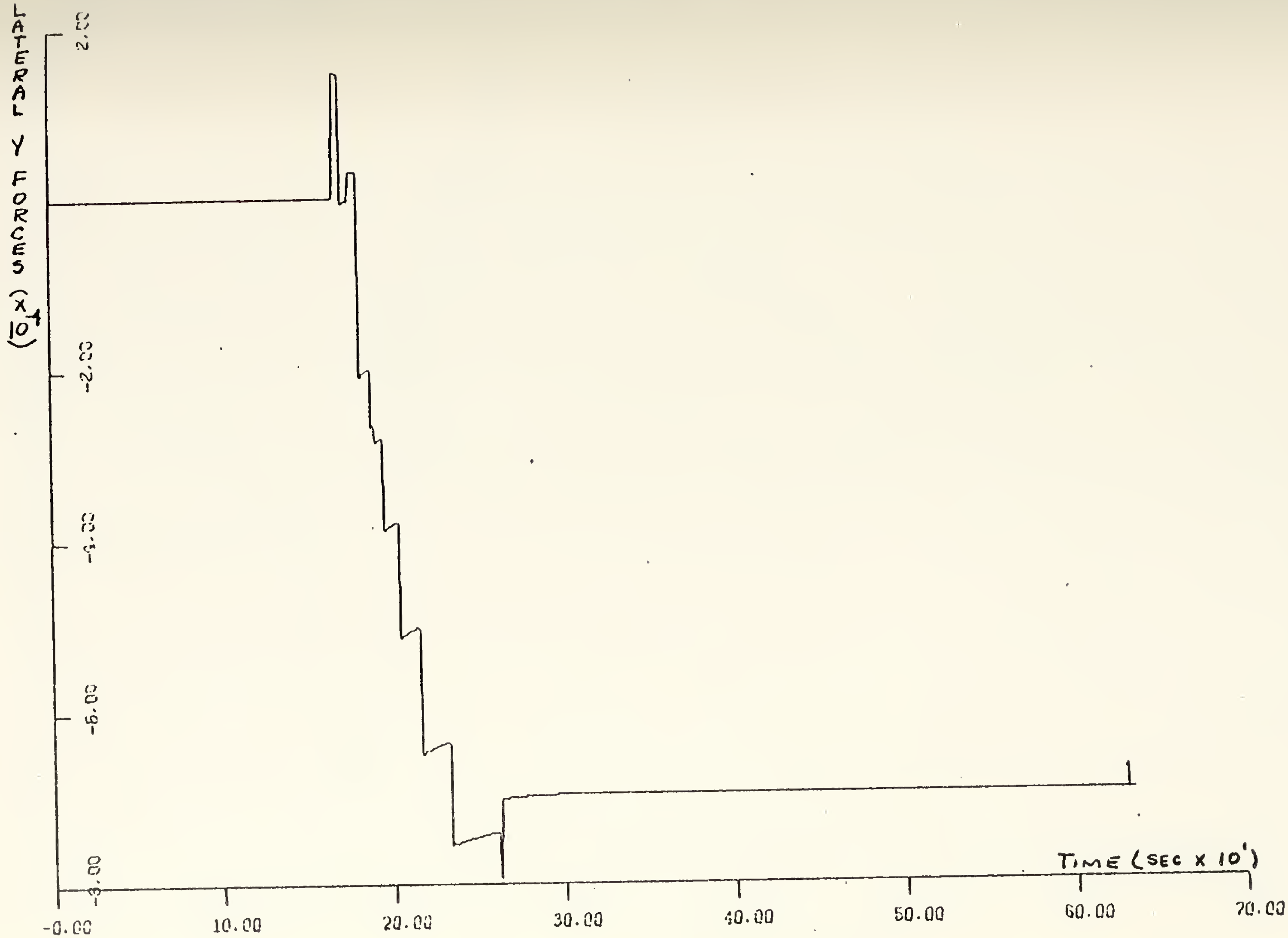


Figure III-59

Approach Phase Run #4 Lateral Y Forces

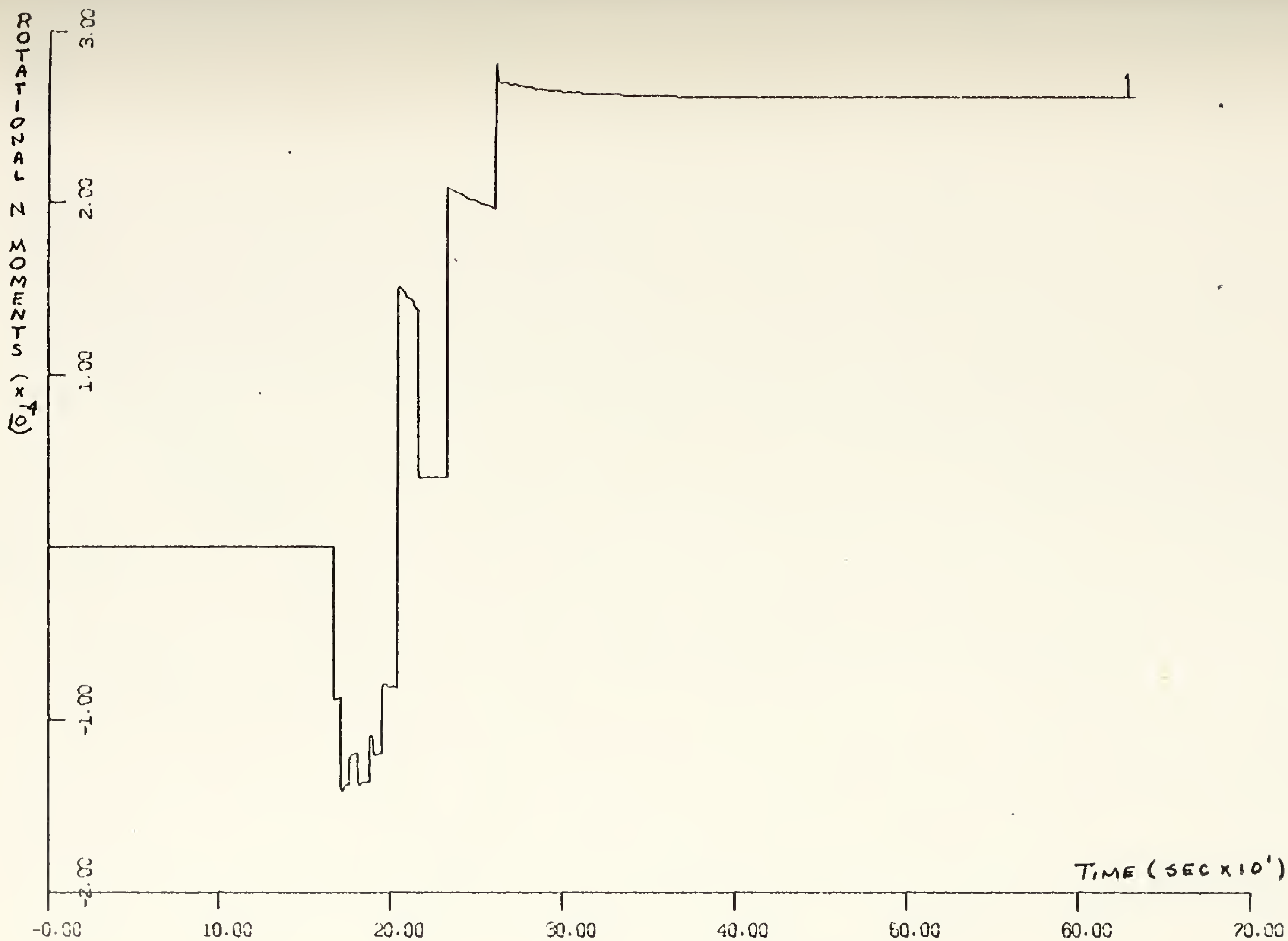


Figure III-60

Approach Phase Run #4 Rotational N Moments

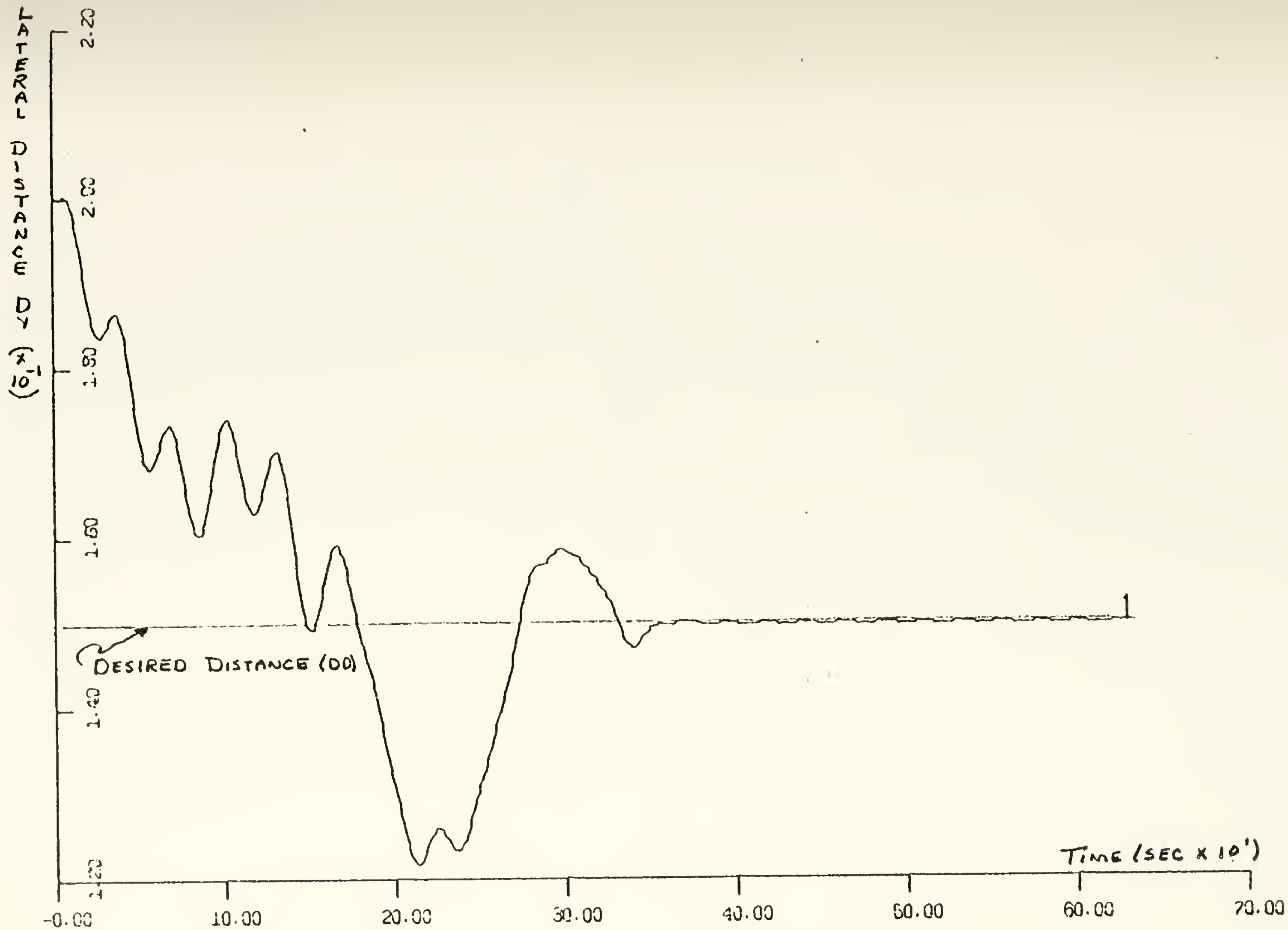


Figure III-61

Approach Phase Run #4 Lateral Distance DY

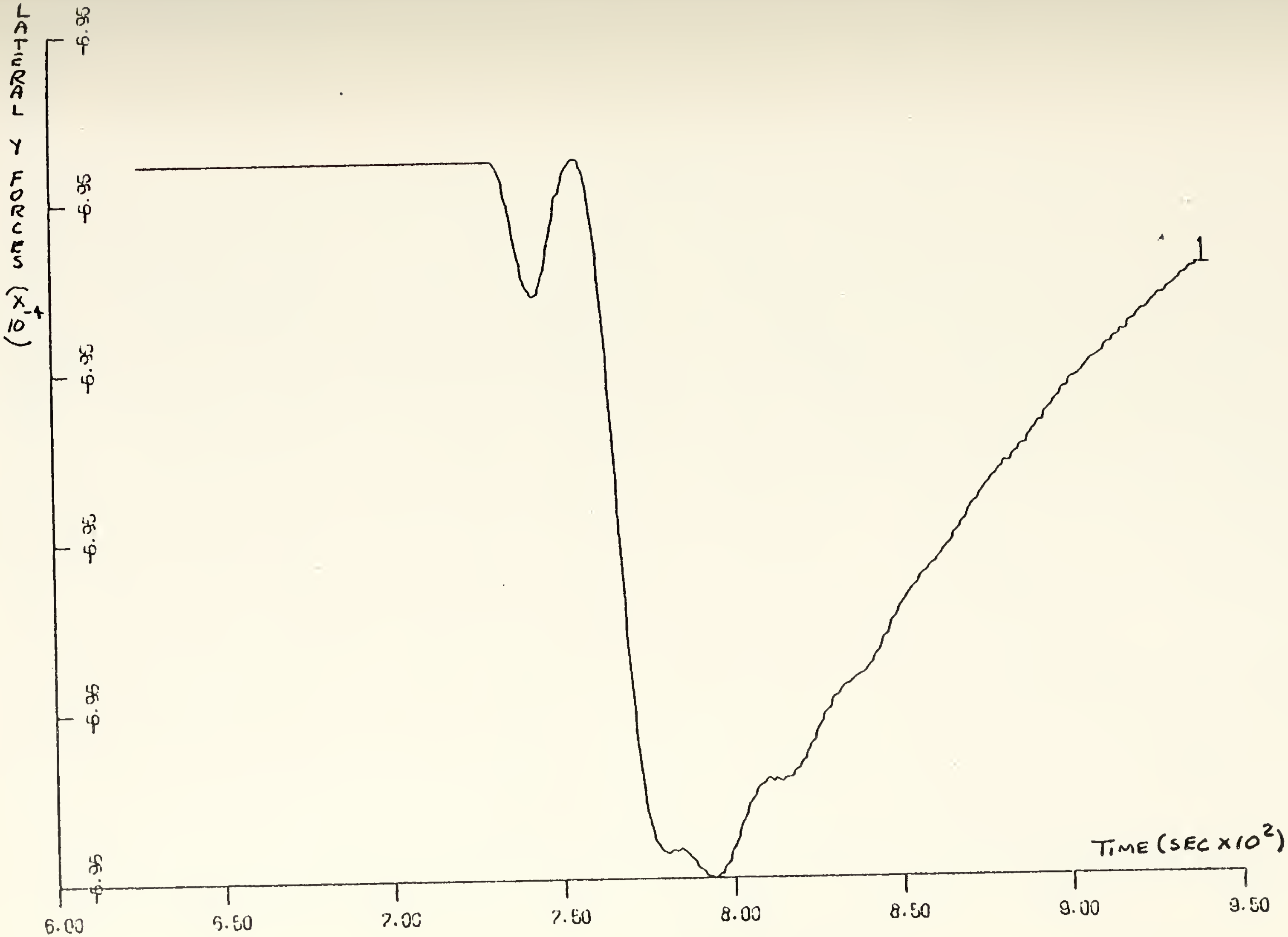


Figure III-62

Turn Phase Run #4 Lateral Y Forces

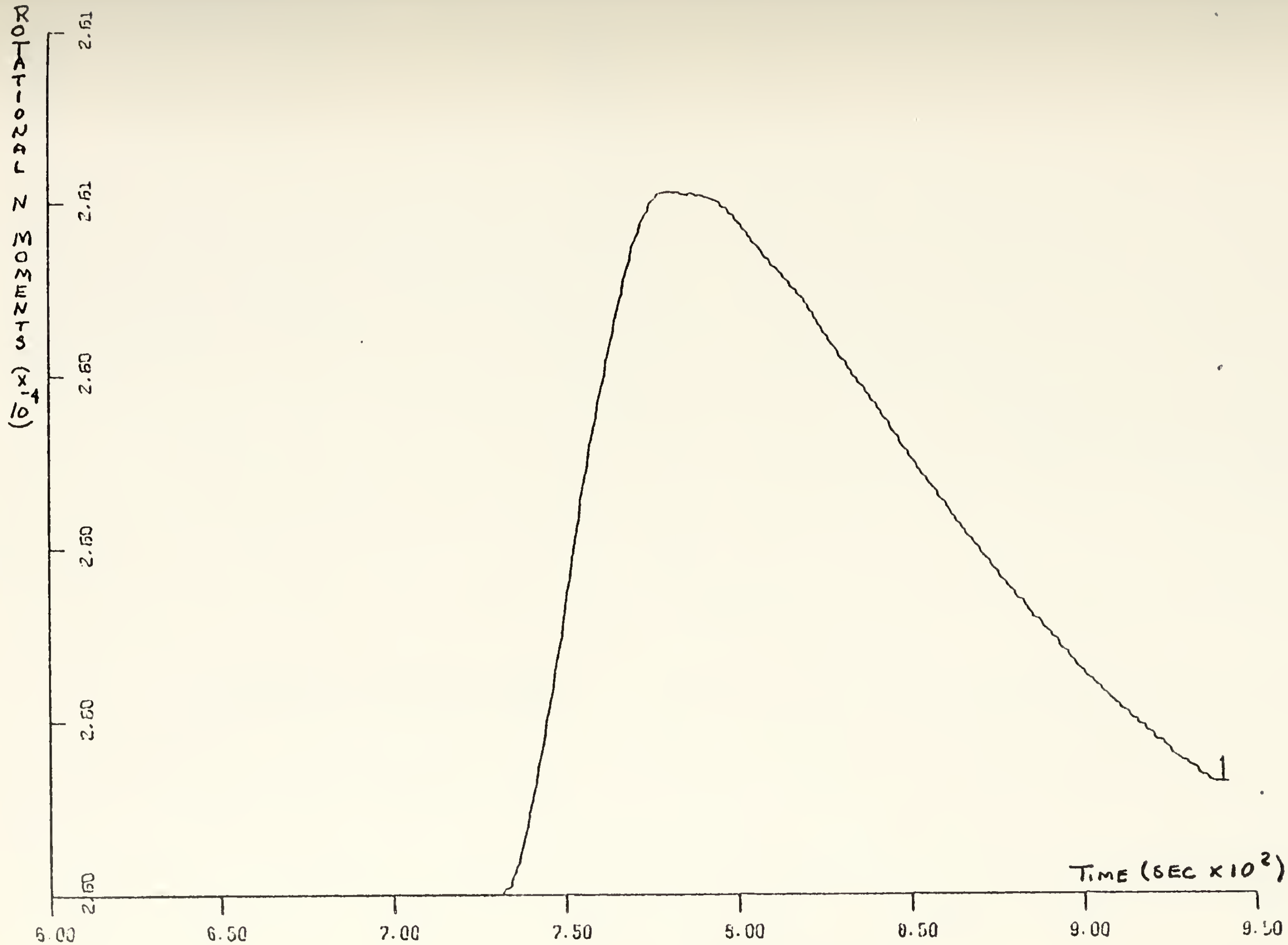


Figure III-63

Turn Phase Run #4 Rotational N Moments

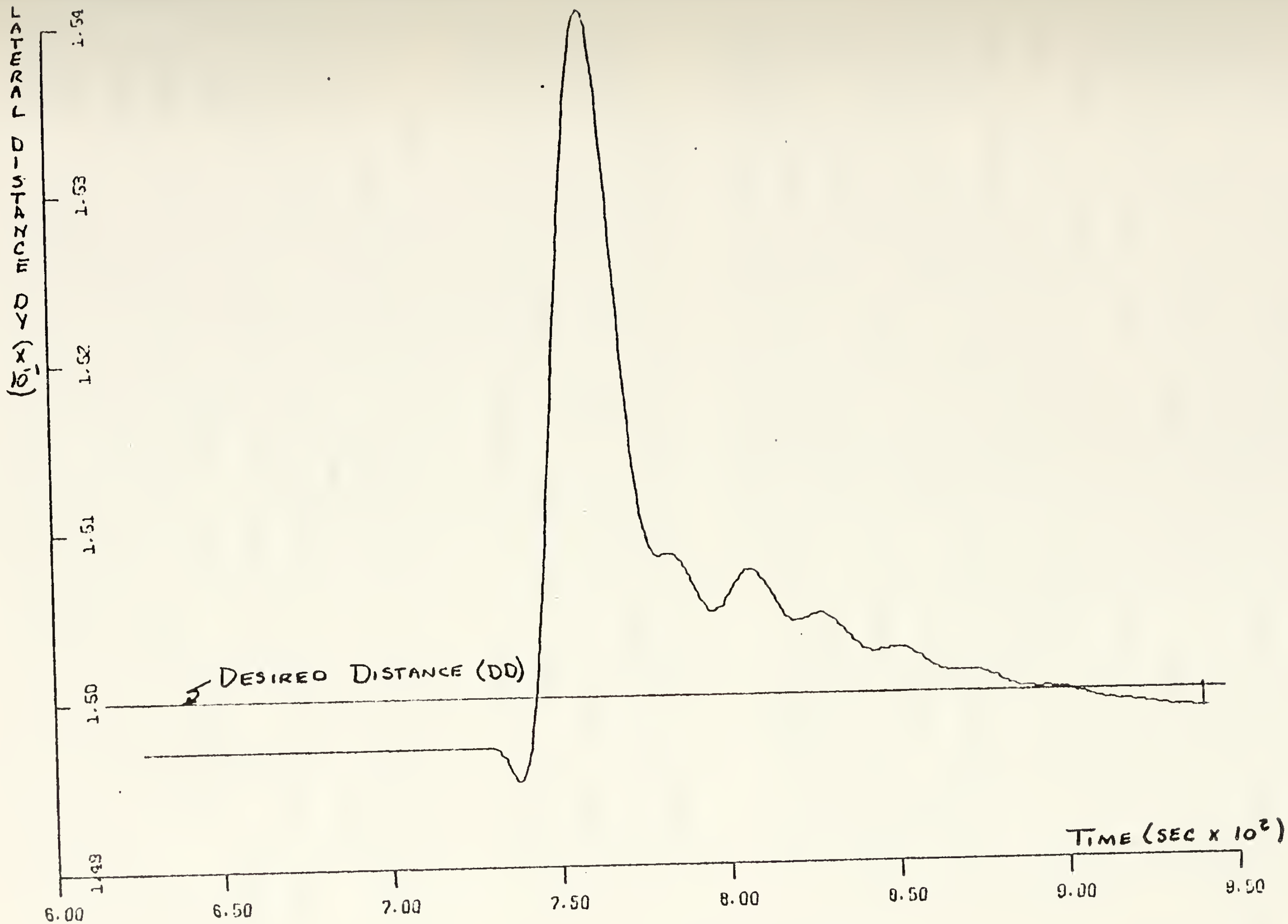


Figure III-64

Turn Phase Run #4 Lateral Distance DY

minimizing the perturbation forces on yaw and lateral direction, a smoother RAS can be accomplished thus aiding safety and comfort during the actual transfer. The wave force maximum is taken as 0.05685. Runs were simulated which used maximum wave forces in the range 0.1137 to 0.05685, wave lengths from 0.5 to 1.5 ship lengths and wave directions 015 to -015 degrees off the initial replenishment course. The control system handled all of the perturbations well except for the cases of a wave length of 1.5. This length of wave with a force of 0.05685 exceeded the control systems capability in that the steady state conditions were not met before a turn was commenced. Figure III-65 shows this instability in the lateral distance DY of the turn phase. It is felt that the modeling inadequacies of the sea state development of chapter II coupled with a simple adaptive gain scheme are the source of the problem. This same phenomenon is covered in greater detail in the longitudinal position offset testing portion of the velocity control section of this chapter.

Problems of this type also manifest themselves in some cases when the wave force maximum (WFMA) was close to the 0.1137 value. If the sea state becomes excessive, which this value represents, a different gain schedule or, at best, a more complex adaptive gain scheme is called for.

The plots produced by computer program #7 are presented as a representative indication of the effectiveness of the control system in the presence of a sea state. Figure III-66 gives the yaw results of the approach phase which indicates the effect of the wave action. The corresponding rudder action of figure III-67 compensates to give the smooth lateral distance shown in figure III-68. The wave profile is shown in figure III-69 with curve 2 being WY and curve 3 being WN. Curve 1 is the WX profile which was not used in this run but will be utilized in the speed control section later in this chapter. Similar curves are portrayed for the turn phase. Figure III-70 is the yaw difference between the two ships (remembering that the reference ship is not being perturbed by the interaction forces or the wave forces). Figure III-71 is the lateral distance DY maintained by the rudder response of figure III-72. The maximum lateral separation in the turn phase is 0.0037 ship lengths (1.95 feet). The wave profile is shown in figure III-73 with the same wave force curve sequence as the approach phase.

As can be seen from these plots, the control system operates very effectively in the presence of a sea state. Again, the development of a much more complex adaptive gain scheme is required to allow exceptionally high sea state. It is felt that the control system presented in this thesis is adequate for most situations that are encountered in the RAS environment. Only the extreme perturbations that chance would allow must be accounted for in a more complex adaptive gain scheme.

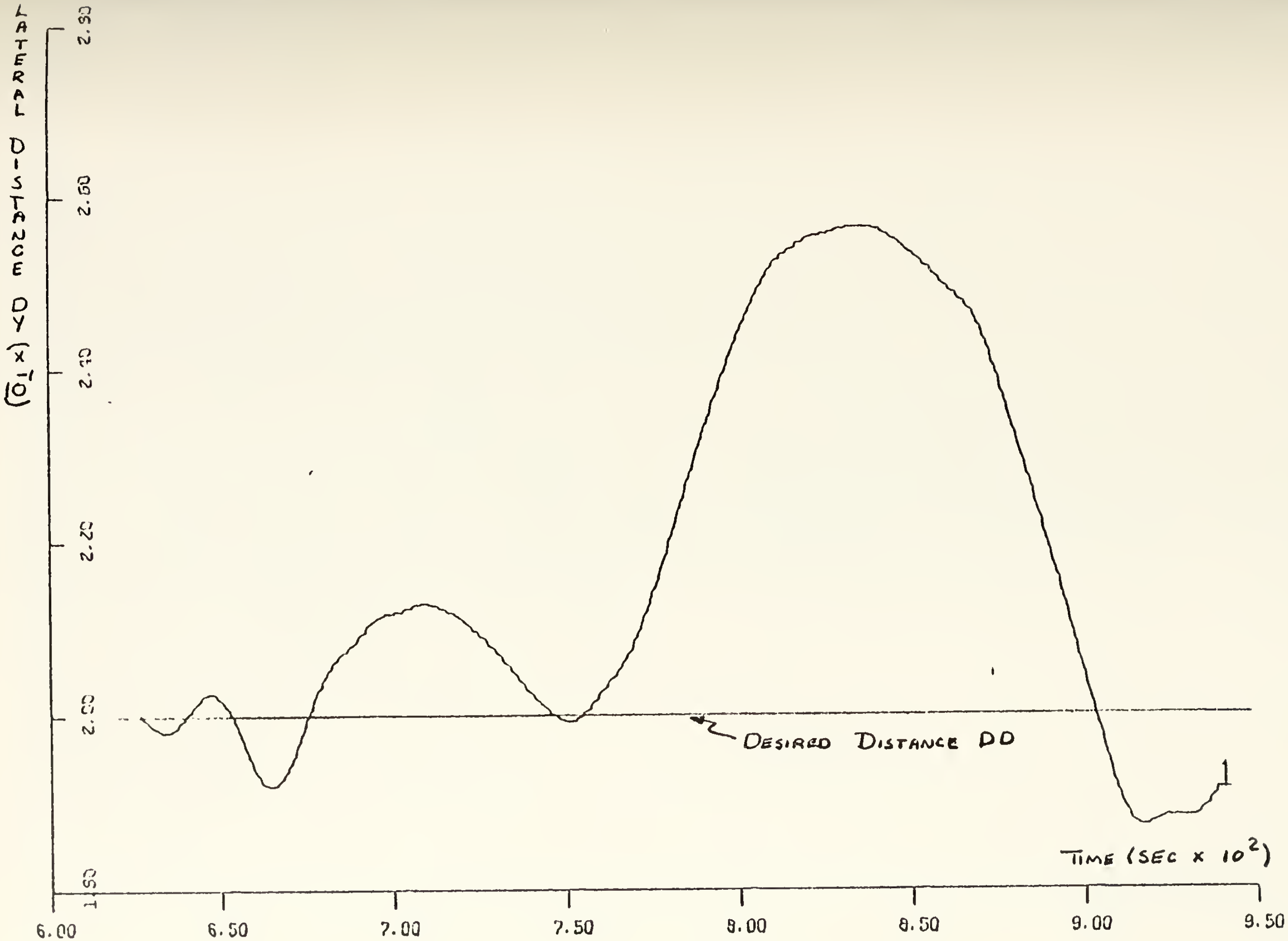


Figure III-65

Wave Effect on Turn Phase Lateral Distance (DY) WL=1.5

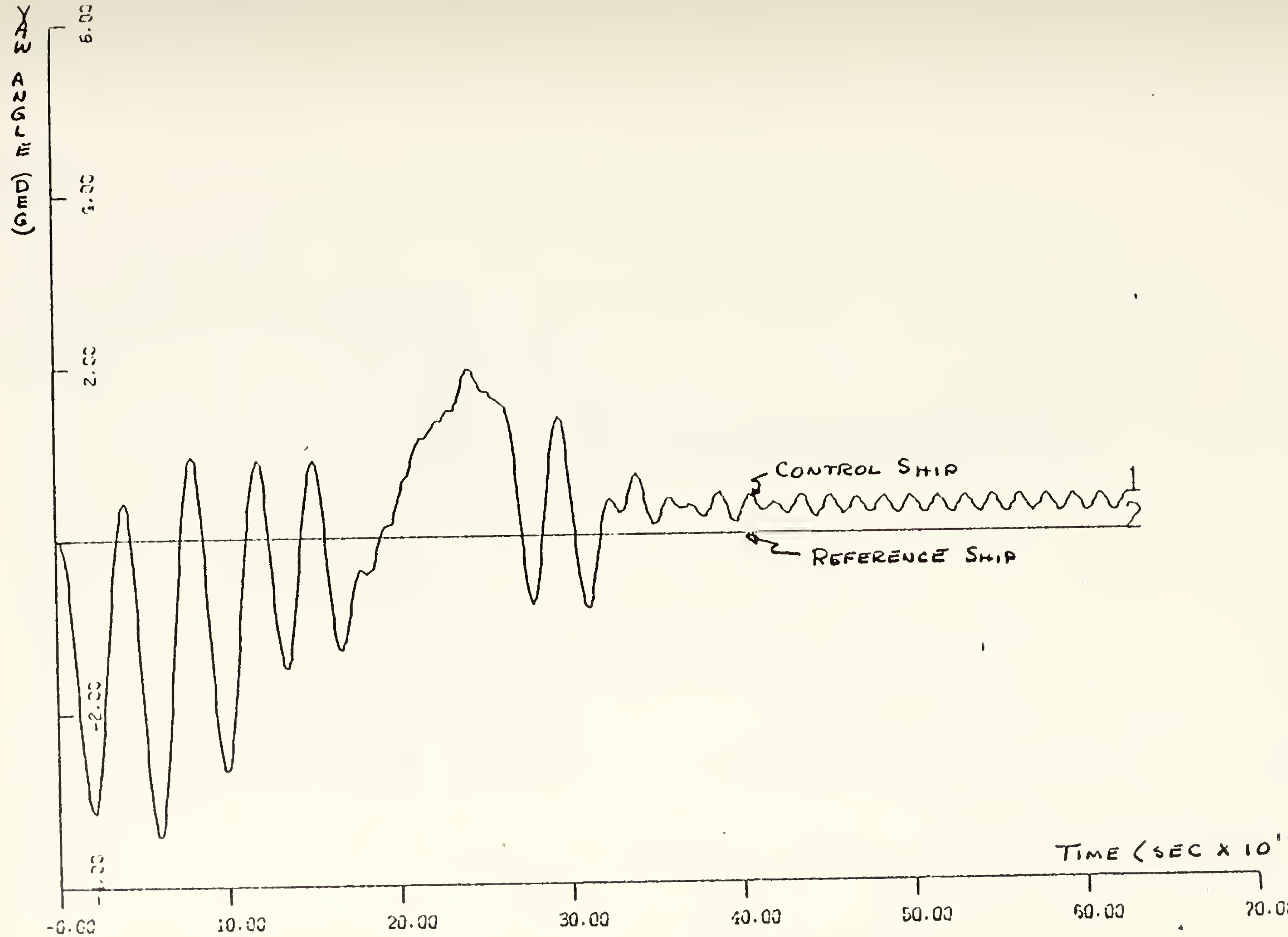


Figure III-66

Wave Effect on Approach Phase Yaw WL=1.0

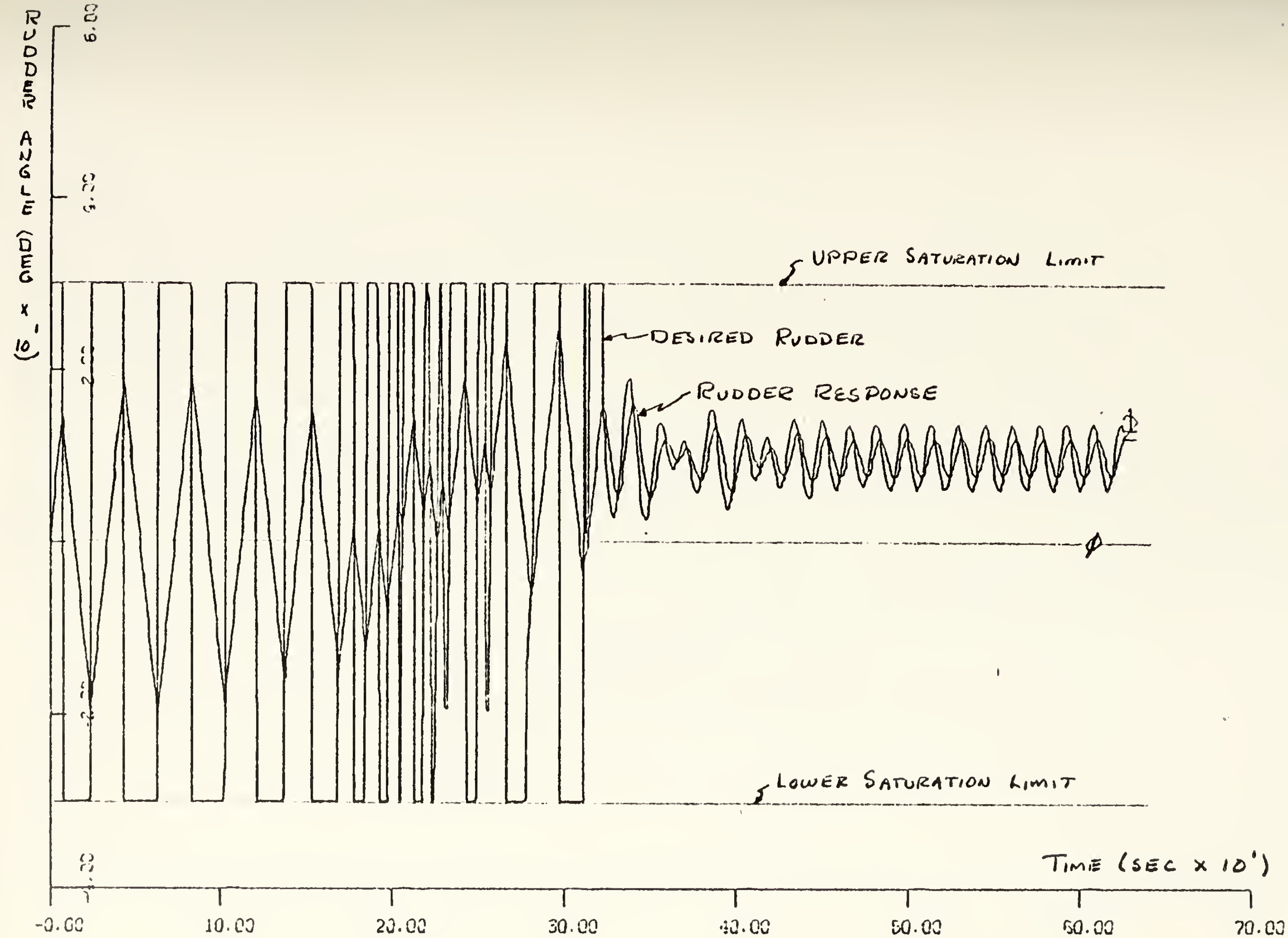


Figure III-67

Approach Phase Rudder Response to Waves WL=1.0

Figure III-68

Wave Effect on Approach Phase Lateral Distance (DY) WL=1.0

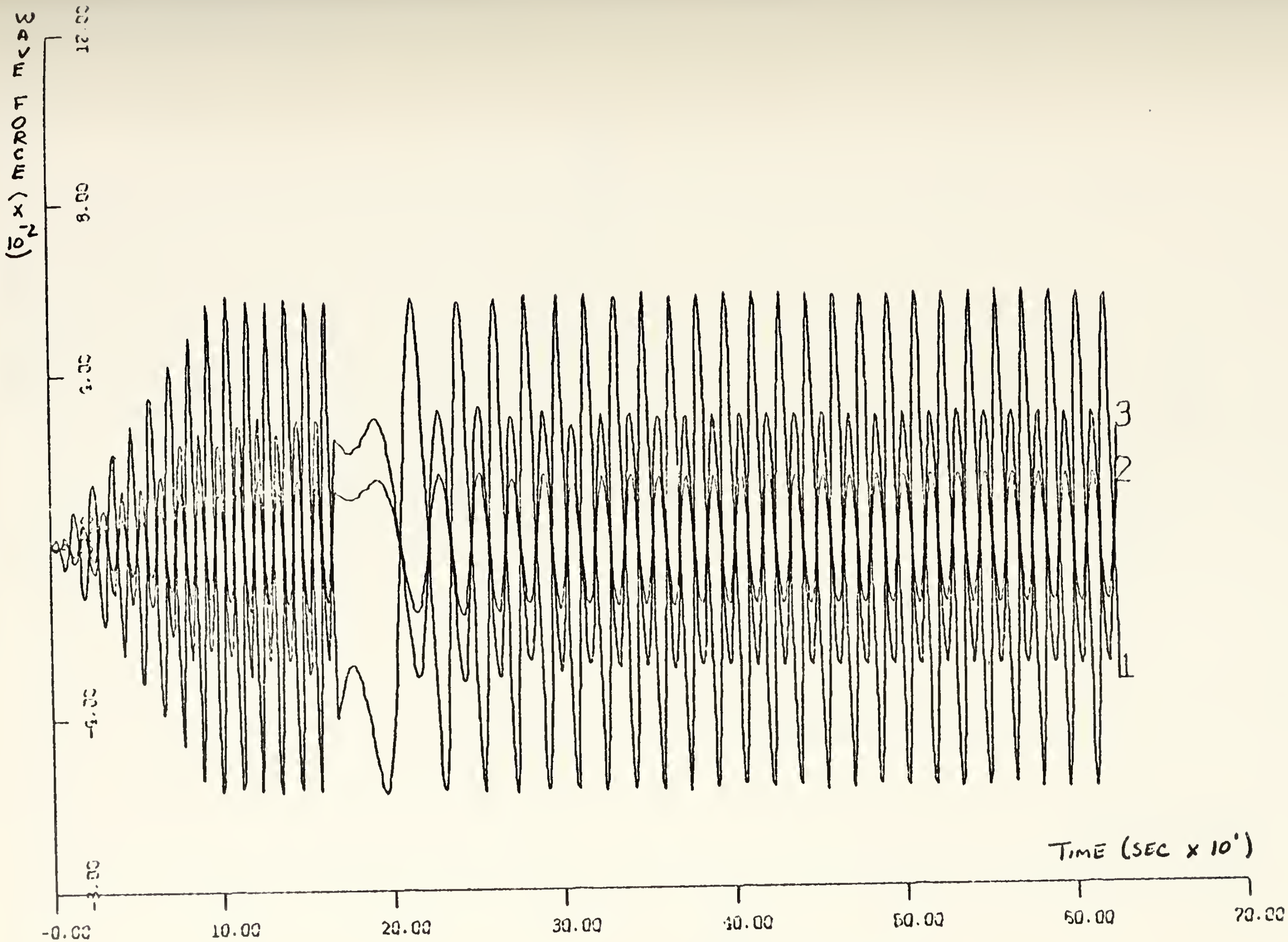


Figure III-69

Approach Phase Wave Profile WL=1.0

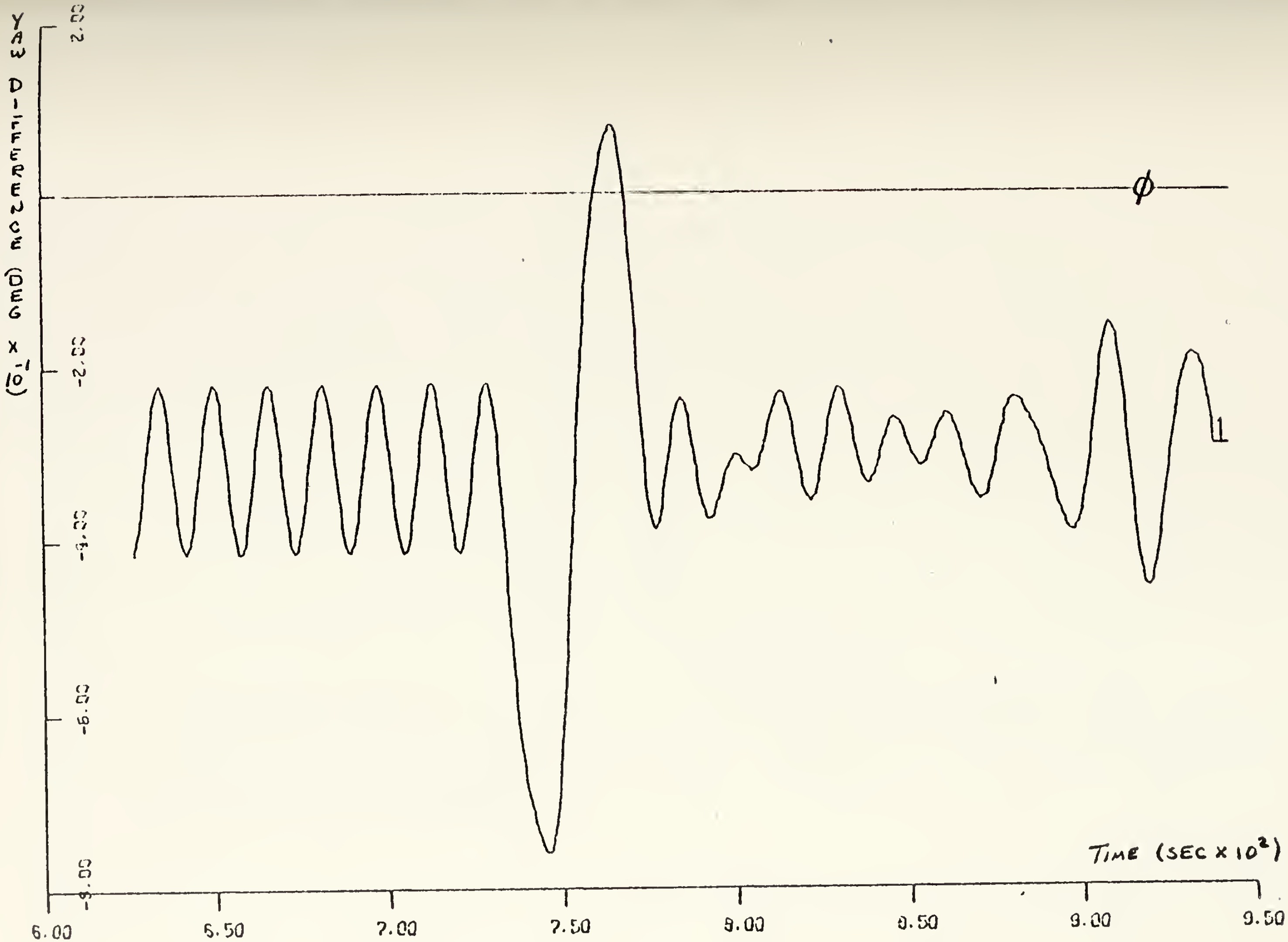


Figure III-70

Wave Effect on Turn Phase Yaw WL=1.0

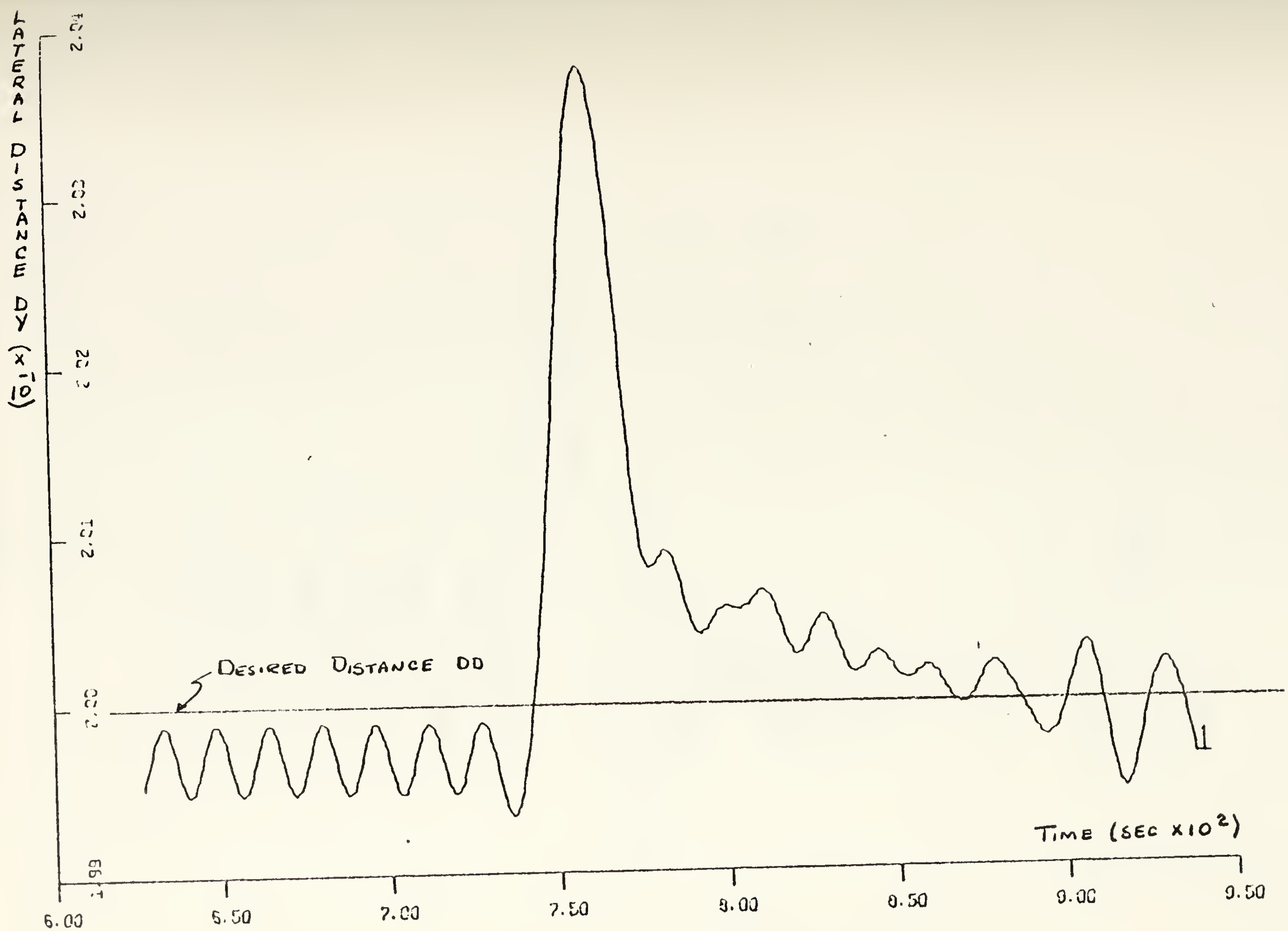


Figure III-71

Wave Effect on Turn Phase Lateral Distance (DY) WL=1.0

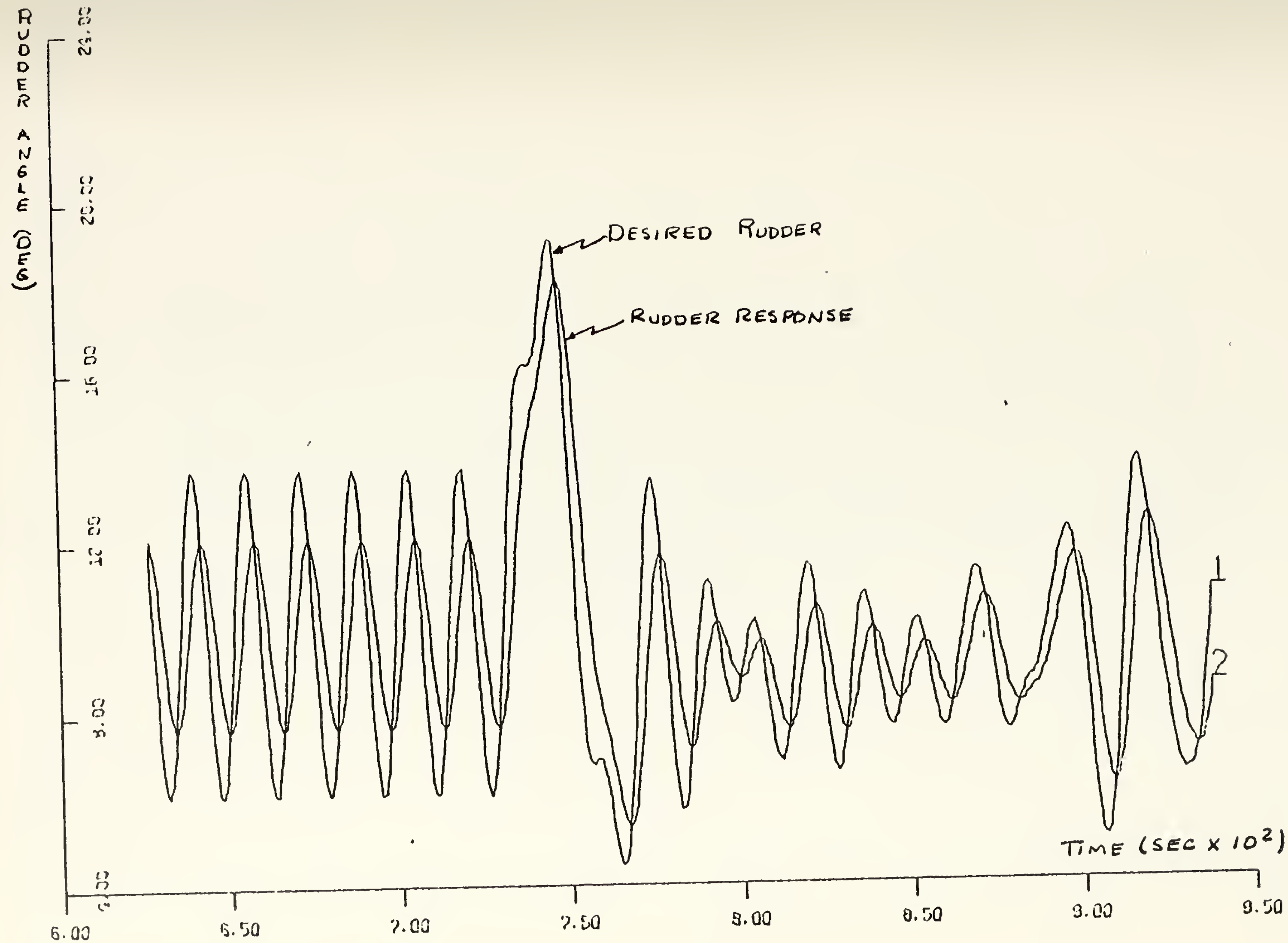


Figure III-72

Turn Phase Rudder Response to Waves WL=1.0

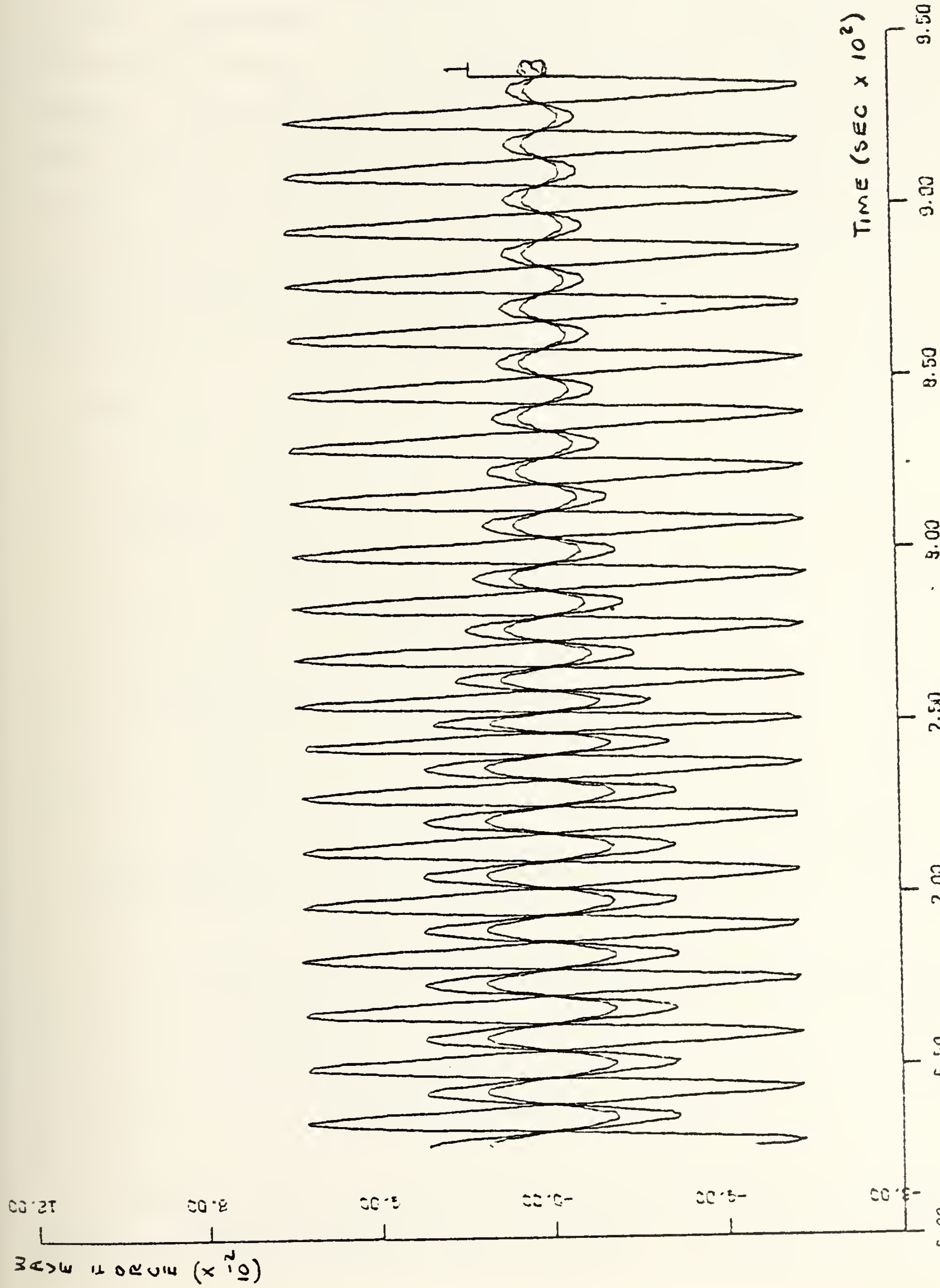


Figure III-73

Turn Phase Wave Profile WL=1.0

B. VELOCITY CONTROL

One advantage derived from using the linearized equations of motion is the decoupling of the velocity components from the remaining equations of motion. This allows separation of the design procedures for lateral separation control and velocity control. Section A of this chapter designed the lateral separation control using the simple speed control algorithm shown in figure III-74. This control output was used directly as the ship's speed (CDCT2) in the model simulation where no attempt was made to use the engine response developed in chapter II. Function SPDCTR of appendix A shows the control used.

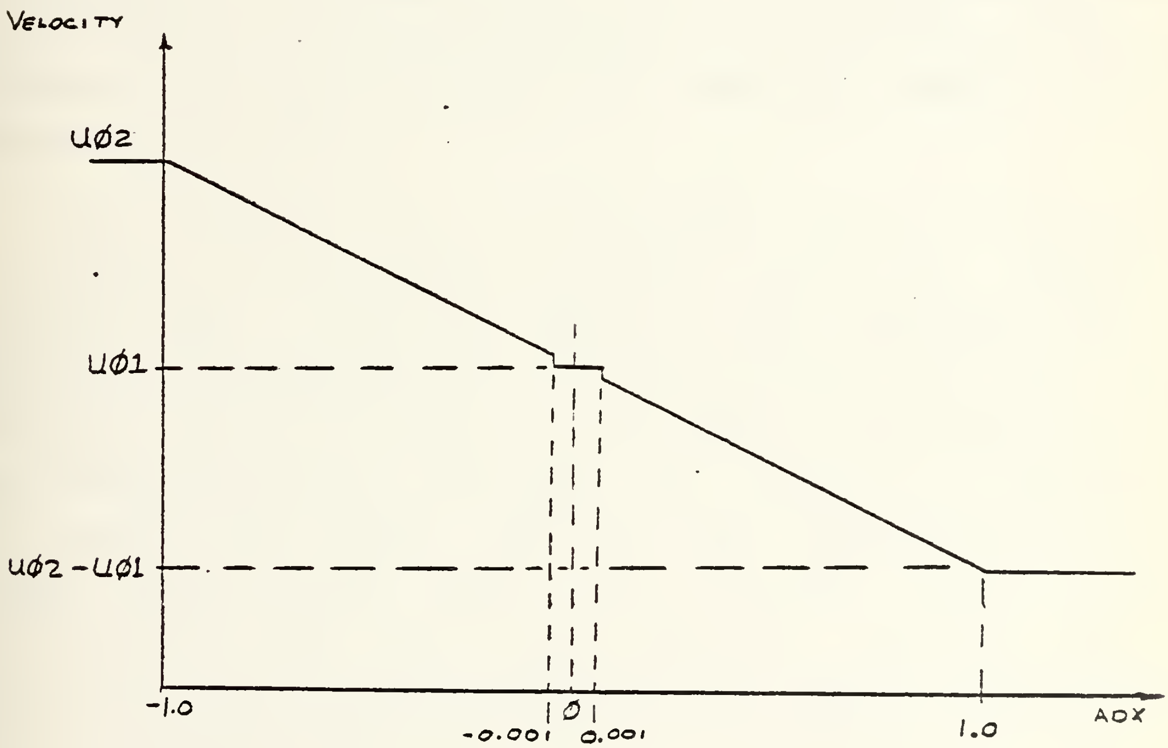


Figure III-74
Non-optimum Speed Law

Because of this decoupling assumption, any valid approach speed control can be used, if used consistantly, for such a design. However, in the RAS environment, complete disassociation is not possible. Recombination occurs in the interactive forces and moments which depend upon the longitudinal distance as well as the lateral distance. Consequently, speed, which is directly responsible for the longitudinal distance, has a direct relation to the lateral distance attainment and maintainment.

The remaining parts of this chapter deal with the development of a viable speed control algorithm and the testing of the designed system.

1. Type of Control

Whenever two ships maneuver for replenishment at sea (RAS), the prime considerations are the time required for approach and the accuracy of position keeping plus conservation of fuel.

The nonlinear control law of figure III-75 is designed to maintain a preselected approach speed for minimum approach time. The proper location of the switching point increases the complexity of the solution since the time of switching from this speed is determined by the dynamics of the nonlinear position attainment loop. Once this position is reached, the speed controller is switched down to a linear portion of the control law to allow control for perturbations about the operating position. However, small perturbations about this operating point can be tolerated and, in fact, are desired to allow for conservation of fuel. Selection of this dead zone is wholly dependent on the accuracy required for final position. Figure III-75 indicates a dead zone extending to ±0.001 normalized distance which in this case translates to ±0.53 feet. Systems for which fuel considerations are not a motivating factor may be designed without this part of the control law to allow finer tracking in the position loop.

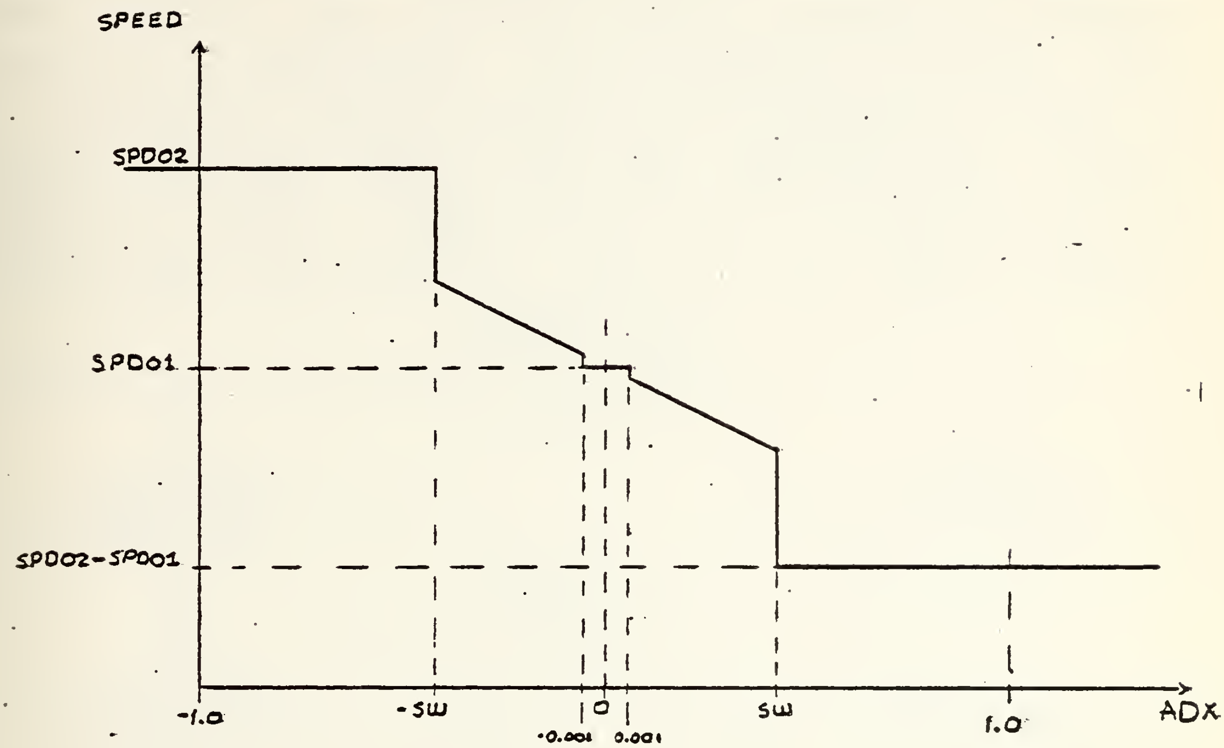


Figure III-75
Speed Control Law

The speed control law as explained above is shown in figure III-75 for an initial approach speed of SPD02 and a final estimated reference speed of SPD01, with ADX being the dynamic position feedback defined as the longitudinal distance between centers of the ships referenced to the controlled ship's heading. Analytically, the linear portion of the control law is written as:

SPDCTR = -ADX*(SPD02-SPD01)+SPD01

Symetric continuation of the control law accounts for operation on both sides of the operating point.

2. Optimization

Using this much simplified model of chapter II and the basic control law of figure III-75, the desired switching curve can be established. An optimization subroutine such as Subroutine BOXPLX can be used to iteratively obtain the optimum switching position (SW) for representative initial approach speeds. Figure III-76 is a flow chart of the subroutines and functions required for speed control

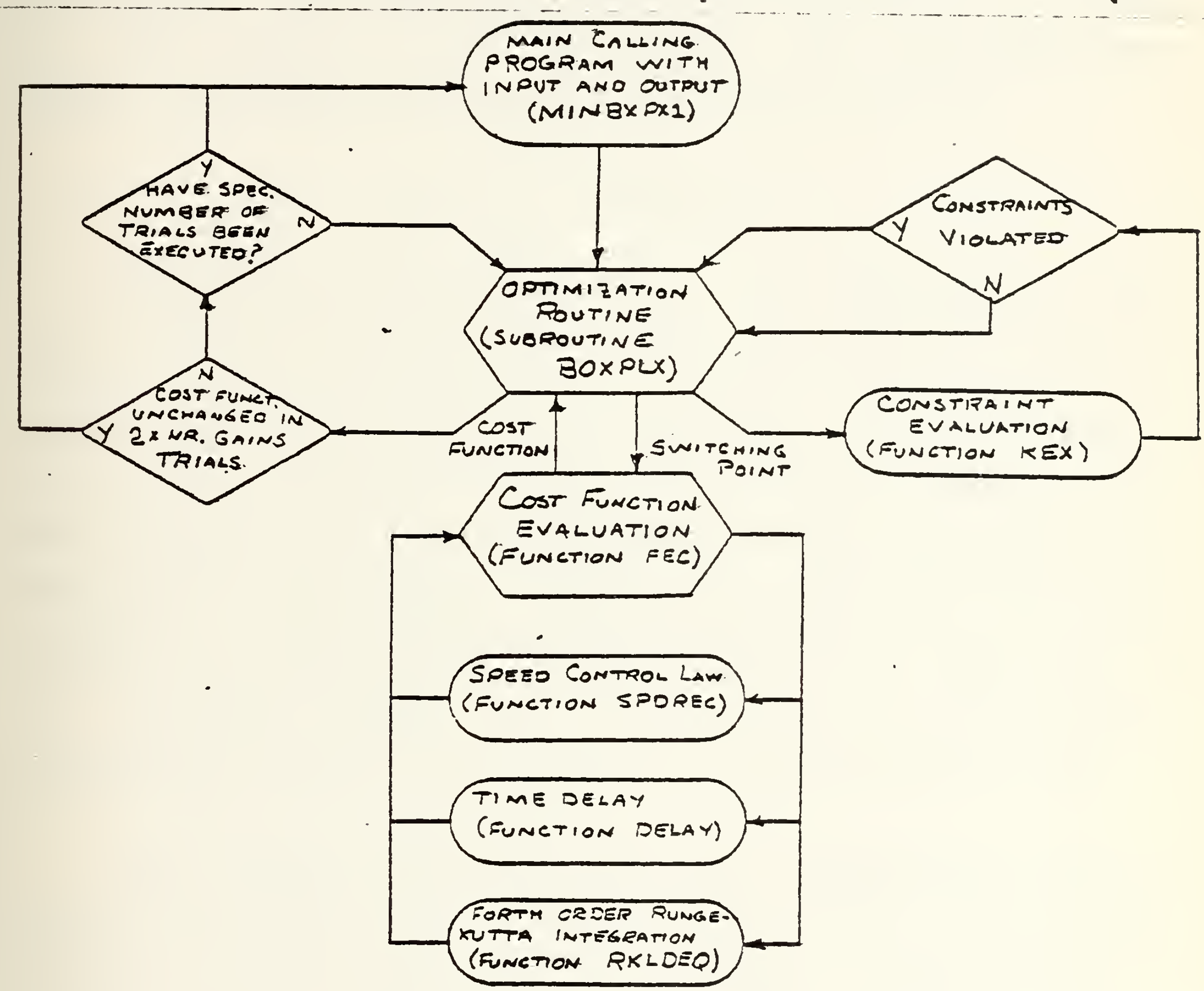


Figure III-76
Optimization Flow Chart

optimization. The major merit of this nonlinear control law stems from the predetermination of the switching point for all possible conditions of initial speeds. This apriori knowlege allows for offline computation of the switching position prior to commencing the approach. The cost function used for optimization is the ITAE which accomplishes two objectives. First, it forces the approach to be accomplished in minimum time. Secondly, it insures that the fuel expenditure will be optimized in the elimination of most overshoot and bang-bang control in the dead zone portion of the control law. The final value of the position error must be within the specified dead zone and the terminal speed must match the reference speed (SPDO1). The cost function has the following form:

$$J = \int_{t_0}^{t_f} (t \bullet |ADX|)\,dt$$

Table III-4 is a comparison of the optimization runs with various initial speeds. The values shown for SW must be multiplied by the speed differential (SPDO2-SPDO1) to obtain the corresponding value of ADX. The max/min values show the band of values which produce the optimum cost. This range of values is attributed to the integration step size used in the optimization program. Experience with this particular optimization program indicates that erroneous values of the switching point are found if the step size is not carefully chosen. The step size may be adequate for integration, but not for location of the switching point.

The points obtained from the optimization runs are plotted in figure III-77. These points define the nonlinear switching curve which must be stored in the computer to insure optimal operation of the speed control for all approach speeds. From here there are many procedure options open. These options have as a goal some usable form for predicting the optimal switching point for any set of initial conditions. One may choose linear straight line segments with an interpolation routine, or a closed form switching curve polynomial. Due to the availability of a

INITIAL CURVE POINTS					
SPDO2	SPDO1	SW MAX	SW MIN	SW	COST
1.1		.545*	.545*	.545	22.340515**
1.2		.58705	.58424	.585	5.733367
1.3		.62656	.6256	.626	2.700768
1.4		.6845	.68234	.683	1.672599
1.5	1.0	.73169	.7283	.729	1.223071
1.6		.7644	.76142	.763	0.992283
1.7		.7945	.7926	.7936	0.861552
1.8		.82178	.81945	.82	0.774621
1.9		.8501	.8439	.85	0.757244
2.0		.8673	.86375	.865	0.730168
CURVE CHECK POINTS					
1.5	1.1	.6859	.6823	.683	1.668055
1.5	1.2	.6307	.6297	.6302	2.691659
1.6	1.2	.67965	.67906	.6793	1.659956

* cpu usage over 4 min. - run not complete.

** cost function based on 20 min problem time
all others based on 10 min problem time.

Table III-4

Optimization Results

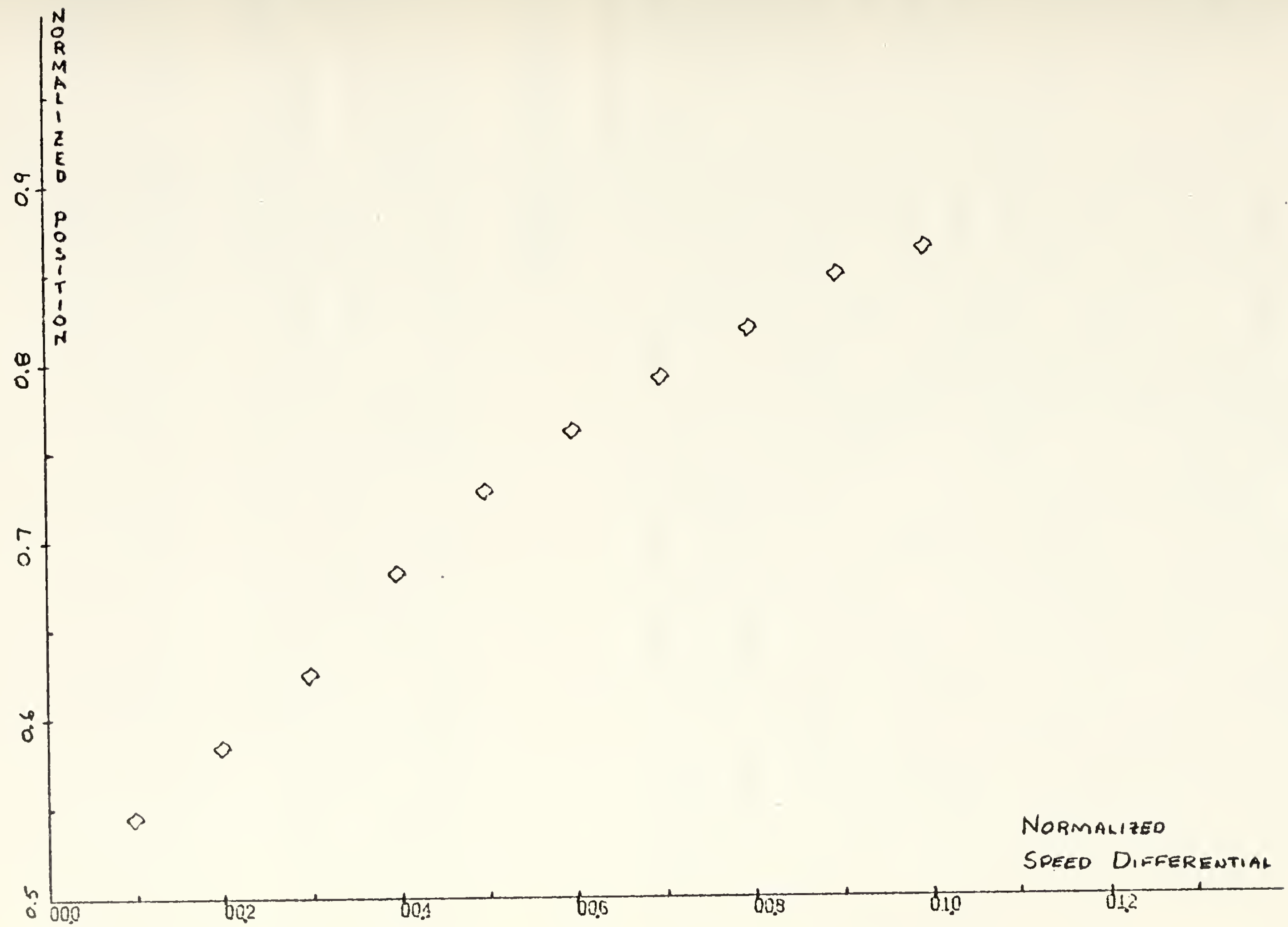


Figure III-77

Switching Curve Minimization Results

hybrid configured XDS 9300 digital computer and AGT-10 graphics terminal, the latter course was taken.

A polynomial curve fitting algorithm was used to obtain the required polynomial coefficients of best fit. This was done for polynomials of order 1 thru 5. The coefficients and the sum of the squares of deviation from the original points are tabulated in table III-5. The selection of the order to be used is highly dependent on the degree of accuracy required. In the RAS problem, the average error introduced for a first order fit is 8.0 feet(1.07 sec), while the fifth order fit introduces an average error of 1.35 feet(0.180 sec). Prior acceptance of errors introduced by an integration (and problem) step size of 0.8 sec allows for use of a second order fit without any degradation of simulation accuracy [second order average error is 2.848 feet (0.38 sec)]. The graphic display of figure III-78 indicates very little difference in the switching curves for second to fifth order polynomial fits. For the sake of accuracy, and owing to the computer control methods of this thesis, the fifth order polynomial fit shown separately in figure III-79 is used for determination of the switching point location.

3. Control Testing

A true test of the control law is accomplished when it is introduced in a computer program for a complete RAS simulation. Considering the performance of this controller in a complex environment of full scale RAS simulation allows maximum verification of the controller design.

The scenario for this simulation initially positions the ships such that the ship being controlled starts an approach 5 ship lengths (2639 feet) behind the reference

POLYNOMIAL DEGREE	COEFFICIENTS OF POWER:						SUM OF SQUARES OF DEVIATION
	5	4	3	2	1	0	
1					0.367928	0.524	2.2911×10^{-3}
2				-0.194621	0.582011	0.481	2.9117×10^{-4}
3			-0.174164	0.0927511	0.449472	0.496	1.9748×10^{-4}
4		0.748543	-1.82096	1.29791	0.120113	0.521	1.0517×10^{-4}
5	-2.24869	6.93243	-8.04233	4.08065	-0.409977	0.554	6.5732×10^{-5}

Table III-5

Polynomial Curve Fit Results

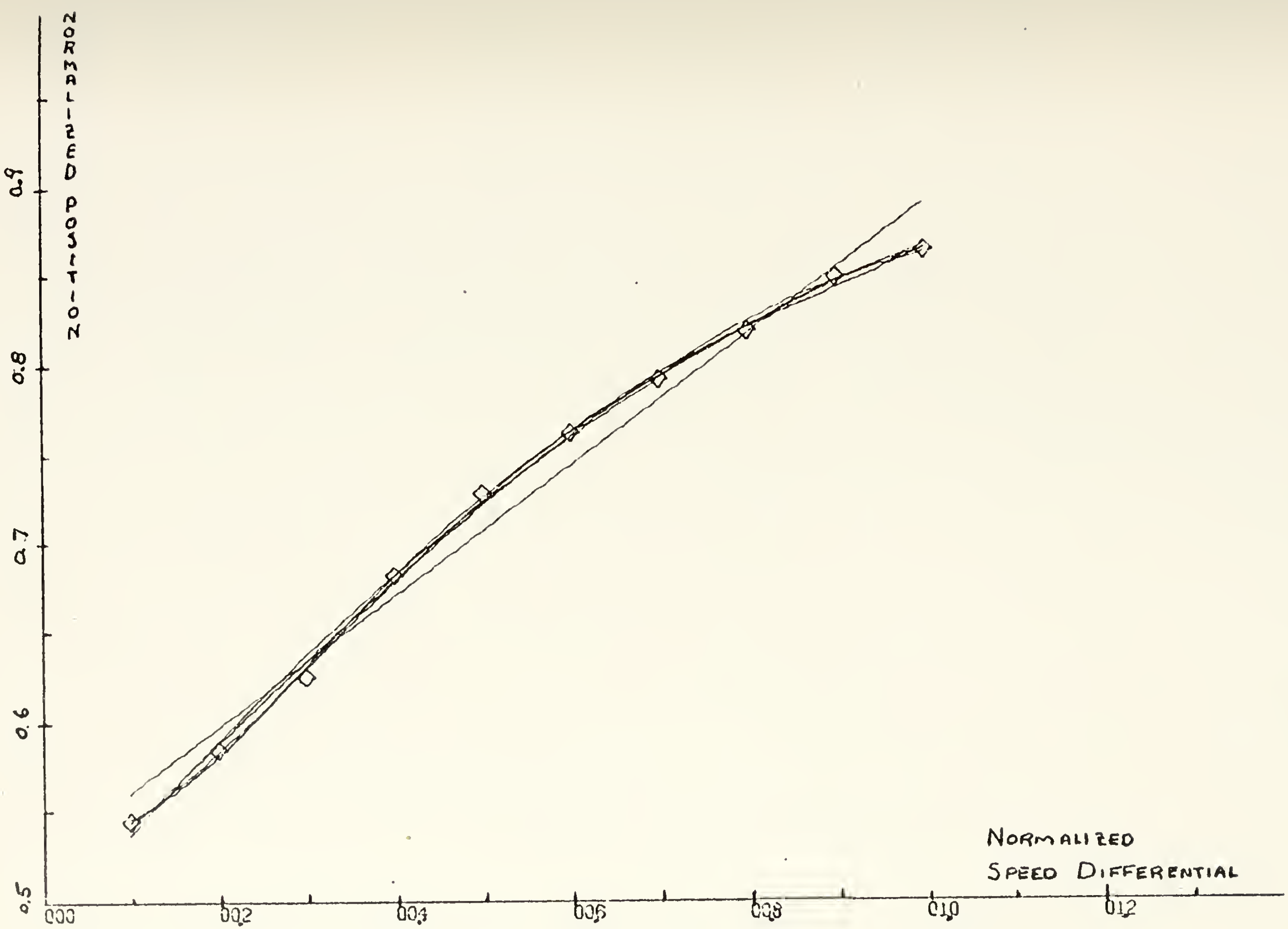


Figure III-78

First Thru Fifth Order Polynomial Curve Fit Results

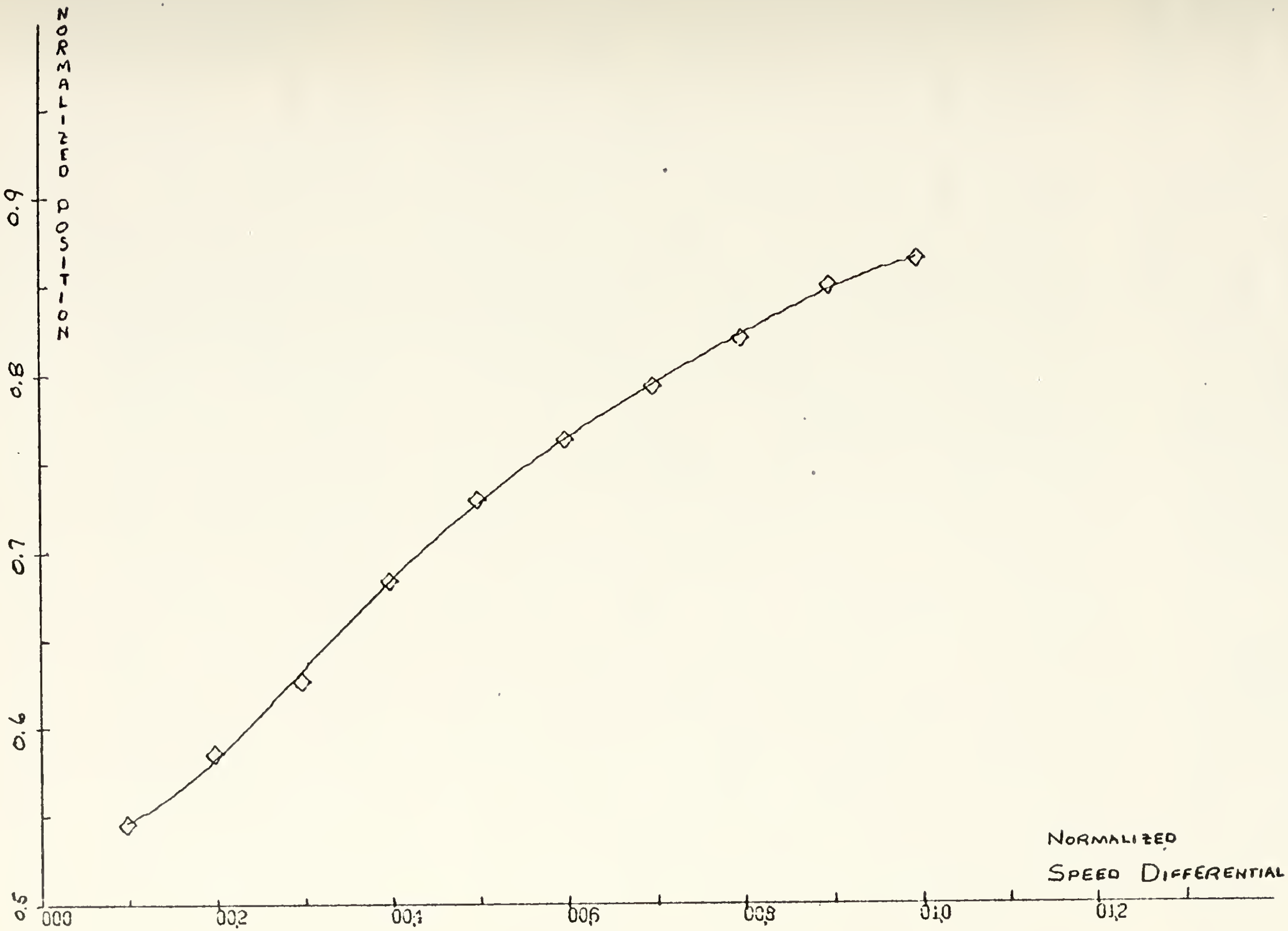


Figure III-79
Fifth Order Polynomial Curve Fit

ship and displaced 0.4 ship lengths (211 feet) to the right. The desired final position is alongside and displaced 0.2 ship lengths (106 feet). The heading control system used is developed in section A of this chapter.

The approach phase is accomplished with the speed desired and speed acquired shown in figure III-80 with the corresponding position attainment exhibited in figure III-81. These plots show excellent switching and optimal position attainment.

The next step is to insure that the position keeping loop will maintain the desired position with an induced perturbation. This is accomplished by turning the reference ship away from the control ship a total of 15 degrees to observe the reaction of the speed control loop. The reference ship's turn causes the relative motion between the ships to be altered, making the control ship lag the desired position. The nonlinear control system is designed to correct this situation as soon as the actual position is outside the limits of the dead zone. Figure III-82 displays the desired speed and acquired speed for the control ship. Figure III-83 indicates that the corresponding position deviates from the desired by 0.0154 ship lengths (8.13 feet) at the maximum excursion. This is well within the limits of acceptability for such a drastic perturbation.

The introduction of velocity control was accomplished by combining the simplified engine response of chapter II and the speed control law developed here. By setting the speed desired (SPDDES) equal to the output of Function SPDREC and scaling the speed error (SPDERR) to the nondimensional equations of motion, the velocity loop is initiated. The auxillary equations added to those presented in chapter II are:

Figure III-80

RAS Speed Control Approach Phase

Speed Desired (1) and Speed Acquired (2) vs Real Time

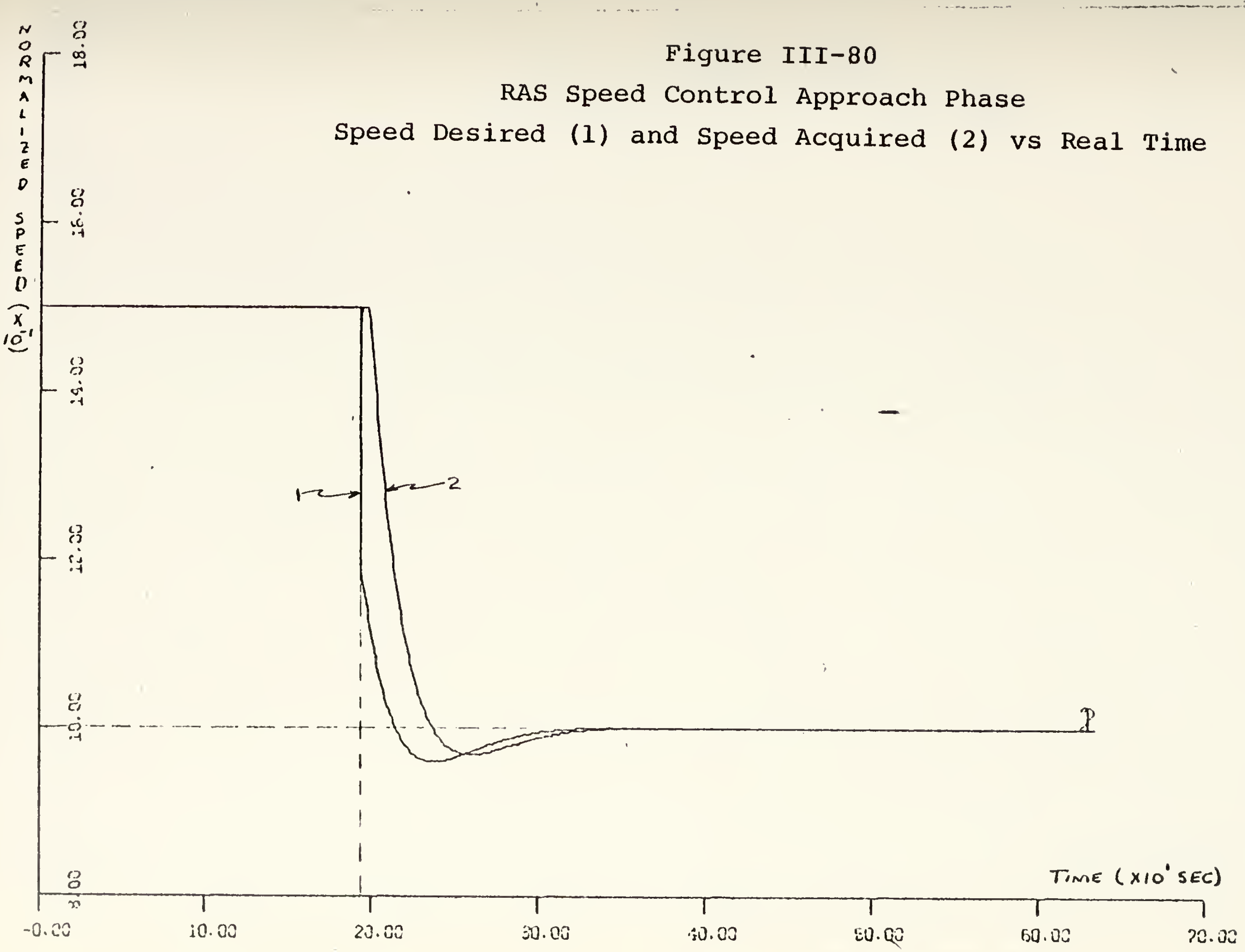


Figure III-81

RAS Speed Control Approach Phase

Position Attainment vs Real Time

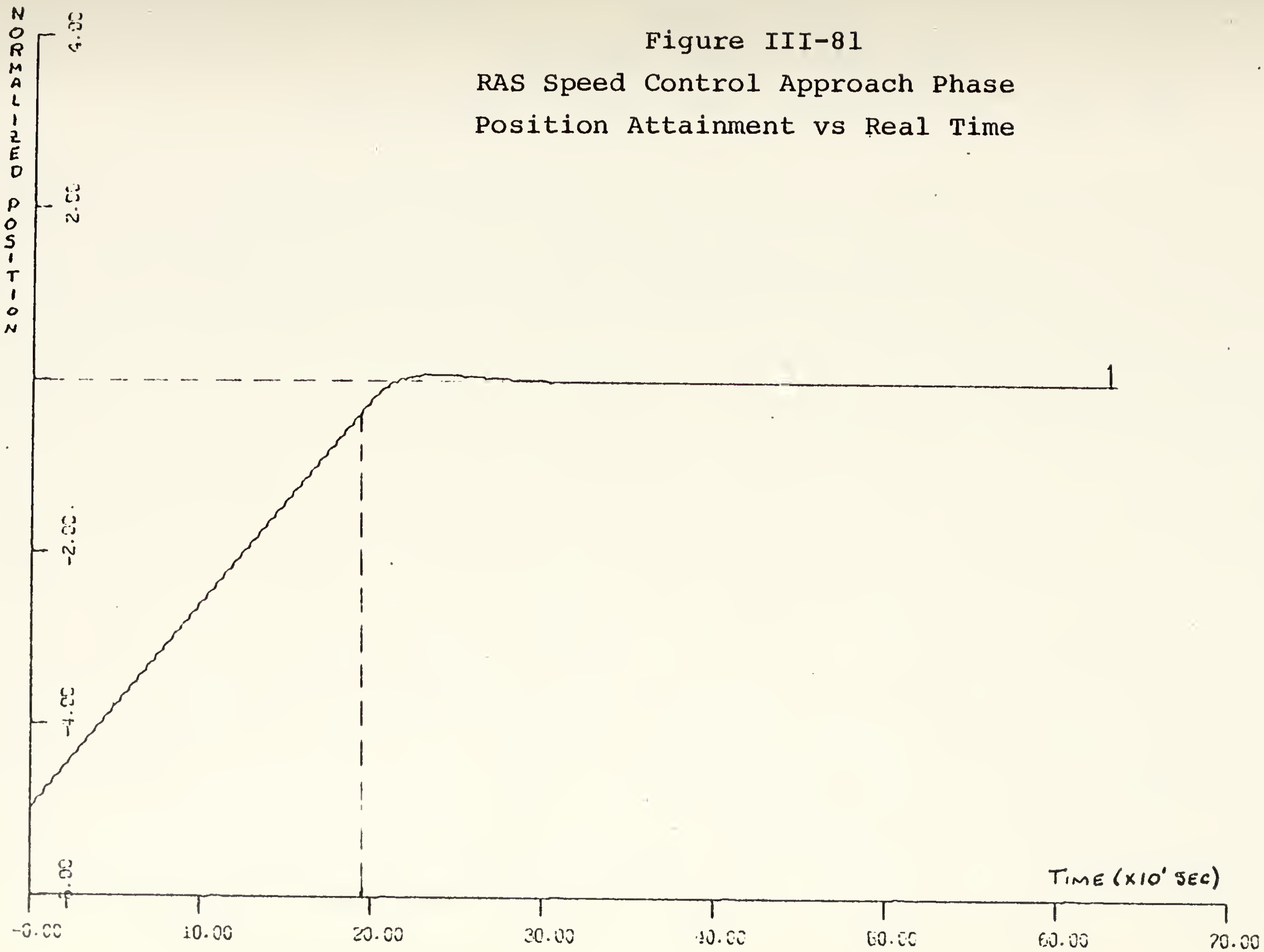


Figure III-82

RAS Speed Control Turn Phase

Speed Desired (1) and Speed Acquired (2) vs Real Time

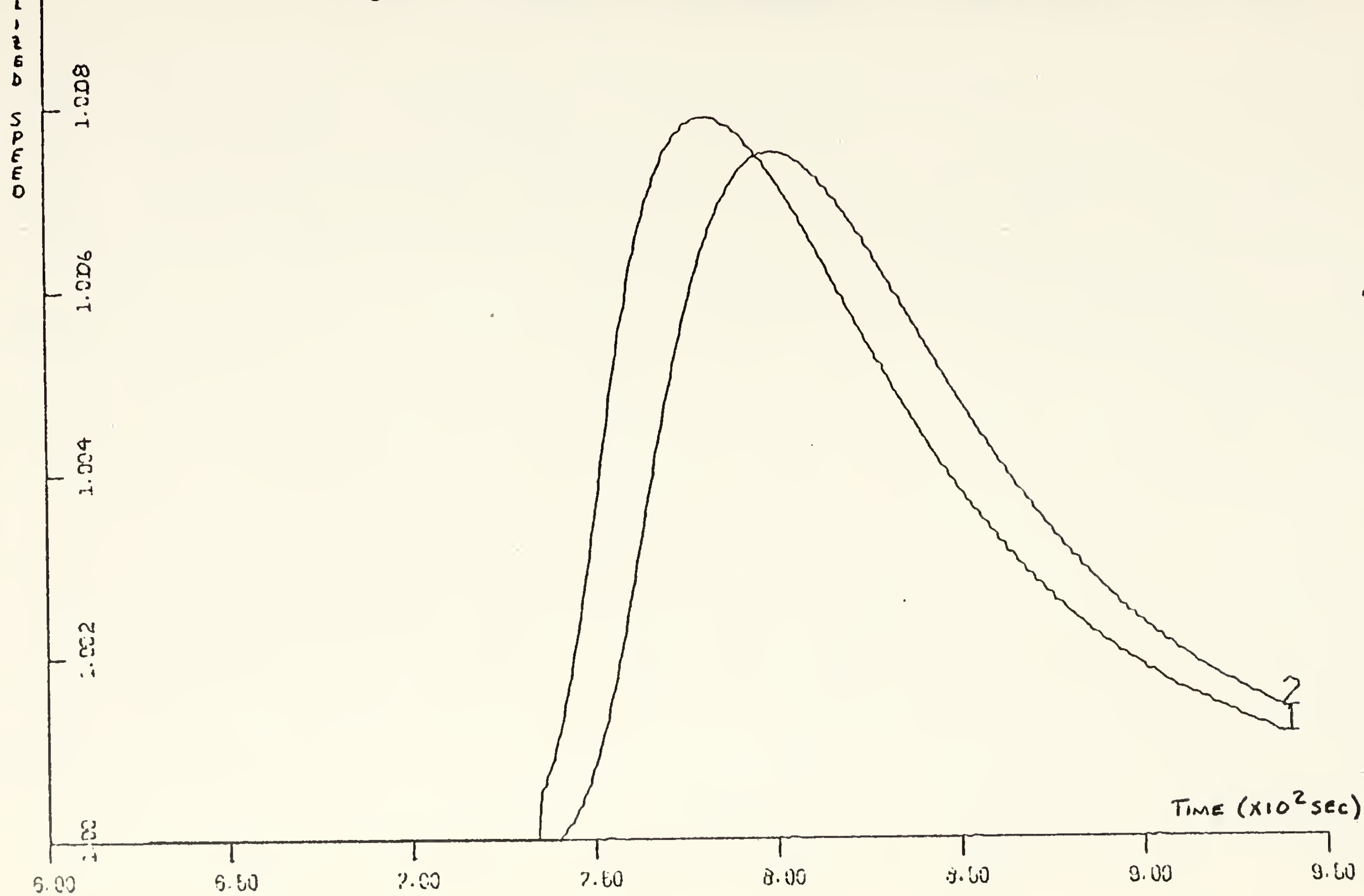
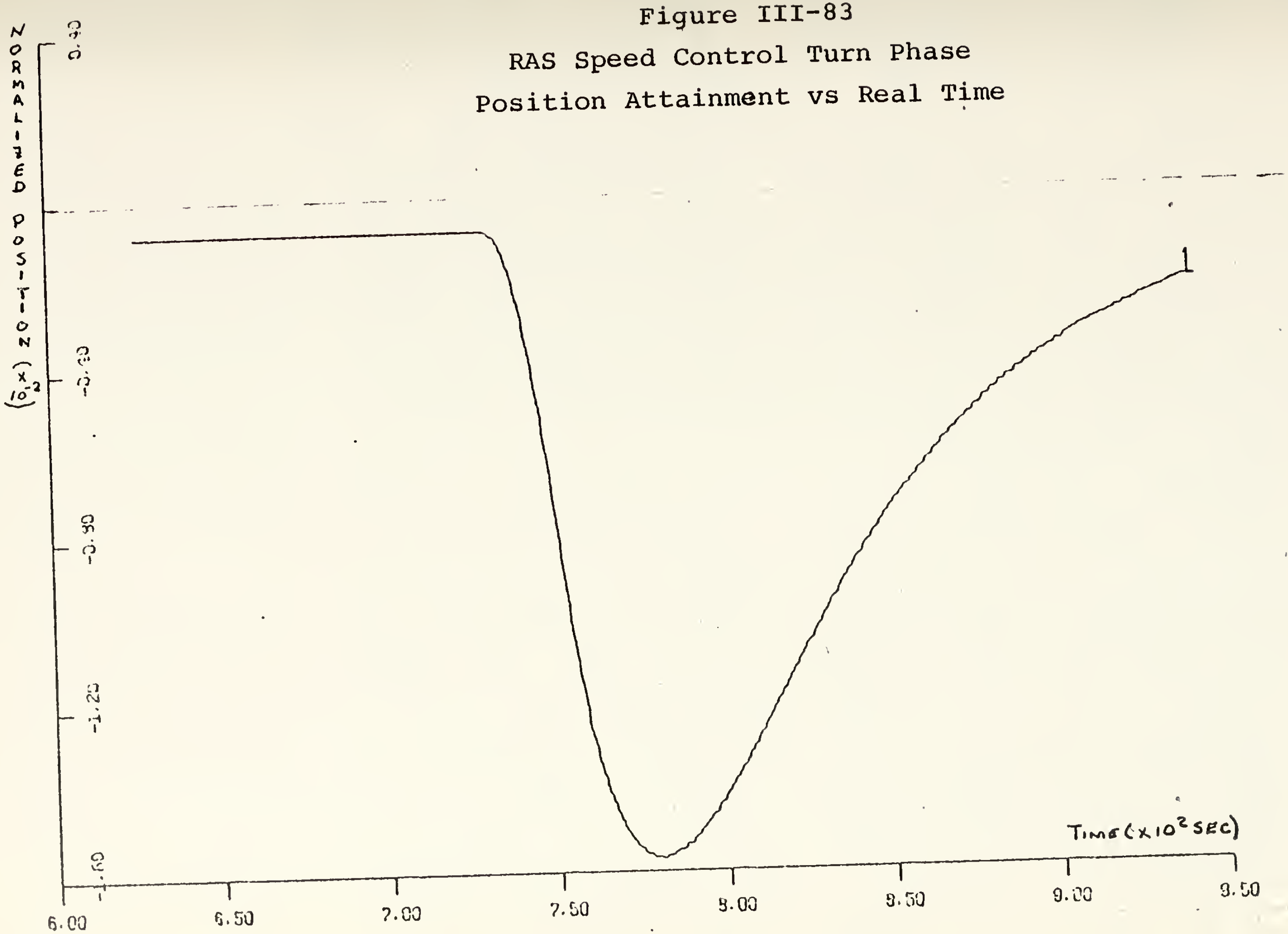


Figure III-83

RAS Speed Control Turn Phase

Position Attainment vs Real Time



```
SPDDES =  SPDREC(ADX,SPDO1,SPDO2,SW)
CDCT2 = INTGRL(UO2,SPDERR*LUC)
```

These equations are introduced in computer program #8 to produce figures III-80 thru III-83.

Further system study indicates that the reference ship speed must be known to a fairly high degree of accuracy. Without apriori knowlege of the reference ship speed, a constant bias is introduced. The amount of bias allowable defines the permissible uncertainty in the reference ship's initial speed. This bias can amount to as much as 0.1 ship lengths (84.48 feet) for a reference speed inaccuracy of 2.5 knots (0.1 normalized speed). However, it is felt that the reference speed in any practical situation will initially be known to within 0.5 knots (0.02 normalized speed). This more practical error will introduce a bias of only 16 feet.

Other feedback parameters can be used to offset the lack of apriori knowlege of the reference ship speed. Since the reference ship is tracked with a high accuracy range and bearing device and the controlled ship's speed is measured, a decoupled multivariable scheme is used to further refine the reference ship speed. With high resolution devices presently available[14], it is estimated that this can be done practically to within 0.05 knots (0.002 normalized speed). This would bring the offset bias to 1.6 feet; well within previously defined errors introduced by integration step size.

4. Longitudinal Position Offset

Throughout the development of the heading control and speed control, the scenario has followed the condition that the final position would be longitudinally alongside. Although this is a good assumption for ships of the same type, it does not account for RAS station differences for different ship types. To alleviate this disparity, function SPDREC was redesigned to allow pre-planned offset condition to exist. Function SPDOFC of appendix A is a result of this redesign.

Simulation runs, with a change of the speed control function only, resulted in some unstable conditions existing in the heading control loop. The cause of this phenomenon stems back to the adaptive gain scheme used and the changes made to force the control loop to a steady state value prior to a turn. By using a favorite ploy of experienced conning officers, this problem is alleviated. The ploy is to take the ship alongside and then either drop back to station or surge forward to station. This method is accomplished by setting the initial offset (XOFS) to 0.0. The final desired offset (XOFSD) is stored and not used until the ship is settled out alongside. It is subsequently used as shown in the following Fortran code:

```
IF (ATIME.GT.450.0) XOFS = XOFSD
```

This method solved the gain transition problem. It did not, however, give a completely stable simulation run. Unstable conditions still existed at the end of the turn phase. This is not suprising, considering the heading control optimization method used. The set of gains previously found were for the alongside scenario only.

Different interactive forces and moments at the offset position cause these gains to be no longer optimal.

By relaxing the control loop in the heading velocity feedback gain (VFBG), sub-optimal control at all practical offset positions is achieved. The gain VFBG was changed from 0.084028 to 0.1 in the turn phase adaptive gain schedule without significant loss of control efficiency for alongside operation (2.3 feet maximum excursion vice 2.0 feet previously obtained). Subroutine SWTCHF of appendix A reflects the gain change and offset calculations required. Computer program #9 incorporates the changes required for offset simulation. Table III-6 is a cross reference listing of the plots obtained. From these figures, the effect of different longitudinal positions is readily apparent. An offset of 0.1, equating to 52.8 feet, causes greater lateral excursions when astern (XOFSD = -0.1) of the alongside position than when ahead (XOFSD = 0.1). The longitudinal position maintainment, however, is essentially the same in all cases.

Run	Approach Phase Plots A	Approach Phase Plots B	Approach Phase Plots C	Turn Phase Plots A	Turn Phase Plots B	Turn Phase Plots C
XOFSD	0.0	0.1	-0.1	0.0	0.1	-0.1
Lateral Distance DY	84	86	90	94	96	99
Yaw Difference	85	87	91	95	97	100
Speed Response	80*	88	92	82*	82*	82*
Longitudinal Position DX	81*	89	93	83*	98	101

*Note: These plots are the same as those obtained from computer program #8 and are not repeated here.

Table III-6

Position Offset Testing Cross Reference

An alternative to the method shown here is again a completely adaptive gain scheme which would achieve optimal control instead of the sub-optimal control settled for here. The alternative may become even more important if the nonlinear terms of the equations of motion are considered. This would couple the heading and speed control designs to a larger extent than encountered in the interactive forces and moments.

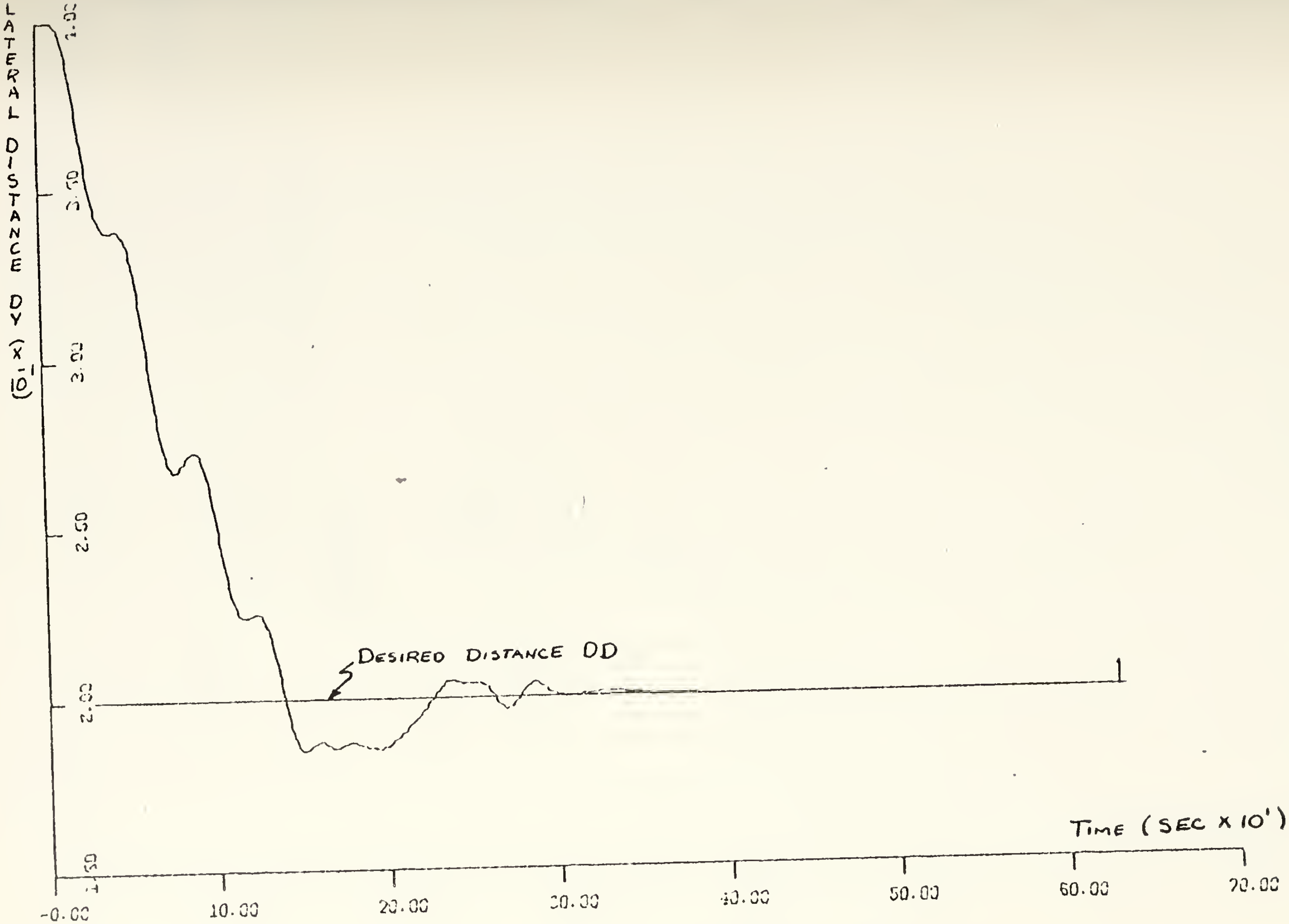


Figure III-84

Approach Phase Run A Lateral Distance DY

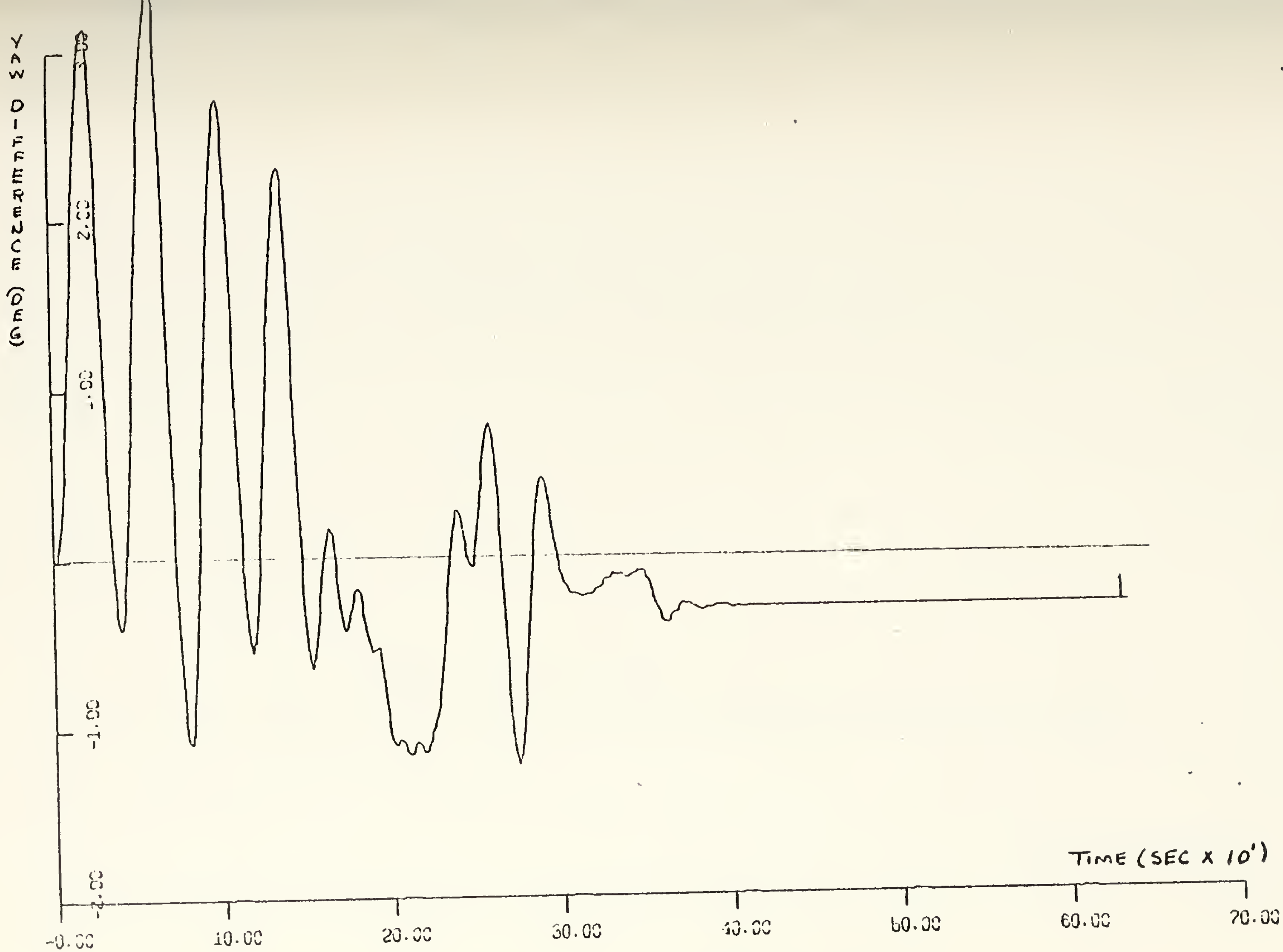


Figure III-85

Approach Phase Run A Yaw Difference

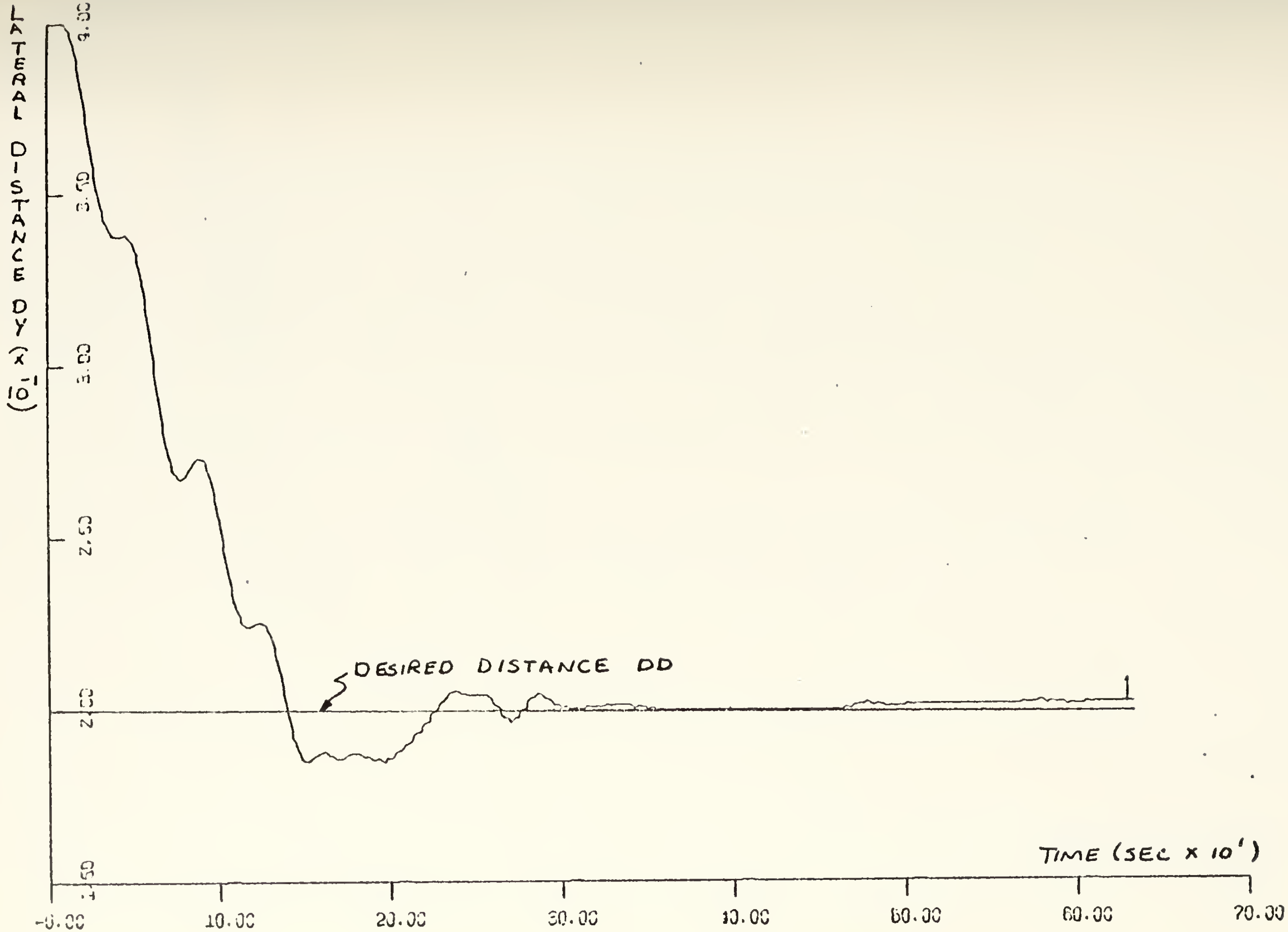


Figure III-86

Approach Phase Run B Lateral Distance DY

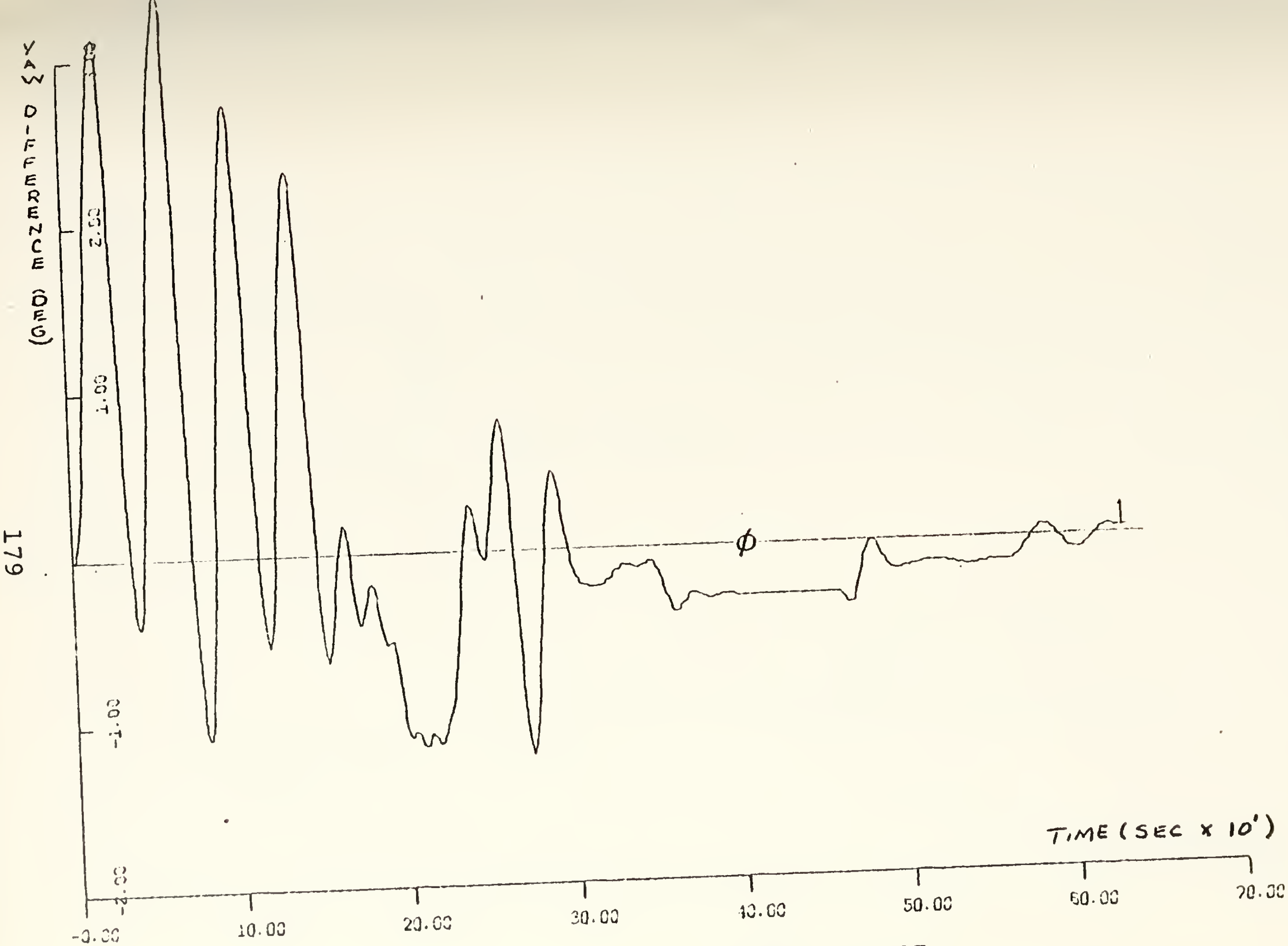


Figure III-87

Approach Phase Run B Yaw Difference

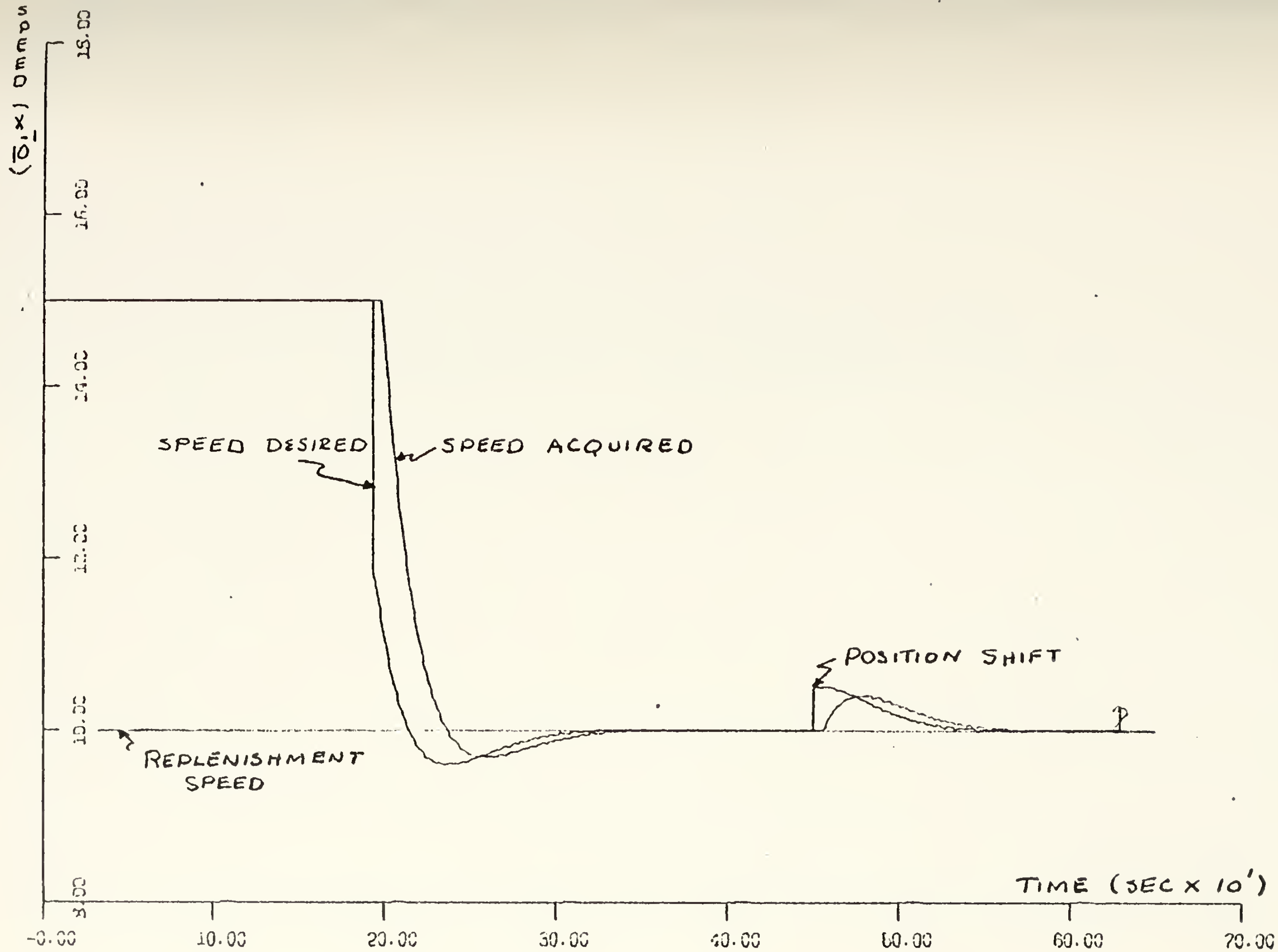


Figure III-88

Approach Phase Run B Speed Response

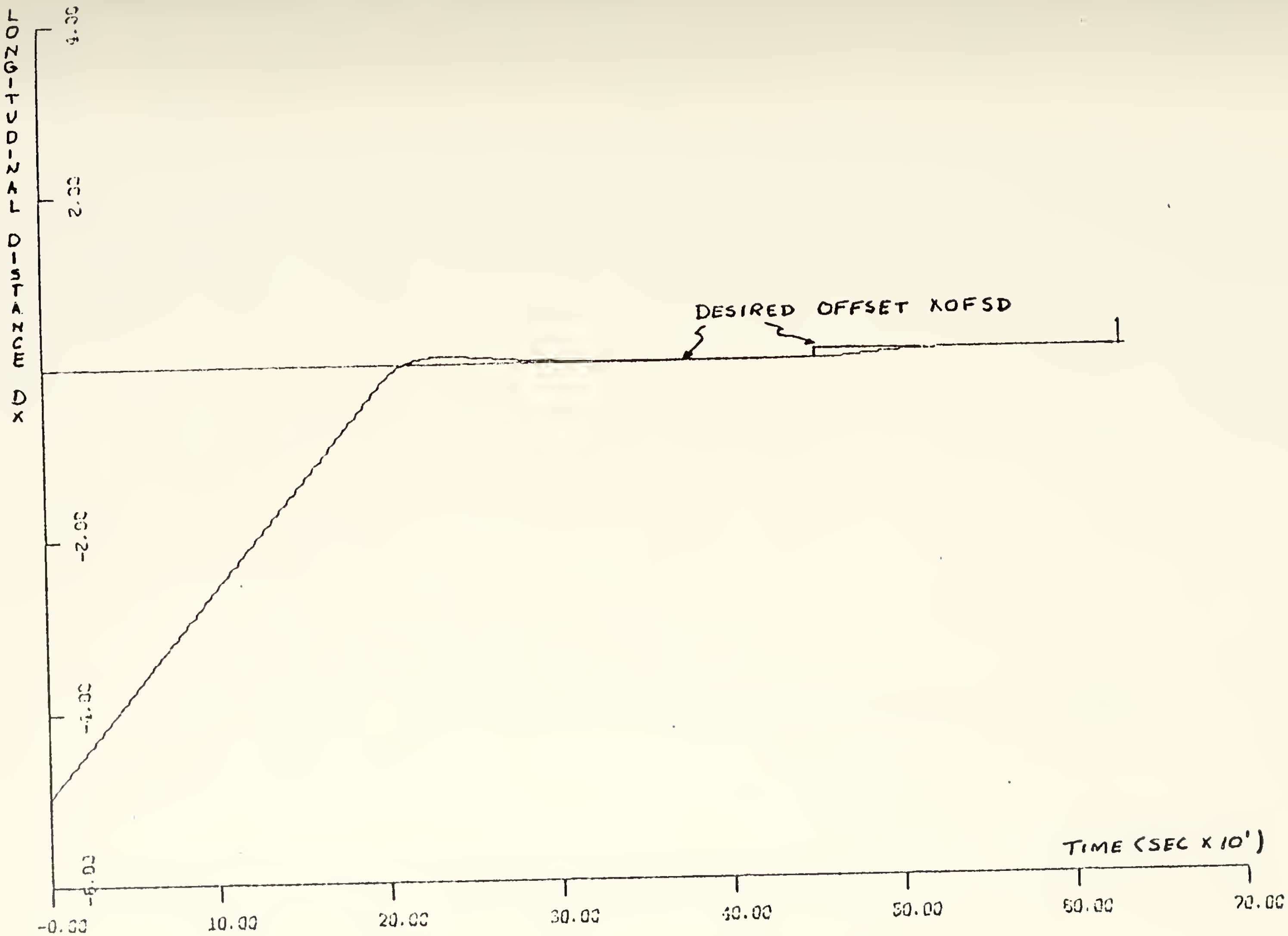


Figure III-89

Approach Phase Run B Longitudinal Position DX

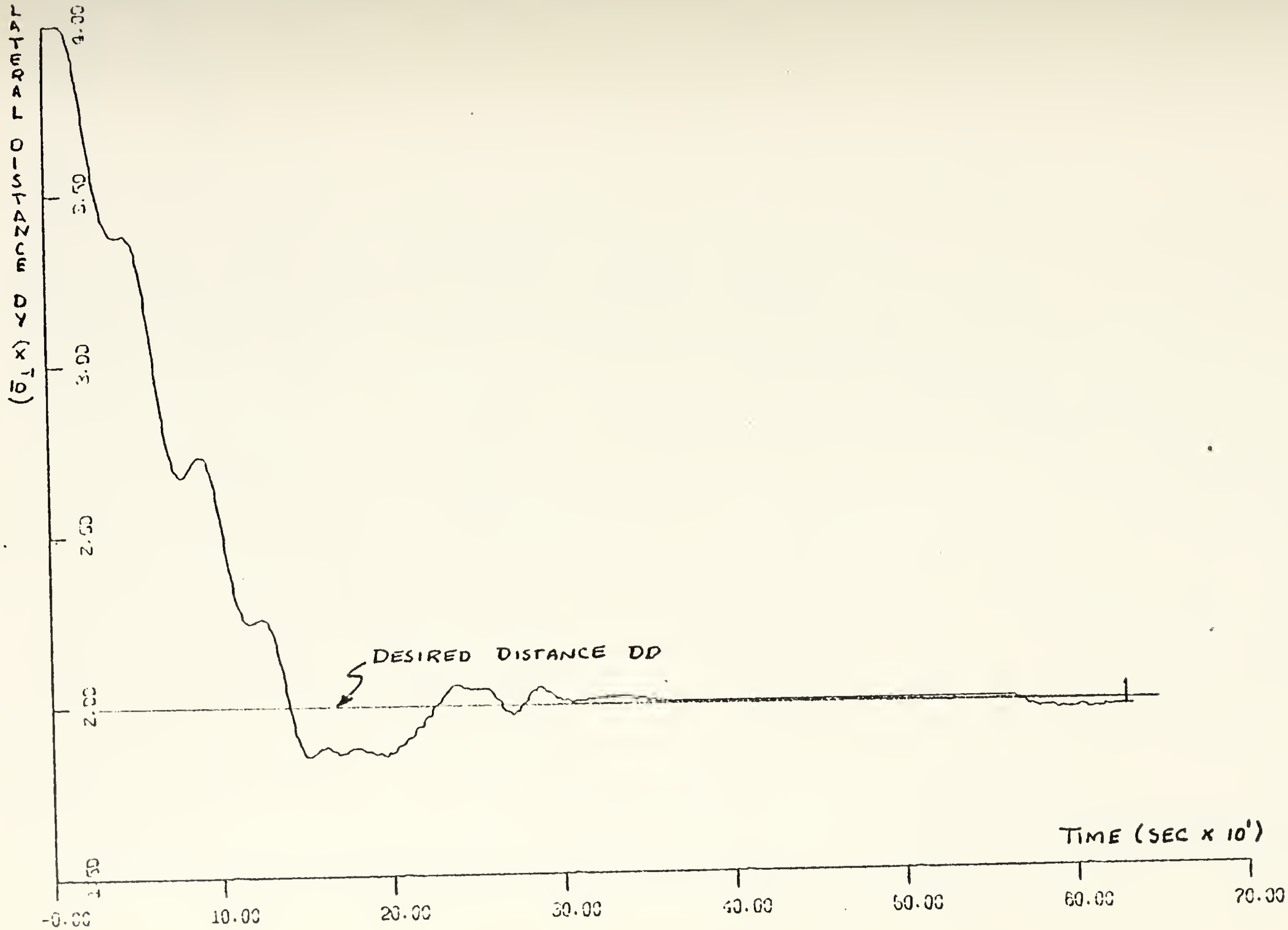


Figure III-90

Approach Phase Run C Lateral Distance DY

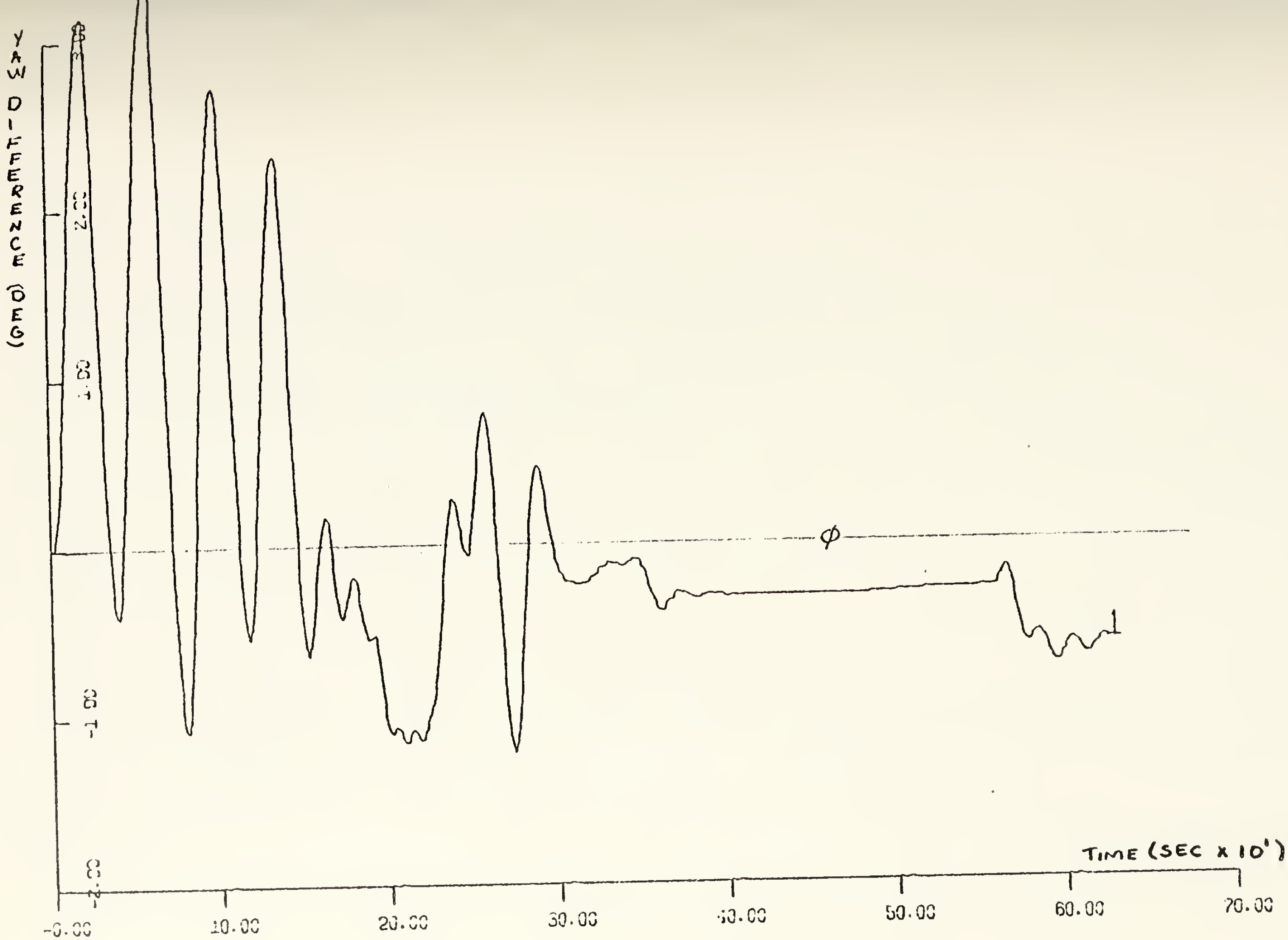


Figure III-91

Approach Phase Run C Yaw Difference

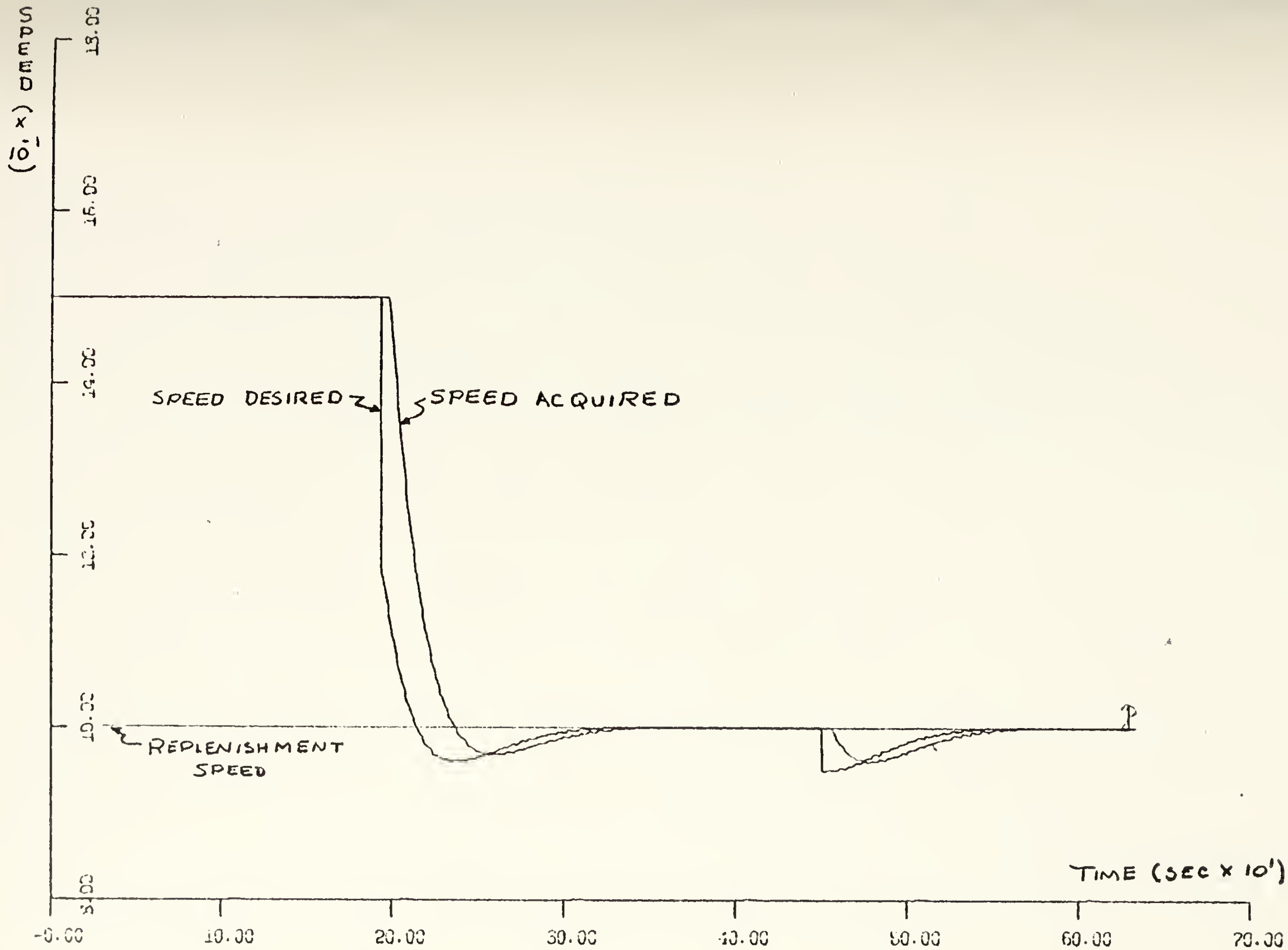


Figure III-92

Approach Phase Run C Speed Response

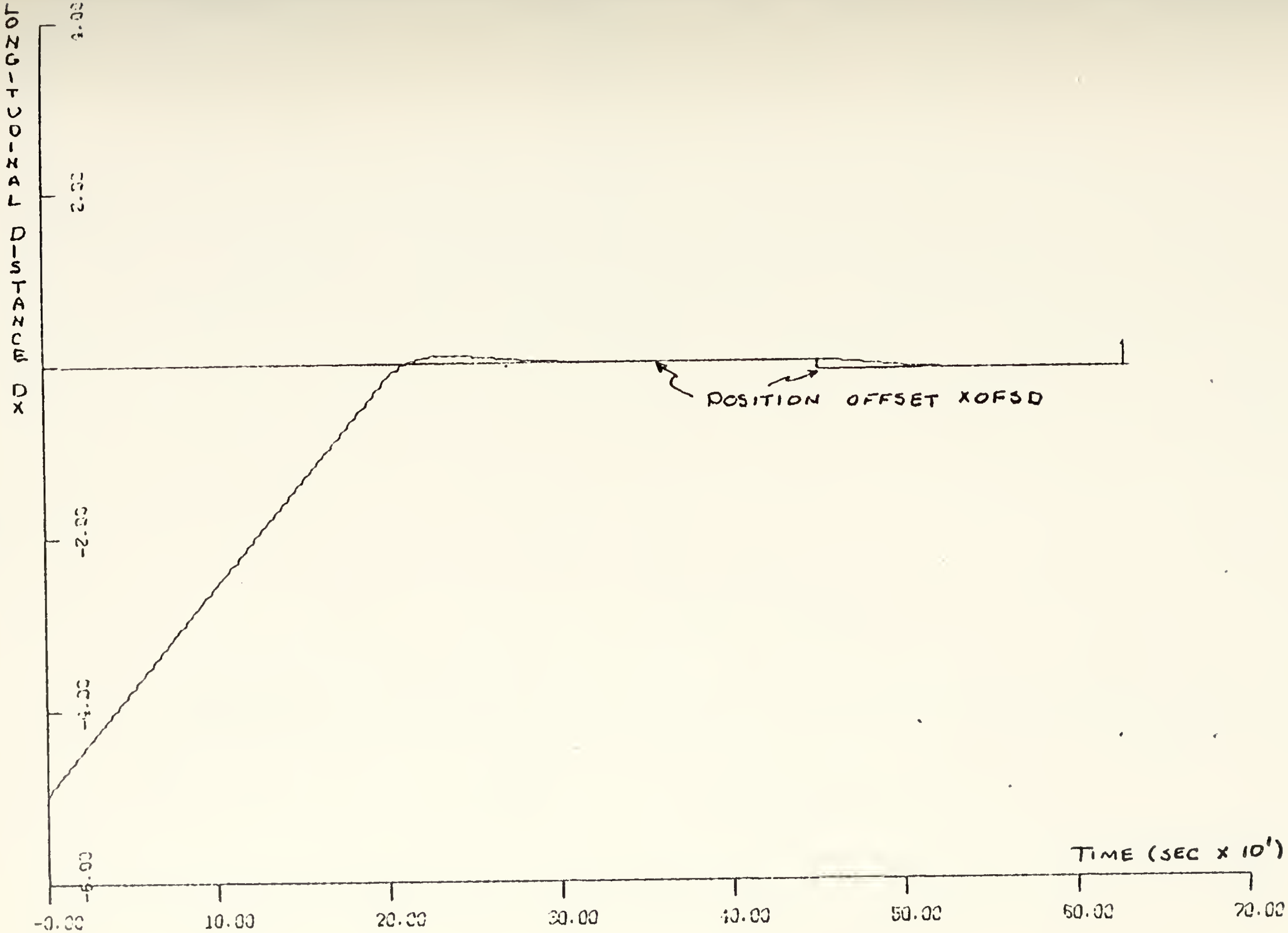


Figure III-93

Approach Phase Run C Longitudinal Position DX

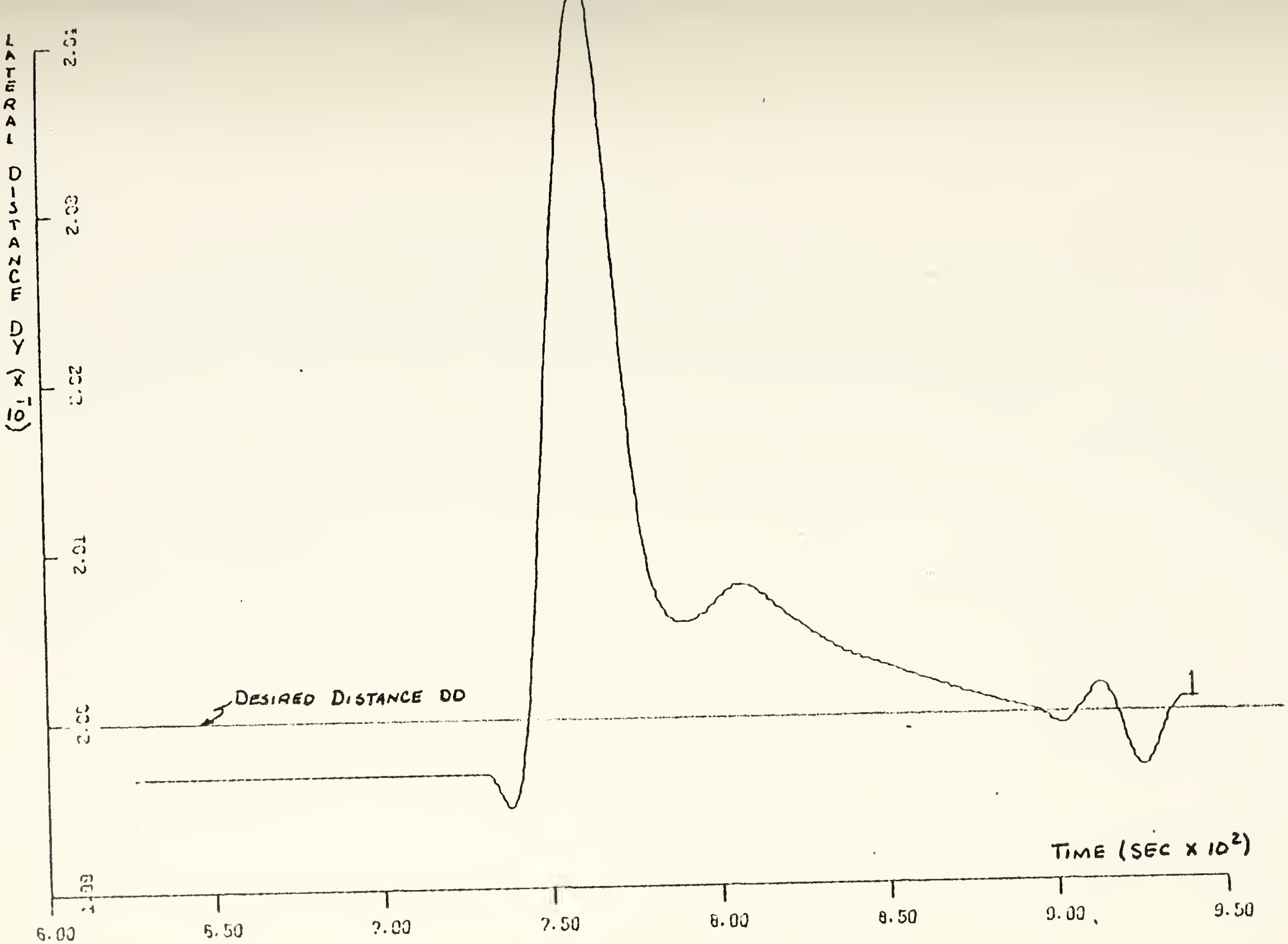


Figure III-94

Turn Phase Run A Lateral Distance DY

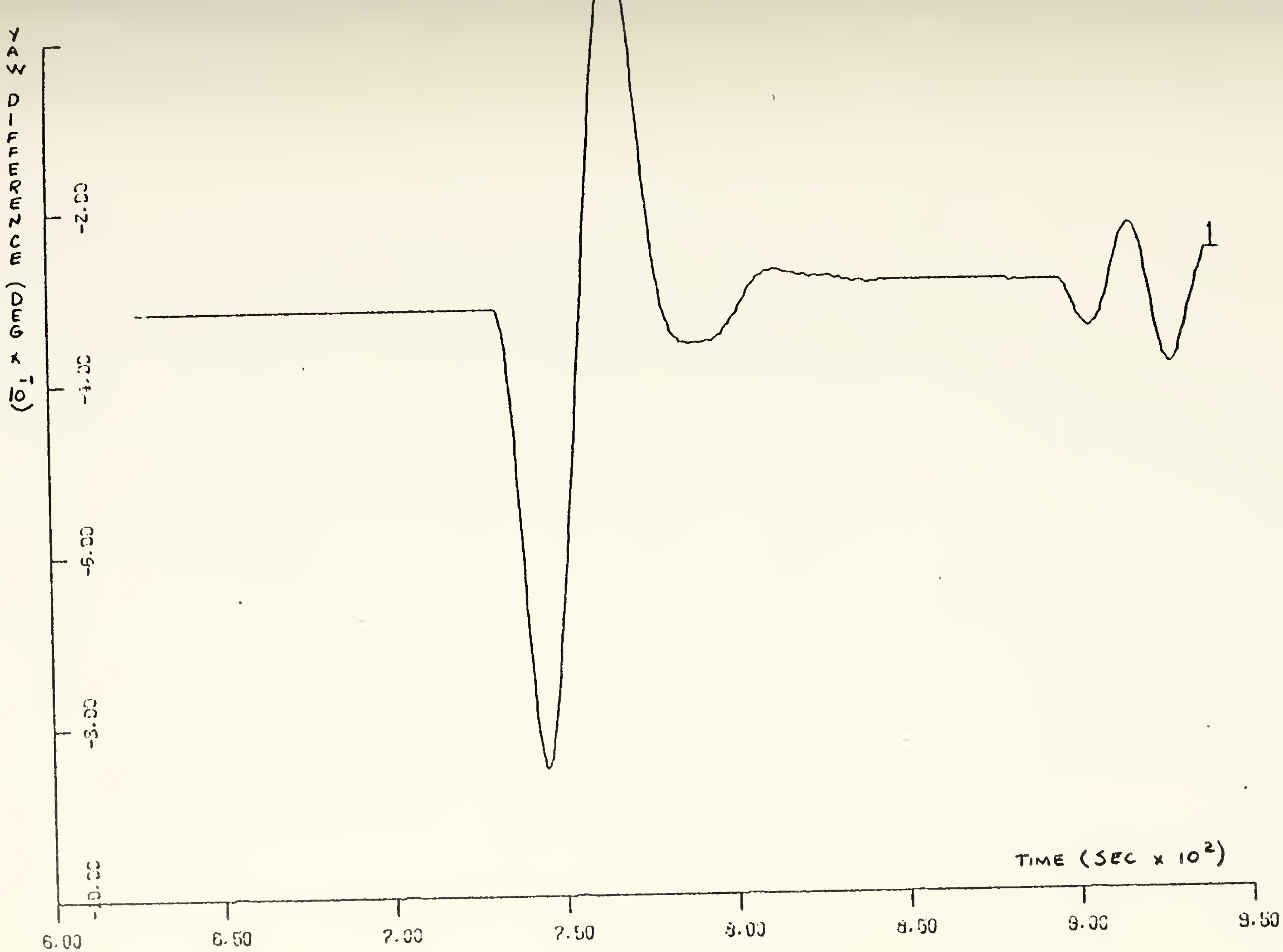


Figure III-95

Turn Phase Run A Yaw Difference

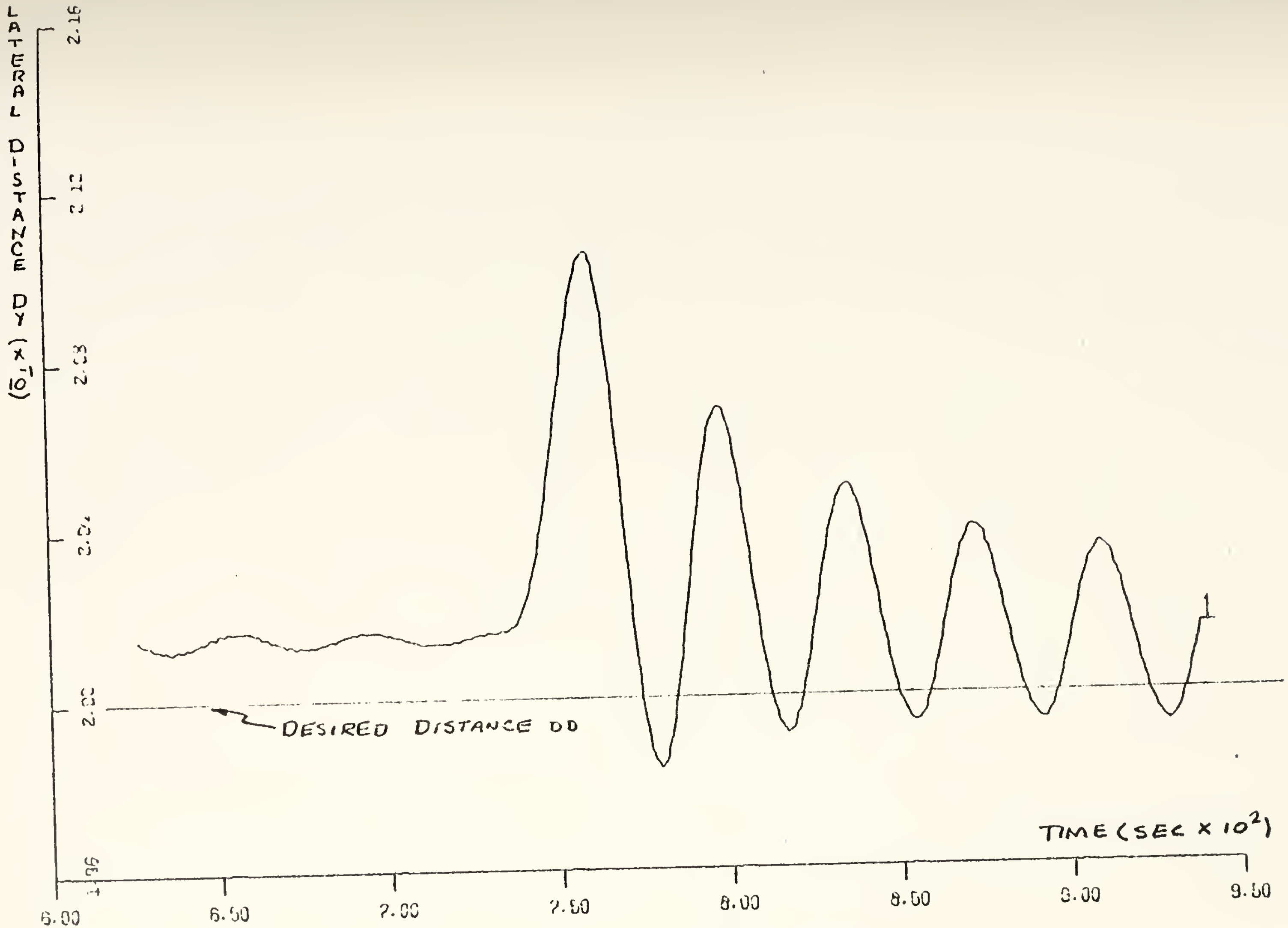


Figure III-96

Turn Phase Run B Lateral Distance DY

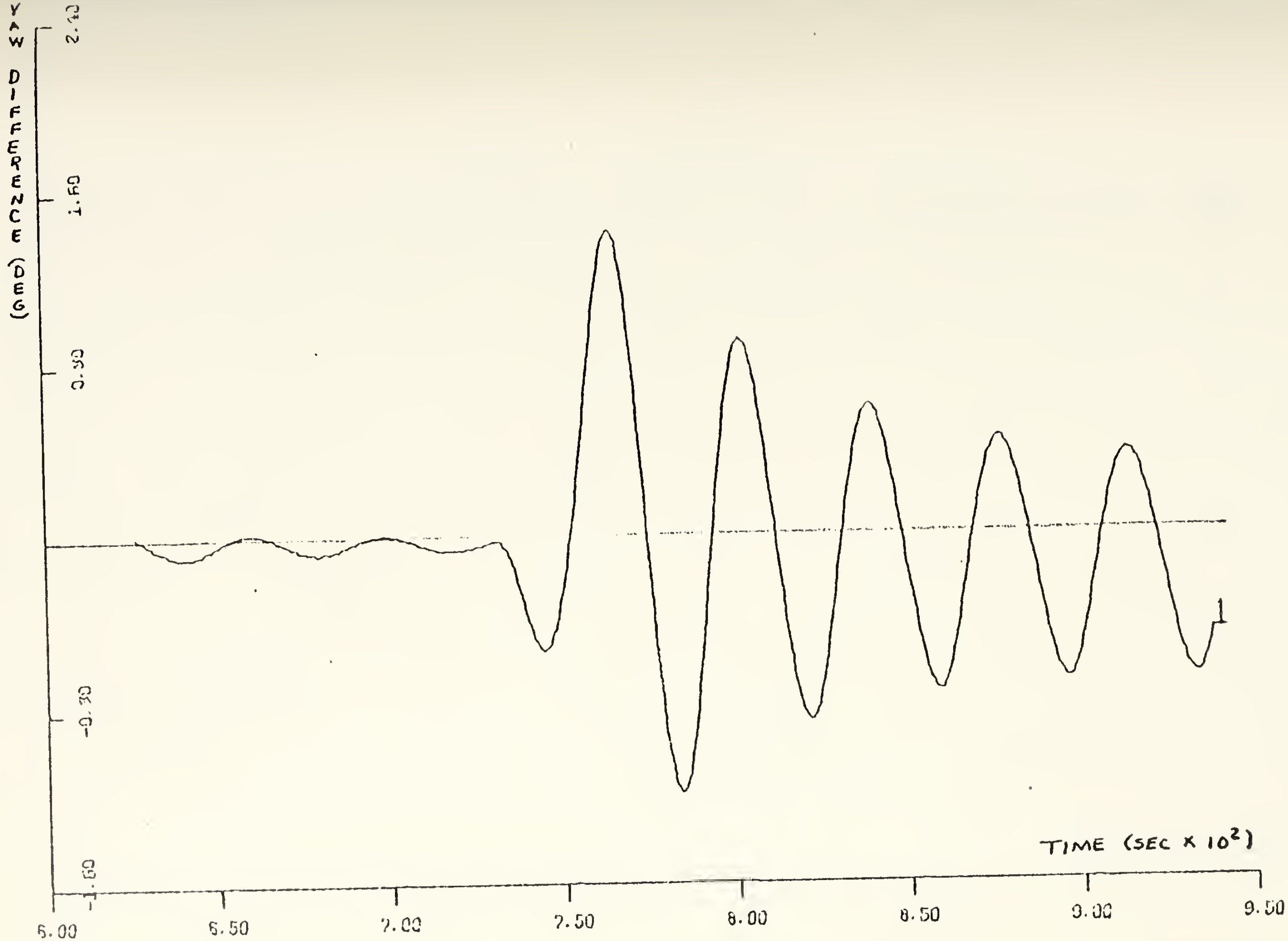


Figure III-97

Turn Phase Run B Yaw Difference

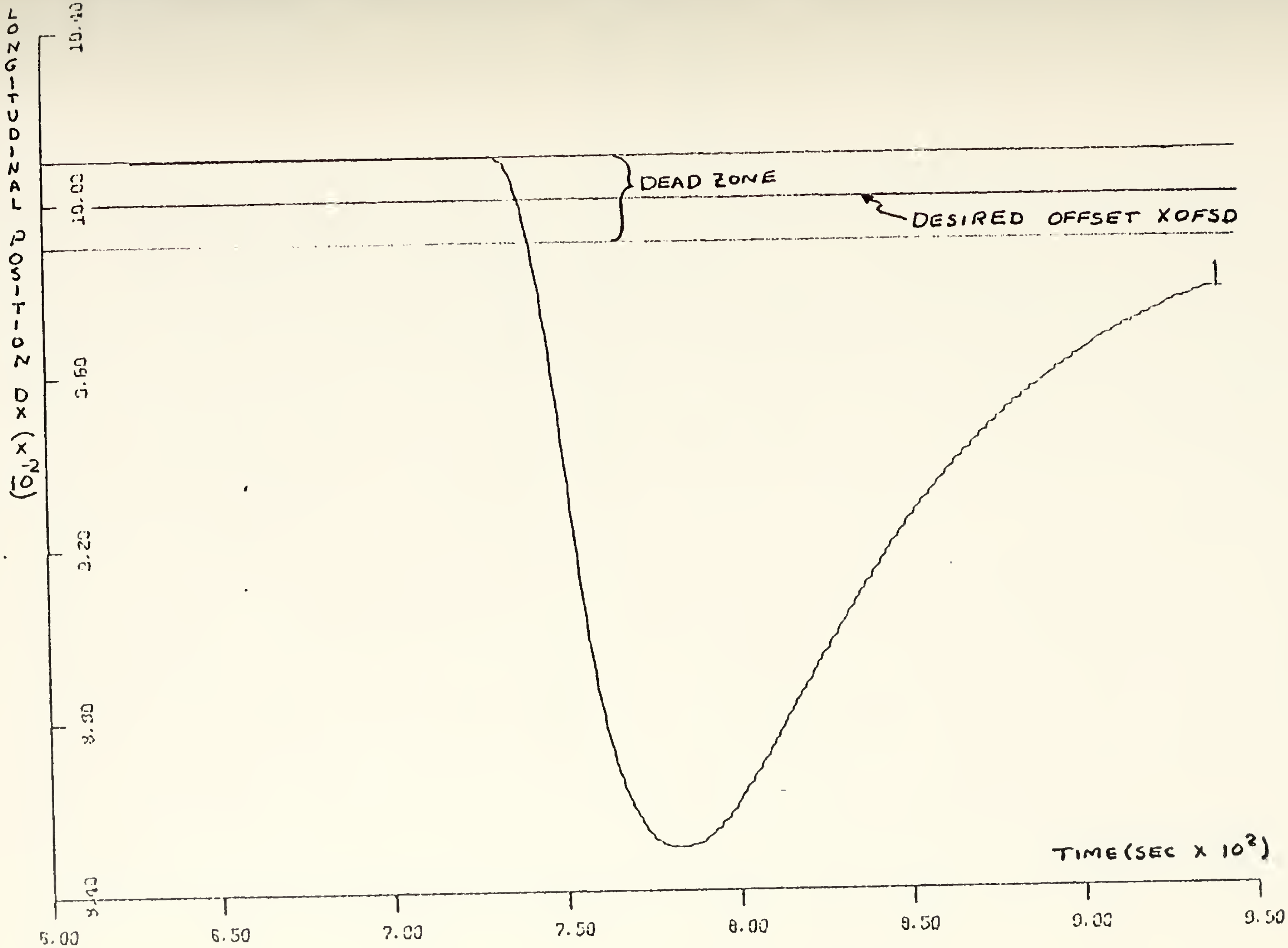


Figure III-98

Turn Phase Run B Longitudinal Position DX

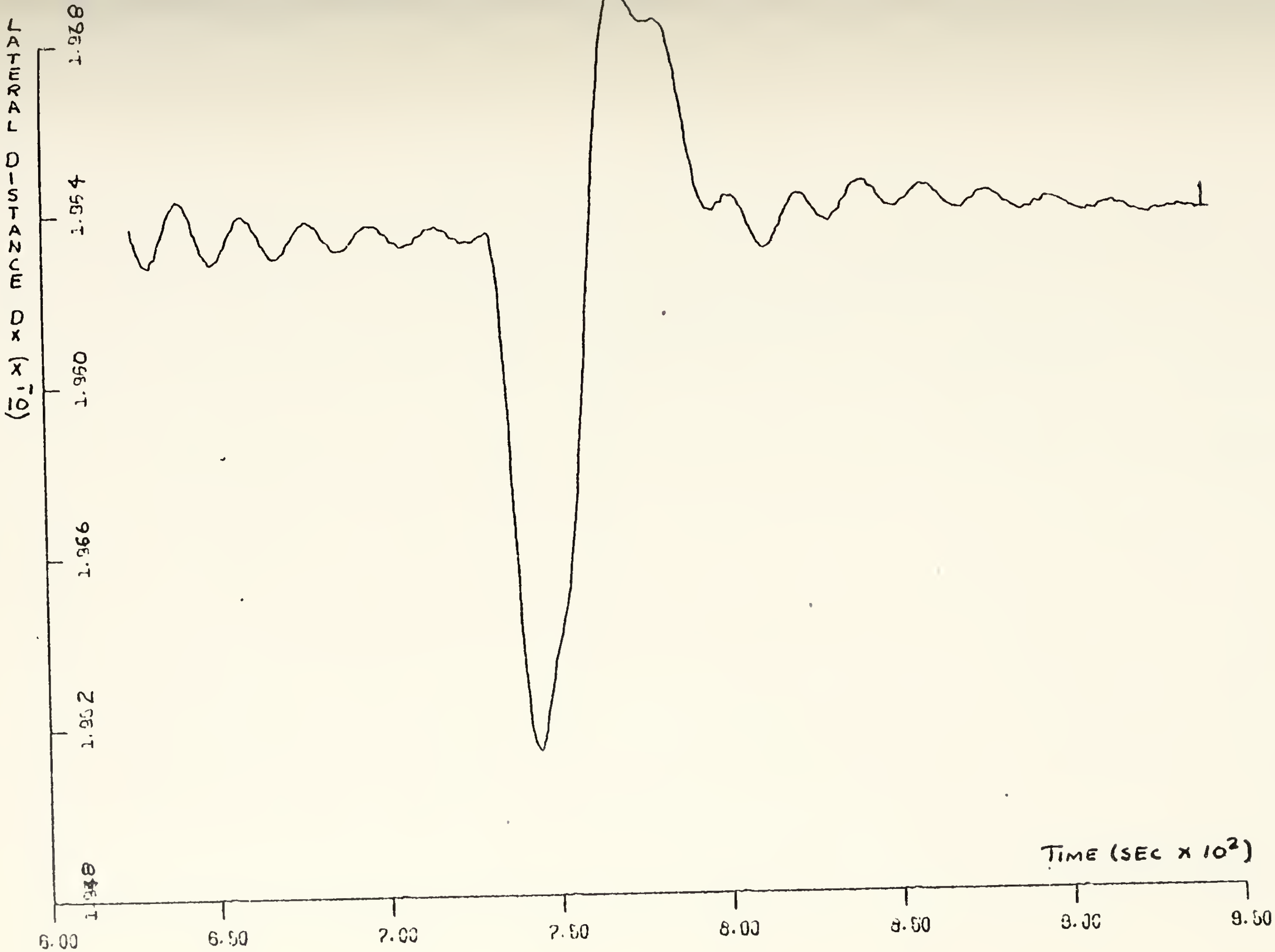


Figure III-99

Turn Phase Run C Lateral Distance DY

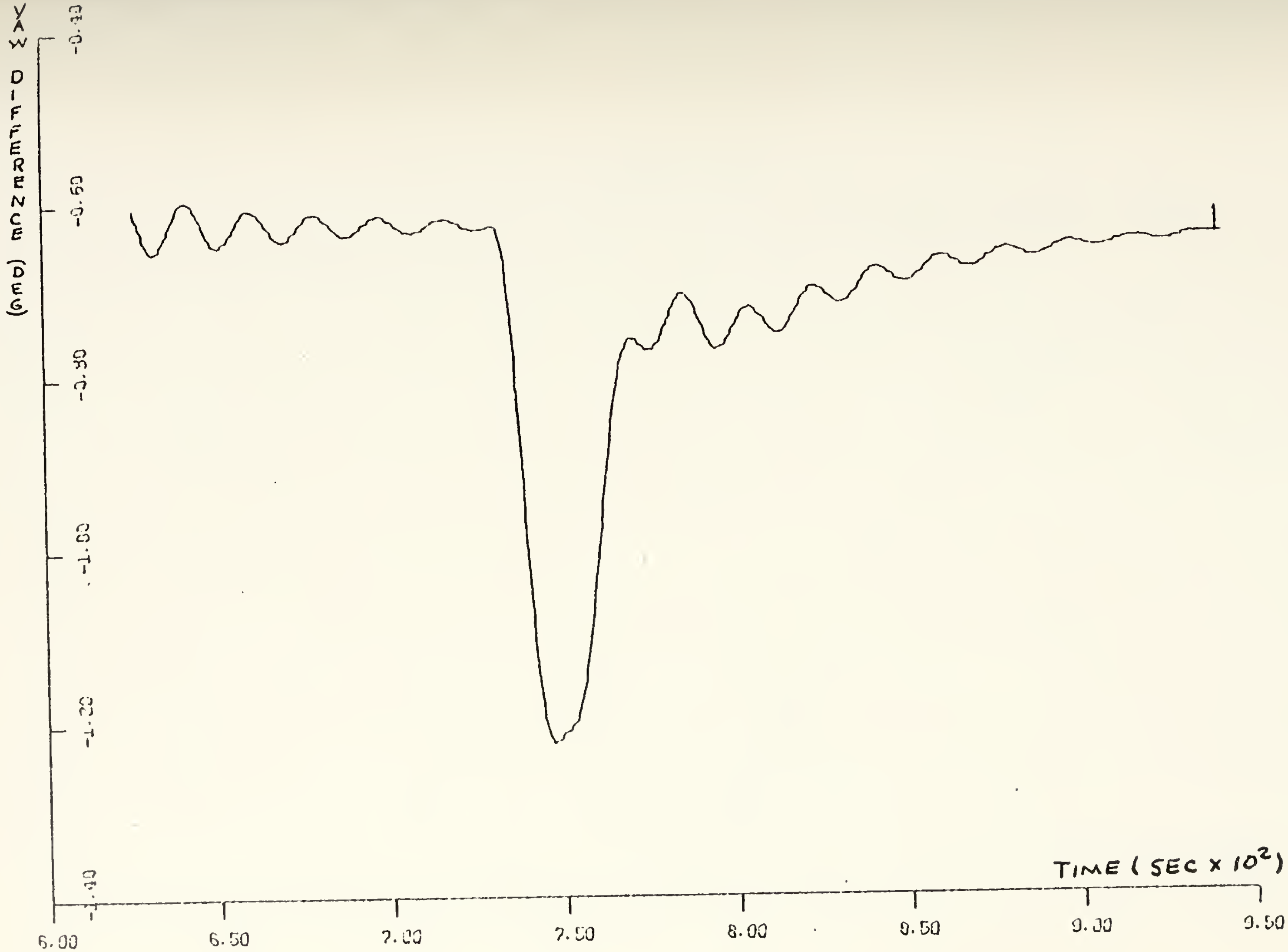


Figure III-100

Turn Phase Run C Yaw Difference

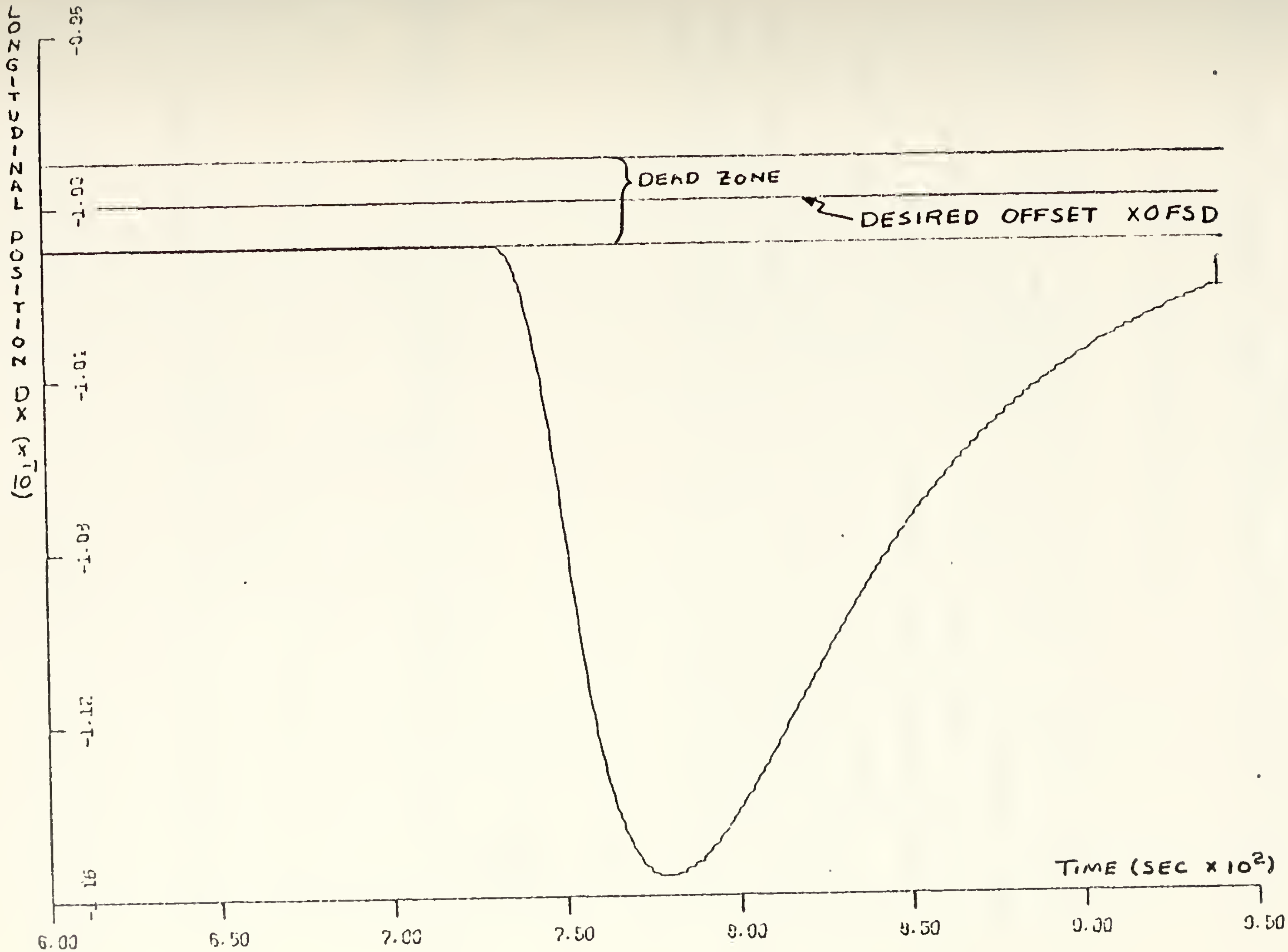


Figure III-101

Turn Phase Run C Longitudinal Position DX

5. Wave Effects on Velocity Control

The final testing procedure involves validation of the speed control system in the presence of waves. This perturbation testing continues that started in section A. of this chapter for heading control. In chapter II the WX force was modeled thru the intermediate force IF32 as:

$$IF32 = KC1 \bullet D2 + NC2 + KC1 \bullet WX$$

By introducing the force in this way, a severe limitation is placed on the magnitude of the force. In the mariner model used, the KC1 coefficient (XDELR) is considered negligable or, at best, only 0.00005. This translates, in the original equations of motion, to a maximum speed perturbation of only 0.0355 kts. for the wave amplitude chosen. The second drawback of this method, with even greater consequences, is that the perturbation is introduced before the control loop. Delay of the wave perturbation is produced making it out of phase with the other wave force (WY) and moment (WN).

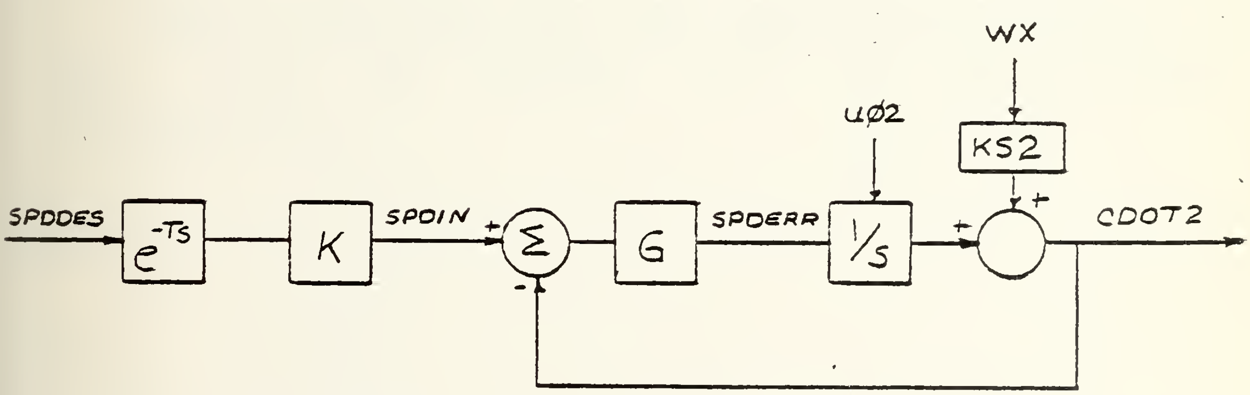


Figure III-102
Block Diagram of Wave Introduction in Speed Loop

In order to bring about uniform introduction of this wave force, its effect is inserted just past the integrator of the speed control loop as shown in figure III-102. This is coded in the DSL simulation program as:

```
CDOT2 = INTGRL(U02,SPDERR*LUC)+KS2*WX
```

A value of -1.0 for KS2 will give a maximum wave perturbation of 0.85275 kts. (a much more realistic perturbation for the high sea state simulated). Figure III-103 portrays the speed desired and speed acquired for the approach phase in the presence of sea state. From this it can be seen that the speed acquired is very dependent upon the sea state present. The control law, however, presents a very stable reference for the speed loop which gives an approach longitudinal position (DX) plot indistinguishable from that of figure III-81. More prominent perturbation results are evident in the turn phase plots of figures III-104 and III-105. The speed response of figure III-104 allows a maximum longitudinal position excursion of 9.5 feet (0.018 normalized position). as compared with 8.286 feet (0.0157 normalized) in calm sea.

These results show that the speed control system is very stable and corrects well for large external perturbations.

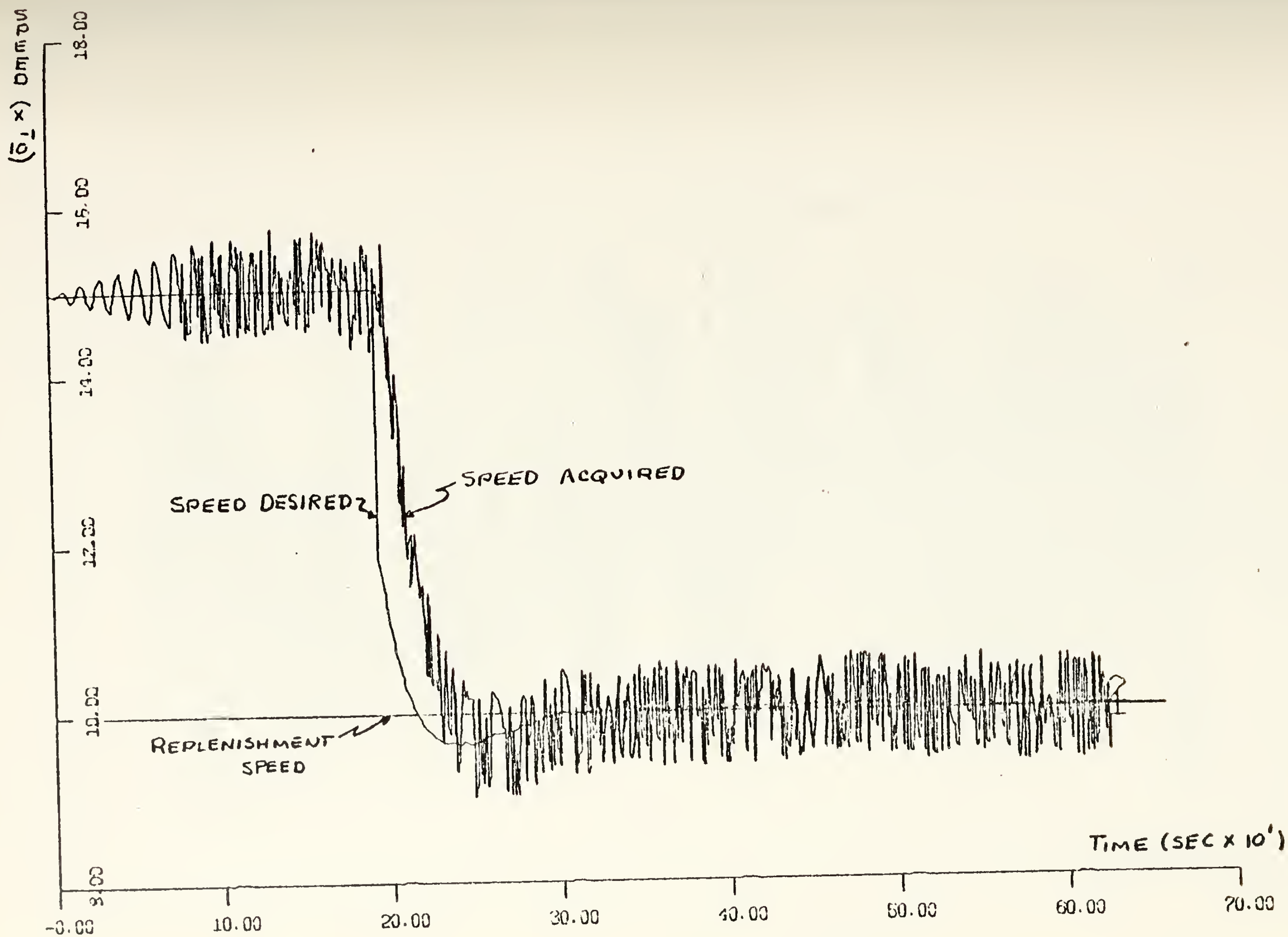


Figure III-103

Approach Phase Speed Response in Waves

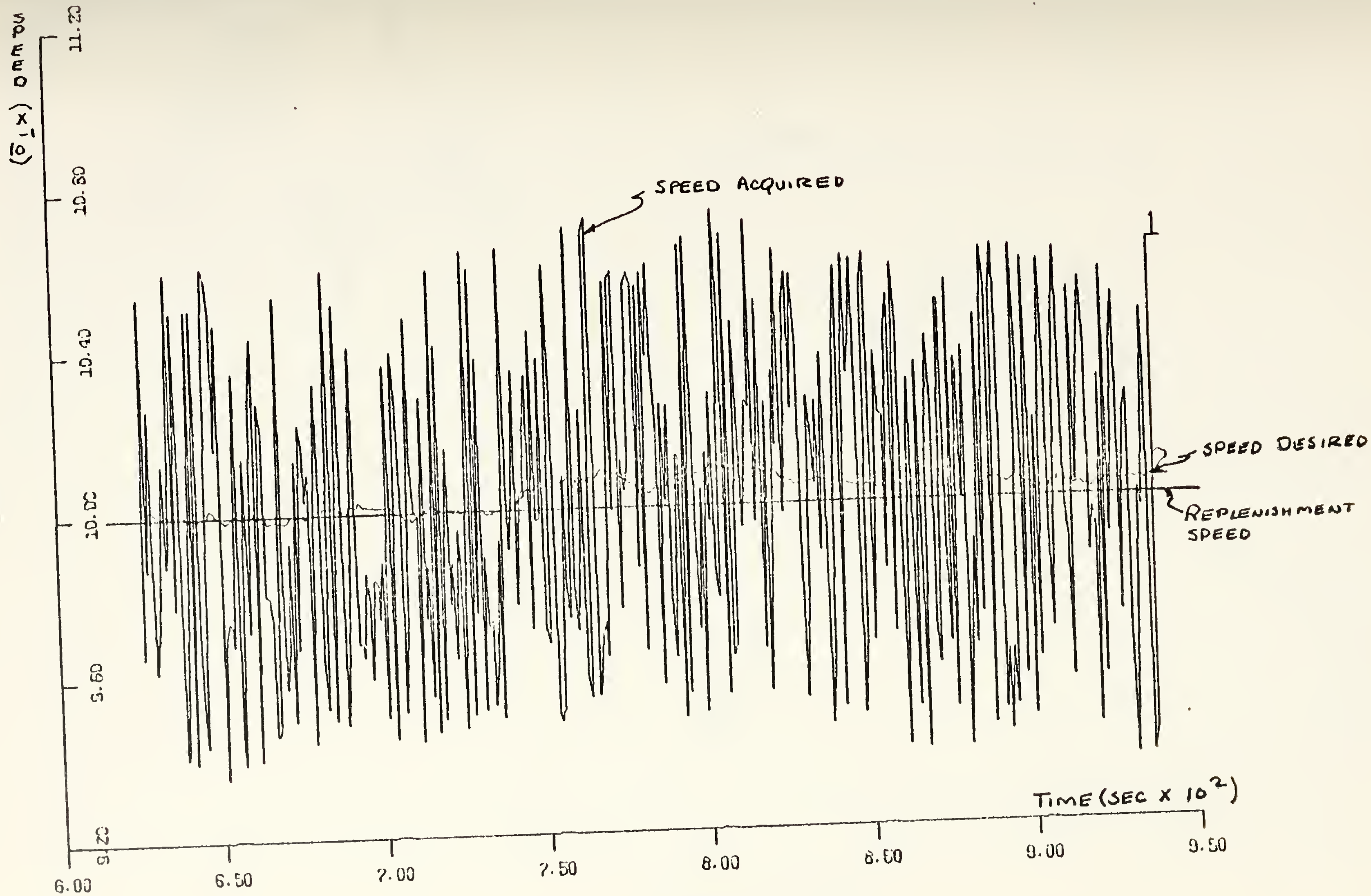


Figure III-104

Turn Phase Speed Response in Waves

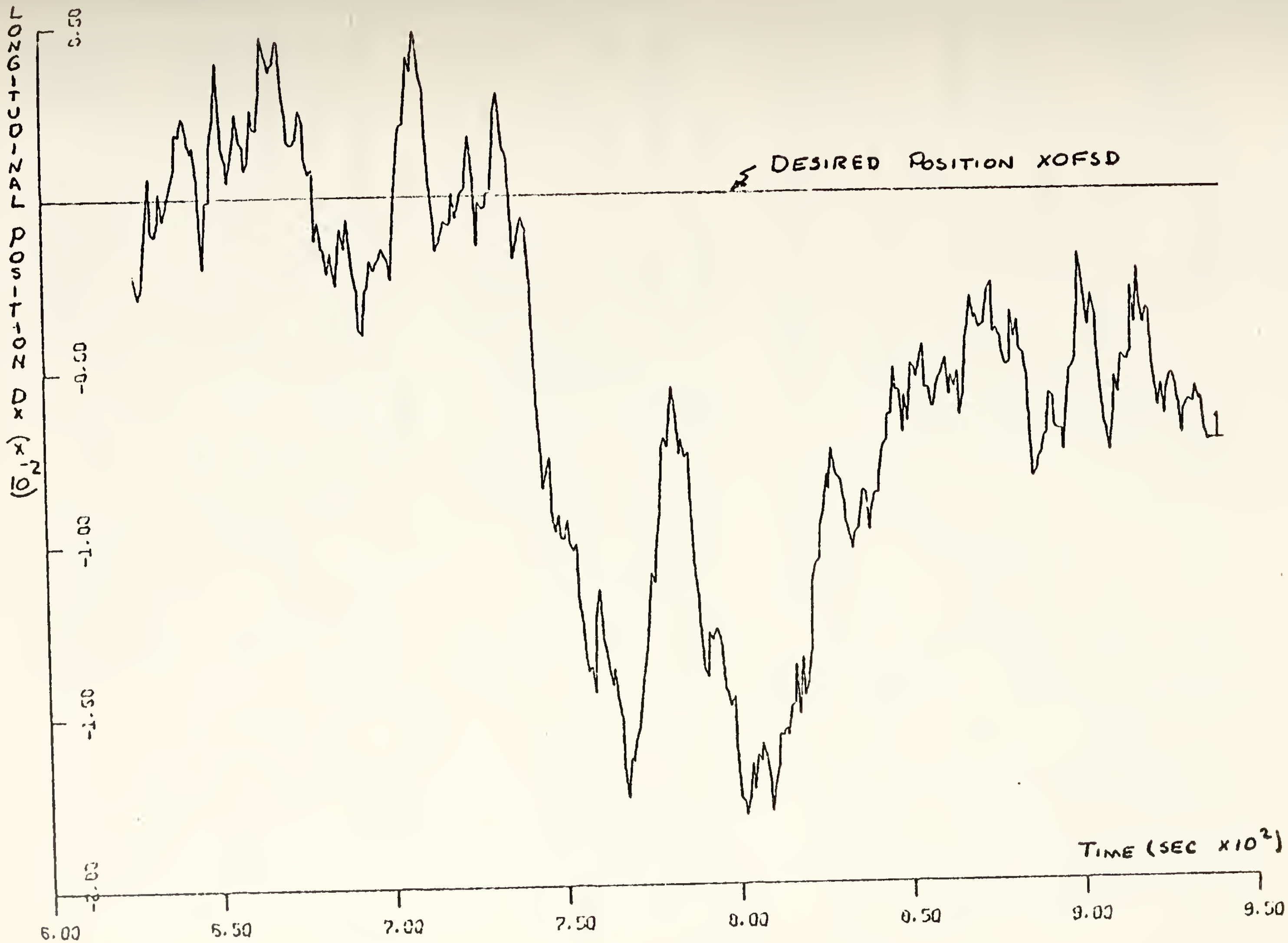


Figure III-105

Turn Phase Longitudinal Position DX in Waves

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The results of this design study have been most gratifying. The basic concepts initially perceived for the RAS control have been realized. The decoupled ship control in the RAS environment is a viable and plausible idea. This thesis contains a workable system for implementation of computer controlled RAS. The achievement of 2.3 foot maximum excursion for lateral distance while both ships are in a turn, and longitudinally offset by 53 feet is a phenomenal achievement. Having this kind of accuracy in RAS operations, can vastly increase the safety of this complicated and dangerous maneuver.

The approach phase of RAS can be a very hair raising experience. Night replenishment and sea state complicate the "seaman's eye" method now employed in the fleet. Having a system that automatically handles the approach regardless of the adverse conditions can, again, do nothing more than increase the safety of the RAS maneuver.

Schemes for computer control of nonlinear systems and the purposeful introduction of nonlinear control laws are becoming more practical with the technological advances in micro processors. The ever increasing number of U.S. Navy ships with computer systems installed, makes digital computer ship control realizable in the present time frame. A good micro computer or an existing installed computer (such as one used for the NTDS system) can be used in this vein. Procurement of the hardware required for this RAS system can be dissipated over time periods contingent on the funding available. The supply ship requires only two reflectors for the range and bearing devices stationed on the receiving ship. All ships can be outfitted with such reflectors at a minimal cost, while the bulk of the hardware can be introduced to the ships at regularly scheduled yard periods.

In the initial conception of this thesis, a section was planned for open ocean maneuvering. After some research on this facet of ship control, it was determined that work in this area has already been documented[24][25]. The existence of NTDS outputs for station attainment and single ship control systems, made design in this area redundant.

The concept of integrated centralized ship control has been in the background for over a decade[26]. Although given a low priority due to funding considerations, its implementation seems to be just around the corner[27]. However, a review of ref. 27 indicates a lack of RAS capability. Whether this is an oversight in the article or neglected in the design criteria is unknown. If it has been neglected in the design, a very real problem has been overlooked. The recent incidences of ship collisions while conducting RAS [28] emphasizes the need for inclusion of this very dangerous maneuver in the "Integrated Bridge System." Lack of technology can no longer be used as an excuse. This thesis and other research reports [29] have advanced the implementation feasibility to a level that cannot be ignored. With these projects finalized in practical terms, their incorporation into fleet use is the next imperative step.

A major effort in this area must be made. The ever increasing complexities of today's naval ships and the loads being placed on the officers and men are such that computer control must be used; and used now! We cannot afford the luxury of time to prove these systems worth, but must make concerted efforts to get them implemented before the lives of 300+ men are lost.

Whenever a complicated system such as a ship in the RAS situation is encountered, many facets have to be concurrently analyzed. This fact has caused inclusion of many diagrams in this thesis to illustrate the total picture. Each run, with a different condition, requires many plots to analyze the differences in the responses and the causes of the differences. The computer programs shown do not reflect the actual run times in the JCL shown. As many as twenty plots were output in these programs in the times listed. Analysis of the actual computation times show that the algorithms run considerably under the time required for real time operation. The sampled data rate used in the simulations was 0.11 seconds. This is well within the realizable data rates available in even the slowest of today's computers and microprocessors. The thrust of this consideration is that there are no problems envisioned in converting RAS simulation to real world RAS control.

B. RECOMMENDATIONS

In the heading control design section of this thesis, the need for a completely adaptive gain scheme was cited. Again in the velocity control section, when a longitudinal offset was introduced, this need became even more evident. The first and most important recommendation for further study is the development of just such an adaptive gain scheme.

The linear equations of motion should be replaced with nonlinear equations to validate the control designs advanced in this thesis. Along with this, the hydrodynamic coefficients for the Navy's modern ships are required to be able to design these control systems for todays vessels.

It is further recommended that a concerted effort be made to obtain data on the interactive forces and moments between ships of dissimilar types and sizes. These forces and moments must also be available for sea state conditions. In fact, the whole area on sea state effects on the various ship types in the RAS situation and in open ocean maneuvering needs attention. Not enough data was available for this researcher to be able to pinpoint sea state effects on ship hulls. Since replenishment at sea is rarely conducted in the sterile condition of calm sea, these considerations are of utmost importance to allow testing of any control system in the simulation stage of development.

The intent here is not to imply that the control systems portrayed in this thesis are the best for the RAS scenario, but that the procedures used can be applied to any control scheme desired and bench marked to the ones contained here. As previously mentioned, much meaningful research and design must be accomplished to allow system reliability and, more important, system acceptability by the officers and men who will ultimately trust their lives to it. This is a task that must not be taken lightly.

APPENDIX A

Due to the lengthy nature of the computer programs presented in this thesis, many functions and subroutines were developed to simplify their presentation. This appendix lists these functions and subroutines in alphabetical order. The computer programs reference this appendix and indicate the placement of the required functions and subroutines.

A brief description for each listing is given to aid the reader in determining their purpose and use. The following is a listing of the functions and subroutines contained in this appendix in the order presented:

SUBROUTINE BOXPLX
FUNCTION DEGRAD
FUNCTION DELAY
FUNCTION FE - RUN A (FEA)
FUNCTION FE - RUN B (FEB)
FUNCTION FE - RUN C (FEC)
SUBROUTINE HDGRAS
FUNCTION KE
MAIN PROGRAM FOR FUNCTION MINIMIZATIONS (MINIBXPX)
SUBROUTINE RBMEAS
FUNCTION RKLDEQ
SUBROUTINE SLOPES
FUNCTION SPINIT
FUNCTION SPDCTR
FUNCTION SPDOFC
FUNCTION SPDREC
FUNCTION SWCL
SUBROUTINE SWTCH

SUBROUTINE SWTCHF
SUBROUTINE TRANS
FUNCTION XLIMIT

SUBROUTINE BOXPLX

This subroutine was used for all optimization runs in heading control and speed control. It was programmed locally and is part of the IBM 360 SSP library at the Naval Postgraduate School. A full explaination and description is shown in the first few pages of the subroutine listing.

```
C     ..................................................................BXPX0010
C                                                                       BXPX0020
C        SUBROUTINE BOXPLX              (CATEGORY HO)                   BXPX0030
C                                                                       BXPX0040
C        PURPOSE                                                        BXPX0050
C                                                                       BXPX0060
C           BOXPLX IS A SUBROUTINE USED TO SOLVE THE PROBLEM OF LOCATING   BXPX0070
C           A MINIMUM (OR MAXIMUM) OF AN ARBITRARY OBJECTIVE FUNCTION    BXPX0080
C           SUBJECT TO ARBITRARY EXPLICIT AND/OR IMPLICIT CONSTRAINTS BY   BXPX0090
C           THE COMPLEX METHOD OF M.J. BOX.  EXPLICIT CONSTRAINTS ARE    BXPX0100
C           DEFINED AS UPPER AND LOWER BOUNDS ON THE INDEPENDENT VARIABLES.BXPX0110
C           IMPLICIT CONSTRAINTS MAY BE ARBITRARY FUNCTIONS OF THE VAR-   BXPX0120
C           IABLES.  TWO FUNCTION SUBPROGRAMS TO EVALUATE THE OBJECTIVE   BXPX0130
C           FUNCTION AND IMPLICIT CONSTRAINTS, RESPECTIVELY, MUST BE     BXPX0140
C           SUPPLIED BY THE USER (SEE EXAMPLE BELOW).  BOXPLX ALSO HAS    BXPX0150
C           THE OPTION TO PERFORM INTEGER PROGRAMMING, WHERE THE VALUES   BXPX0160
C           OF THE INDEPENDENT VARIABLES ARE RESTRICTED TO INTEGERS.     BXPX0170
C                                                                       BXPX0180
C        USAGE                                                          BXPX0190
C                                                                       BXPX0200
C           CALL BOXPLX (NV,NAV,NPR,NTA,R,XS,IP,XU,XL,YMN,IER)          BXPX0210
C                                                                       BXPX0220
C                                                                       BXPX0230
C        DESCRIPTION OF PARAMETERS                                      BXPX0240
C                                                                       BXPX0250
C           NV    AN INTEGER INPUT DEFINING THE NUMBER OF INDEPENDENT     BXPX0260
C                 VARIABLES OF THE OBJECTIVE FUNCTION TO BE MINIMIZED.    BXPX0270
C                 NOTE:  MAXIMUM NV + NAV IS PRESENTLY 50.  MAXIMIM NV IS   BXPX0280
C                 25.  IF THESE LIMITS MUST BE EXCEEDED, PUNCH A SOURCE   BXPX0290
C                 DECK IN THE USUAL MANNER, AND CHANGE THE DIMENSION      BXPX0300
C                 STATEMENTS.                                           BXPX0310
C                                                                       BXPX0320
C           NAV   AN INTEGER INPUT DEFINING THE NUMBER OF AUXILIARY VAR-  BXPX0330
C                 IABLES THE USER WISHES TO DEFINE FOR HIS OWN CONVENIENCE. BXPX0340
C                 TYPICALLY HE MAY WISH TO DEFINE THE VALUE OF EACH IMPLICITBXPX0350
C                 CONSTRAINT FUNCTION AS AN AUXILIARY VARIABLE.  IF THIS   BXPX0360
C                 IS DONE, THE OPTIONAL OUTPUT FEATURE OF BOXPLX CAN BE    BXPX0370
C                 USED TO OBSERVE THE VALUES OF THOSE CONSTRAINTS AS THE   BXPX0380
C                 SOLUTION PROGRESSES.  AUXILIARY VARIABLES, IF USED,     BXPX0390
C                 SHOULD BE EVALUATED IN FUNCTION KE (DEFINED BELOW).     BXPX0400
C                 NAV MAY BE ZERO.                                      BXPX0410
C                                                                       BXPX0420
C           NPR   INPUT INTEGER CONTROLLING THE FREQUENCY OF OUTPUT DESIRED BXPX0430
```

```
C           FOR DIAGNOSTIC PURPOSES.  IF NPR .LE. 0, NO OUTPUT WILL BEBXPX0440
C           PRODUCED BY BOXPLX.  OTHERWISE, THE CURRENT COMPLEX OF    BXPX0450
C           K= 2*NV VERTICES AND THEIR CENTROID WILL BE OUTPUT AFTER  BXPX0460
C           EACH NPR PERMISSIBLE TRIALS.  THE NUMBER OF TOTAL TRIALS, BXPX0470
C           NUMBER OF FEASIBLE TRIALS, NUMBER OF FUNCTION EVALUATIONS BXPX0480
C           AND NJMBER OF IMPLICIT CONSTRAINT EVALUATIONS ARE IN-     BXPX0490
C           CLUDED IN THE OUTPUT.                                     BXPX0500
C           ADDITIONALLY, (WHEN NPR .GT. 0) THE SAME INFORMATION      BXPX0510
C           WILL BE OUTPUT:                                           BXPX0520
C                                                                     BXPX0530
C              1) IF THE INITIAL POINT IS NOT FEASIBLE,               BXPX0540
C              2) AFTER THE FIRST COMPLETE COMPLEX IS GENERATED,      BXPX0550
C              3) IF A FEASIBLE VERTEX CANNOT BE FOUND AT SOME TRIAL, BXPX0560
C              4) IF THE OBJECTIVE VALUE OF A VERTEX CANNOT BE MADE   BXPX0570
C                 NO-LONGER-WORST.                                    BXPX0580
C              5) IF THE LIMIT ON TRIALS (NTA) IS REACHED AND,        BXPX0590
C              6) WHEN THE OBJECTIVE FUNCTION HAS BEEN UNCHANGED FOR  BXPX0600
C                 2*NV TRIALS, INDICATING A LOCAL MINIMUM HAS BEEN    BXPX0610
C                 FOUND.                                              BXPX0620
C                                                                     BXPX0630
C           IF THE USER WISHES TO TRACE THE PROGRESS OF A SOLUTION,   BXPX0640
C           A CHOICE OF NPR = 25, 50 OR 100 IS RECOMMENDED.           BXPX0650
C                                                                     BXPX0660
C     NTA   INTEGER INPUT OF LIMIT ON THE NUMBER OF TRIALS ALLOWED    BXPX0670
C           IN THE CALCULATION.  IF THE USER INPUTS NTA .LE. 0, A     BXPX0680
C           DEFAULT VALUE OF 2000 IS USED.  WHEN THIS LIMIT IS REACHEDBXPX0690
C           CONTROL RETURNS TO THE CALLING PROGRAM WITH THE BEST      BXPX0700
C           ATTAINED OBJECTIVE FUNCTION VALUE IN YMN, AND THE BEST    BXPX0710
C           ATTAINED SOLUTION POINT IN XS.                            BXPX0720
C                                                                     BXPX0730
C     R     A REAL NUMBER INPUT TO DEFINE THE FIRST RANDOM NUMBER     BXPX0740
C           USED IN DEVELOPING THE INITIAL COMPLEX OF 2*NV VERTICIES. BXPX0750
C           (0. .GT. R .LT. 1.) IF R IS NOT WITHIN THESE BOUNDS,      BXPX0760
C           IT WILL BE REPLACED BY 1./3. .                            BXPX0770
C                                                                     BXPX0780
C     XS    INPUT REAL ARRAY DIMENSIONED AT LEAST NV+NAV. THE FIRST   BXPX0790
C           NV MUST CONTAIN A FEASIBLE ORIGIN FOR STARTING THE CAL-   BXPX0800
C           CULATION.  THE LAST NAV NEED NOT BE INITIALIZED.  UPON    BXPX0810
C           RETURN FROM BOXPLX, THE FIRST NV ELEMENTS OF THE ARRAY    BXPX0820
C           CONTAIN THE COORDINATES OF THE MINIMUM OBJECTIVE FUNCTION,BXPX0830
C           AND THE REMAINING NAV (NAV .GE. 0) CONTAIN THE VALUES OF  BXPX0840
C           THE CORRESPONDING AUXILIARY VARIABLES.                    BXPX0850
C                                                                     BXPX0860
C     IP    INTEGER INPUT FOR OPTIONAL INTEGER PROGRAMMING.  IF IP=1, BXPX0870
C           THE VALUES OF THE INDEPENDENT VARIABLES WILL BE REPLACED  BXPX0880
C           WITH INTEGER VALUES (STILL STORED AS REAL*4).             BXPX0890
C                                                                     BXPX0900
C     XU    A REAL ARRAY DIMENSIONED AT LEAST NV INPUTTING THE UPPER  BXPX0910
```

```
C                 BOUND ON EACH INDEPENDENT VARIABLE, (EACH EXPLICIT CON-   BXPX0920
C                 STRAINT).  INPUT VALUES ARE SLIGHTLY ALTERED BY BOXPLX.   BXPX0930
C                                                                           BXPX0940
C            XL   A REAL ARRAY DIMENSIONED AT LEAST NV INPUTTING THE LOWER  BXPX0950
C                 BOUND ON EACH INDEPENDENT VARIABLE, (EACH EXPLICIT CON-   BXPX0960
C                 STRAINT).  NOTE:  FOR BOTH XU AND XL CHOOSE REASONABLE    BXPX0970
C                 VALUES IF NONE ARE GIVEN, NOT VALUES WHICH ARE MAGNITUDES BXPX0980
C                 ABOVE OR BELOW THE EXPECTED SOLUTION.  INPUT VALUES ARE   BXPX0990
C                 SLIGHTLY ALTERED BY BOXPLX.                               BXPX1000
C                                                                           BXPX1010
C            YMN  THIS OUTPUT IS THE VALUE (REAL*4) OF THE OBJECTIVE FUNC-  BXPX1020
C                 TION, CORRESPONDING TO THE SOLUTION POINT OUTPUT IN XS.   BXPX1030
C                                                                           BXPX1040
C            IER  INTEGER ERROR RETURN.  TO BE INTERROGATED UPON RETURN     BXPX1050
C                 FROM BOXPLX.  IER WILL BE ONE OF THE FOLLOWING:           BXPX1060
C                                                                           BXPX1070
C                   =-1   CANNOT FIND FEASIBLE VERTEX OR FEASIBLE CENTROID  BXPX1080
C                         AT THE START OR A RESTART (SEE 'METHOD' BELOW).   BXPX1090
C                   =0    FUNCTION VALUE UNCHANGED FOR 'N' TRIALS.  (WHERE  BXPX1100
C                         N=6*NV+10)  THIS IS THE NORMAL RETURN PARAMETER.  BXPX1110
C                   =1    CANNOT DEVELOP FEASIBLE VERTEX.                   BXPX1120
C                   =2    CANNOT DEVELOP A NO-LONGER-WORST VERTEX.          BXPX1130
C                   =3    LIMIT ON TRIALS REACHED.  (NTA EXCEEDED)          BXPX1140
C                   NOTE:  VALID RESULTS MAY BE RETURNED IN ANY OF THE      BXPX1150
C                         ABOVE CASES.                                      BXPX1160
C                                                                           BXPX1170
C         EXAMPLE OF USAGE                                                  BXPX1180
C                                                                           BXPX1190
C            THIS EXAMPLE MINIMIZES THE OBJECTIVE FUNCTION SHOWN IN THE     BXPX1200
C            EXTERNAL FUNCTION FE(X).  THERE ARE TWO INDEPENDENT VAR-       BXPX1210
C            IABLES X(1) & X(2), AND TWO IMPLICIT CONSTRAINT FUNCTIONS      BXPX1220
C            X(3) & X(4) WHICH ARE EVALUATED AS AUXILIARY VARIABLES (SEE    BXPX1230
C            EXTERNAL FUNCTION KE(X) ).                                     BXPX1240
C                                                                           BXPX1250
C         DIMENSION  XS(4),XU(2),XL(2)                                      BXPX1260
CC                                                                          BXPX1270
CC     STARTING GUESS                                                       BXPX1280
C         XS(1) = 1.0                                                       BXPX1290
C         XS(2) = 0.5                                                       BXPX1300
CC     UPPER LIMITS                                                         BXPX1310
C         XU(1) = 6.0                                                       BXPX1320
C         XU(2) = 6.0                                                       BXPX1330
CC     LOWER LIMITS                                                         BXPX1340
C         XL(1) = 0.0                                                       BXPX1350
C         XL(2) = 0.0                                                       BXPX1360
CC                                                                          BXPX1370
C         R = 9./13.                                                        BXPX1380
C         NTA = 5000                                                        BXPX1390
```

```
C         NPR = 50
C         NAV = 2
C         NV = 2
C         IP = 0
CC
C          CALL BOXPLX (NV,NAV,NPR,NTA,R,XS,IP,XU,XL,YMN,IER)
C         WRITE(6,1) ((XS(I),I=1,4),YMN,IER)
C       1 FORMAT (////,'    THE POINT IS LOCATED AT (XS(I)=) ',4(E13.7,5X),
C        1//,'    AND THE FUNCTION VALUE IS ',E13.7,' IER = ',I5)
CC
C         STOP
C         END
CC
CC
C         FUNCTION  KE(X)
CC      EVALUATE CONSTRAINTS.  SET KE=0 IF NO IMPLICIT CONSTRAINT IS
CC      VIOLATED, OR SET KE=1 IF ANY IMPLICIT CONSTRAINT IS VIOLATED.
C         DIMENSION X(4)
C         X1 = X(1)
C         X2 = X(2)
C         KE = 0
C         X(3) = X1 + 1.732051*X2
C         IF (X(3) .LT. 0.  .OR. X(3) .GT. 6.) GO TO 1
C         X(4) = X1/1.732051 -X2
C         IF (X(4) .GE. 0.)  RETURN
CC
C       1 KE = 1
C         RETURN
C         END
CC
CC
C         FUNCTION  FE(X)
C         DIMENSION X(4)
CC
CC      THIS IS THE OBJECTIVE FUNCTION.
C         FE= -(X(2)**3 *(9.-(X(1)-3.)**2)/(46.76538))
C         RETURN
C         END
C
C         METHOD
C
C              THE COMPLEX METHOD IS AN EXTENSION AND ADAPTION OF THE SIM-
C              PLEX METHOD OF LINEAR PROGRAMMING.  STARTING WITH ANY ONE
C              FEASIBLE POINT IN N-DIMENSION SPACE A "COMPLEX" OF 2*N
C              VERTICES IS CONSTRUCTED BY SELECTING RANDOM POINTS WITHIN THE
C              FEASIBLE REGION.  FOR THIS PURPOSE N COORDINATES ARE FIRST
C              RANDOMLY CHOSEN WITHIN THE SPACE BOUNDED BY EXPLICIT CON-
C              STRAINTS.  THIS DEFINES A TRIAL INITIAL VERTEX.  IT IS THEN
```

```
C        CHECKED FOR POSSIBLE VIOLATION OF IMPLICIT CONSTRAINTS.  IF     BXPX1880
C        ONE OR MORE ARE VIOLATED, THE TRIAL INITIAL VERTEX IS DISPLACEDBXPX1890
C        HALF OF ITS DISTANCE FROM THE CENTROID OF PREVIOUSLY SELECTED   BXPX1900
C        INITIAL VERTICES.  IF NECESSARY THIS DISPLACEMENT PROCESS IS    BXPX1910
C        REPEATED UNTIL THE VERTEX HAS BECOME FEASIBLE.  IF THIS FAILS   BXPX1920
C        TO HAPPEN AFTER 5*N+10 DISPLACEMENTS, THE SOLUTION IS ABAND-    BXPX1930
C        ONED.  AFTER EACH VERTEX IS ADDED TO THE COMPLEX, THE CURRENT   BXPX1940
C        CENTROID IS CHECKED FOR FEASIBILITY.  IF IT IS INFEASIBLE,      BXPX1950
C        THE LAST TRIAL VERTEX IS ABANDONED AND AN EFFORT TO GENERATE    BXPX1960
C        AN ALTERNATIVE TRIAL VERTEX IS MADE.  IF 5*N+10 VERTICES ARE    BXPX1970
C        ABANDONED CONSECUTIVELY, THE SOLUTION IS TERMINATED.            BXPX1980
C                                                                        BXPX1990
C        IF AN INITIAL COMPLEX IS ESTABLISHED, THE BASIC COMPUTATION     BXPX2000
C        LOOP IS INITIATED.  THESE INSTRUCTIONS FIND THE CURRENT WORST   BXPX2010
C        VERTEX, THAT IS, THE VERTEX WITH THE LARGEST CORRESPONDING      BXPX2020
C        VALUE FOR THE OBJECTIVE FUNCTION, AND REPLACE THAT VERTEX BY    BXPX2030
C        ITS OVER-REFLECTION THROUGH THE CENTROID OF ALL OTHER VERTICES.BXPX2040
C        (IF THE VERTEX TO BE REPLACED IS CONSIDERED AS A VECTOR IN      BXPX2050
C        N-SPACE, ITS OVER-REFLECTION IS OPPOSITE IN DIRECTION, IN-      BXPX2060
C        CREASED IN LENGTH BY THE FACTOR 1.3, AND COLLINEAR WITH         BXPX2070
C        THE REPLACED VERTEX AND CENTROID OF ALL OTHER VERTICES.)        BXPX2080
C                                                                        BXPX2090
C        WHEN AN OVER-REFLECTION IS NOT FEASIBLE OR REMAINS WORST, IT    BXPX2100
C        IS CONSIDERED NOT-PERMISSIBLE AND IS DISPLACED HALFWAY TOWARD   BXPX2110
C        THE CENTROID.  AFTER FOUR SUCH ATTEMPTS ARE MADE UNSUCCESSFULLYBXPX2120
C        EVERY FIFTH ATTEMPT IS MADE BY REFLECTING THE OFFENDING VERTEX  BXPX2130
C        THROUGH THE PRESENT BEST VERTEX, INSTEAD OF THROUGH THE CEN-    BXPX2140
C        TROID.  IF 5*N+10 DISPLACEMENTS AND OVER-REFLECTIONS OCCUR      BXPX2150
C        WITHOUT A SUCCESSFUL (PERMISSIBLE) RESULT, THE CURRENT BEST     BXPX2160
C        VERTEX IS TAKEN AS AN INITIAL FEASIBLE POINT FOR A RESTART      BXPX2170
C        RUN OF THE COMPLETE PROCESS.  RESTARTING IS ALSO UNDERTAKEN     BXPX2180
C        WHEN 6*NV+10 CONSECUTIVE TRIALS HAVE BEEN MADE WITH NO SIGNIF-  BXPX2190
C        ICANT CHANGE IN THE VALUE OF THE OBJECTIVE FUNCTION.  IN ALL    BXPX2200
C        CASES, RESTARTING IS INHIBITED IF THE LAST RESTART DID NOT      BXPX2210
C        PRODUCE A SIGNIFICANT IMPROVEMENT IN THE MINIMUM ATTAINED.      BXPX2220
C                                                                        BXPX2230
C        IT IS RECOMMENDED THAT THE USER READ THE REFERENCE FOR          BXPX2240
C        FURTHER USEFJL INFORMATION.  IT SHOULD BE NOTED THAT THE        BXPX2250
C        ALGORITHM DEFINED THERE HAS BEEN ALTERED TO FIND THE            BXPX2260
C        CONSTRAINED MINIMUM, RATHER THAN THE MAXIMUM.                   BXPX2270
C                                                                        BXPX2280
C                                                                        BXPX2290
C                                                                        BXPX2300
C     REMARKS                                                            BXPX2310
C                                                                        BXPX2320
C        THE INTEGER PROGRAMMING OPTION WAS ADDED TO THIS PROGRAM        BXPX2330
C        AS SUGGESTED IN REFERENCE (2).  A MIXED INTEGER/CONTINUOUS      BXPX2340
C        VARIABLE VERSION OF BOXPLX WOULD BE EASY TO CREATE BY DE-       BXPX2350
```

```
C        CLARING "IP" TO BE AN ARRAY OF NV CONTROL VARIABLES WHERE IP   BXPX2360
C        (I)=1 WOULD INDICATE THAT THE I-TH VARIABLE IS TO BE CONFINED  BXPX2370
C        TO INTEGER VALUES.  EACH STATEMENT OF THE FORM 'IF (IP .EQ.    BXPX2380
C        1)' ETC. WOULD THEN NEED TO BE ALTERED TO 'IF (IP(I) .EQ. 1)'  BXPX2390
C        ETC., WHERE THE SUBSCRIPT IS APPROPRIATELY CHOSEN.  NORMALLY,  BXPX2400
C        XU AND XL VALUES ARE ALTERED TO BE AN EPSILON 'WITHIN' ACTUAL  BXPX2410
C        VALUES DECLARED BY THE USER.  THIS ADJUSTMENT IS NOT MADE      BXPX2420
C        WHEN IP=1.                                                     BXPX2430
C                                                                       BXPX2440
C        NOTE:  NO NON-LINEAR PROGRAMMING ALGORITHM CAN GUARANTEE THAT  BXPX2450
C        THE ANSWER FOUND IS THE GLOBAL MINIMUM, RATHER THAN JUST A     BXPX2460
C        LOCAL MINIMUM.  HOWEVER, ACCORDING TO REF.2, THE COMPLEX       BXPX2470
C        METHOD HAS AN ADVANTAGE IN THAT IT TENDS TO FIND THE GLOBAL    BXPX2480
C        MINIMUM MORE FREQUENTLY THAN MANY OTHER NON-LINEAR PROGRAM-    BXPX2490
C        MING ALGORITHMS.                                               BXPX2500
C                                                                       BXPX2510
C        IT SHOULD BE NOTED THAT THE AUXILIARY VARIABLE FEATURE CAN     BXPX2520
C        ALSO BE USED TO DEAL WITH PROBLEMS CONTAINING EQUALITY CON-    BXPX2530
C        STRAINTS.  ANY EQUALITY CONSTRAINT IMPLIES THAT A GIVEN VAR-   BXPX2540
C        IABLE IS NOT TRULY INDEPENDENT.  THEREFORE, IN GENERAL, ONE    BXPX2550
C        VARIABLE INVOLVED IN AN EQUALITY CONSTRAINT CAN BE RENUMBERED  BXPX2560
C        FROM THE SET OF NV INDEPENDENT VARIABLES AND ADDED TO THE SET  BXPX2570
C        OF NAV AUXILIARY VARIABLES.  THIS USUALLY INVOLVES RENUMBERING BXPX2580
C        THE INDEPENDENT VARIABLES OF THE GIVEN PROBLEM.                BXPX2590
C                                                                       BXPX2600
C     SUBROUTINES AND FUNCTIONS REQUIRED                                BXPX2610
C                                                                       BXPX2620
C        SUBROUTINE 'BOUT' AND FUNCTION 'FBV' ARE INTEGRAL PARTS OF     BXPX2630
C        THE BOXPLX PACKAGE.                                            BXPX2640
C                                                                       BXPX2650
C        TWO FUNCTIONS MUST BE SUPPLIED BY THE USER.  THE FIRST, KE(X), BXPX2660
C        IS USED TO EVALUATE THE IMPLICIT CONSTRAINTS.  SET KE=0 AT THE BXPX2670
C        BEGINING OF THE FUNCTION, THEN EVALUATE THE IMPLICIT CON-      BXPX2680
C        STRAINTS.  IN THE EXAMPLE ABOVE, THE FIRST CONSTRAINT, X(3),   BXPX2690
C        MUST BE WITHIN THE RANGE (0. .LE. X(3) .LE. 6.).  THE SECOND   BXPX2700
C        CONSTRAINT X(4), MUST BE .GE. 0. .  IF EITHER CONSTRAINT IS    BXPX2710
C        NOT WITHIN THESE BOUNDS, CONTROL IS TRANSFERRED TO STATEMENT 1,BXPX2720
C        AND KE IS SET TO "1" AND CONTROL IS RETURNED TO BOXPLX.        BXPX2730
C                                                                       BXPX2740
C        THE SECOND FUNCTION THE USER MUST PROVIDE EVALUATES THE OB-    BXPX2750
C        JECTIVE FUNCTION.  IT IS CALLED FE(X) AS SHOWN IN THE EXAM-    BXPX2760
C        PLE ABOVE, AND FE MUST BE SET TO THE VALUE OF THE OBJECTIVE    BXPX2770
C        FUNCTION CORRESPONDING TO CURRENT VALUES OF THE NV INDEPENDENT BXPX2780
C        VARIABLES IN ARRAY 'X'.                                        BXPX2790
C                                                                       BXPX2800
C     REFERENCES                                                        BXPX2810
C                                                                       BXPX2820
C        BOX, M. J., "A NEW METHOD OF CONSTRAINED OPTIMIZATION AND A    BXPX2830
```

```
C         COMPARISON WITH OTHER METHODS", COMPUTER JOURNAL, 8 APR. '65,
C         PP. 45-52.
C
C         BEVERIDGE G., AND SCHECHTER R., "OPTIMIZATION: THEORY AND
C         PRACTICE", MCGRAW-HILL, 1970.
C
C      PROGRAMMER
C
C         R.R. HILLEARY 1/1966.
C         REVISED FOR SYSTEM 360 4/1967
C         CORRECTED  1/1969
C         REVISED/EXTENDED BY L.NOLAN/R.HILLEARY  2/1975
C
C
C      ...................................................................
C
C
       SUBROUTINE BOXPLX (NV,NAV,NPR,NTZ,RZ,XS,IP,BU,BL,YMN,IER)
C
       DIMENSION V(50,50), FUN(50), SUM(25), CEN(25), XS(NV), BU(NV), BL(
      1NV)
C
       KV = 5
       EP = 1.E-6
       NTA = 2000
       IF (NTZ.GT.0) NTA = NTZ
       R = RZ
       IF (R.LE.0..OR.R.GE.1.) R=1./3.
       NVT = NV+NAV
C
C            TOTAL VARS, EXPLICIT PLUS IMPLICIT
       NT = 0
C            CURRENT TRIAL NO.
       NPT = 0
C            CURRENT NO. OF PERMISSIBLE TRIALS
       NTFS = 0
C            CURRENT NO. OF TIMES F HAS BEEN ALMOST UNCHANGED
C
C              CHECK FEASIBILITY OF START POINT
C
       DO 4 I=1,NV
       VT = XS(I)
       IF (BL(I).LE.VT) GO TO 1
       II = -I
       VT = BL(I)
       GO TO 2
     1 IF (BU(I).GE.VT) GO TO 3
       II = I
```

```
         VT = BU(I)                                                       BXPX3320
       2 IF (NPR.GT.0) WRITE (6,49) II                                    BXPX3330
       3 V(I,1) = VT                                                      BXPX3340
         CEN(I) = VT                                                      BXPX3350
         IF (IP.EQ.1) GO TO 4                                             BXPX3360
         BL(I) = BL(I)+AMAX1(EP,EP*ABS(BL(I)))                            BXPX3370
         BU(I) = BU(I)-AMAX1(EP,EP*ABS(BU(I)))                            BXPX3380
       4 SUM(I) = VT                                                      BXPX3390
C                                                                         BXPX3400
C                                                                         BXPX3410
         NCE = 1                                                          BXPX3420
C              NUMBER OF CONSTRAINT EVALUATIONS                           BXPX3430
         I = 1                                                            BXPX3440
         IF (KE(V(1,1)).EQ.0) GO TO 5                                     BXPX3450
         IF (NPR.LE.0) GO TO 12                                           BXPX3460
         WRITE (6,50)                                                     BXPX3470
         GO TO 12                                                         BXPX3480
       5 NFE = 1                                                          BXPX3490
C                                                                         BXPX3500
C      NUMBER OF VERTICES (K) = 2 TIMES NO. OF VARIABLES.                 BXPX3510
         K = 2*NV                                                         BXPX3520
C                                                                         BXPX3530
C      NUMBER OF DISPLACEMENTS ALLOWED.                                   BXPX3540
         NLIM = 5*NV+10                                                   BXPX3550
C                                                                         BXPX3560
C      NUMBER OF CONSECUTIVE TRIALS WITH UNCHANGED FE TO TERMINATE.       BXPX3570
         NCT = NLIM+NV                                                    BXPX3580
         ALPHA = 1.3                                                      BXPX3590
         FK = K                                                           BXPX3600
         FKM = FK-1.                                                      BXPX3610
         BETA = ALPHA+1.                                                  BXPX3620
C                                                                         BXPX3630
C      INSURE SEED OF RANDOM NUMBER GENERATOR IS ODD.                     BXPX3640
         IQR = R*1.E7                                                     BXPX3650
         IF (MOD(IQR,2).EQ.0) IQR=IQR+101                                 BXPX3660
C                                                                         BXPX3670
C                         SET UP INITIAL VERTICES                         BXPX3680
         FUN(1) = FE(V(1,1))                                              BXPX3690
         YMN = FUN(1)                                                     BXPX3700
       6 FI = 1.                                                          BXPX3710
         FUNOLD = FUN(1)                                                  BXPX3720
C                                                                         BXPX3730
         DO 15 I=2,K                                                      BXPX3740
         FI = FI+1.                                                       BXPX3750
         LIMT = 0                                                         BXPX3760
       7 LIMT = LIMT+1                                                    BXPX3770
C                                                                         BXPX3780
C      END CALCULATION IF FEASIBLE CENTROID CANNOT BE FOUND.              BXPX3790
```

```
      IF (LIMT.GE.NLIM) GO TO 11                                        BXPX3800
C                                                                       BXPX3810
      DO 8 J=1,NV                                                       BXPX3820
C                                                                       BXPX3830
C    RANDOM NUMBER GENERATOR  (RANDU)                                   BXPX3840
      IQR = IQR*65539                                                   BXPX3850
      IF (IQR.LT.0) IQR = IQR+2147483647+1                              BXPX3860
      RQX = IQR                                                         BXPX3870
      RQX = RQX*.4656613E-9                                             BXPX3880
      V(J,I) = BL(J)+RQX*(BU(J)-BL(J))                                  BXPX3890
      IF (IP.EQ.1) V(J,I)=AINT(V(J,I)+.5)                               BXPX3900
    8 CONTINUE                                                          BXPX3910
C                                                                       BXPX3920
      DO 10 L=1,NLIM                                                    BXPX3930
      NCE = NCE+1                                                       BXPX3940
      IF (KE(V(1,I)).EQ.0) GO TO 13                                     BXPX3950
C                                                                       BXPX3960
      DO 9 J=1,NV                                                       BXPX3970
      VT = .5*(V(J,I)+CEN(J))                                           BXPX3980
      IF (IP.EQ.1) VT = AINT(VT+.5)                                     BXPX3990
      V(J,I) = VT                                                       BXPX4000
    9 CONTINUE                                                          BXPX4010
C                                                                       BXPX4020
   10 CONTINUE                                                          BXPX4030
C                                                                       BXPX4040
   11 IF (NPR.LE.0) GO TO 12                                            BXPX4050
      WRITE (6,51) I                                                    BXPX4060
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,I,FUN,CEN,I)                   BXPX4070
   12 IER = -1                                                          BXPX4080
      GO TO 48                                                          BXPX4090
C                                                                       BXPX4100
   13 DO 14 J=1,NV                                                      BXPX4110
      SUM(J) = SUM(J)+V(J,I)                                            BXPX4120
   14 CEN(J) = SUM(J)/FI                                                BXPX4130
C                                                                       BXPX4140
C    TRY TO ASSURE FEASIBLE CENTROID FOR STARTING.                      BXPX4150
      NCE = NCE+1                                                       BXPX4160
      IF (KE(CEN).NE.0) GO TO 7                                         BXPX4170
      NFE = NFE+1                                                       BXPX4180
      FUN(I) = FE(V(1,I))                                               BXPX4190
   15 CONTINUE                                                          BXPX4200
C                                                                       BXPX4210
C    END OF LOOP SETTING OF INITIAL COMPLEX.                            BXPX4220
      IF (NPR.LE.0) GO TO 17                                            BXPX4230
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,0)                   BXPX4240
C                                                                       BXPX4250
C    FIND THE WORST VERTEX, THE 'J'TH.                                  BXPX4260
      J = 1                                                             BXPX4270
```

```
C                                                                       BXPX4280
      DO 16 I=2,K                                                       BXPX4290
      IF (FUN(J).GE.FUN(I)) GO TO 16                                    BXPX4300
      J = I                                                             BXPX4310
   16 CONTINUE                                                          BXPX4320
C                                                                       BXPX4330
C    BASIC LOOP.  ELIMINATE EACH WORST VERTEX IN TURN.  IT MUST BECOME  BXPX4340
C    NO LONGER WORST, NOT MERELY IMPROVED.  FIND NEXT-TO-WORST VERTEX,  BXPX4350
C    THE 'JN'TH ONE.                                                    BXPX4360
   17 JN = 1                                                            BXPX4370
      IF (J.EQ.1) JN = 2                                                BXPX4380
C                                                                       BXPX4390
      DO 18 I=1,K                                                       BXPX4400
      IF (I.EQ.J) GO TO 18                                              BXPX4410
      IF (FUN(JN).GE.FUN(I)) GO TO 18                                   BXPX4420
      JN = I                                                            BXPX4430
   18 CONTINUE                                                          BXPX4440
C                                                                       BXPX4450
C    LIMT = NUMBER OF MOVES DURING THIS TRIAL TOWARD THE CENTROID       BXPX4460
C     DUE TO FUNCTION VALUE.                                            BXPX4470
      LIMT = 1                                                          BXPX4480
C                                                                       BXPX4490
C    COMPUTE CENTROID AND OVER REFLECT WORST VERTEX.                    BXPX4500
C                                                                       BXPX4510
      DO 19 I=1,NV                                                      BXPX4520
      VT = V(I,J)                                                       BXPX4530
      SUM(I) = SUM(I)-VT                                                BXPX4540
      CEN(I) = SUM(I)/FKM                                               BXPX4550
      VT = BETA*CEN(I)-ALPHA*VT                                         BXPX4560
      IF (IP.EQ.1) VT = AINT(VT+.5)                                     BXPX4570
C                                                                       BXPX4580
C    INSURE THE EXPLICIT CONSTRAINTS ARE OBSERVED.                      BXPX4590
   19 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))                             BXPX4600
C                                                                       BXPX4610
      NT = NT+1                                                         BXPX4620
C                                                                       BXPX4630
C    CHECK FOR IMPLICIT CONSTRAINT VIOLATION.                           BXPX4640
C                                                                       BXPX4650
   20 DO 25 N=1,NLIM                                                    BXPX4660
      NCE = NCE+1                                                       BXPX4670
      IF (KE(V(1,J)).EQ.0) GO TO 26                                     BXPX4680
C                                                                       BXPX4690
C    EVERY 'KV'TH TIME, OVER-REFLECT THE OFFENDING VERTEX THROUGH THE   BXPX4700
C    BEST VERTEX.                                                       BXPX4710
      IF (MOD(N,KV).NE.0) GO TO 22                                      BXPX4720
      CALL FBV (K,FUN,M)                                                BXPX4730
C                                                                       BXPX4740
      DO 21 I=1,NV                                                      BXPX4750
```

```
      VT = BETA*V(I,M)-ALPHA*V(I,J)                                     BXPX4760
      IF (IP.EQ.1) VT = AINT(VT+.5)                                     BXPX4770
   21 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))                             BXPX4780
C                                                                       BXPX4790
      GO TO 24                                                          BXPX4800
C                                                                       BXPX4810
C   CONSTRAINT VIOLATION:  MOVE NEW POINT TOWARD CENTROID.              BXPX4820
C                                                                       BXPX4830
   22 DO 23 I=1,NV                                                      BXPX4840
      VT = .5*(CEN(I)+V(I,J))                                           BXPX4850
      IF (IP.EQ.1) VT = AINT(VT+.5)                                     BXPX4860
      V(I,J) = VT                                                       BXPX4870
   23 CONTINUE                                                          BXPX4880
C                                                                       BXPX4890
   24 NT = NT+1                                                         BXPX4900
   25 CONTINUE                                                          BXPX4910
C                                                                       BXPX4920
      IER = 1                                                           BXPX4930
C                                                                       BXPX4940
C   CANNOT GET FEASIBLE VERTEX BY MOVING TOWARD CENTROID,               BXPX4950
C   OR BY OVER-REFLECTING THRU THE BEST VERTEX.                         BXPX4960
      IF (NPR.LE.0) GO TO 42                                            BXPX4970
      WRITE (6,52) NT,J                                                 BXPX4980
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,J)                   BXPX4990
      GO TO 42                                                          BXPX5000
C                                                                       BXPX5010
C   FEASIBLE VERTEX FOUND, EVALUATE THE OBJECTIVE FUNCTION.             BXPX5020
   26 NFE = NFE+1                                                       BXPX5030
      FUNTRY = FE(V(1,J))                                               BXPX5040
C                                                                       BXPX5050
C   TEST TO SEE IF FUNCTION VALUE HAS NOT CHANGED.                      BXPX5060
      AFO = ABS(FUNTRY-FUNOLD)                                          BXPX5070
      AMX = AMAX1(ABS(EP*FUNOLD),EP)                                    BXPX5080
C                                                                       BXPX5090
C   ACTIVATE THE FOLLOWING TWO STATEMENTS FOR DIAGNOSTIC PURPOSES ONLY. BXPX5100
C     WRITE (6,99) J,AFO,AMX,FUNTRY,FUNOLD,FUN(J),FUN(JN),NTFS,N        BXPX5110
C  99 FORMAT (1X,I3,6E15.7,2I5)                                         BXPX5120
      IF (AFO.GT.AMX) GO TO 27                                          BXPX5130
      NTFS = NTFS+1                                                     BXPX5140
      IF (NTFS.LT.NCT) GO TO 28                                         BXPX5150
      IER = 0                                                           BXPX5160
      IF (NPR.LE.0) GO TO 42                                            BXPX5170
      WRITE (6,53) K                                                    BXPX5180
      GO TO 42                                                          BXPX5190
   27 NTFS = 0                                                          BXPX5200
C                                                                       BXPX5210
C   IS THE NEW VERTEX NO LONGER WORST?                                  BXPX5220
   28 IF (FUNTRY.LT.FUN(JN)) GO TO 34                                   BXPX5230
```

```
C                                                                       BXPX5240
C    TRIAL VERTEX IS STILL WORST; ADJUST TOWARD CENTROID.                BXPX5250
C    EVERY 'KV'TH TIME, OVER-REFLECT THE OFFENDING VERTEX THROUGH THE    BXPX5260
C    BEST VERTEX.                                                        BXPX5270
      LIMT = LIMT+1                                                      BXPX5280
      IF (MOD(LIMT,KV).NE.0) GO TO 30                                    BXPX5290
      CALL FBV (K,FUN,M)                                                 BXPX5300
C                                                                       BXPX5310
      DO 29 I=1,NV                                                       BXPX5320
      VT = BETA*V(I,M)-ALPHA*V(I,J)                                      BXPX5330
      IF (IP.EQ.1) VT = AINT(VT+.5)                                      BXPX5340
   29 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))                              BXPX5350
C                                                                       BXPX5360
      GO TO 32                                                           BXPX5370
C                                                                       BXPX5380
   30 DO 31 I=1,NV                                                       BXPX5390
      VT = .5*(CEN(I)+V(I,J))                                            BXPX5400
      IF (IP.EQ.1) VT = AINT(VT+.5)                                      BXPX5410
      V(I,J) = VT                                                        BXPX5420
   31 CONTINUE                                                           BXPX5430
C                                                                       BXPX5440
   32 IF (LIMT.LT.NLIM) GO TO 33                                         BXPX5450
C                                                                       BXPX5460
C    CANNOT MAKE THE 'J'TH VERTEX NO LONGER WORST BY DISPLACING TOWARD   BXPX5470
C    THE CENTROID OR BY OVER-REFLECTING THRU THE BEST VERTEX.            BXPX5480
      IER = 2                                                            BXPX5490
      IF (NPR.GT.0) WRITE (6,52) NT,J                                    BXPX5500
      GO TO 42                                                           BXPX5510
   33 NT = NT+1                                                          BXPX5520
      GO TO 20                                                           BXPX5530
C                                                                       BXPX5540
C    SUCCESS:  WE HAVE A REPLACEMENT FOR VERTEX J.                       BXPX5550
   34 FUN(J) = FUNTRY                                                    BXPX5560
      FUNOLD = FUNTRY                                                    BXPX5570
      NPT = NPT+1                                                        BXPX5580
C                                                                       BXPX5590
C    EVERY 100'TH PERMISSIBLE TRIAL, RECOMPUTE CENTROID SUMMATION TO     BXPX5600
C    AVOID CREEPING ERROR.                                               BXPX5610
      IF (MOD(NPT,100).NE.0) GO TO 37                                    BXPX5620
C                                                                       BXPX5630
      DO 36 I=1,NV                                                       BXPX5640
      SUM(I) = 0.                                                        BXPX5650
C                                                                       BXPX5660
      DO 35 N=1,K                                                        BXPX5670
   35 SUM(I) = SUM(I)+V(I,N)                                             BXPX5680
C                                                                       BXPX5690
      CEN(I) = SUM(I)/FK                                                 BXPX5700
   36 CONTINUE                                                           BXPX5710
```

```
C                                                                          BXPX5720
      LC = 0                                                               BXPX5730
      GO TO 39                                                             BXPX5740
C                                                                          BXPX5750
   37 DO 38 I=1,NV                                                         BXPX5760
   38 SUM(I) = SUM(I)+V(I,J)                                               BXPX5770
C                                                                          BXPX5780
      LC = J                                                               BXPX5790
C                                                                          BXPX5800
   39 IF (NPR.LE.0) GO TO 40                                               BXPX5810
      IF (MOD(NPT,NPR).NE.0) GO TO 40                                      BXPX5820
C                                                                          BXPX5830
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,LC)                     BXPX5840
C                                                                          BXPX5850
C    HAS THE MAX. NUMBER OF TRIALS BEEN REACHED WITHOUT CONVERGENCE?       BXPX5860
C    IF NOT, GO TO NEW TRIAL.                                              BXPX5870
   40 IF (NT.GE.NTA) GO TO 41                                              BXPX5880
C                                                                          BXPX5890
C    NEXT-TO-WORST VERTEX NOW BECOMES WORST.                               BXPX5900
      J = JN                                                               BXPX5910
      GO TO 17                                                             BXPX5920
   41 IER = 3                                                              BXPX5930
      IF (NPR.GT.0) WRITE (6,54)                                           BXPX5940
C                                                                          BXPX5950
C    COLLECTOR POINT FOR ALL ENDINGS.                                      BXPX5960
C     1)  CANNOT DEVELOP FEASIBLE VERTEX.               IER = 1            BXPX5970
C     2)  CANNOT DEVELOP A NO-LONGER-WORST VERTEX.      IER = 2            BXPX5980
C     3)  FUNCTION VALUE UNCHANGED FOR K TRIALS.        IER = 0            BXPX5990
C     4)  LIMIT ON TRIALS REACHED.                      IER = 3            BXPX6000
C     5)  CANNOT FIND FEASIBLE VERTEX AT START.         IER = -1           BXPX6010
   42 CONTINUE                                                             BXPX6020
C                                                                          BXPX6030
C    FIND BEST VERTEX.                                                     BXPX6040
      CALL FBV (K,FUN,M)                                                   BXPX6050
      IF (IER.GE.3) GO TO 44                                               BXPX6060
C                                                                          BXPX6070
C    RESTART IF THIS SOLUTION IS SIGNIFICANTLY BETTER THAN THE PREVIOUS,   BXPX6080
C    OR IF THIS IS THE FIRST TRY.                                          BXPX6090
      IF (NPR.LE.0) GO TO 43                                               BXPX6100
      WRITE (6,55) (M,YMN,FUN(M))                                          BXPX6110
   43 IF (FUN(M).GE.YMN) GO TO 47                                          BXPX6120
      IF (ABS(FUN(M)-YMN).LE.AMAX1(EP,EP*YMN)) GO TO 47                    BXPX6130
C                                                                          BXPX6140
C    GIVE IT ANOTHER TRY UNLESS LIMIT ON TRIALS REACHED.                   BXPX6150
   44 YMN = FUN(M)                                                         BXPX6160
      FUN(1) = FUN(M)                                                      BXPX6170
C                                                                          BXPX6180
      DO 45 I=1,NV                                                         BXPX6190
```

```
      CEN(I) = V(I,M)                                                   BXPX6200
      SUM(I) = V(I,M)                                                   BXPX6210
   45 V(I,1) = V(I,M)                                                   BXPX6220
C                                                                       BXPX6230
      DO 46 I=1,NVT                                                     BXPX6240
   46 XS(I) = V(I,M)                                                    BXPX6250
C                                                                       BXPX6260
      IF (IER.LT.3) GO TO 6                                             BXPX6270
   47 IF (NPR.LE.0) GO TO 48                                            BXPX6280
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,V(1,M),-1)               BXPX6290
      WRITE (6,56) FUN(M)                                               BXPX6300
   48 RETURN                                                            BXPX6310
C                                                                       BXPX6320
   49 FORMAT (50HOINDEX AND DIRECTION OF OUTLYING VARIABLE AT STARTI5)  BXPX6330
   50 FORMAT (50HOIMPLICIT CONSTRAINT VIOLATED AT START. DEAD END. )    BXPX6340
   51 FORMAT ('OCANNOT FIND FEASIBLE',I4,'TH VERTEX OR CENTROID AT STARTBXPX6350
     1.')                                                               BXPX6360
   52 FORMAT (10HOAT TRIAL I4,54H CANNOT FIND FEASIBLE VERTEX WHICH IS NBXPX6370
     1O LONGER WORST,I4,15X,'RESTART FROM BEST VERTEX.')                BXPX6380
   53 FORMAT (40HOFUNCTION HAS BEEN ALMOST UNCHANGED FOR I5,7H TRIALS)  BXPX6390
   54 FORMAT (27HOLIMIT ON TRIALS EXCEEDED. )                           BXPX6400
   55 FORMAT ('OBEST VERTEX IS NO.',I3,'  OLD MIN WAS ',E15.7,          BXPX6410
     1 '  NEW MIN IS ',E15.7)                                           BXPX6420
   56 FORMAT ('OMIN OBJECTIVE FUNCTION IS ',E15.7)                      BXPX6430
      END                                                               BXPX6440
      SUBROUTINE FBV (K,FUN,M)                                          BXPX6450
      DIMENSION FUN(50)                                                 BXPX6460
      M = 1                                                             BXPX6470
C                                                                       BXPX6480
      DO 1 I=2,K                                                        BXPX6490
      IF (FUN(M).LE.FUN(I)) GO TO 1                                     BXPX6500
      M = I                                                             BXPX6510
    1 CONTINUE                                                          BXPX6520
C                                                                       BXPX6530
      RETURN                                                            BXPX6540
      END                                                               BXPX6550
      SUBROUTINE BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FN,C,IK)               BXPX6560
      DIMENSION V(50,50), FN(50), C(25)                                 BXPX6570
      WRITE (6,4) NT,NPT,NFE,NCE                                        BXPX6580
C                                                                       BXPX6590
      DO 1 I=1,K                                                        BXPX6600
      WRITE (6,5) FN(I),(V(J,I),J=1,NV)                                 BXPX6610
      IF (NVT.LE.NV) GO TO 1                                            BXPX6620
      NVP = NV+1                                                        BXPX6630
      WRITE (6,6) (V(J,I),J=NVP,NVT)                                    BXPX6640
    1 CONTINUE                                                          BXPX6650
C                                                                       BXPX6660
      IF (IK.NE.0) GO TO 2                                              BXPX6670
```

```
C                                                                       BXPX6680
      WRITE (6,7) (C(I),I=1,NV)                                         BXPX6690
      RETURN                                                            BXPX6700
    2 IF (IK.GE.0) GO TO 3                                              BXPX6710
      WRITE (6,8) (C(I),I=1,NV)                                         BXPX6720
      RETURN                                                            BXPX6730
    3 WRITE (6,9) IK,(C(I),I=1,NV)                                      BXPX6740
      RETURN                                                            BXPX6750
C                                                                       BXPX6760
    4 FORMAT ('ONO. TOTAL TRIALS = ',I5,4X,'NO. FEASIBLE TRIALS = ',    BXPX6770
     1I5,4X,'NO. FUNCTION EVALUATIONS = ',I5,4X,'NO. CONSTRAINT EVALUATIBXPX6780
     2ONS = ',I5/'0     FUNCTION VALUE',6X,'INDEPENDENT VARIABLES/DEPENDBXPX6790
     3ENT OR IMPLICIT CONSTRAINTS')                                     BXPX6800
    5 FORMAT (1H ,E18.7,2X,7E14.7/(21X,7E14.7))                         BXPX6810
    6 FORMAT (21X,7E14.7)                                               BXPX6820
    7 FORMAT (10HOCENTROID 11X,7E14.7/(21X,7E14.7))                     BXPX6830
    8 FORMAT ('0  BEST VERTEX',7X,7E14.7/(21X,7E14.7))                  BXPX6840
    9 FORMAT ('OCENTROID LESS VX',I2,2X,7E14.7/(21X,7E14.7))            BXPX6850
      END                                                               BXPX6860
```

FUNCTION DEGRAD

This function was programmed to convert degrees to radians and radians to degrees. A third purpose is to convert degrees to a range of 0-360. It is used extensively throughout the programs, functions, and subroutines listed in this thesis.

FUNCTION DEGRAD

```
      FUNCTION DEGRAD (NDRFLG,NSHFLG,FUNCT)
C
C     FUNCTION TO CONVERT DEGREES TO RADIANS AND RADIANS TO
C     DEGREES AND SHIFT THE DEGREES TO A RANGE BETWEEN 000.0
C     AND 360.0 IN ACCORDANCE WITH THE FOLLOWING FLAG
C     DEFINITIONS
C     NDRFLG
C          0 - CONVERT RADIANS TO DEGREES
C          1 - CONVERT DEGREES TO RADIANS
C          2 - DO NOT CONVERT
C     NSHFLG
C          0 - CONVERT ANGLES(DEG) TO THE RANGE BETWEEN 000.0
C                AND 360.0
C          1 - DO NOT CONVERT
C
      FUNCU = FUNCT
      IF (NDRFLG.EQ.2) GO TO 2
      IF (NDRFLG.EQ.1) GO TO 1
      FUNCU = FUNCU*180.0/3.141592654
      GO TO 2
    1 FUNCU = FUNCU*3.141592654/180.0
      GO TO 5
    2 IF (NSHFLG.EQ.1) GO TO 5
    3 IF (FUNCU.LT.360.0) GO TO 4
      FUNCU = FUNCU-360.0
      GO TO 3
    4 IF (FUNCU.GE.0.0) GO TO 5
      FUNCU = FUNCU+360.0
      GO TO 4
    5 DEGRAD = FUNCU
      RETURN
      END
```

FUNCTION DELAY

This function is used as the time delay in the speed control optimization runs. It was designed to be used as an equivalence to DELY in DSL simulation.The following variables are defined:

- E(I) is the storage array (should be initialized before the first function call)
- K is the delay step count
- SPDDES is the variable to be delayed for K steps
- P is the flag for delay or no delay
  - $P \geq 0.0$ delay SPDDES
  - $P < 0.0$ function output equal to SPDDES

The function stores the input value (SPDDES) in E(M) and decrements the value in array E(I) at each call of the function until the value is in the position of E(1). The value is then output from the function delayed K intervals.

FUNCTION DELAY

```
      FUNCTION DELAY (K,P,SPDDES,E)
      DIMENSION E(10)
      M = K+1
      E(M) = SPDDES
C
      DO 1 I=1,K
    1 E(I) = E(I+1)
C
      IF (P.LT.0.0) GO TO 2
      DELAY = E(1)
      RETURN
    2 DELAY = SPDDES
      RETURN
      END
```

FUNCTION FE - RUN A

FEA

This function is the simulation for heading control optimization of the approach phase. It is called by subroutine BOXPLX. The integration step size is 0.04 with a final time of 20.0. In this function all initial conditions are set to zero except initial geographic location and speed. The reference ship maintains a straight course and the control ship starts its approach 5 ship lengths astern and 0.4 ship lengths laterally displaced to starboard of the reference ship.

The function is referred to as function FEA in the text.

```
      FUNCTION FE (Z)
C     EVALUATION OF COST FUNCTION AS A FUNCTION OF RSENS,WTSENS,RGN
      DIMENSION Z(8), Y(20), YDOT(20), X(20), XDOT(20)
      REAL *8XDOT,X,DTD,TD
C     HYDRODYNAMIC COEFFICIENTS
      A11 = 0.015
      B11 = 0.01243
      A21 = 0.00027
      B21 = 0.0051
      A12 = 0.000197
      B12 = 0.00351
      A22 = 0.00068
      B22 = 0.00227
      A33 = 0.0085
      B33 = 0.0012
      XKA = 0.0027
      XKB = -0.00126
      XNC = 0.0012
      D = A11*A22-A21*A12
      XKC = 0.0
      XLUC = 20.84765
      DLTDM = 2.0
      DLTEM = 7.0
      XKG = DLTDM/DLTEM
      D2D = 0.0
C     IDENTIFICATION OF GAINS TO BE FOUND
      RSENS = Z(1)
      WTSENS = Z(2)
      RGN = Z(3)
      VFBG = 0.0
C     INITIAL CONDITIONS
C
      DO 1 J=1,14
    1 Y(J) = 0.0
C
C     INITIAL GEOGRAPHIC LOCATION
      Y(5) = 5.0
      Y(6) = 0.0
      Y(10) = 0.0
      Y(11) = 0.4
C     OTHER INITIALIZATIONS
      N = 1
      RD = 1.0
```

```
      IS = 1
      DD = 0.2
      D1 = 0.0
      D2 = 0.0
      YY1 = 0.0
      YY2 = 0.0
      YN1 = 0.0
      YN2 = 0.0
C     SPEED INITIALIZATIONS
      Y(4) = 1.0
      U02 = 1.5
      CDOT2 = 1.5
C     DISTANCE INITIALIZATION
      DYO = Y(10)-Y(5)
      DXO = Y(11)-Y(6)
      CALL TRANS (Y(2),DXO,DYO,ADX,ADY)
      CALL SLOPES (ADX,ADY,YY1,YY2,YN1,YN2)
C     INITIALIZE TIME
      T = 0.0
      DT = 0.04
      JT = 0
C     SHIP A
    2 XIF11 = XKA*D1
      XIF21 = XKB*D1
      XIF31 = XKC*D1+XNC
      XI11 = -B11*Y(1)-B21*Y(3)+XIF11
      XI21 = -B12*Y(1)-B22*Y(3)+XIF21
      XI31 = -B33*Y(4)+XIF31
      YDOT(1) = (XI11*A22-XI21*A21)/D
      YDOT(2) = Y(3)
      YDOT(3) = (XI21*A11-XI11*A12)/D
      YDOT(4) = XI31/A33
      YDOT(5) = Y(4)*COS(Y(2))-Y(1)*SIN(Y(2))
      YDOT(6) = Y(4)*SIN(Y(2))+Y(1)*COS(Y(2))
C     SHIP B
      XIF12 = XKA*D2+YY2
      XIF22 = XKB*D2+YN2
      XIF32 = XKC*D2+XNC
      XI12 = -B11*Y(7)-B21*Y(9)+XIF12
      XI22 = -B12*Y(7)-B22*Y(9)+XIF22
      CDOT2 = SPDCTR(ADX,U02)
      XI32 = -B33*CDOT2+XIF32
      YDOT(7) = (XI12*A22-XI22*A21)/D
      YDOT(8) = Y(9)
      YDOT(9) = (XI22*A11-XI12*A12)/D
      YDOT(10) = CDOT2*COS(Y(8))-Y(7)*SIN(Y(8))
      YDOT(11) = CDOT2*SIN(Y(8))+Y(7)*COS(Y(8))
      DX = Y(10)-Y(5)
```

```
      DY = Y(11)-Y(6)
      CALL TRANS (Y(2),DX,DY,ADX,ADY)
      CALL SLOPES (ADX,ADY,YY1,YY2,YN1,YN2)
      YAWD2 = DEGRAD(0,0,Y(8))
      CALL RBMEAS (N,Y(2),Y(5),Y(6),Y(8),Y(10),Y(11),RD,R1,B1,BB1,R2,B2,
     1BB2)
      CALL HDGRAS (N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,Y(8),PSIDFD,PSIADD,PSI
     1DED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)
      BDOT2D = DEGRAD(0,1,Y(9))
      BDOTFB = VFBG*BDOT2D
      DDUMB = YAWD2-PSIDED+BDOTFB
      IF (DDUMB.GT.180.0) DDUMB=DDUMB-360.0
      IF (DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
      DLTS = XLIMIT(-30.0,30.0,DDUMB*RGN)
      DLTE = DLTS-D2D
      DLTBE = XLIMIT(-DLTEM,DLTEM,DLTE)
      YDOT(14) = XKG*DLTBE*XLUC
      D2D = Y(14)
      D2 = DEGRAD(1,1,D2D)
      DTRAN = T*ABS(D2)
      YDOT(12) = DTRAN
      DISTE = T*10.0*ABS(DD-ADY)
      YDOT(13) = DISTE
      OBJ = Y(12)+Y(13)
C
      DO 3 J=1,14
      X(J) = DBLE(Y(J))
    3 XDOT(J) = DBLE(YDOT(J))
C
      TD = DBLE(T)
      DTD = DBLE(DT)
      ZS = RKLDEQ(14,X,XDOT,TD,DTD,JT)
C
      DO 4 J=1,14
      Y(J) = SNGL(X(J))
    4 YDOT(J) = SNGL(X(J))
C
      T = SNGL(TD)
      DT = SNGL(DTD)
      IF (ZS-1.) 5,2,6
    5 WRITE (6,8)
      STOP
    6 IF (T.GT.20.0) GO TO 7
      GO TO 2
    7 FE = OBJ
      WRITE (6,9) OBJ,RSENS,WTSENS,RGN
      RETURN
C
```

```
8 FORMAT (' RKLDEQ RETURNED VALUE LT 1.0, INTEGRATION PROBLEM')       FEZA1400
9 FORMAT (' EXIT FUNCTION FE(Z)     OBJ=',F15.8,'     RSENS=',F15.8,' FEZA1410
 1      WTSENS=',F15.8,'      RGN=',F15.8)                             FEZA1420
  END                                                                  FEZA1430
```

FUNCTION FE - RUN B

FEB

This function is the simulation for heading control optimization of the turn phase. It is called by subroutine BOXPLX. The integration step size is 0.04 with a final time of 20.0. In this function, the following initial conditions are non-zero:

control ship rudder angle D2D & Y(14) = 8.7 degrees
lateral displacement Y(11) = 0.2
reference ship's speed U01 & Y(4) = 1.0
control ship's speed U02 & CDCT2 = 1.5 (after first
step becomes 1.0)

The reference ship's rudder is activated to 5.0 degrees between time 4.0 and 5.0. The runs were for port side replenishment.

The function is referred to as function FEB in the text.

```
      FUNCTION FE (Z)                                                   FEZB  10
C     EVALUATION OF COST FUNCTION AS A FUNCTION OF RSENS,WTSENS,RGN     FEZB  20
      DIMENSION Z(8), Y(20), YDOT(20), X(20), XDOT(20)                  FEZB  30
      REAL *8XDOT,X,DTD,TD                                              FEZB  40
C     HYDRODYNAMIC COEFFICIENTS                                         FEZB  50
      A11 = 0.015                                                       FEZB  60
      B11 = 0.01243                                                     FEZB  70
      A21 = 0.00027                                                     FEZB  80
      B21 = 0.0051                                                      FEZB  90
      A12 = 0.000197                                                    FEZB 100
      B12 = 0.00351                                                     FEZB 110
      A22 = 0.00068                                                     FEZB 120
      B22 = 0.00227                                                     FEZB 130
      A33 = 0.0085                                                      FEZB 140
      B33 = 0.0012                                                      FEZB 150
      XKA = 0.0027                                                      FEZB 160
      XKB = -0.00125                                                    FEZB 170
      XNC = 0.0012                                                      FEZB 180
      D = A11*A22-A21*A12                                               FEZB 190
      XKC = 0.0                                                         FEZB 200
      XLUC = 20.84765                                                   FEZB 210
      DLTDM = 2.0                                                       FEZB 220
      DLTEM = 7.0                                                       FEZB 230
      XKG = DLTDM/DLTEM                                                 FEZB 240
      D1D = 0.0                                                         FEZB 250
      D2D = 8.7                                                         FEZB 260
C     IDENTIFICATION OF GAINS TO BE FOUND                               FEZB 270
      RSENS = Z(1)                                                      FEZB 280
      WTSENS = Z(2)                                                     FEZB 290
      RGN = Z(3)                                                        FEZB 300
      VFBG = Z(4)                                                       FEZB 310
C     INITIAL CONDITIONS                                                FEZB 320
C                                                                       FEZB 330
      DO 1 J=1,14                                                       FEZB 340
    1 Y(J) = 0.0                                                        FEZB 350
C                                                                       FEZB 360
C     INITIAL GEOGRAPHIC LOCATION                                       FEZB 370
      Y(5) = 0.0                                                        FEZB 380
      Y(6) = 0.0                                                        FEZB 390
      Y(10) = 0.0                                                       FEZB 400
      Y(11) = 0.2                                                       FEZB 410
      Y(14) = 8.7                                                       FEZB 420
C     OTHER INITIALIZATIONS                                             FEZB 430
```

```
      N = 1
      RD = 1.0
      IS = 1
      DD = 0.2
      D1 = 0.0
      D2 = DEGRAD(1,1,D2D)
      YY1 = 0.0
      YY2 = 0.0
      YN1 = 0.0
      YN2 = 0.0
C     SPEED INITIALIZATIONS
      Y(4) = 1.0
      U01 = 1.0
      U02 = 1.5
      CDOT2 = 1.5
C     DISTANCE INITIALIZATION
      DY0 = Y(10)-Y(5)
      DX0 = Y(11)-Y(6)
      CALL TRANS (Y(2),DX0,DY0,ADX,ADY)
      CALL SLOPES (ADX,ADY,YY1,YY2,YN1,YN2)
C     INITIALIZE TIME
      T = 0.0
      DT = 0.04
      JT = 0
C     SHIP A
    2 D1DES = 0.0
      IF ((T.GE.4.0).AND.(T.LE.5.0)) D1DES=5.0
      DLTS1 = XLIMIT(-30.0,30.0,D1DES)
      DLTE1 = DLTS1-D1D
      DLTBE1 = XLIMIT(-DLTEM,DLTEM,DLTE1)
      YDOT(12) = XKG*DLTBE1*XLUC
      D1D = Y(12)
      D1 = DEGRAD(1,1,D1D)
      XIF11 = XKA*D1
      XIF21 = XKB*D1
      XIF31 = XKC*D1+XNC
      XI11 = -B11*Y(1)-B21*Y(3)+XIF11
      XI21 = -B12*Y(1)-B22*Y(3)+XIF21
      XI31 = -B33*Y(4)+XIF31
      YDOT(1) = (XI11*A22-XI21*A21)/D
      YDOT(2) = Y(3)
      YDOT(3) = (XI21*A11-XI11*A12)/D
      YDOT(4) = XI31/A33
      YDOT(5) = Y(4)*COS(Y(2))-Y(1)*SIN(Y(2))
      YDOT(6) = Y(4)*SIN(Y(2))+Y(1)*COS(Y(2))
      YAWD1 = DEGRAD(0,0,Y(2))
C     SHIP B
      XIF12 = XKA*D2+YY2
```

```
      XIF22 = XKB*D2+YN2                                                FEZB 920
      XIF32 = XKC*D2+XNC                                                FEZB 930
      XI12 = -B11*Y(7)-B21*Y(9)+XIF12                                   FEZB 940
      XI22 = -B12*Y(7)-B22*Y(9)+XIF22                                   FEZB 950
      CDOT2 = SPDCTR(ADX,U01,U02)                                       FEZB 960
      XI32 = -B33*CDOT2+XIF32                                           FEZB 970
      YDOT(7) = (XI12*A22-XI22*A21)/D                                   FEZB 980
      YDOT(8) = Y(9)                                                    FEZB 990
      YDOT(9) = (XI22*A11-XI12*A12)/D                                   FEZB1000
      YDOT(10) = CDOT2*COS(Y(8))-Y(7)*SIN(Y(8))                         FEZB1010
      YDOT(11) = CDOT2*SIN(Y(8))+Y(7)*COS(Y(8))                         FEZB1020
      DX = Y(10)-Y(5)                                                   FEZB1030
      DY = Y(11)-Y(6)                                                   FEZB1040
      CALL TRANS (Y(2),DX,DY,ADX,ADY)                                   FEZB1050
      CALL SLOPES (ADX,ADY,YY1,YY2,YN1,YN2)                             FEZB1060
      YAWD2 = DEGRAD(0,0,Y(8))                                          FEZB1070
      CALL RBMEAS (N,Y(2),Y(5),Y(6),Y(8),Y(10),Y(11),RD,R1,B1,BB1,R2,B2,FEZB1080
     1BB2)                                                              FEZB1090
      CALL HDGRAS (N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,Y(8),PSIDFD,PSIADD,PSIFEZB1100
     1DED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)                               FEZB1110
      BDOT2D = DEGRAD(0,1,Y(9))                                         FEZB1120
      BDOTFB = VFBG*BDOT2D                                              FEZB1130
      DDUMB = YAWD2-PSIDED+BDOTFB                                       FEZB1140
      IF (DDUMB.GT.180.0) DDUMB=DDUMB-360.0                             FEZB1150
      IF (DDUMB.LT.-180.0) DDUMB=360.0+DDUMB                            FEZB1160
      DLTS = XLIMIT(-30.0,30.0,DDUMB*RGN)                               FEZB1170
      DLTE = DLTS-D2D                                                   FEZB1180
      DLTBE = XLIMIT(-DLTEM,DLTEM,DLTE)                                 FEZB1190
      YDOT(14) = XKG*DLTBE*XLUC                                         FEZB1200
      D2D = Y(14)                                                       FEZB1210
      D2 = DEGRAD(1,1,D2D)                                              FEZB1220
      DISTE = ABS(DD-ADY)                                               FEZB1230
      YDOT(13) = DISTE                                                  FEZB1240
      OBJ = Y(13)                                                       FEZB1250
C                                                                       FEZB1260
      DO 3 J=1,14                                                       FEZB1270
      X(J) = DBLE(Y(J))                                                 FEZB1280
    3 XDOT(J) = DBLE(YDOT(J))                                           FEZB1290
C                                                                       FEZB1300
      TD = DBLE(T)                                                      FEZB1310
      DTD = DBLE(DT)                                                    FEZB1320
      ZS = RKLDEQ(14,X,XDOT,TD,DTD,JT)                                  FEZB1330
C                                                                       FEZB1340
      DO 4 J=1,14                                                       FEZB1350
      Y(J) = SNGL(X(J))                                                 FEZB1360
    4 YDOT(J) = SNGL(X(J))                                              FEZB1370
C                                                                       FEZB1380
      T = SNGL(TD)                                                      FEZB1390
```

```
      DT = SNGL(DTD)                                                    FEZB1400
      IF (ZS-1.) 5,2,6                                                  FEZB1410
    5 WRITE (6,8)                                                       FEZB1420
      STOP                                                              FEZB1430
    6 IF (T.GT.20.0) GO TO 7                                            FEZB1440
      GO TO 2                                                           FEZB1450
    7 FE = OBJ                                                          FEZB1460
      WRITE (6,9) OBJ,ADY,ADX,YAWD1,YAWD2,D2D                           FEZB1470
      RETURN                                                            FEZB1480
C                                                                       FEZB1490
    8 FORMAT (' RKLDEQ RETURNED VALUE LT 1.0, INTEGRATION PROBLEM')     FEZB1500
    9 FORMAT ('  EXIT FUNCTION FE(Z),OBJ=',F15.8,',ADY=',F15.8,',ADX=', FEZB1510
     1 F15.8,/,5X,'YAWD1=',F15.8,',YAWD2=',F15.8,',D2D=',F15.8)         FEZB1520
      END                                                               FEZB1530
```

FUNCTION FE - RUN C

FEC

This function is the simplified simulation for speed control optimization of the switching function SW. It is called by subroutine BOXPLX. The function shown is for approach speed of 1.1 and a replenishment speed of 1.0. The runs were made for various realistic combinations to obtain an optimum switching curve.

The run used a step size of 0.04 and a final time of 10.0. The two ships were run linearily with only the longitudinal direction and motion of any concern.

This function is referred to as function FEC in the text.

FUNCTION FE - RUN C

```
      FUNCTION FE (Z)
      DIMENSION Z(8), X(10), XDOT(10), Y(10), YDOT(10), E(10)
      REAL *8XDOT,X,DTD,TD
C     IDENTIFICATION OF GAINS
      SW = Z(1)
C     INITIAL CONDITIONS
      N = 2
      APOSX = 1.1
      BPOSX = 0.0
C
      DO 1 I=1,N
      Y(I) = 0.0
    1 YDOT(I) = 0.0
C
C
      DO 2 I=1,10
    2 E(I) = 0.0
C
      SPDO1 = 1.0
      SPDO2 = 1.1
      Y(1) = 1.1
      XK1 = 1.012425
      P = -1.0
      A = 22.0
      G = 0.092
      XLUC = 20.84765
      UF = 21.73
      XK = UF/A
      T = 0.0
      DT = 0.04
      JT = 0
      SPDDER = XK1*SPDO1
      ADX = BPOSX-APOSX
C     ITERATION LOOP
    3 IF (T.GT.0.24) P = 4.88/XLUC
      IF (ABS(ADX).GT.2.0) GO TO 9
      SPDDES = XK1*(SPDREC(ADX,SPDO1,SPDO2,SW)-SPDO1)
      SPDDEL = DELAY(6,P,SPDDES,E)
      SPDIN = XK*(SPDDEL+SPDDER)
      SPDERR = G*(SPDIN-Y(1))
      YDOT(1) = SPDERR*XLUC
      YDOT(2) = T*ABS(ADX)
      OBJ = Y(2)
```

```
C
      DO 4 I=1,N
      X(I) = DBLE(Y(I))
    4 XDOT(I) = DBLE(YDOT(I))
C
      TD = DBLE(T)
      DTD = DBLE(DT)
      ZS = RKLDEQ(N,X,XDOT,TD,DTD,JT)
C
      DO 5 I=1,N
      Y(I) = SNGL(X(I))
    5 YDOT(I) = SNGL(XDOT(I))
C
      T = SNGL(TD)
      DT = SNGL(DTD)
      IF (ZS-1.) 6,3,7
    6 WRITE (6,10)
      STOP
    7 IF (T.GT.10.0) GO TO 8
      APOSX = APOSX+SPDO1*DT
      BPOSX = BPOSX+Y(1)*DT
      ADX = BPOSX-APOSX
      GO TO 3
    8 FE = OBJ
      WRITE (6,11) OBJ,SW,ADX,Y(1)
      RETURN
    9 FE = 1.0E06
      WRITE (6,12) ADX
      RETURN
C
   10 FORMAT (' RKLDEQ RETURNED ZS FLAG LT 0.0, INTEGRATION PROBLEM')
   11 FORMAT ('  EXIT FUNCTION FE(Z)  OBJ=',F15.8,5X,'SW=',F15.8,/,
     1 22X,'ADX=',F15.8,5X,'FINAL SPEED=',F15.8)
   12 FORMAT ('  EXIT FUNCTION FE(Z)   ABS(ADX).GT.2.0,   ADX=',F15.8)
      END
```

SUBROUTINE HDGRAS

This subroutine was programmed to calculate the desired heading (PSIDES) for RAS heading control. It uses the outputs of subroutine RBMEAS to calculate this heading with gains RSENS and WTSENS. The large number of outputs in the subroutine call statement were made for ease of DSL printed output for tracking of simulation accuracy.

The subroutine also incorporates a loop to avoid computer precision problems in the ARSIN function.

SUBROUTINE HDGRAS

```
      SUBROUTINE HDGRAS (N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,PSIB,PSIDFD,PSIA
     1DD,PSIDED,DA,A1D,B1D,B2D,WTSENS,DD,D)
C     SUBROUTINE TO CALCULATE DESIRED HEADING FOR RAS
C     IF N SET TO 1 HDGRAS WILL BE USED
C
      IF (N.NE.1) GO TO 4
C
C     IS DENOTES THE SIDE OF APPROACH OF RECEIVING SHIP IS=1 PORT,
C        IS=0 STBD
C     RSENS - ADDITIONAL HEADING SENSITIVITY DUE TO SEPARATION DISTANCE
C                RSENS=1.0 CORRESPONDS TO 10.87 DEG/10FT FO A 527 FT
C                SHIP
C     PSIDFD - ADDITIONAL HEADING DUE TO HEADING DIFFERENCE(WEIGHTED)
C     PSIADD - ADDITIONAL HEADING DUE TO DISTANCE ERROR(WEIGHTED)
C     PSIDED - TOTAL DESIRED HEADING=PSIDFD(WEIGHTED)+PSIADD(WEIGHTED)
C              +PSIBD
C     NOTE  ALL RETURNED VALUES ENDING IN "D" ARE IN DEGREES
C     WTSENS - WEIGHTING FACTOR GAIN FOR DIFFERENCE IN HEADINGS
C     DA - CG ABS DISTANCE BETWEEN SHIPS
C     A1D - CG REL BEARING BETWEEN SHIPS
C
C     HEADING DIFFERENCE
      ARG = (R1*SIN(B1)-R2*SIN(B2))/D
      IF (ARG.GT.1.0) GO TO 1
      IF (ARG.LT.-1.0) GO TO 2
      PSIDIF = ARSIN(ARG)
      GO TO 3
    1 PSIDIF = ARSIN(1.0)
      GO TO 3
    2 PSIDIF = ARSIN(-1.0)
    3 CONTINUE
C     CG DISTANCE
      DA = (R1+R2)/2.0
C     CG RELATIVE BEARING
      AA1 = (BB1+BB2)/2.0
C     INITIAL SENSE OF APPROACH SIDE DESIRED DISTANCE
      DDC = DD
      IF (IS.EQ.0) DDC = -DD
C     ADDITIONAL HEADING DUE TO DISTANCE
      PSIADC = RSENS*(DDC+DA*SIN(AA1))
C     TOTAL DESIRED HEADING
      PSIDES = PSIADC+WTSENS*PSIDIF+PSIB
C     CONVERT RADIANS TO DEGREES
```

```
      PSIDFD = DEGRAD(0,0,PSIDIF)
      PSIADD = DEGRAD(0,0,PSIADC)
      PSIDED = DEGRAD(0,0,PSIDES)
      A1 = 6.283185307+AA1
      A1D = DEGRAD(0,0,A1)
      B1D = DEGRAD(0,0,B1)
      B2D = DEGRAD(0,0,B2)
    4 RETURN
      END
```

FUNCTION KE

This function is required by all optimization runs. It is the function that contains constraints for subroutine BOXPLX. No constraints are present, consequently function KE is set equal to 0.

FUNCTION KE

```
      FUNCTION KE (X)                                    KEX
      DIMENSION X(8)                                     KEX
      KE = 0                                             KEX
      RETURN                                             KEX
      END                                                KEX
```

MAIN PROGRAM FOR FUNCTION MINIMIZATIONS

MINIBXPX

This is a generalized program which calls subroutine BOXPLX. Its main purpose is input and output of the values required in the optimization runs. This is referred to as MINIBXPX in the text.

MAIN PROGRAM FOR FUNCTION MINIMIZATIONS

```
      DIMENSION X(8), XS(8), BU(8), BL(8)
      CALL ERRSET (257,256,0,1,1)
      READ (5,4) N,NAV,IP
      READ (5,3) (BU(I),I=1,N)
      READ (5,3) (BL(I),I=1,N)
      READ (5,3) (XS(I),I=1,N)
      READ (5,4) NT,NPR
      WRITE (6,5) N
C
      DO 1 I=1,N
    1 WRITE (6,6) I,BU(I),I,BL(I),I,XS(I)
C
      WRITE (6,7) NT
      WRITE (6,8)
      CALL BOXPLX (N,NAV,NPR,NT,2.0,XS,IP,BU,BL,OBJ,IER)
      WRITE (6,9)
C
      DO 2 I=1,N
    2 WRITE (6,10) I,XS(I)
C
      WRITE (6,11) OBJ
      WRITE (6,12) IER
      WRITE (6,8)
      STOP
C
    3 FORMAT (8F10.5)
    4 FORMAT (3I10)
    5 FORMAT ('1',' VARIABLES - UPPER LIMIT(BU), LOWER LIMIT(BL),  START
     1ING VALUE(XS) - NUMBER OF VARIABLES=',I5,//)
    6 FORMAT (' BU(',I2,')=',F10.5,'     BL(',I2,')=',F10.5,'     XS(',I
     12,')=',F10.5,/)
    7 FORMAT (/,' NUMBER OF TRIALS TO BE ALLOWED=',I10)
    8 FORMAT ('1')
    9 FORMAT ('1',' OUTPUT RESULTS',///,' COORDINATES OF MINIMUM',/)
   10 FORMAT (/,' Z(',I2,')=',1P1E15.8)
   11 FORMAT (//,' TOTAL COST IS =',1P1E15.8)
   12 FORMAT (//,' ERROR CODE IS =',I2)
      END
```

SUBROUTINE RBMEAS

This subroutine measures the range and bearing of the forward and after stations which is required of subroutine HDGRAS. This is done with trignometric functions as shown in chapter II. The subroutine is specifically designed to circumvent any ambiguities usually associated with these functions.

It is the basis of the decoupling of the two RAS ships that this thesis is based.

SUBROUTINE RBMEAS

```
      SUBROUTINE RBMEAS (N,PSIA,X1,Y1,PSIB,X2,Y2,D,R1,B1,BB1,R2,B2,BB2)  RBMS
C     SUBROUTINE TO CALCULATE THE RELATIVE POSITIONS OF SHIP A (SUPPLY)  RBMS
C          TO SHIP B (RECEIVING)                                         RBMS
C     PSIA - HEADING OF SHIP A (RAD)                                     RBMS
C     PSIB - HEADING OF SHIP B (RAD)                                     RBMS
C     X1,Y1 - COORDINATES OF SHIP A                                      RBMS
C     X2,Y2 - COORDINATES OF SHIP B                                      RBMS
C     D - DISTANCE BETWEEN SENSORS(ALSO BETWEEN REFLECTORS)(SAME ON      RBMS
C            BOTH SHIPS)                                                 RBMS
C     R1,R2 - FWD,AFT MEASURED DISTANCE                                  RBMS
C     B1,B2 - FWD,AFT RELATIVE BEARINGS(RAD)                             RBMS
C     BB1,BB2 - FWD,AFT RELATIVE BEARINGS(RAD) WITH SIGN                 RBMS
C     IF N SET TO 1 RBMEAS SUBROUTINE WILL BE USED                       RBMS
      IF (N.NE.1) GO TO 7                                                RBMS
C     ANGLES AND DISTANCES FOR SENSOR INPUT                              RBMS
      SDF1 = (D/2.0)*SIN(PSIA)                                           RBMS
      SDF2 = (D/2.0)*SIN(PSIB)                                           RBMS
      SDF3 = (D/2.0)*COS(PSIA)                                           RBMS
      SDF4 = (D/2.0)*COS(PSIB)                                           RBMS
C     FWD DISTANCE                                                       RBMS
      ADFX = SDF3+X1-SDF4-X2                                             RBMS
      ADFY = SDF1+Y1-SDF2-Y2                                             RBMS
      R1 = SQRT(ADFX**2+ADFY**2)                                         RBMS
C     AFT DISTANCE                                                       RBMS
      ADAX = -SDF3+X1+SDF4-X2                                            RBMS
      ADAY = -SDF1+Y1+SDF2-Y2                                            RBMS
      R2 = SQRT(ADAX**2+ADAY**2)                                         RBMS
C     FWD ANGLE                                                          RBMS
      IF (ADFY.EQ.0.0) GO TO 2                                           RBMS
      BB1 = ATAN2(ADFY,ADFX)                                             RBMS
    1 BB1 = BB1-PSIB                                                     RBMS
      B1 = BB1+6.283185307                                               RBMS
      GO TO 3                                                            RBMS
    2 BB1 = 1.570796327                                                  RBMS
      IF (ADFY.LE.0.0) BB1=-BB1                                          RBMS
      GO TO 1                                                            RBMS
C     AFT ANGLE                                                          RBMS
    3 IF (ADAX.EQ.0.0) GO TO 5                                           RBMS
      BB2 = ATAN2(ADAY,ADAX)                                             RBMS
    4 BB2 = BB2-PSIB                                                     RBMS
      B2 = BB2+6.283185307                                               RBMS
      GO TO 6                                                            RBMS
    5 BB2 = 1.570796327                                                  RBMS
```

```
      IF (ADAY.LE.0.0) BB2=-BB2                                         RBMS 440
      GO TO 4                                                           RBMS 450
    6 CONTINUE                                                          RBMS 460
    7 RETURN                                                            RBMS 470
      END                                                               RBMS 480
```

FUNCTION RKLDEQ

This function is the Runge-Kutta-Gill forth-order integration used in all optimization runs. It is programmed locally and is part of the IBM 360 SSP library. A full explaination and description is shown in the first few pages of the function listing.

FUNCTION RKLDEQ

```
C
C     FUNCTION RKLDEQ    (FORTRAN 4,G/H) OR ASSEMBLER LANGUAGE)            RKLDF003
C                                                                          RKLDF004
C     IDENTIFICATION D2-NPS-RKLDEQ, CHECKED OUT BY R. HILLEARY, 4/67.      RKLDF005
C                                                                          RKLDF006
C     PURPOSE                                                              RKLDF007
C         THIS ROUTINE SOLVES A SYSTEM OF N FIRST-ORDER ORDINARY DIF-      RKLDF008
C         FERENTIAL EQUATIONS BY THE RUNGE-KUTTA-GILL FOURTH-ORDER ME-     RKLDF009
C         THOD. ALL CALCULATIONS ARE IN DOUBLE-PRECISION.                  RKLDF010
C                                                                          RKLDF011
C     USAGE -(WHEN USED BY FORTRAN CALLING PROGRAM)                        RKLDF012
C                                                                          RKLDF013
C         S = RKLDEQ (N,Y,F,X,H,NT )                                       RKLDF014
C                                                                          RKLDF015
C         FOUR ENTRIES ARE REQUIRED TO ADVANCE THE SOLUTION FROM X TO      RKLDF016
C         X + H WHERE H IS THE INCREMENT. SEE SAMPLE PROBLEM FOR MORE      RKLDF017
C         INFORMATION                                                      RKLDF018
C                                                                          RKLDF019
C     DESCRIPTION OF PARAMETERS                                            RKLDF020
C         N - NUMBER OF FIRST-ORDER EQUATIONS IN SYSTEM TO BE SOLVED.      RKLDF021
C             (O.LE.N.LE.25).                                              RKLDF022
C         Y - NAME OF LINEAR ARRAY OF LENGTH AT LEAST N, IN WHICH SOLU-    RKLDF023
C             TION VALUES WILL BE STORED BY RKLDEQ. THE CALLING PROGRAM    RKLDF024
C             SHOULD SUPPLY INITIAL VALUES BEFORE FIRST ENTRY.             RKLDF025
C         F - NAME OF LINEAR ARRAY OF LENGTH AT LEAST N, IN WHICH THE      RKLDF026
C             DERIVATIVES, COMPUTED IN USER'S CALLING PROGRAM, ARE STORERKLDF027
C         X - THE INDEPENDENT VARIABLE, WHICH IS ADVANCED WITHIN RKLDEQ.RKLDF028
C         H - THE INCREMENT FOR X, WHICH MAY BE CHANGED AT THE END         RKLDF029
C              OF ANY INTERVAL. (WHEN S=2.0)                               RKLDF030
C         NT - AN INTEGER WHICH COUNTS THE NUMBER OF TIMES ENTRY TO        RKLDF031
C              RKLDEQ HAS BEEN MADE DURING CURRENT INTERVAL. IT MUST       RKLDF032
C              BE INITIALLY SET TO ZERO BY USER BEFORE FIRST CALL OF       RKLDF033
C              RKLDEQ.  SUBSEQUENTLY IT SHOULD NOT BE CHANGED BY USER.     RKLDF034
C         S - A SWITCH TO BE TESTED BY USER UPON RETURN FROM RKLDEQ.       RKLDF035
C              IF S = 1.0, THE CALLING PROGRAM SHOULD NOW COMPUTE          RKLDF036
C                          VALUES OF F, USING CURRENT VALUES OF X AND Y,   RKLDF037
C                          AND THEN RETURN TO RKLDEQ.                      RKLDF038
C              IF S=2.0, AND END OF PRESENT INTERVAL HAS BEEN REACHED.     RKLDF039
C                          USER SHOULD STORE AND/OR OUTPUT CURRENT X AND/RKLDF040
C                          OR Y AND TEST FOR END OF COMPUTATION.           RKLDF041
C                                                                          RKLDF042
C                          SEE SAMPLE PROBLEM.                             RKLDF043
C                                                                          RKLDF044
```

```
C       REMARKS                                                         RKLDF045
C           Y,F,X,AND H ARE DOUBLE PRECISION (REAL*8), RKLDEQ IS REAL*4.  RKLDF046
C           MAXIMUM N IS NOW 25.                                        RKLDF047
C                                                                       RKLDF048
C       NOTE                                                            RKLDF049
C          TWO DECKS EXIST FOR THIS FUNCTION. ONE IS IN F4 SOURCE LANGUAGERKLDF050
C          THE OTHER IS ASSEMBLER LANGUAGE AND IS TO BE CALLED WITH A     RKLDF051
C          FORTRAN TYPE CALLING SEQUENCE. RKLDEQ IS RETURNED IN F.P.REG.0 RKLDF052
C                                                                       RKLDF053
C       SAMPLE PROBLEM                                                  RKLDF054
C                                                                       RKLDF055
C       DIMENSION Y(2), F(2)                                            RKLDF056
C       REAL*8 Y,F,X,H                                                  RKLDF057
C       Y(1) =0.D0                                                      RKLDF058
C       Y(2) = 1.D0                                                     RKLDF059
C       X = 1.D0                                                        RKLDF060
C       H = .01D0                                                       RKLDF061
C       NT = 0                                                          RKLD06A
C       WRITE (6,11) X,Y(1),Y(2)                                        RKLDF062
C     CALCULATION OF DERIVATIVESO                                       RKLDF063
C     1 F(1) = X * Y(1) - Y(2)**2                                       RKLDF064
C       F(2) = Y(1) + DSQRT(X)                                          RKLDF065
C                                                                       RKLDF066
C       S = RKLDEQ(2,Y,F,X,H,NT)                                        RKLDF067
C                                                                       RKLDF068
C       IF (S- 1.0) 10,1,2                                              RKLDF069
C                                                                       RKLDF070
C     ERROR STOPO                                                       RKLDF071
C    10 STOP                                                            RKLDF072
C                                                                       RKLDF073
C     2 WRITE (6,11) X, Y(1), Y(2)                                      RKLDF074
C    11 FORMAT (1X,3D25.13)                                             RKLDF075
C                                                                       RKLDF076
C     TEST FOR END OF COMPUTATIONO                                      RKLDF077
C       IF  (DABS(X - 2.5D0) - 1.D-5 ) 3,3,1                            RKLDF078
C     3 STOP                                                            RKLDF079
C       END                                                             RKLDF080
C                                                                       RKLDF081
C       ...............................................................RKLDF082
C           FORTRAN 4 VERSION OF RUNGE-KUTTA-GILL ROUTINE               RKLDF083
C         X,Y,F,H  ARE DOUBLE-PRECISION. MAX N = 25                     RKLDF084
        FUNCTION RKLDEQ (N,Y,F,X,H,NT)                                  RKLDF085
        REAL*8 Y,F,X,H,Q,H1,H2,H3,H6                                    RKLDF086
        DIMENSION Y(1), F(1), Q(25)                                     RKLDF087
C                                                                       RKLDF088
        NT = NT +1                                                      RKLDF089
        GO TO (1,2,3,4),NT                                              RKLDF090
      1 H1 = H                                                          RKLDF091
```

```
      H2 = H1  *  .5D0                                               RKLDF092
      H3 = H1  *  2.D0                                               RKLDF093
      H6 = H1/6.D0                                                   RKLDF094
      DO 11 J =1,N                                                   RKLDF095
   11 Q(J) = 0.D0                                                    RKLDF096
      A = .5D0                                                       RKLDF097
      X = X + H2                                                     RKLDF098
      GO TO 5                                                        RKLDF099
C                                                                    RKLDF100
    2 A = .2928932188134525                                          RKLDF101
      GO TO 5                                                        RKLDF102
C                                                                    RKLDF103
    3 A = 1.7071067811865475                                         RKLDF104
      X = X + H2                                                     RKLDF105
      GO TO 5                                                        RKLDF106
C                                                                    RKLDF107
    4 DO 41 I = 1,N                                                  RKLDF108
   41 Y(I) = Y(I) + H6 * F(I) -Q(I)/3.D0                             RKLDF109
      NT = 0                                                         RKLDF110
      RKLDEQ =2.                                                     RKLDF111
      GO TO 6                                                        RKLDF112
C                                                                    RKLDF113
    5 DO 51 L = 1,N                                                  RKLDF114
      Y(L) = Y(L) + A *(H * F(L) -Q(L))                              RKLDF115
   51 Q(L) = H3 * A *F(L) +(1.D0-3.D0*A) *Q(L)                       RKLDF116
      RKLDEQ =1.                                                     RKLDF117
C                                                                    RKLDF118
    6 RETURN                                                         RKLDF119
      END                                                            RKLDF120
```

SUBROUTINE SLOPES

This subroutine contains the table look-up and interpolation scheme for the interactive forces and moments presented in the RAS environment. It is long and must be pre-compiled for most of the DSL simulation programs shown in this thesis.

```
      SUBROUTINE SLOPES (DX,DY,YY1,YY2,YN1,YN2)                          SLOP  10
C     TABLE LOOK-UP AND INTERPOLATION                                    SLOP  20
      DIMENSION Z(23,16), W(23,16), X(23), Y(16)                         SLOP  30
      X(1) = -1.1                                                        SLOP  40
      X(2) = -1.                                                         SLOP  50
      X(3) = -.9                                                         SLOP  60
      X(4) = -.8                                                         SLOP  70
      X(5) = -.7                                                         SLOP  80
      X(6) = -.6                                                         SLOP  90
      X(7) = -.5                                                         SLOP 100
      X(8) = -.4                                                         SLOP 110
      X(9) = -.3                                                         SLOP 120
      X(10) = -.2                                                        SLOP 130
      X(11) = -.1                                                        SLOP 140
      X(12) = 0.                                                         SLOP 150
      X(13) = .1                                                         SLOP 160
      X(14) = .2                                                         SLOP 170
      X(15) = .3                                                         SLOP 180
      X(16) = .4                                                         SLOP 190
      X(17) = .5                                                         SLOP 200
      X(18) = .6                                                         SLOP 210
      X(19) = .7                                                         SLOP 220
      X(20) = .8                                                         SLOP 230
      X(21) = .9                                                         SLOP 240
      X(22) = 1.                                                         SLOP 250
      X(23) = 1.1                                                        SLOP 260
      Y(1) = 0.10                                                        SLOP 270
      Y(2) = 0.12                                                        SLOP 280
      Y(3) = 0.14                                                        SLOP 290
      Y(4) = 0.16                                                        SLOP 300
      Y(5) = 0.18                                                        SLOP 310
      Y(6) = 0.2                                                         SLOP 320
      Y(7) = 0.22                                                        SLOP 330
      Y(8) = 0.24                                                        SLOP 340
      Y(9) = 0.26                                                        SLOP 350
      Y(10) = 0.28                                                       SLOP 360
      Y(11) = 0.30                                                       SLOP 370
      Y(12) = 0.32                                                       SLOP 380
      Y(13) = 0.34                                                       SLOP 390
      Y(14) = 0.36                                                       SLOP 400
      Y(15) = 0.38                                                       SLOP 410
      Y(16) = 0.40                                                       SLOP 420
      Z(1,1) = 0.                                                        SLOP 430
```

```
      Z(1,2) = 0.                                                       SLOP 440
      Z(1,3) = 0.                                                       SLOP 450
      Z(1,4) = 0.                                                       SLOP 460
      Z(1,5) = 0.                                                       SLOP 470
      Z(1,6) = 0.                                                       SLOP 480
      Z(1,7) = 0.                                                       SLOP 490
      Z(1,8) = 0.                                                       SLOP 500
      Z(1,9) = 0.                                                       SLOP 510
      Z(1,10) = 0.                                                      SLOP 520
      Z(1,11) = 0.                                                      SLOP 530
      Z(1,12) = 0.                                                      SLOP 540
      Z(1,13) = 0.                                                      SLOP 550
      Z(1,14) = 0.                                                      SLOP 560
      Z(1,15) = 0.                                                      SLOP 570
      Z(1,16) = 0.                                                      SLOP 580
      Z(2,1) = -24.                                                     SLOP 590
      Z(2,2) = -21.                                                     SLOP 600
      Z(2,3) = -19.                                                     SLOP 610
      Z(2,4) = -16.                                                     SLOP 620
      Z(2,5) = -15.                                                     SLOP 630
      Z(2,6) = -14.                                                     SLOP 640
      Z(2,7) = -12.                                                     SLOP 650
      Z(2,8) = -10.                                                     SLOP 660
      Z(2,9) = -8.                                                      SLOP 670
      Z(2,10) = -6.                                                     SLOP 680
      Z(2,11) = -4.                                                     SLOP 690
      Z(2,12) = -2.                                                     SLOP 700
      Z(2,13) = 0.                                                      SLOP 710
      Z(2,14) = 0.                                                      SLOP 720
      Z(2,15) = 0.                                                      SLOP 730
      Z(2,16) = 0.                                                      SLOP 740
      Z(3,1) = -36.                                                     SLOP 750
      Z(3,2) = -29.                                                     SLOP 760
      Z(3,3) = -26.                                                     SLOP 770
      Z(3,4) = -22.                                                     SLOP 780
      Z(3,5) = -18.                                                     SLOP 790
      Z(3,6) = -16.                                                     SLOP 800
      Z(3,7) = -14.                                                     SLOP 810
      Z(3,8) = -12.                                                     SLOP 820
      Z(3,9) = -10.                                                     SLOP 830
      Z(3,10) = -8.                                                     SLOP 840
      Z(3,11) = -6.                                                     SLOP 850
      Z(3,12) = -4.                                                     SLOP 860
      Z(3,13) = -2.                                                     SLOP 870
      Z(3,14) = 0.                                                      SLOP 880
      Z(3,15) = 0.                                                      SLOP 890
      Z(3,16) = 0.                                                      SLOP 900
      Z(4,1) = -39.                                                     SLOP 910
```

```
      Z(4,2)  = -33.                                                    SLOP 920
      Z(4,3)  = -28.                                                    SLOP 930
      Z(4,4)  = -23.                                                    SLOP 940
      Z(4,5)  = -18.                                                    SLOP 950
      Z(4,6)  = -16.                                                    SLOP 960
      Z(4,7)  = -14.                                                    SLOP 970
      Z(4,8)  = -12.                                                    SLOP 980
      Z(4,9)  = -10.                                                    SLOP 990
      Z(4,10)  = -8.                                                    SLOP1000
      Z(4,11)  = -6.                                                    SLOP1010
      Z(4,12)  = -4.                                                    SLOP1020
      Z(4,13)  = -2.                                                    SLOP1030
      Z(4,14)  = 0.                                                     SLOP1040
      Z(4,15)  = 0.                                                     SLOP1050
      Z(4,16)  = 0.                                                     SLOP1060
      Z(5,1)  = -38.                                                    SLOP1070
      Z(5,2)  = -31.                                                    SLOP1080
      Z(5,3)  = -26.                                                    SLOP1090
      Z(5,4)  = -21.                                                    SLOP1100
      Z(5,5)  = -18.                                                    SLOP1110
      Z(5,6)  = -14.                                                    SLOP1120
      Z(5,7)  = -12.                                                    SLOP1130
      Z(5,8)  = -10.                                                    SLOP1140
      Z(5,9)  = -8.                                                     SLOP1150
      Z(5,10)  = -6.                                                    SLOP1160
      Z(5,11)  = -4.                                                    SLOP1170
      Z(5,12)  = -2.                                                    SLOP1180
      Z(5,13)  = 0.                                                     SLOP1190
      Z(5,14)  = 0.                                                     SLOP1200
      Z(5,15)  = 0.                                                     SLOP1210
      Z(5,16)  = 0.                                                     SLOP1220
      Z(6,1)  = -28.                                                    SLOP1230
      Z(6,2)  = -24.                                                    SLOP1240
      Z(6,3)  = -20.                                                    SLOP1250
      Z(6,4)  = -15.                                                    SLOP1260
      Z(6,5)  = -12.                                                    SLOP1270
      Z(6,6)  = -10.                                                    SLOP1280
      Z(6,7)  = -8.                                                     SLOP1290
      Z(6,8)  = -6.                                                     SLOP1300
      Z(6,9)  = -4.                                                     SLOP1310
      Z(6,10)  = -2.                                                    SLOP1320
      Z(6,11)  = 0.                                                     SLOP1330
      Z(6,12)  = 0.                                                     SLOP1340
      Z(6,13)  = 0.                                                     SLOP1350
      Z(6,14)  = 0.                                                     SLOP1360
      Z(6,15)  = 0.                                                     SLOP1370
      Z(6,16)  = 0.                                                     SLOP1380
      Z(7,1)  = -15.                                                    SLOP1390
```

```
      Z(7,2) = -11.                                                     SLOP1400
      Z(7,3) = -9.                                                      SLOP1410
      Z(7,4) = -6.                                                      SLOP1420
      Z(7,5) = -4.                                                      SLOP1430
      Z(7,6) = -2.                                                      SLOP1440
      Z(7,7) = 0.                                                       SLOP1450
      Z(7,8) = 0.                                                       SLOP1460
      Z(7,9) = 0.                                                       SLOP1470
      Z(7,10) = 0.                                                      SLOP1480
      Z(7,11) = 0.                                                      SLOP1490
      Z(7,12) = 0.                                                      SLOP1500
      Z(7,13) = 0.                                                      SLOP1510
      Z(7,14) = 0.                                                      SLOP1520
      Z(7,15) = 0.                                                      SLOP1530
      Z(7,16) = 0.                                                      SLOP1540
      Z(8,1) = 4.                                                       SLOP1550
      Z(8,2) = 6.                                                       SLOP1560
      Z(8,3) = 7.                                                       SLOP1570
      Z(8,4) = 8.                                                       SLOP1580
      Z(8,5) = 9.                                                       SLOP1590
      Z(8,6) = 10.                                                      SLOP1600
      Z(8,7) = 7.                                                       SLOP1610
      Z(8,8) = 6.                                                       SLOP1620
      Z(8,9) = 4.                                                       SLOP1630
      Z(8,10) = 2.                                                      SLOP1640
      Z(8,11) = 0.                                                      SLOP1650
      Z(8,12) = 0.                                                      SLOP1660
      Z(8,13) = 0.                                                      SLOP1670
      Z(8,14) = 0.                                                      SLOP1680
      Z(8,15) = 0.                                                      SLOP1690
      Z(8,16) = 0.                                                      SLOP1700
      Z(9,1) = 27.                                                      SLOP1710
      Z(9,2) = 25.                                                      SLOP1720
      Z(9,3) = 23.                                                      SLOP1730
      Z(9,4) = 22.                                                      SLOP1740
      Z(9,5) = 21.                                                      SLOP1750
      Z(9,6) = 20.                                                      SLOP1760
      Z(9,7) = 18.                                                      SLOP1770
      Z(9,8) = 16.                                                      SLOP1780
      Z(9,9) = 14.                                                      SLOP1790
      Z(9,10) = 12.                                                     SLOP1800
      Z(9,11) = 10.                                                     SLOP1810
      Z(9,12) = 8.                                                      SLOP1820
      Z(9,13) = 6.                                                      SLOP1830
      Z(9,14) = 4.                                                      SLOP1840
      Z(9,15) = 2.                                                      SLOP1850
      Z(9,16) = 0.                                                      SLOP1860
      Z(10,1) = 52.                                                     SLOP1870
```

```
      Z(10,2) = 47                                                      SLOP1880
      Z(10,3) = 43.                                                     SLOP1890
      Z(10,4) = 39.                                                     SLOP1900
      Z(10,5) = 36.                                                     SLOP1910
      Z(10,6) = 34.                                                     SLOP1920
      Z(10,7) = 32.                                                     SLOP1930
      Z(10,8) = 30.                                                     SLOP1940
      Z(10,9) = 28.                                                     SLOP1950
      Z(10,10) = 26.                                                    SLOP1960
      Z(10,11) = 24.                                                    SLOP1970
      Z(10,12) = 22.                                                    SLOP1980
      Z(10,13) = 20.                                                    SLOP1990
      Z(10,14) = 18.                                                    SLOP2000
      Z(10,15) = 16.                                                    SLOP2010
      Z(10,16) = 14.                                                    SLOP2020
      Z(11,1) = 72.                                                     SLOP2030
      Z(11,2) = 64.                                                     SLOP2040
      Z(11,3) = 58.                                                     SLOP2050
      Z(11,4) = 52.                                                     SLOP2060
      Z(11,5) = 46.                                                     SLOP2070
      Z(11,6) = 43.                                                     SLOP2080
      Z(11,7) = 40.                                                     SLOP2090
      Z(11,8) = 37.                                                     SLOP2100
      Z(11,9) = 34.                                                     SLOP2110
      Z(11,10) = 31.                                                    SLOP2120
      Z(11,11) = 28.                                                    SLOP2130
      Z(11,12) = 25.                                                    SLOP2140
      Z(11,13) = 22.                                                    SLOP2150
      Z(11,14) = 19.                                                    SLOP2160
      Z(11,15) = 17.                                                    SLOP2170
      Z(11,16) = 15.                                                    SLOP2180
      Z(12,1) = 86.                                                     SLOP2190
      Z(12,2) = 75.                                                     SLOP2200
      Z(12,3) = 67.                                                     SLOP2210
      Z(12,4) = 60.                                                     SLOP2220
      Z(12,5) = 53.                                                     SLOP2230
      Z(12,6) = 48.                                                     SLOP2240
      Z(12,7) = 43.                                                     SLOP2250
      Z(12,8) = 39.                                                     SLOP2260
      Z(12,9) = 35.                                                     SLOP2270
      Z(12,10) = 32.                                                    SLOP2280
      Z(12,11) = 29                                                     SLOP2290
      Z(12,12) = 26.                                                    SLOP2300
      Z(12,13) = 23.                                                    SLOP2310
      Z(12,14) = 20.                                                    SLOP2320
      Z(12,15) = 18.                                                    SLOP2330
      Z(12,16) = 16.                                                    SLOP2340
      Z(13,1) = 89.                                                     SLOP2350
```

```
      Z(13,2) = 78.                                                     SLOP2360
      Z(13,3) = 69.                                                     SLOP2370
      Z(13,4) = 60.                                                     SLOP2380
      Z(13,5) = 53.                                                     SLOP2390
      Z(13,6) = 48.                                                     SLOP2400
      Z(13,7) = 43.                                                     SLOP2410
      Z(13,8) = 38.                                                     SLOP2420
      Z(13,9) = 34.                                                     SLOP2430
      Z(13,10) = 31.                                                    SLOP2440
      Z(13,11) = 28.                                                    SLOP2450
      Z(13,12) = 25.                                                    SLOP2460
      Z(13,13) = 22.                                                    SLOP2470
      Z(13,14) = 19.                                                    SLOP2480
      Z(13,15) = 17.                                                    SLOP2490
      Z(13,16) = 15.                                                    SLOP2500
      Z(14,1) = 80.                                                     SLOP2510
      Z(14,2) = 70.                                                     SLOP2520
      Z(14,3) = 63.                                                     SLOP2530
      Z(14,4) = 55.                                                     SLOP2540
      Z(14,5) = 50.                                                     SLOP2550
      Z(14,6) = 45.                                                     SLOP2560
      Z(14,7) = 40.                                                     SLOP2570
      Z(14,8) = 36.                                                     SLOP2580
      Z(14,9) = 33.                                                     SLOP2590
      Z(14,10) = 30.                                                    SLOP2600
      Z(14,11) = 29.                                                    SLOP2610
      Z(14,12) = 26.                                                    SLOP2620
      Z(14,13) = 23.                                                    SLOP2630
      Z(14,14) = 20.                                                    SLOP2640
      Z(14,15) = 18.                                                    SLOP2650
      Z(14,16) = 16.                                                    SLOP2660
      Z(15,1) = 64.                                                     SLOP2670
      Z(15,2) = 56.                                                     SLOP2680
      Z(15,3) = 51.                                                     SLOP2690
      Z(15,4) = 46.                                                     SLOP2700
      Z(15,5) = 41.                                                     SLOP2710
      Z(15,6) = 38.                                                     SLOP2720
      Z(15,7) = 36.                                                     SLOP2730
      Z(15,8) = 34.                                                     SLOP2740
      Z(15,9) = 32.                                                     SLOP2750
      Z(15,10) = 30.                                                    SLOP2760
      Z(15,11) = 28.                                                    SLOP2770
      Z(15,12) = 26.                                                    SLOP2780
      Z(15,13) = 24.                                                    SLOP2790
      Z(15,14) = 22.                                                    SLOP2800
      Z(15,15) = 20.                                                    SLOP2810
      Z(15,16) = 18.                                                    SLOP2820
      Z(16,1) = 45.                                                     SLOP2830
```

```
      Z(16,2) = 41.                                                     SLOP2840
      Z(16,3) = 38.                                                     SLOP2850
      Z(16,4) = 35.                                                     SLOP2860
      Z(16,5) = 33.                                                     SLOP2870
      Z(16,6) = 30.                                                     SLOP2880
      Z(16,7) = 28.                                                     SLOP2890
      Z(16,8) = 26.                                                     SLOP2900
      Z(16,9) = 24.                                                     SLOP2910
      Z(16,10) = 22.                                                    SLOP2920
      Z(16,11) = 20.                                                    SLOP2930
      Z(16,12) = 18.                                                    SLOP2940
      Z(16,13) = 16.                                                    SLOP2950
      Z(16,14) = 14.                                                    SLOP2960
      Z(16,15) = 12.                                                    SLOP2970
      Z(16,16) = 10.                                                    SLOP2980
      Z(17,1) = 30.                                                     SLOP2990
      Z(17,2) = 28.                                                     SLOP3000
      Z(17,3) = 26.                                                     SLOP3010
      Z(17,4) = 24.                                                     SLOP3020
      Z(17,5) = 23.                                                     SLOP3030
      Z(17,6) = 22.                                                     SLOP3040
      Z(17,7) = 20.                                                     SLOP3050
      Z(17,8) = 18.                                                     SLOP3060
      Z(17,9) = 16.                                                     SLOP3070
      Z(17,10) = 14.                                                    SLOP3080
      Z(17,11) = 12.                                                    SLOP3090
      Z(17,12) = 10.                                                    SLOP3100
      Z(17,13) = 8.                                                     SLOP3110
      Z(17,14) = 6.                                                     SLOP3120
      Z(17,15) = 4.                                                     SLOP3130
      Z(17,16) = 2.                                                     SLOP3140
      Z(18,1) = 17.                                                     SLOP3150
      Z(18,2) = 17.                                                     SLOP3160
      Z(18,3) = 16.                                                     SLOP3170
      Z(18,4) = 16.                                                     SLOP3180
      Z(18,5) = 16.                                                     SLOP3190
      Z(18,6) = 16.                                                     SLOP3200
      Z(18,7) = 15.                                                     SLOP3210
      Z(18,8) = 15.                                                     SLOP3220
      Z(18,9) = 15.                                                     SLOP3230
      Z(18,10) = 14.                                                    SLOP3240
      Z(18,11) = 14.                                                    SLOP3250
      Z(18,12) = 14.                                                    SLOP3260
      Z(18,13) = 12.                                                    SLOP3270
      Z(18,14) = 12.                                                    SLOP3280
      Z(18,15) = 12.                                                    SLOP3290
      Z(18,16) = 10.                                                    SLOP3300
      Z(19,1) = 6.                                                      SLOP3310
```

```
      Z(19,2) = 7.                                                      SLOP3320
      Z(19,3) = 7.                                                      SLOP3330
      Z(19,4) = 8.                                                      SLOP3340
      Z(19,5) = 8.                                                      SLOP3350
      Z(19,6) = 9.                                                      SLOP3360
      Z(19,7) = 9.                                                      SLOP3370
      Z(19,8) = 8.                                                      SLOP3380
      Z(19,9) = 8.                                                      SLOP3390
      Z(19,10) = 7.                                                     SLOP3400
      Z(19,11) = 6.                                                     SLOP3410
      Z(19,12) = 5.                                                     SLOP3420
      Z(19,13) = 4.                                                     SLOP3430
      Z(19,14) = 3.                                                     SLOP3440
      Z(19,15) = 2.                                                     SLOP3450
      Z(19,16) = 0.                                                     SLOP3460
      Z(20,1) = -5.                                                     SLOP3470
      Z(20,2) = -4.                                                     SLOP3480
      Z(20,3) = -3.                                                     SLOP3490
      Z(20,4) = -1.                                                     SLOP3500
      Z(20,5) = 0.                                                      SLOP3510
      Z(20,6) = 0.                                                      SLOP3520
      Z(20,7) = 0.                                                      SLOP3530
      Z(20,8) = 0.                                                      SLOP3540
      Z(20,9) = 0.                                                      SLOP3550
      Z(20,10) = 0.                                                     SLOP3560
      Z(20,11) = 0.                                                     SLOP3570
      Z(20,12) = 0.                                                     SLOP3580
      Z(20,13) = 0.                                                     SLOP3590
      Z(20,14) = 0.                                                     SLOP3600
      Z(20,15) = 0.                                                     SLOP3610
      Z(20,16) = 0.                                                     SLOP3620
      Z(21,1) = -5.                                                     SLOP3630
      Z(21,2) = -4.                                                     SLOP3640
      Z(21,3) = -3.                                                     SLOP3650
      Z(21,4) = -1.                                                     SLOP3660
      Z(21,5) = 0.                                                      SLOP3670
      Z(21,6) = 0.                                                      SLOP3680
      Z(21,7) = 0.                                                      SLOP3690
      Z(21,8) = 0.                                                      SLOP3700
      Z(21,9) = 0.                                                      SLOP3710
      Z(21,10) = 0.                                                     SLOP3720
      Z(21,11) = 0.                                                     SLOP3730
      Z(21,12) = 0.                                                     SLOP3740
      Z(21,13) = 0.                                                     SLOP3750
      Z(21,14) = 0.                                                     SLOP3760
      Z(21,15) = 0.                                                     SLOP3770
      Z(21,16) = 0.                                                     SLOP3780
      Z(22,1) = -6.                                                     SLOP3790
```

```
      Z(22,2) = -5.                                                     SLOP3800
      Z(22,3) = -5.                                                     SLOP3810
      Z(22,4) = -4.                                                     SLOP3820
      Z(22,5) = -4.                                                     SLOP3830
      Z(22,6) = -4.                                                     SLOP3840
      Z(22,7) = -3.                                                     SLOP3850
      Z(22,8) = -3.                                                     SLOP3860
      Z(22,9) = -2.                                                     SLOP3870
      Z(22,10) = -2.                                                    SLOP3880
      Z(22,11) = -1.                                                    SLOP3890
      Z(22,12) = -1.                                                    SLOP3900
      Z(22,13) = -1.                                                    SLOP3910
      Z(22,14) = 0.                                                     SLOP3920
      Z(22,15) = 0.                                                     SLOP3930
      Z(22,16) = 0.                                                     SLOP3940
      Z(23,1) = 0.                                                      SLOP3950
      Z(23,2) = 0.                                                      SLOP3960
      Z(23,3) = 0.                                                      SLOP3970
      Z(23,4) = 0.                                                      SLOP3980
      Z(23,5) = 0.                                                      SLOP3990
      Z(23,6) = 0.                                                      SLOP4000
      Z(23,7) = 0.                                                      SLOP4010
      Z(23,8) = 0.                                                      SLOP4020
      Z(23,9) = 0.                                                      SLOP4030
      Z(23,10) = 0.                                                     SLOP4040
      Z(23,11) = 0.                                                     SLOP4050
      Z(23,12) = 0.                                                     SLOP4060
      Z(23,13) = 0.                                                     SLOP4070
      Z(23,14) = 0.                                                     SLOP4080
      Z(23,15) = 0.                                                     SLOP4090
      Z(23,16) = 0.                                                     SLOP4100
      W(1,1) = 0.                                                       SLOP4110
      W(1,2) = 0.                                                       SLOP4120
      W(1,3) = 0.                                                       SLOP4130
      W(1,4) = 0.                                                       SLOP4140
      W(1,5) = 0.                                                       SLOP4150
      W(1,6) = 0.                                                       SLOP4160
      W(1,7) = 0.                                                       SLOP4170
      W(1,8) = 0.                                                       SLOP4180
      W(1,9) = 0.                                                       SLOP4190
      W(1,10) = 0.                                                      SLOP4200
      W(1,11) = 0.                                                      SLOP4210
      W(1,12) = 0.                                                      SLOP4220
      W(1,13) = 0.                                                      SLOP4230
      W(1,14) = 0.                                                      SLOP4240
      W(1,15) = 0.                                                      SLOP4250
      W(1,16) = 0.                                                      SLOP4260
      W(2,1) = 11.                                                      SLOP4270
```

```
      W(2,2) = 10.                                          SLOP4280
      W(2,3) = 10.                                          SLOP4290
      W(2,4) = 9.                                           SLOP4300
      W(2,5) = 9.                                           SLOP4310
      W(2,6) = 8.                                           SLOP4320
      W(2,7) = 6.                                           SLOP4330
      W(2,8) = 4.                                           SLOP4340
      W(2,9) = 2.                                           SLOP4350
      W(2,10) = 0.                                          SLOP4360
      W(2,11) = 0.                                          SLOP4370
      W(2,12) = 0.                                          SLOP4380
      W(2,13) = 0.                                          SLOP4390
      W(2,14) = 0.                                          SLOP4400
      W(2,15) = 0.                                          SLOP4410
      W(2,16) = 0.                                          SLOP4420
      W(3,1) = 14.                                          SLOP4430
      W(3,2) = 14.                                          SLOP4440
      W(3,3) = 11.                                          SLOP4450
      W(3,4) = 10.                                          SLOP4460
      W(3,5) = 9.                                           SLOP4470
      W(3,6) = 8.                                           SLOP4480
      W(3,7) = 7.                                           SLOP4490
      W(3,8) = 6.                                           SLOP4500
      W(3,9) = 5.                                           SLOP4510
      W(3,10) = 4.                                          SLOP4520
      W(3,11) = 3.                                          SLOP4530
      W(3,12) = 2.                                          SLOP4540
      W(3,13) = 1.                                          SLOP4550
      W(3,14) = 0.                                          SLOP4560
      W(3,15) = 0.                                          SLOP4570
      W(3,16) = 0.                                          SLOP4580
      W(4,1) = 14.                                          SLOP4590
      W(4,2) = 12.                                          SLOP4600
      W(4,3) = 10.                                          SLOP4610
      W(4,4) = 8.                                           SLOP4620
      W(4,5) = 7.                                           SLOP4630
      W(4,6) = 6.                                           SLOP4640
      W(4,7) = 5.                                           SLOP4650
      W(4,8) = 4.                                           SLOP4660
      W(4,9) = 3.                                           SLOP4670
      W(4,10) = 2.                                          SLOP4680
      W(4,11) = 1.                                          SLOP4690
      W(4,12) = 0.                                          SLOP4700
      W(4,13) = 0.                                          SLOP4710
      W(4,14) = 0.                                          SLOP4720
      W(4,15) = 0.                                          SLOP4730
      W(4,16) = 0.                                          SLOP4740
      W(5,1) = 7.                                           SLOP4750
```

```
      W(5,2) = 6.                                                       SLOP4760
      W(5,3) = 5.                                                       SLOP4770
      W(5,4) = 4.                                                       SLOP4780
      W(5,5) = 3.                                                       SLOP4790
      W(5,6) = 2.                                                       SLOP4800
      W(5,7) = 1.                                                       SLOP4810
      W(5,8) = 0.                                                       SLOP4820
      W(5,9) = 0.                                                       SLOP4830
      W(5,10) = 0.                                                      SLOP4840
      W(5,11) = 0.                                                      SLOP4850
      W(5,12) = 0.                                                      SLOP4860
      W(5,13) = 0.                                                      SLOP4870
      W(5,14) = 0.                                                      SLOP4880
      W(5,15) = 0.                                                      SLOP4890
      W(5,16) = 0.                                                      SLOP4900
      W(6,1) = -5.                                                      SLOP4910
      W(6,2) = -5.                                                      SLOP4920
      W(6,3) = -4.                                                      SLOP4930
      W(6,4) = -4.                                                      SLOP4940
      W(6,5) = -4.                                                      SLOP4950
      W(6,6) = -4.                                                      SLOP4960
      W(6,7) = -3.                                                      SLOP4970
      W(6,8) = -3.                                                      SLOP4980
      W(6,9) = -3.                                                      SLOP4990
      W(6,10) = -2.                                                     SLOP5000
      W(6,11) = -2.                                                     SLOP5010
      W(6,12) = -1.                                                     SLOP5020
      W(6,13) = -1.                                                     SLOP5030
      W(6,14) = 0.                                                      SLOP5040
      W(6,15) = 0.                                                      SLOP5050
      W(6,16) = 0.                                                      SLOP5060
      W(7,1) = -18.                                                     SLOP5070
      W(7,2) = -16.                                                     SLOP5080
      W(7,3) = -14.                                                     SLOP5090
      W(7,4) = -12.                                                     SLOP5100
      W(7,5) = -10.                                                     SLOP5110
      W(7,6) = -9.                                                      SLOP5120
      W(7,7) = -8.                                                      SLOP5130
      W(7,8) = -7.                                                      SLOP5140
      W(7,9) = -6.                                                      SLOP5150
      W(7,10) = -5.                                                     SLOP5160
      W(7,11) = -4.                                                     SLOP5170
      W(7,12) = -3.                                                     SLOP5180
      W(7,13) = -2.                                                     SLOP5190
      W(7,14) = -1.                                                     SLOP5200
      W(7,15) = 0.                                                      SLOP5210
      W(7,16) = 0.                                                      SLOP5220
      W(8,1) = -30.                                                     SLOP5230
```

```
      W(8,2) = -26.                                                      SLOP5240
      W(8,3) = -22.                                                      SLOP5250
      W(8,4) = -19.                                                      SLOP5260
      W(8,5) = -17.                                                      SLOP5270
      W(8,6) = -15.                                                      SLOP5280
      W(8,7) = -13.                                                      SLOP5290
      W(8,8) = -11.                                                      SLOP5300
      W(8,9) = -9.                                                       SLOP5310
      W(8,10) = -7.                                                      SLOP5320
      W(8,11) = -5.                                                      SLOP5330
      W(8,12) = -3.                                                      SLOP5340
      W(8,13) = -1.                                                      SLOP5350
      W(8,14) = 0.                                                       SLOP5360
      W(8,15) = 0.                                                       SLOP5370
      W(8,16) = 0.                                                       SLOP5380
      W(9,1) = -41.                                                      SLOP5390
      W(9,2) = -35.                                                      SLOP5400
      W(9,3) = -30.                                                      SLOP5410
      W(9,4) = -25.                                                      SLOP5420
      W(9,5) = -22.                                                      SLOP5430
      W(9,6) = -18.                                                      SLOP5440
      W(9,7) = -16.                                                      SLOP5450
      W(9,8) = -14.                                                      SLOP5460
      W(9,9) = -12.                                                      SLOP5470
      W(9,10) = -10.                                                     SLOP5480
      W(9,11) = -8.                                                      SLOP5490
      W(9,12) = -6.                                                      SLOP5500
      W(9,13) = -4.                                                      SLOP5510
      W(9,14) = -2.                                                      SLOP5520
      W(9,15) = 0.                                                       SLOP5530
      W(9,16) = 0.                                                       SLOP5540
      W(10,1) = -44.                                                     SLOP5550
      W(10,2) = -38.                                                     SLOP5560
      W(10,3) = -33.                                                     SLOP5570
      W(10,4) = -28.                                                     SLOP5580
      W(10,5) = -25.                                                     SLOP5590
      W(10,6) = -20.                                                     SLOP5600
      W(10,7) = -17.                                                     SLOP5610
      W(10,8) = -14.                                                     SLOP5620
      W(10,9) = -11.                                                     SLOP5630
      W(10,10) = -9.                                                     SLOP5640
      W(10,11) = -7.                                                     SLOP5650
      W(10,12) = -5.                                                     SLOP5660
      W(10,13) = -3.                                                     SLOP5670
      W(10,14) = -2.                                                     SLOP5680
      W(10,15) = -1.                                                     SLOP5690
      W(10,16) = 0.                                                      SLOP5700
      W(11,1) = -43.                                                     SLOP5710
```

```
      W(11,2) = -37.                                                    SLOP5720
      W(11,3) = -33.                                                    SLOP5730
      W(11,4) = -28.                                                    SLOP5740
      W(11,5) = -25.                                                    SLOP5750
      W(11,6) = -20.                                                    SLOP5760
      W(11,7) = -17.                                                    SLOP5770
      W(11,8) = -14.                                                    SLOP5780
      W(11,9) = -11.                                                    SLOP5790
      W(11,10) = -9.                                                    SLOP5800
      W(11,11) = -7.                                                    SLOP5810
      W(11,12) = -5.                                                    SLOP5820
      W(11,13) = -3.                                                    SLOP5830
      W(11,14) = -2.                                                    SLOP5840
      W(11,15) = -1.                                                    SLOP5850
      W(11,16) = 0.                                                     SLOP5860
      W(12,1) = -37.                                                    SLOP5870
      W(12,2) = -33.                                                    SLOP5880
      W(12,3) = -30.                                                    SLOP5890
      W(12,4) = -26.                                                    SLOP5900
      W(12,5) = -23.                                                    SLOP5910
      W(12,6) = -20.                                                    SLOP5920
      W(12,7) = -17.                                                    SLOP5930
      W(12,8) = -14.                                                    SLOP5940
      W(12,9) = -11.                                                    SLOP5950
      W(12,10) = -9.                                                    SLOP5960
      W(12,11) = -7.                                                    SLOP5970
      W(12,12) = -5.                                                    SLOP5980
      W(12,13) = -3.                                                    SLOP5990
      W(12,14) = -2.                                                    SLOP6000
      W(12,15) = -1.                                                    SLOP6010
      W(12,16) = 0.                                                     SLOP6020
      W(13,1) = -29.                                                    SLOP6030
      W(13,2) = -26.                                                    SLOP6040
      W(13,3) = -25.                                                    SLOP6050
      W(13,4) = -22.                                                    SLOP6060
      W(13,5) = -19.                                                    SLOP6070
      W(13,6) = -17.                                                    SLOP6080
      W(13,7) = -15.                                                    SLOP6090
      W(13,8) = -13.                                                    SLOP6100
      W(13,9) = -11.                                                    SLOP6110
      W(13,10) = -9.                                                    SLOP6120
      W(13,11) = -7.                                                    SLOP6130
      W(13,12) = -5.                                                    SLOP6140
      W(13,13) = -3.                                                    SLOP6150
      W(13,14) = -2.                                                    SLOP6160
      W(13,15) = -1.                                                    SLOP6170
      W(13,16) = 0.                                                     SLOP6180
      W(14,1) = -18.                                                    SLOP6190
```

```
      W(14,2) = -16.                                                    SLOP6200
      W(14,3) = -15.                                                    SLOP6210
      W(14,4) = -13.                                                    SLOP6220
      W(14,5) = -12.                                                    SLOP6230
      W(14,6) = -10.                                                    SLOP6240
      W(14,7) = -9.                                                     SLOP6250
      W(14,8) = -8.                                                     SLOP6260
      W(14,9) = -7.                                                     SLOP6270
      W(14,10) = -6.                                                    SLOP6280
      W(14,11) = -5.                                                    SLOP6290
      W(14,12) = -4.                                                    SLOP6300
      W(14,13) = -3.                                                    SLOP6310
      W(14,14) = -2.                                                    SLOP6320
      W(14,15) = -1.                                                    SLOP6330
      W(14,16) = 0.                                                     SLOP6340
      W(15,1) = -5.                                                     SLOP6350
      W(15,2) = -4.                                                     SLOP6360
      W(15,3) = -4.                                                     SLOP6370
      W(15,4) = -4.                                                     SLOP6380
      W(15,5) = -4.                                                     SLOP6390
      W(15,6) = -4.                                                     SLOP6400
      W(15,7) = -3.                                                     SLOP6410
      W(15,8) = -3.                                                     SLOP6420
      W(15,9) = -3.                                                     SLOP6430
      W(15,10) = -3.                                                    SLOP6440
      W(15,11) = -2.                                                    SLOP6450
      W(15,12) = -2.                                                    SLOP6460
      W(15,13) = -1.                                                    SLOP6470
      W(15,14) = -1.                                                    SLOP6480
      W(15,15) = 0.                                                     SLOP6490
      W(15,16) = 0.                                                     SLOP6500
      W(16,1) = -4.                                                     SLOP6510
      W(16,2) = -4.                                                     SLOP6520
      W(16,3) = -3.                                                     SLOP6530
      W(16,4) = -3.                                                     SLOP6540
      W(16,5) = -2.                                                     SLOP6550
      W(16,6) = -2.                                                     SLOP6560
      W(16,7) = -2.                                                     SLOP6570
      W(16,8) = -1.                                                     SLOP6580
      W(16,9) = -1.                                                     SLOP6590
      W(16,10) = -1.                                                    SLOP6600
      W(16,11) = 0.                                                     SLOP6610
      W(16,12) = 0.                                                     SLOP6620
      W(16,13) = 0.                                                     SLOP6630
      W(16,14) = 0.                                                     SLOP6640
      W(16,15) = 0.                                                     SLOP6650
      W(16,16) = 0.                                                     SLOP6660
      W(17,1) = 11.                                                     SLOP6670
```

```
      W(17,2) = 10.                                                     SLOP6680
      W(17,3) = 9.                                                      SLOP6690
      W(17,4) = 8.                                                      SLOP6700
      W(17,5) = 7.                                                      SLOP6710
      W(17,6) = 6.                                                      SLOP6720
      W(17,7) = 5.                                                      SLOP6730
      W(17,8) = 4.                                                      SLOP6740
      W(17,9) = 3.                                                      SLOP6750
      W(17,10) = 2.                                                     SLOP6760
      W(17,11) = 1.                                                     SLOP6770
      W(17,12) = 0.                                                     SLOP6780
      W(17,13) = 0.                                                     SLOP6790
      W(17,14) = 0.                                                     SLOP6800
      W(17,15) = 0.                                                     SLOP6810
      W(17,16) = 0.                                                     SLOP6820
      W(18,1) = 14.                                                     SLOP6830
      W(18,2) = 12.                                                     SLOP6840
      W(18,3) = 11.                                                     SLOP6850
      W(18,4) = 10.                                                     SLOP6860
      W(18,5) = 9.                                                      SLOP6870
      W(18,6) = 8.                                                      SLOP6880
      W(18,7) = 6.                                                      SLOP6890
      W(18,8) = 4.                                                      SLOP6900
      W(18,9) = 2.                                                      SLOP6910
      W(18,10) = 0.                                                     SLOP6920
      W(18,11) = 0.                                                     SLOP6930
      W(18,12) = 0.                                                     SLOP6940
      W(18,13) = 0.                                                     SLOP6950
      W(18,14) = 0.                                                     SLOP6960
      W(18,15) = 0.                                                     SLOP6970
      W(18,16) = 0.                                                     SLOP6980
      W(19,1) = 13.                                                     SLOP6990
      W(19,2) = 12.                                                     SLOP7000
      W(19,3) = 11.                                                     SLOP7010
      W(19,4) = 10.                                                     SLOP7020
      W(19,5) = 9.                                                      SLOP7030
      W(19,6) = 8.                                                      SLOP7040
      W(19,7) = 6.                                                      SLOP7050
      W(19,8) = 4.                                                      SLOP7060
      W(19,9) = 2.                                                      SLOP7070
      W(19,10) = 0.                                                     SLOP7080
      W(19,11) = 0.                                                     SLOP7090
      W(19,12) = 0.                                                     SLOP7100
      W(19,13) = 0.                                                     SLOP7110
      W(19,14) = 0.                                                     SLOP7120
      W(19,15) = 0.                                                     SLOP7130
      W(19,16) = 0.                                                     SLOP7140
      W(20,1) = 12.                                                     SLOP7150
```

```
      W(20,2) = 10.                                                     SLOP7160
      W(20,3) = 9.                                                      SLOP7170
      W(20,4) = 8.                                                      SLOP7180
      W(20,5) = 7.                                                      SLOP7190
      W(20,6) = 7.                                                      SLOP7200
      W(20,7) = 6.                                                      SLOP7210
      W(20,8) = 5.                                                      SLOP7220
      W(20,9) = 4.                                                      SLOP7230
      W(20,10) = 3.                                                     SLOP7240
      W(20,11) = 2.                                                     SLOP7250
      W(20,12) = 1.                                                     SLOP7260
      W(20,13) = 0.                                                     SLOP7270
      W(20,14) = 0.                                                     SLOP7280
      W(20,15) = 0.                                                     SLOP7290
      W(20,16) = 0.                                                     SLOP7300
      W(21,1) = 10.                                                     SLOP7310
      W(21,2) = 9.                                                      SLOP7320
      W(21,3) = 7.                                                      SLOP7330
      W(21,4) = 6.                                                      SLOP7340
      W(21,5) = 5.                                                      SLOP7350
      W(21,6) = 4.                                                      SLOP7360
      W(21,7) = 3.                                                      SLOP7370
      W(21,8) = 2.                                                      SLOP7380
      W(21,9) = 1.                                                      SLOP7390
      W(21,10) = 0.                                                     SLOP7400
      W(21,11) = 0.                                                     SLOP7410
      W(21,12) = 0.                                                     SLOP7420
      W(21,13) = 0.                                                     SLOP7430
      W(21,14) = 0.                                                     SLOP7440
      W(21,15) = 0.                                                     SLOP7450
      W(21,16) = 0.                                                     SLOP7460
      W(22,1) = 5.                                                      SLOP7470
      W(22,2) = 4.                                                      SLOP7480
      W(22,3) = 3.                                                      SLOP7490
      W(22,4) = 3.                                                      SLOP7500
      W(22,5) = 2.                                                      SLOP7510
      W(22,6) = 2.                                                      SLOP7520
      W(22,7) = 1.                                                      SLOP7530
      W(22,8) = 1.                                                      SLOP7540
      W(22,9) = 0.                                                      SLOP7550
      W(22,10) = 0.                                                     SLOP7560
      W(22,11) = 0.                                                     SLOP7570
      W(22,12) = 0.                                                     SLOP7580
      W(22,13) = 0.                                                     SLOP7590
      W(22,14) = 0.                                                     SLOP7600
      W(22,15) = 0.                                                     SLOP7610
      W(22,16) = 0.                                                     SLOP7620
      W(23,1) = 0.                                                      SLOP7630
```

```
      W(23,2) = 0.                                                      SLOP7640
      W(23,3) = 0.                                                      SLOP7650
      W(23,4) = 0.                                                      SLOP7660
      W(23,5) = 0.                                                      SLOP7670
      W(23,6) = 0.                                                      SLOP7680
      W(23,7) = 0.                                                      SLOP7690
      W(23,8) = 0.                                                      SLOP7700
      W(23,9) = 0.                                                      SLOP7710
      W(23,10) = 0.                                                     SLOP7720
      W(23,11) = 0.                                                     SLOP7730
      W(23,12) = 0.                                                     SLOP7740
      W(23,13) = 0.                                                     SLOP7750
      W(23,14) = 0.                                                     SLOP7760
      W(23,15) = 0.                                                     SLOP7770
      W(23,16) = 0.                                                     SLOP7780
      IF (ABS(DY).GE..4734) GO TO 1                                     SLOP7790
      IF (ABS(DX).GT.1.15) GO TO 1                                      SLOP7800
      I = IFIX((DX+1.1002)/.1)+1.                                       SLOP7810
      J = IFIX((ABS(DY)-.1)/.02)+1.                                     SLOP7820
      IF (J.LT.1) J = 1                                                 SLOP7830
      IF (I.LT.1) I = 1                                                 SLOP7840
      IF (I.GT.23) I = 23                                               SLOP7850
      IF (J.GT.16) J = 16                                               SLOP7860
      K = 24-I                                                          SLOP7870
      DELX = DX-X(I)                                                    SLOP7880
      DELY = ABS(DY)-Y(J)                                               SLOP7890
      DELK = DX-X(K)                                                    SLOP7900
      IF ((I.EQ.23).OR.(J.EQ.16)) GO TO 2                               SLOP7910
      DYD1 = DELX*(Z(I+1,J)-Z(I,J))+DELY*(Z(I,J+1)-Z(I,J))              SLOP7920
      DYD2 = DELK*(Z(K+1,J)-Z(K,J))+DELY*(Z(K,J+1)-Z(K,J))              SLOP7930
      DYND1 = DELX*(W(I+1,J)-W(I,J))+DELY*(W(I,J+1)-W(I,J))             SLOP7940
      DYND2 = DELK*(W(K+1,J)-W(K,J))+DELY*(W(K,J+1)-W(K,J))             SLOP7950
      YY1 = (Z(I,J)+DYD1)*1.E-05                                        SLOP7960
      YY2 = -(Z(K,J)+DYD2)*1.E-05                                       SLOP7970
      YN1 = (W(I,J)+DYND1)*1.E-05                                       SLOP7980
      YN2 = -(W(K,J)+DYND2)*1.E-05                                      SLOP7990
      IF (DY.LT.0.) GO TO 3                                             SLOP8000
      RETURN                                                            SLOP8010
    1 YY1 = 0.                                                          SLOP8020
      YY2 = 0.                                                          SLOP8030
      YN1 = 0.                                                          SLOP8040
      YN2 = 0.                                                          SLOP8050
      RETURN                                                            SLOP8060
    2 YY1 = Z(I,J)*1.E-05                                               SLOP8070
      YY2 = -Z(K,J)*1.E-05                                              SLOP8080
      YN1 = W(I,J)*1.E-05                                               SLOP8090
      YN2 = -W(K,J)*1.E-05                                              SLOP8100
      IF (DY.LT.0.) GO TO 3                                             SLOP8110
```

```
      RETURN                                                          SLOP8120
    3 YY1 =  -YY1                                                     SLOP8130
      YY2 =  -YY2                                                     SLOP8140
      YN1 =  -YN1                                                     SLOP8150
      YN2 =  -YN2                                                     SLOP8160
      RETURN                                                          SLOP8170
      END                                                             SLOP8180
```

FUNCTION SPINIT

This function was designed to aid in initialization problems associated with the DSL function DELY. The effect is that the function initializes the delay loop until it can be self-supportive.

FUNCTION SPINIT

```
      FUNCTION SPINIT (SPDDEL,TIME,SPDO)
      IF (TIME.GT.0.5) GO TO 1
      SPINIT = SPDO
      RETURN
    1 SPINIT = SPDDEL
      RETURN
      END
```

FUNCTION SPDCTR

This function is the speed control used during heading control development. It is used directly as the speed of the control ship with information of the speed of the two ships and the longitudinal position ADX. It contains a linear function at ±1.0 ship lengths to a dead zone of ±0.001 centered about the alongside position (0.0).

FUNCTION SPDCTR

```
      FUNCTION SPDCTR (ADX,U01,U02)
      IF (ADX.LT.-1.0) GO TO 1
      IF (ADX.GT.1.0) GO TO 2
      IF (ABS(ADX).LT.0.001) GO TO 3
      SPDCTR = -ADX*(U02-U01)+U01
      RETURN
    1 SPDCTR = U02
      RETURN
    2 SPDCTR = U02-U01
      RETURN
    3 SPDCTR = U01
      RETURN
      END
```

FUNCTION SPDOFC

This function is identical to SPDREC except that the ability to offset the alongside position (0.C) is incorporated. This is the speed control function in its final development form.

FUNCTION SPDOFC

```
      FUNCTION SPDOFC (ADX,SPDO1,SPDO2,SW,XOFS)
      SWTCH = -SW*(SPDO2-SPDO1)
      IF ((ADX-XOFS).LT.SWTCH) GO TO 1
      IF ((ADX-XOFS).GT.-SWTCH) GO TO 2
      IF (ABS(ADX-XOFS).LT.0.001) GO TO 3
      SPDOFC = -(ADX-XOFS)*(SPDO2-SPDO1)+SPDO1
      RETURN
    1 SPDOFC = SPDO2
      RETURN
    2 SPDOFC = SPDO2-SPDO1
      RETURN
    3 SPDOFC = SPDO1
      RETURN
      END
```

FUNCTION SPDREC

This function is similar to SPDCTR except that a switching function is incorporated. This is the function used for optimization of the switching function and is used in the velocity loop simulated in the velocity control section of chapter III.

FUNCTION SPDREC

```
      FUNCTION SPDREC (ADX,SPD01,SPD02,SW)
      SWTCH = -SW*(SPD02-SPD01)
      IF (ADX.LT.SWTCH) GO TO 1
      IF (ADX.GT.-SWTCH) GO TO 2
      IF (ABS(ADX).LT.0.001) GO TO 3
      SPDREC = -ADX*(SPD02-SPD01)+SPD01
      RETURN
    1 SPDREC = SPD02
      RETURN
    2 SPDREC = SPD02-SPD01
      RETURN
    3 SPDREC = SPD01
      RETURN
      END
```

FUNCTION SWCL

This function contains the fifth order polynomial curve fit for the optimal switching position of the speed control loop. Its range of values for SPDDIF are 0.1 to 1.0 normalized speed difference between the two ships.

FUNCTION SWCL

```
      FUNCTION SWCL (SPD01,SPD02)
      SPDDIF = SPD02-SPD01
      SWCL = -2.24869*SPDDIF**5+6.93243*SPDDIF**4-8.04233*SPDDIF**3+4.08
     1065*SPDDIF**2-0.409977*SPDDIF+0.554
      RETURN
      END
```

SUBROUTINE SWTCH

This subroutine contains the gains and mechanisms required for the adaptive gain schedule developed in this thesis. It includes the optimal gains obtained from the heading control optimization runs.

SUBROUTINE SWTCH

```
      SUBROUTINE SWTCH (DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D)
C     SUBROUTINE TO SWITCH RAS GAINS ONCE SHIPS ALONGSIDE
      DDC = DD
      IF (IS.EQ.0) DDC = -DD
      AMDY = DDC+DA*SIN(AA1)
      AMDX = DA*COS(AA1)
      IF (ABS(AMDX).GT.1.0) N=1
      IF (ABS(AMDX).LT.0.5.AND.ABS(AMDY).LT.0.005) GO TO 1
      IF (ABS(AMDY).LT.0.05) GO TO 2
      N = 1
      RSENS = 1.86642
      WTSENS = 2.38692
      RGN = 23.41847
      VFBG = 4.35162
      RETURN
    1 RSENS = 1.99765
      WTSENS = 0.7357
      RGN = 49.97757
      VFBG = 0.084028
      N = N+1
      IF (ABS(BDOT2D).GT.2.0.AND.N.LT.150) VFBG=1.0
      RETURN
    2 RSENS = 4.0
      WTSENS = 2.38692
      RGN = 23.41847
      VFBG = 4.35162
      IF (N.GT.150) GO TO 1
      N = 1
      RETURN
      END
```

SUBROUTINE SWTCHF

This subroutine is identical to SWTCH except that the turn phase gain VFBG is relaxed to allow for offset longitudinal position placement. This is the adaptive gain schedule in its final form.

SUBROUTINE SWTCHF

```
      SUBROUTINE SWTCHF (DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D,XOFS)
C     SUBROUTINE TO SWITCH RAS GAINS ONCE SHIPS ALONGSIDE
      DDC = DD
      IF (IS.EQ.0) DDC = -DD
      AMDY = DDC+DA*SIN(AA1)
      AMDX = DA*COS(AA1)
      IF (ABS(AMDX-XOFS).GT.1.0) N=1
      IF (ABS(AMDX-XOFS).LT.0.5.AND.ABS(AMDY).LT.0.005) GO TO 1
      IF (ABS(AMDY).LT.0.05) GO TO 2
      N = 1
      RSENS = 1.86642
      WTSENS = 2.38692
      RGN = 23.41847
      VFBG = 4.35162
      RETURN
    1 RSENS = 1.99765
      WTSENS = 0.7357
      RGN = 49.97757
      VFBG = 0.1
      IF (ABS(BDOT2D).GT.2.0.AND.N.LE.150) VFBG=1.0
      N = N+1
      RETURN
    2 RSENS = 4.0
      WTSENS = 2.38692
      RGN = 23.41847
      VFBG = 4.35162
      IF (N.GT.150) GO TO 1
      N = 1
      RETURN
      END
```

SUBROUTINE TRANS

This subroutine takes the lateral and longitudinal geographic displacements and converts them to actual displacements referenced to the control ship's head. This is done to gain a more realistic reference for subroutine BEMEAS, subroutine SLOPES, function SPDCTR, function SPDCPC, and function SPDREC.

SUBROUTINE TRANS

```
SUBROUTINE TRANS (PSIA,DX,DY,ADX,ADY)
DXY = SQRT(DX**2+DY**2)
AXY = ARSIN(-DY/DXY)
IF (DX.LT.O.) AXY = 3.141592654-AXY
AT = AXY+PSIA
ADY = -DXY*SIN(AT)
ADX = +DXY*COS(AT)
RETURN
END
```

FUNCTION XLIMIT

This function was developed to allow the LIMIT function of DSL to be incorporated in the optimization runs. It is a saturation amplifier with a gain of 1.0, and upper limit of UL, and a lower limit of XLL.

FUNCTION XLIMIT

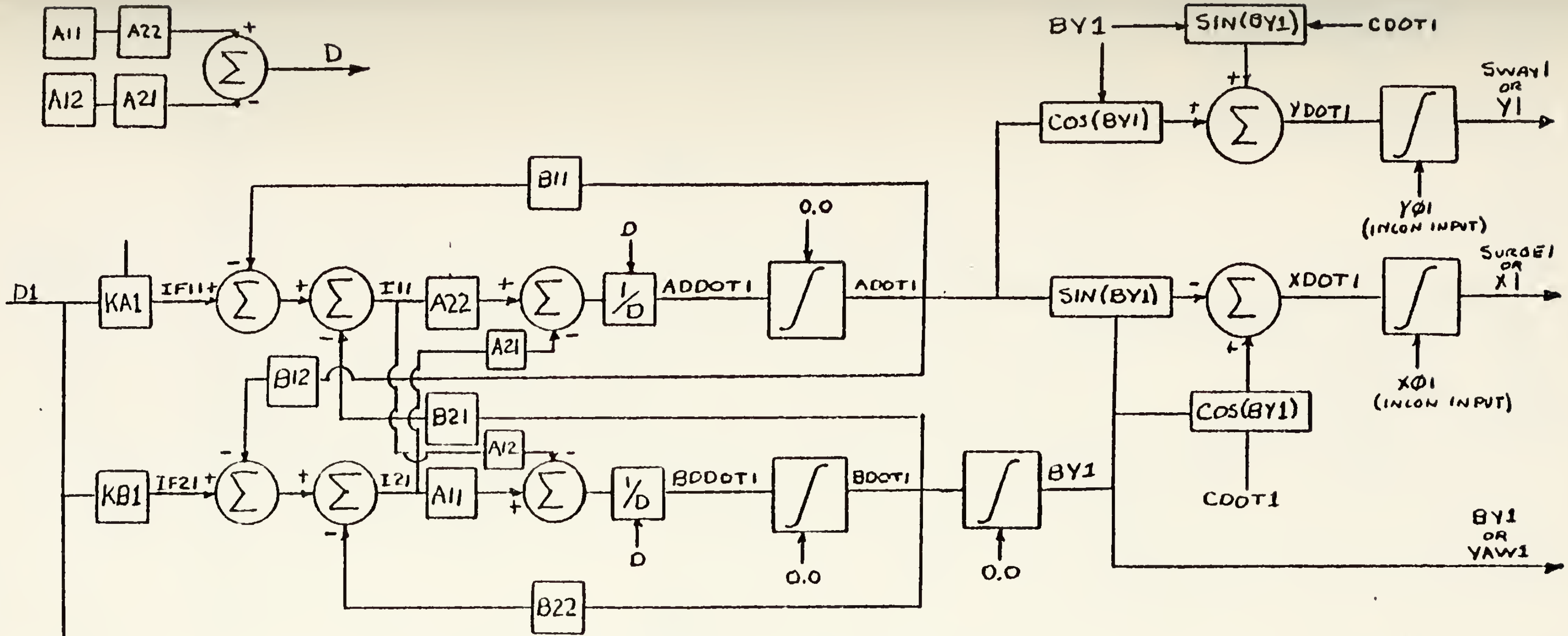
```
      FUNCTION XLIMIT (XLL,UL,FUNCT)
      FUNCU = FUNCT
      IF (FUNCT.LT.XLL) FUNCU=XLL
      IF (FUNCT.GT.UL) FUNCU=UL
      XLIMIT = FUNCU
      RETURN
      END
```

APPENDIX B

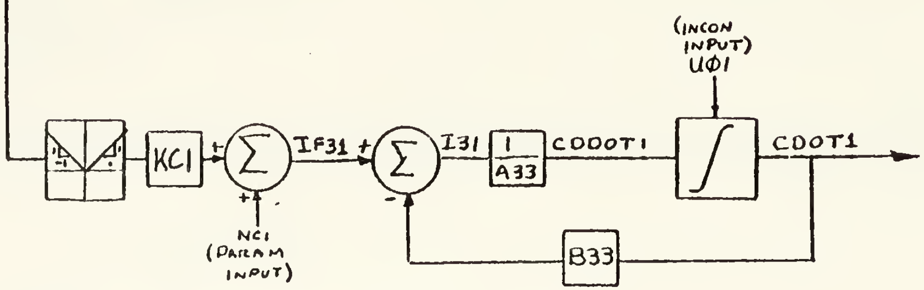
The final form of the simulation program, with all its subroutines and functions, is a very complex and complicated maze. To aid in following its progression, this appendix contains a detailed block diagram of the program with each variable listed in its computer variable name. Each page contains a functional part of the simulation with inputs and outputs shown cross referenced to their origin and destination.

The following is a list of the block diagrams contained in this appendix in the order in which they appear:

Wave Direction
Random Variable
Wave Encounter
Wave Components



Ship A Heading Simulation

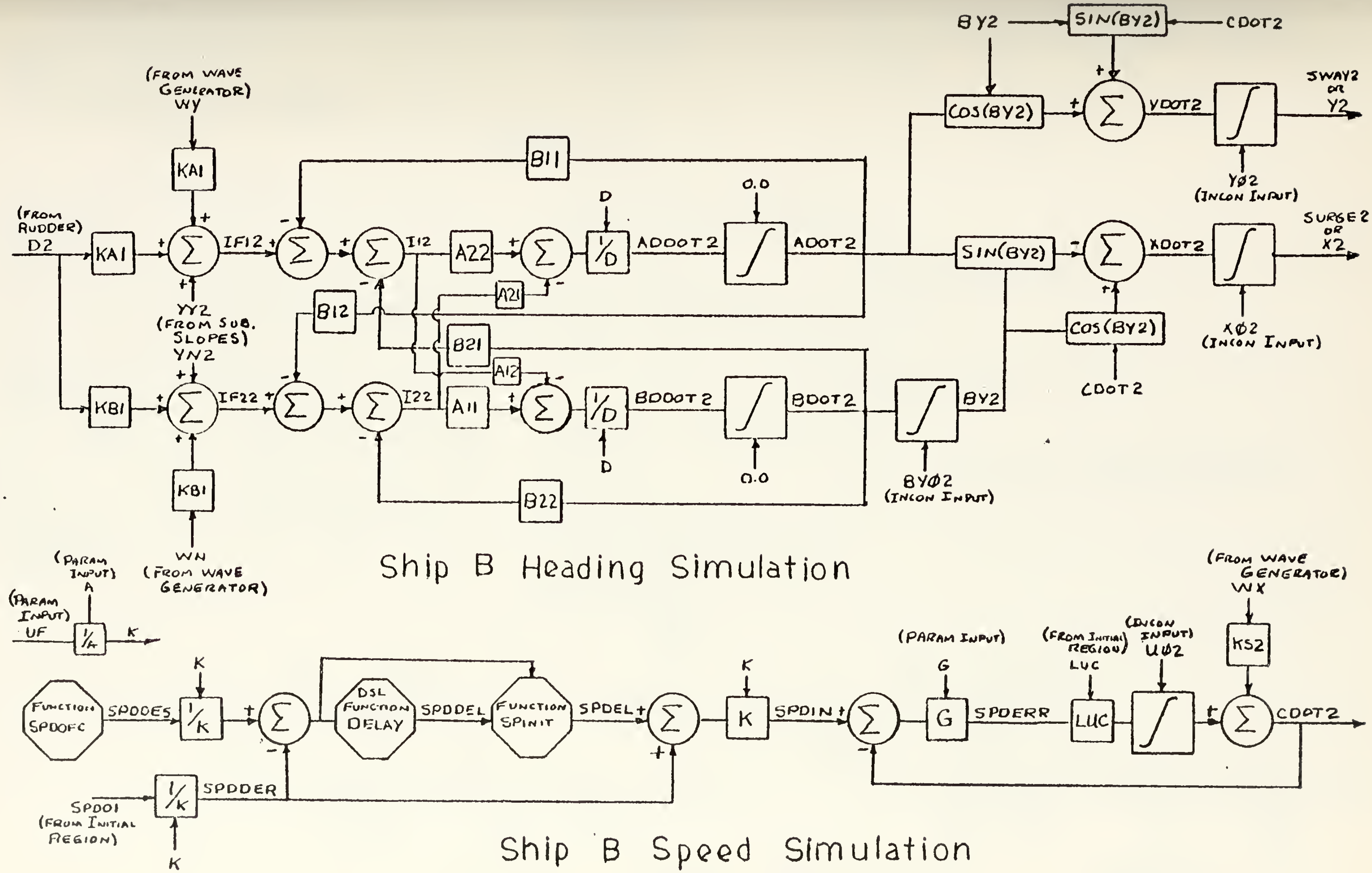


Ship A Speed Simulation

GAIN CROSS REFERENCE

A11 = MYVD
B11 = -YV
A21 = -YRD
B21 = MYR
A12 = -NVD
B12 = -NV
A22 = IZNRD
B22 = -NR
A33 = MXUD
B33 = -XU
KA1 = -YDELR
KB1 = NDELR
KC1 = XDELR
NC1 = -XU

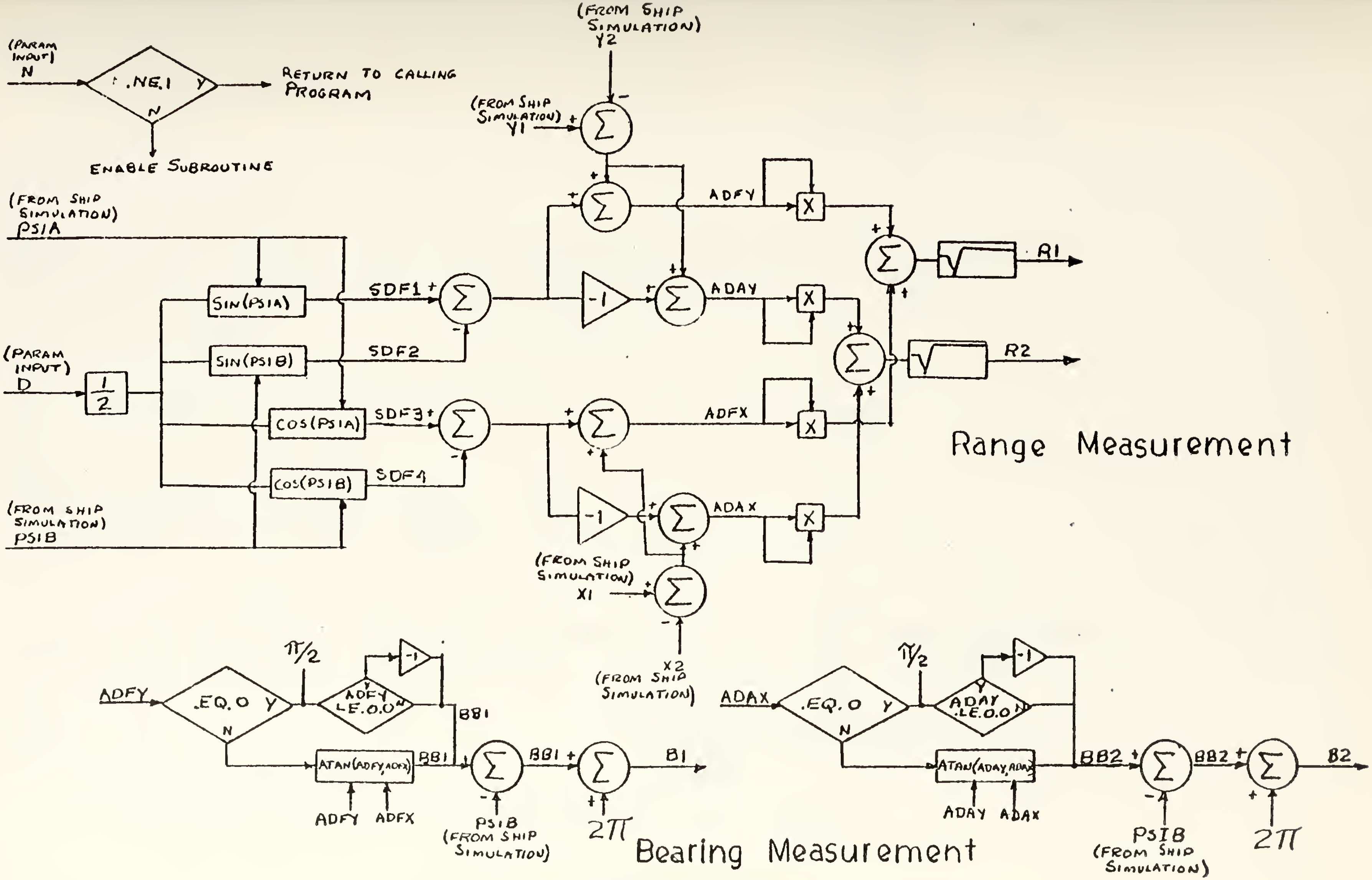
Ship A (Reference Ship) Simulation



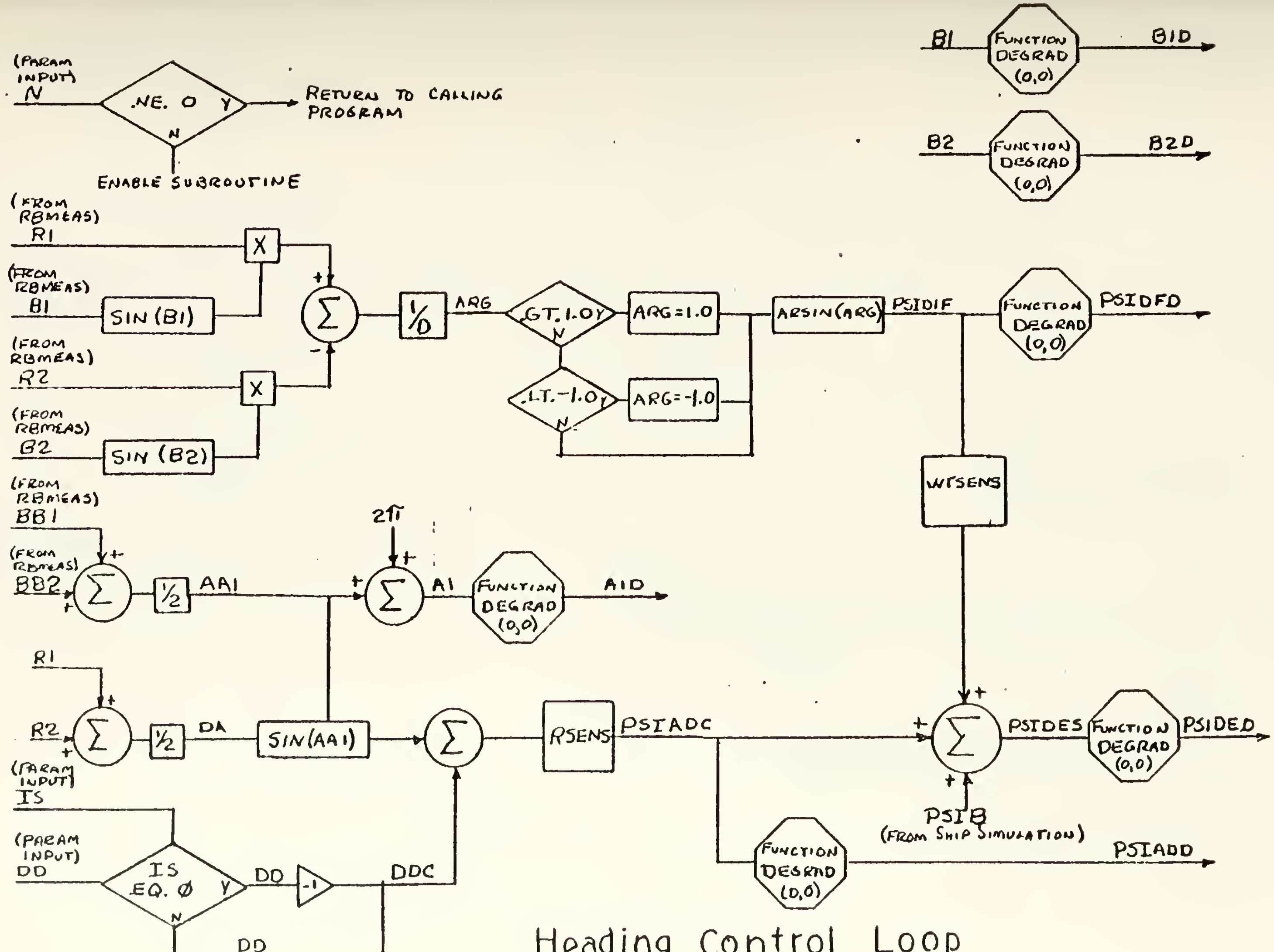
Ship B Heading Simulation

Ship B Speed Simulation

Ship B (Control Ship) Simulation

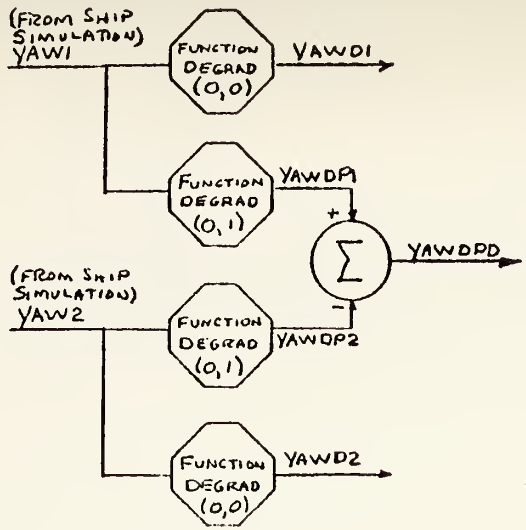


Subroutine RBMEAS

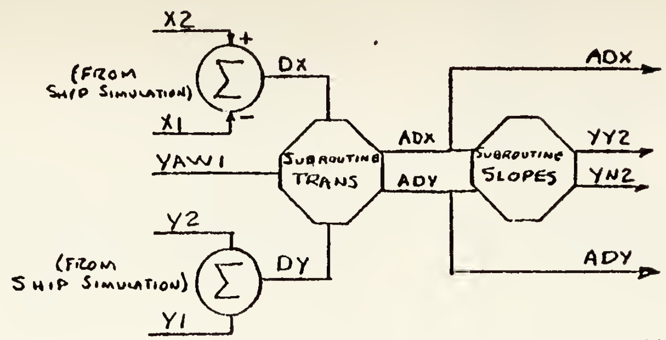


Heading Control Loop

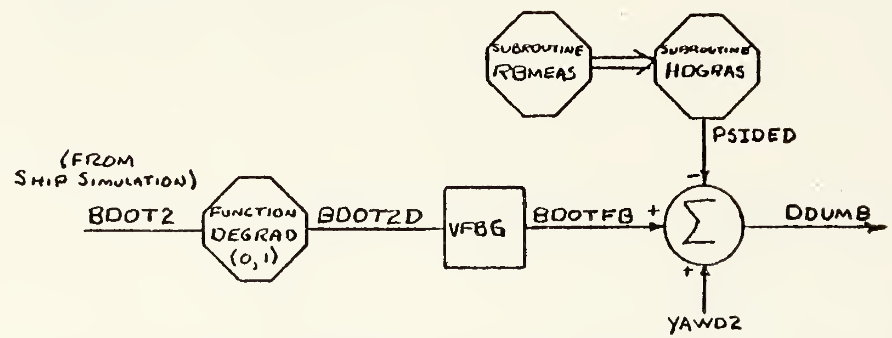
Subroutine HDGRAS



Yaw Conversions

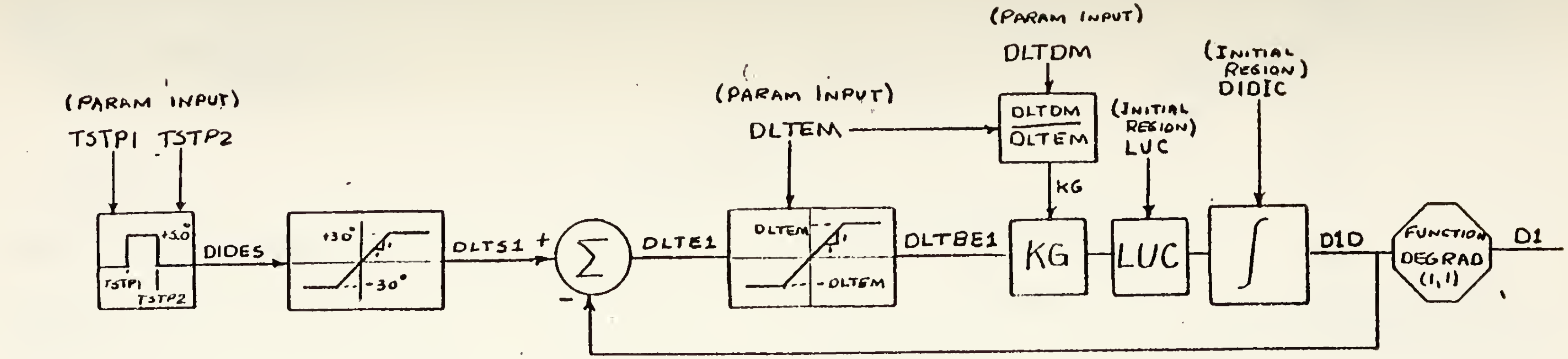


Coordinate Conversion

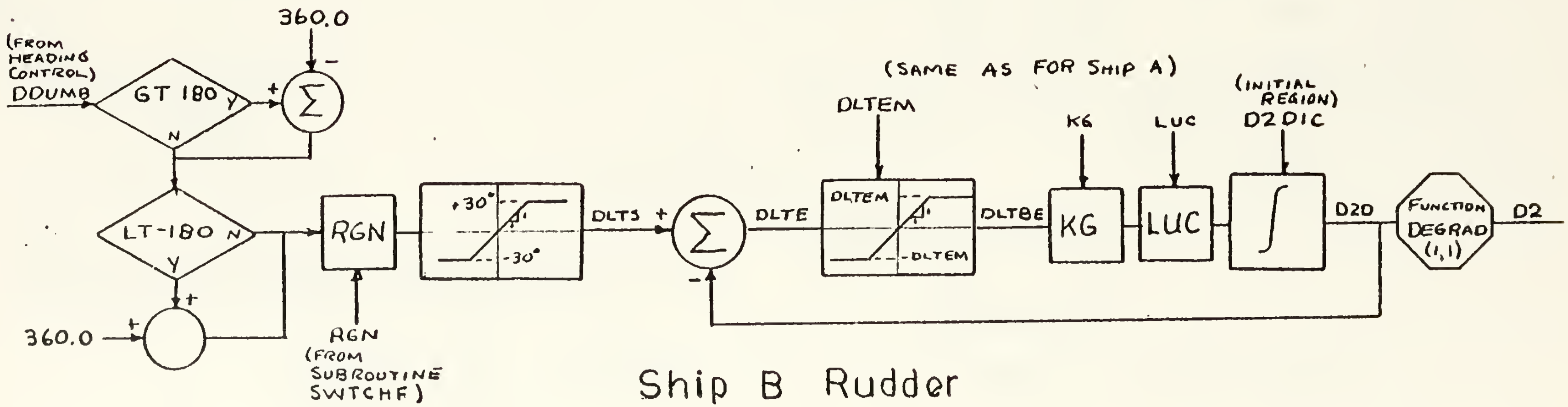


Feedback Loop

Auxiliary Functions

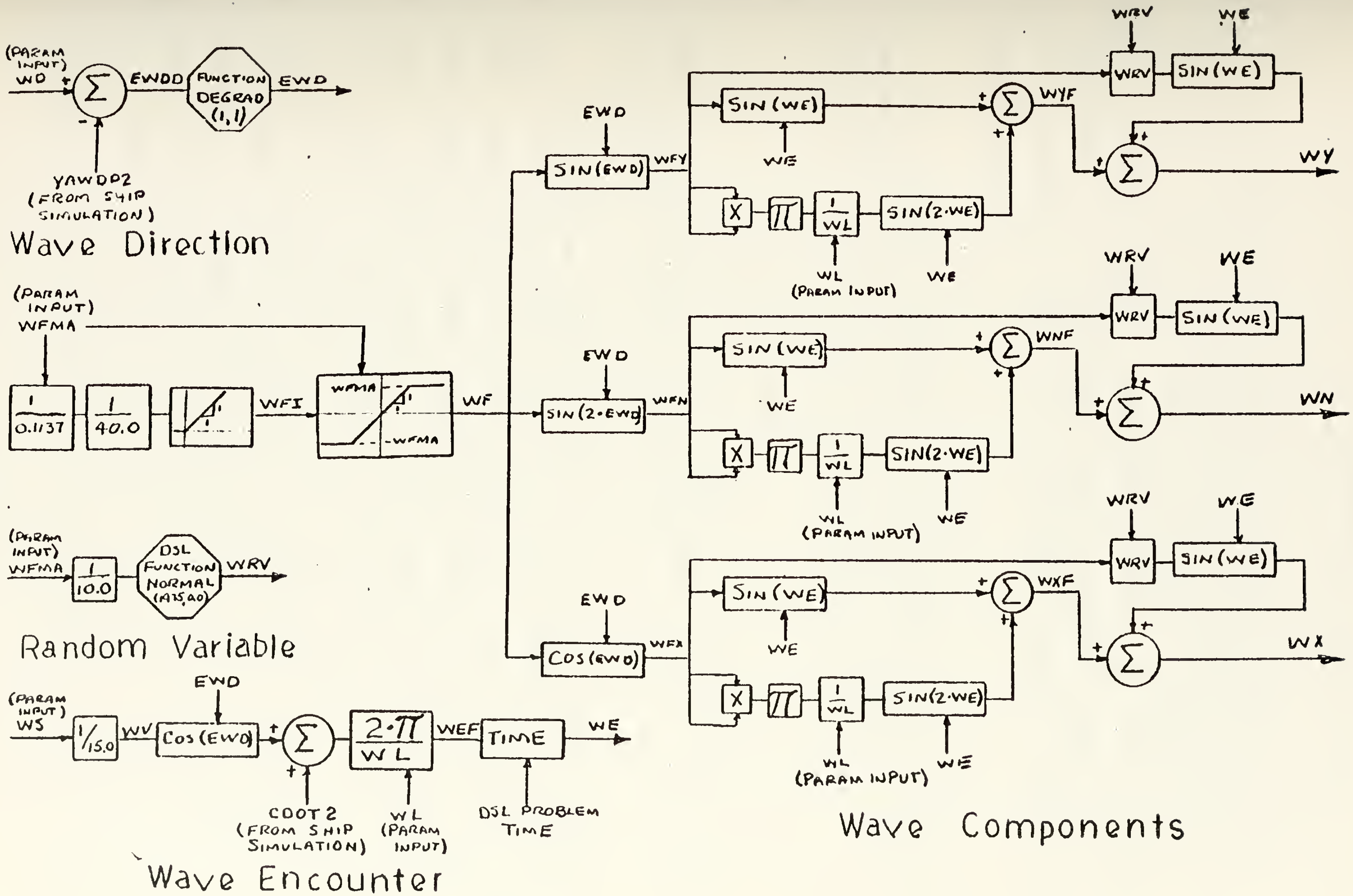


Ship A Rudder



Ship B Rudder

Rudder Modeling



Wave Generator

APPENDIX C

Throughout this thesis subroutine SLOPES has been used to output the interactive forces and moments between ships in the RAS situation. This subroutine, adapted from ref. 11, does not contain a complete picture of the circumstances envisioned. In particular, ship's speeds other than the 15 kt. operating point and different ship lengths are not accounted for.

As stated in chapter II, the speed modification factor can easily be applied for both ships at the same speed and other than 15 kts. with the following expression:

$$SPDP = CDOT^2$$

Ships replenishing with different lengths can also be incorporated as shown in ref. 1.

Subroutine FAMIC listed in this appendix incorporates these two ideas along with a better method of determining the interactive forces and moments. The curves of figures II-11 and II-12 were quantized every 50 feet of DX for all the DY curves shown. These points were then used in the NPGS XDS-930C digital computer and AGT-10 graphics terminal to obtain a family of best fit curves. The best fit criteria is based on the sum of the error squared at each quantized point (modified somewhat by this researcher's evaluation of best fit between points to eliminate spikes and other anomalies). The results of this curve fit process is summarized in tables C-1 and C-2, which includes tabulation of the best fit criteria. These polynomial coefficients are based on the DX distance and are coded in

Power	Y50[YY(1)]	Y60[YY(2)]	Y70[YY(3)]	Y80[YY(4)]	Y90[YY(5)]	Y100[YY(6)]
0	84.324	75.260	67.668	59.223	53.449	47.441
1	.364580E 2	.384289E 2	.305160E 2	.221400E 2	.174604E 2	.115702E 2
2	-.118950E 3	-.116865E 3	-.103683E 3	-.944770E 2	-.740983E 2	-.587039E 2
3	-.197311E 2	-.504085E 2	-.265717E 2	-.710687E 1	-.885819E 0	.165712E 2
4	.513437E 2	.684302E 2	.719316E 2	.103147E 3	.467865E 2	.242239E 2
5	.503561E 1	.466749E 2	.158656E 2	-.286063E 1	-.749374E 1	-.397052E 2
6	-.100337E 2	-.194633E 2	-.344841E 2	-.106332E 3	-.206166E 2	-.164427E 1
7	-.484175E 0	-.238722E 2	-.532674E 1	-.304098E 0	.488631E 1	.345891E 2
8	.750949E 0	-.398443E 0	.101864E 2	.688338E 2	.584863E 1	.677233E 0
9		.655075E 1	.877360E 0	.456701E 1	-.136413E 1	-.158999E 2
10		.181994E 1	-.158465E 1	-.221987E 2	-.941379E 0	-.255071E 1
11		-.916189E 0	-.556587E-1	-.347210E 1	.178935E 0	.405863E 1
12		-.428170E 0	.984583E-1	.160105E 1	.729225E-1	.131265E 1
13		.514317E-1		.114485E 1	-.901537E-2	-.543239E 0
14		.316613E-1		.965145E 0	-.176277E-2	-.256741E 0
15				-.179844E 0		.296983E-1
16				-.256001E 0		.177684E-1
17				.109745E-1		
18				.189745E-1		

Table C-1

Interactive Curve Fit Polynomial Coefficients

Power	Y110[YY(7)]	Y120[YY(8)]	Y130[YY(9)]	Y140[YY(10)]	Y150[YY(11)]
0	40.423	35.484	30.606	26.063	22.876
1	.142822E 2	.146142E 2	.138458E 2	.120761E 2	.108418E 2
2	-.533075E 2	-.458814E 2	-.425663E 2	-.341076E 2	-.196219E 2
3	-.103692E 2	-.158329E 2	-.139724E 2	-.150249E 2	-.127593E 2
4	.389319E 2	.398457E 2	.661640E 2	.574245E 2	.133189E 2
5	.875165E 1	.163639E 2	.863816E 1	.122510E 2	.974262E 1
6	-.200712E 2	-.242537E 2	-.800393E 2	-.702577E 2	-.695905E 1
7	-.502212E 1	-.953711E 1	-.274913E 1	-.333288E 1	-.335866E 1
8	.615393E 1	.820245E 1	.533407E 2	.488103E 2	.214837E 1
9	.154525E 1	.291771E 1	.211749E 1	-.946233E 0	.453764E 0
10	-.972521E 0	-.137319E 1	-.175113E 2	-.195409E 2	-.339142E 0
11	-.237633E 0	-.447954E 0	-.190223E 1	.766196E 0	.407256E-3
12	.607427E-1	.890809E-1	.163223E 1	.448663E 1	.211973E-1
13	.144010E-1	.272508E-1	.777504E 0	-.162184E 0	-.347399E-2
14			.560018E 0	-.548550E 0	
15			-.141969E 0	.116108E-1	
16			-.160683E 0	.276591E-1	
17			.960782E-2		
18			.120067E-1		

Table C-1

Interactive Curve Fit Polynomial Coefficients

Power	N50[YN(1)]	N60[YN(2)]	N70[YN(3)]	N80[YN(4)]	N90[YN(5)]	N100[YN(6)]
0	-37.329	-32.887	-29.637	-25.559	-23.064	-19.946
1	.396089E 2	.326330E 2	.242094E 2	.188936E 2	.173015E 2	.135031E 2
2	.402860E 2	.362146E 2	.371281E 2	.304472E 2	.328997E 2	.311296E 2
3	-.326885E 2	-.256484E 2	-.711639E 1	-.125164E 1	-.559773E 1	-.199635E 1
4	.397695E 1	.348686E 1	-.705366E 1	-.599694E 0	-.167065E 2	-.200747E 2
5	.847350E 1	.747495E 1	-.121721E 2	-.141799E 2	-.505714E 1	-.613206E 1
6	-.188304E 2	-.174840E 2	-.782624E 1	-.133875E 2	.626911E 1	.938014E 1
7	.334908E 0	-.236966E 1	.894936E 1	.915469E 1	.308497E 1	.324620E 1
8	.103131E 2	.102702E 2	.657382E 1	.943453E 1	-.179228E 1	-.279550E 1
9	-.136166E 1	.992354E 0	-.251129E 1	-.251460E 1	-.605389E 0	-.630851E 0
10	-.222623E 1	-.287062E 1	-.224627E 1	-.303990E 1	.301083E 0	.443308E 0
11	.647133E 0	-.217277E 0	.326965E 0	.332880E 0	.410884E-1	.438239e-1
12	.338223E-1	.397733E 0	.366197E 0	.478124E 0	-.206993E-1	-.281713E-1
13	-.132735E 0	.168090E-1	-.164232E-1	-.174528E-1		
14	.553291E-1	-.218380E-1	-.231377E-1	-.294675E-1		
15	.989063E-2					
16	-.574936E-2					

Table C-1

Interactive Curve Fit Polynomial Coefficients

Power	N110[YN(7)]	N120[YN(8)]	N130[YN(9)]	N140[YN(10)]	N150[YN(11)]
0	-16.682	-13.636	-10.637	-8.880	-6.955
1	.101849E 2	.103644E 2	.800782E 1	.603195E 1	.502563E 1
2	.223191E 2	.191665E 2	.104719E 2	.106262E 2	.784026E 1
3	-.108658E 0	-.867847E 1	-.587329E 1	-.222096E 1	-.261221E 1
4	-.296954E 1	-.102589E 2	.423064E 1	-.348743E 1	-.211257E 1
5	-.587218E 1	.793490E 1	.508267E 1	-.205426E 0	.551911E 0
6	-.100911E 2	-.357415E 1	-.106235E 2	.518022E 0	.292597E 0
7	.333381E 1	-.610692E 1	-.388765E 1	.102580E 0	-.213673E 0
8	.830859E 1	-.697142E 0	.702568E 1	-.198157E-1	-.779550E-1
9	-.891852E 0	.235015E 1	.146642E 1	-.592635E-2	.513869E-1
10	-.281000E 1	.575924E-1	-.225402E 1	-.269964E-2	.209270E-1
11	.124323E 0	-.420471E 0	-.258120E 0		-.396563E-2
12	.443430E 0	-.835171E-3	.354734E 0		-.211007E-2
13	-.718710E-2	.283053E-1	.172570E-1		
14	-.268301E-1		-.218733E-1		

Table C-1

Interactive Curve Fit Polynomial Coefficients

Curve Fit	Best Fit			Best Fit (Modified)		
	Order	$\sum e^2$	$\bar{e}$	Order	$\sum e^2$	$\bar{e}$
Y50	15	2.1664	0.307	8	70.345	1.749
Y60	15	2.1462	0.305	14	2.1703	0.307
Y70	15	0.41727	0.135	12	1.1915	0.228
Y80	15	0.71421	0.176	18	0.87588	0.195
Y90	15	1.2793	0.236	14	1.2981	0.238
Y100	16	1.1573	0.224	16	1.1573	0.224
Y110	15	1.2798	0.236	13	1.3400	0.241
Y120	15	0.39722	0.131	13	0.77148	0.183
Y130	15	0.54194	0.154	18	0.66737	0.170
Y140	15	0.77259	0.183	16	1.0620	0.215
Y150	15	0.26589	0.108	13	0.44726	0.139
N50	17	0.80547	0.187	16	0.80572	0.187
N60	18	0.63879	0.167	14	0.73449	0.179
N70	17	0.57433	0.158	14	0.59042	0.160
N80	14	0.43632	0.138	14	0.43632	0.138
N90	17	0.77685	0.184	12	0.83329	0.190
N100	15	0.48934	0.146	12	0.67948	0.172
N110	15	0.25701	0.106	14	0.59247	0.160
N120	15	0.29538	0.113	13	0.62067	0.164
N130	15	0.051807	0.047	14	0.064323	0.053
N140	15	0.13166	0.076	10	0.30835	0.116
N150	15	0.11837	0.072	12	0.16549	0.085
Avg.		0.71425	0.176		3.96169	0.252

Table C-2

Interactive Curve Fit Error Analysis

subroutine FAMIC as YY(1) thru YY(11) and YN(1) thru YN(11). An interpolation algorithm is used to determine the forces and moments at DY points between the curves of each family. Although all the computations are based on measurements from the control ship (ship #2), the interactive forces and moments are also computed for the reference ship (ship #1).

Figures C-1 and C-2 are the interactive forces and moments output to show comparison to figures II-11 and II-12. The speed of this run was the operating point of 15 kts. The ships are of equal length (527.8 feet).

Linear interpolation of the interactive curves for greater than 150 feet DY distance is accomplished from this 150 foot curve to a value of 0.0 at 200 feet. It therefore assumes no force and moment are present outside the 200 foot range. All forces and moments for DY distance of less than 50 feet are taken as that of the 50 foot curve. These two endpoints are by no means exact, but will suffice until more detailed data can be gathered. Another inexact endpoint is produced at the curve families limits of ±550 feet. At these points, the forces and moments are forced to 0.0 since detailed data outside of these limits was not available. A side effect of this abrupt truncation will manifest itself in the instantaneous commencement of the forces and moments during the approach phase run. The endpoint variations in some of the curves of figures C-1 and C-2 are due in part to the curve fitting routine used, but mostly to the differences in computer precision. (curve fits were calculated on a 11 digit precision XDS 9300 while the curves were plotted on single precision 7 digit IBM 360/67)

As previously mentioned, the speed modification for other than the operating point of 15 kts. is only completely valid for the situation where both ships are at the same speeds. Since this thesis considered an approach phase

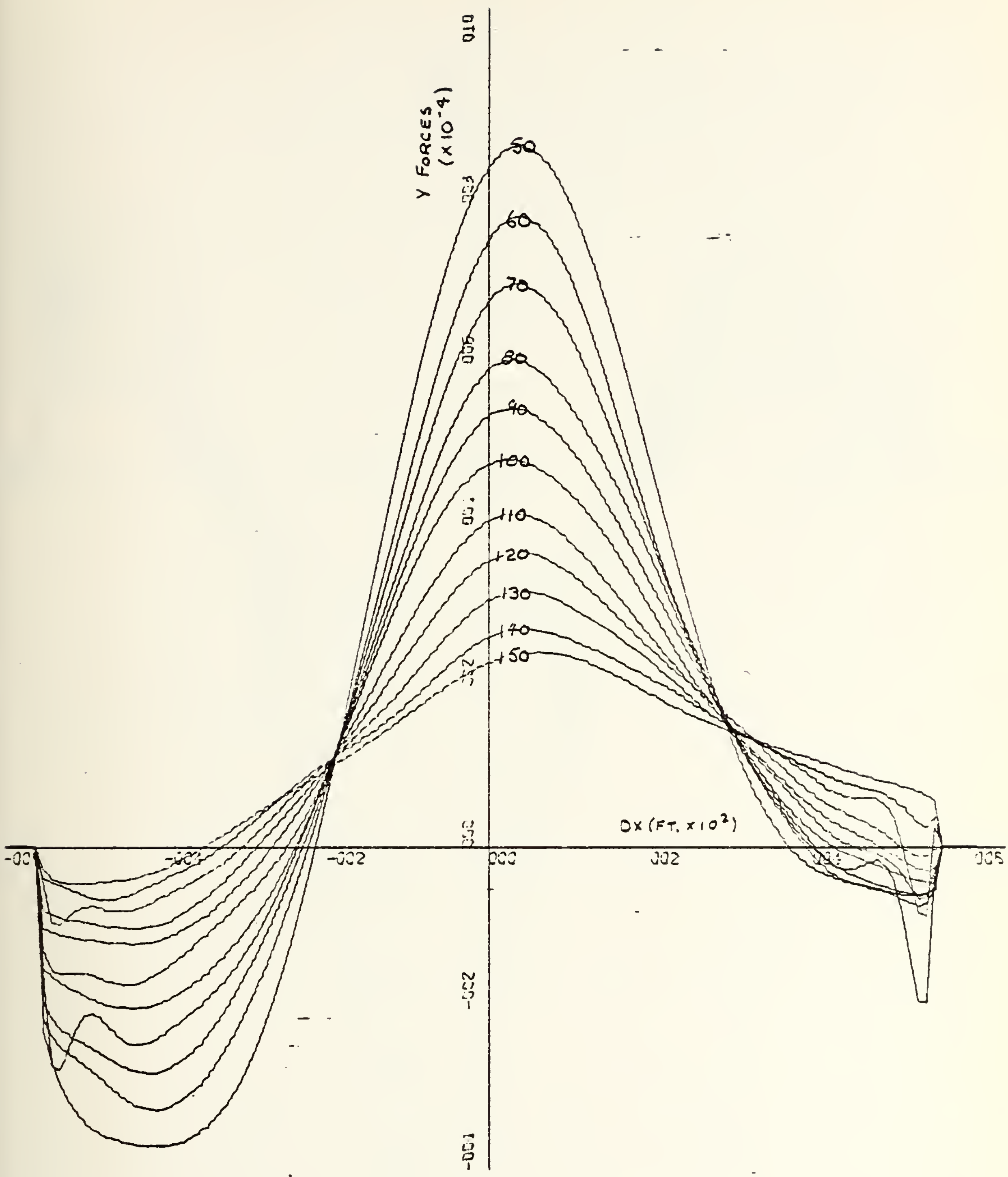


Figure C-1
Curve Fitted Interactive Y Forces

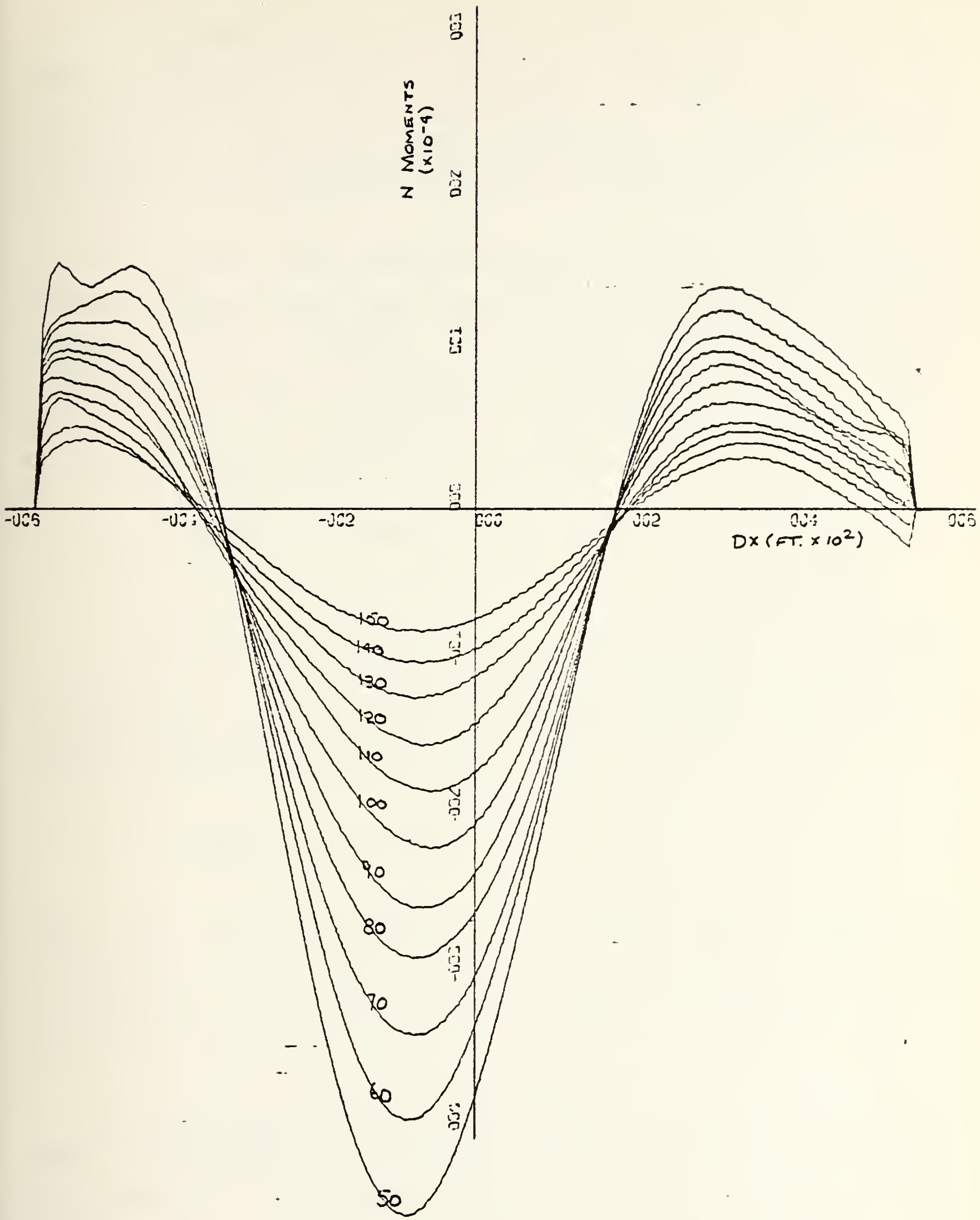


Figure C-2

Curve Fitted Interactive N Moments

where the control ship enters the interactive field at a speed quite different than the reference ship, some modification of the interactive effects should be considered. However, exact relationships are not available to compute the required modification factors.

To dispel any problems with the design of the heading control system, the worst case speed modification factor was chosen. This factor, in effect, considers that the interactive forces and moments are derived from the control ship. This is accomplished in subroutine FAMIC with the following fortran expression:

```
SPDP2 = CDOT2**2
```

As stated in chapter II, it is felt that it is more accurate to consider the interactive forces and moments to be modified by the speed of the reference ship, and can be coded in subroutine FAMIC as:

```
SPDP2 = CDOT1**2
```

With the scenario followed throughout this thesis, this expression would equate to unity throughout the RAS situation, since the reference ship is maintained at 1.0 normalized speed (15 kts.).

For the sake of error analysis, simulation of the worst case modification is performed. This gives rise to forces and moments 2.25 times what they were in the rest of this thesis during a portion of the approach phase when the normalized speed of the control ship is 1.5. Figures C-3 and C-4 show the interactive forces and moments for the approach phase of the simulation. The comparison plots which appear in chapter III as figures III-24 and III-25

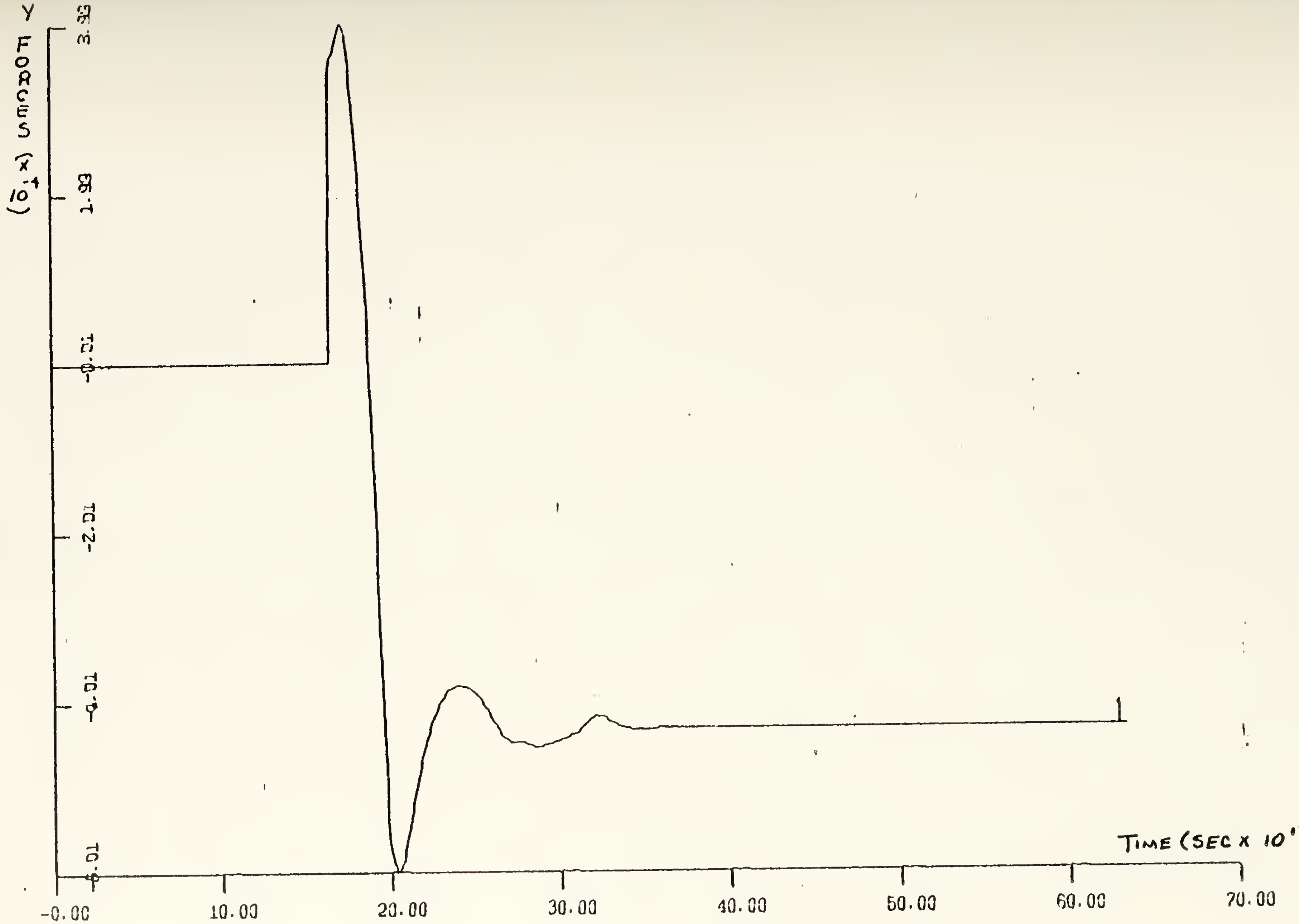


Figure C-3

Approach Phase Curve Fitted Y Forces

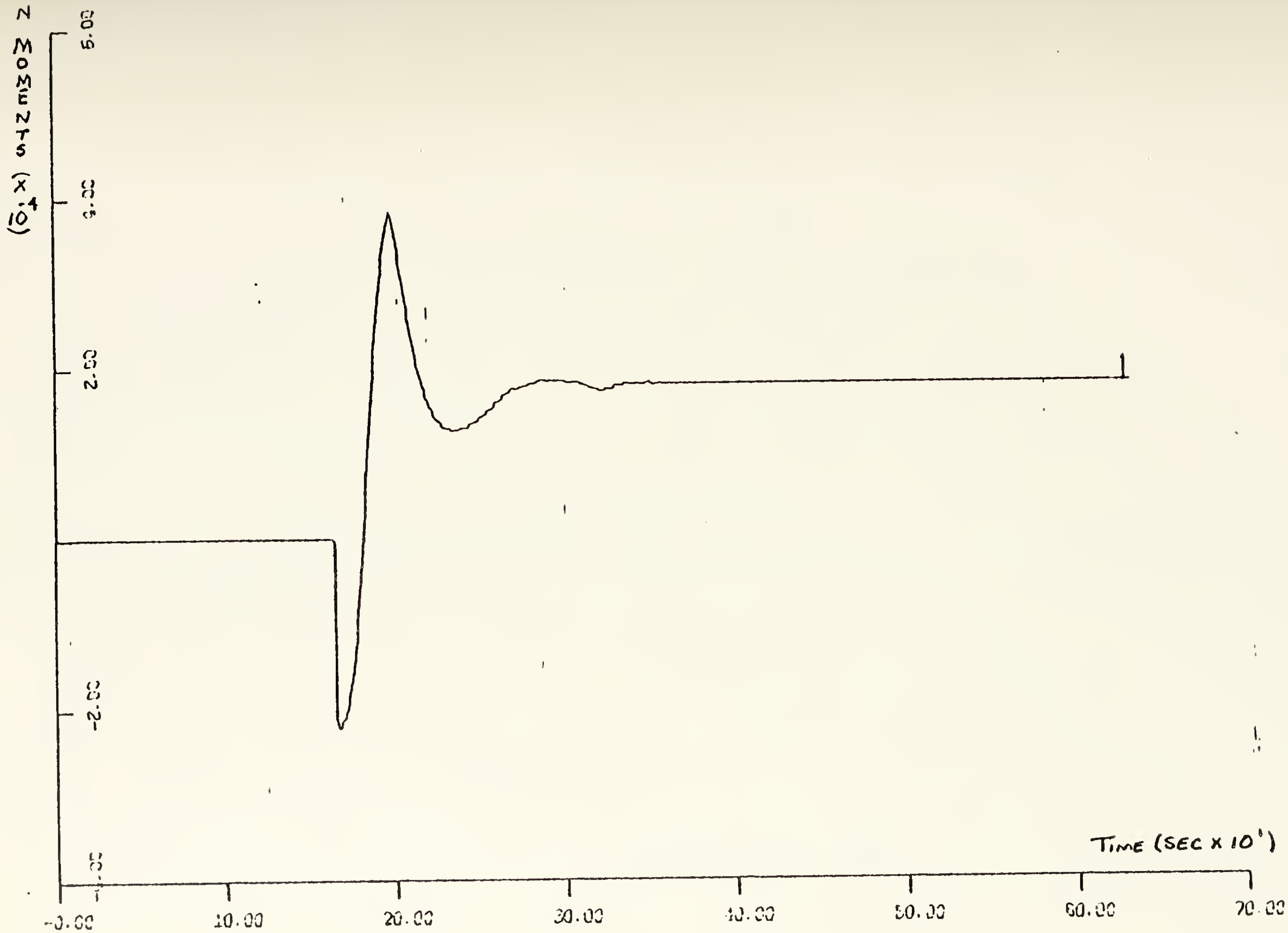


Figure C-4

Approach Phase Curve Fitted N Moments

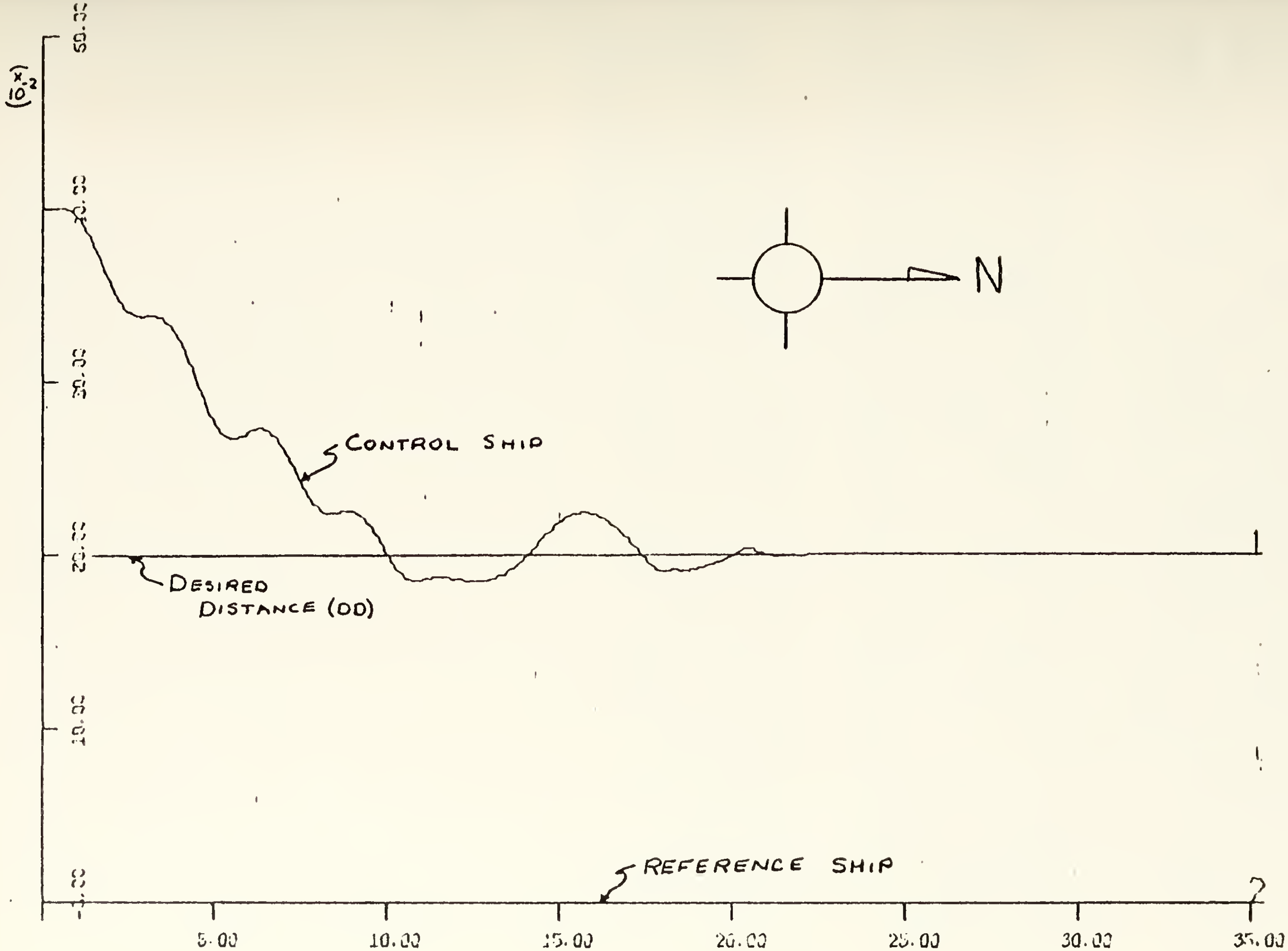


Figure C-5

Approach Phase Geographical Plot From Modified Interactive Effects illustrate the extent of the changes. Most notable is the smoother output of subroutine FAMIC. This more realistically portrays the interactive effects in the RAS environment. Figure C-5 portrays the geographical plot which compares with figure III-26 without speed modification. Although differences exist, figure C-5 illustrates that the interactive effects speed modification factor for the worst case does not drastically alter the approach phase outcome. The heading control system design is still valid in the face of these changes.

For reference, figures C-6 and C-7 show the interactive forces and moments in the turn phase as calculated by subroutine FAMIC. Figure C-8 is the turn phase lateral distance plot produced. It can be seen from this illustration that the maximum excursion is 0.0056 normalized distance (2.96 feet), well within acceptable limits.

In summary, the designed control system will accomodate even the worst case modification of the interactive effects. This insensitivity to a large range of perturbations, makes this control system a more viable design for actual ship installation.

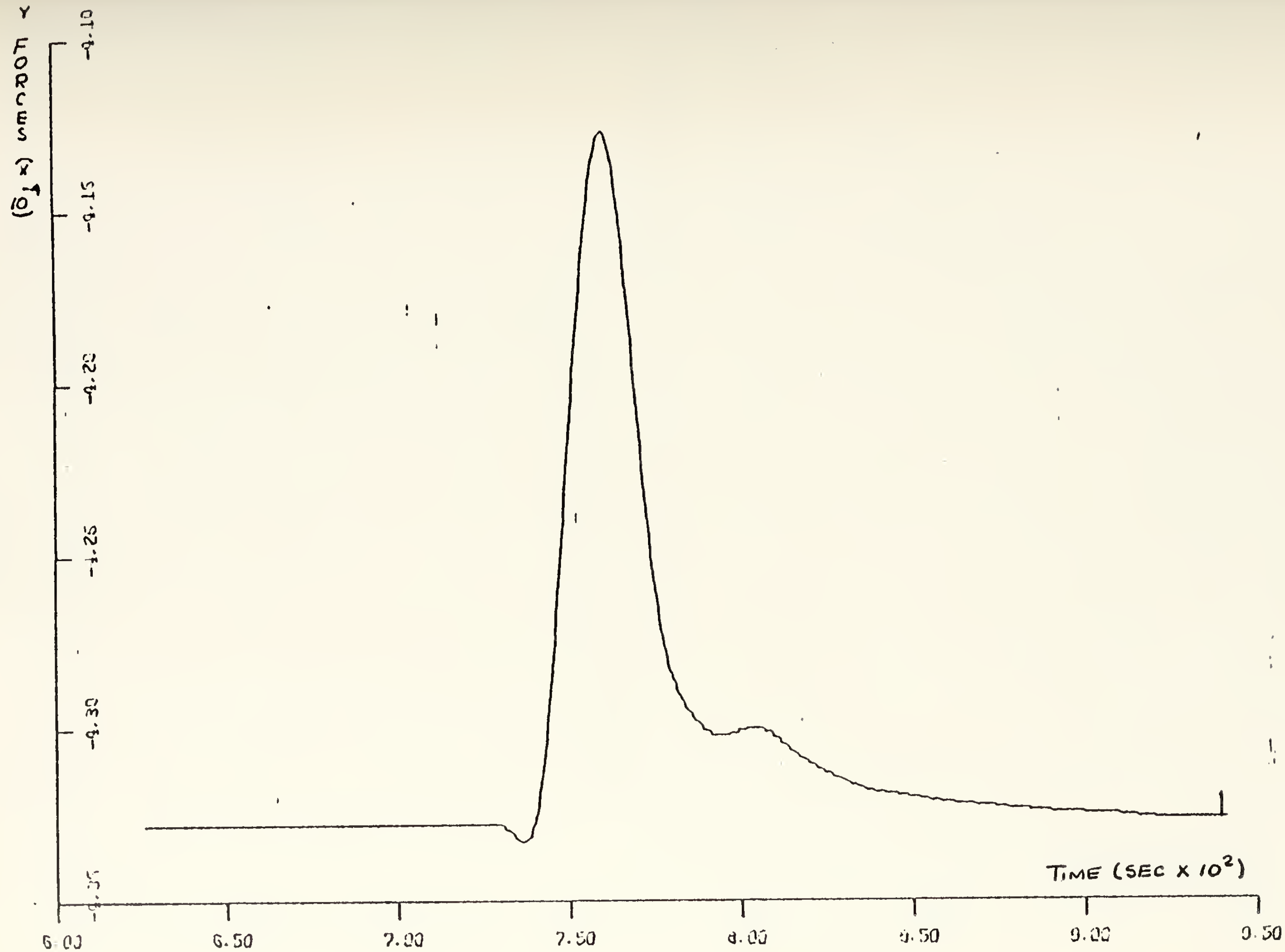


Figure C-6

Turn Phase Curve Fitted Y Forces

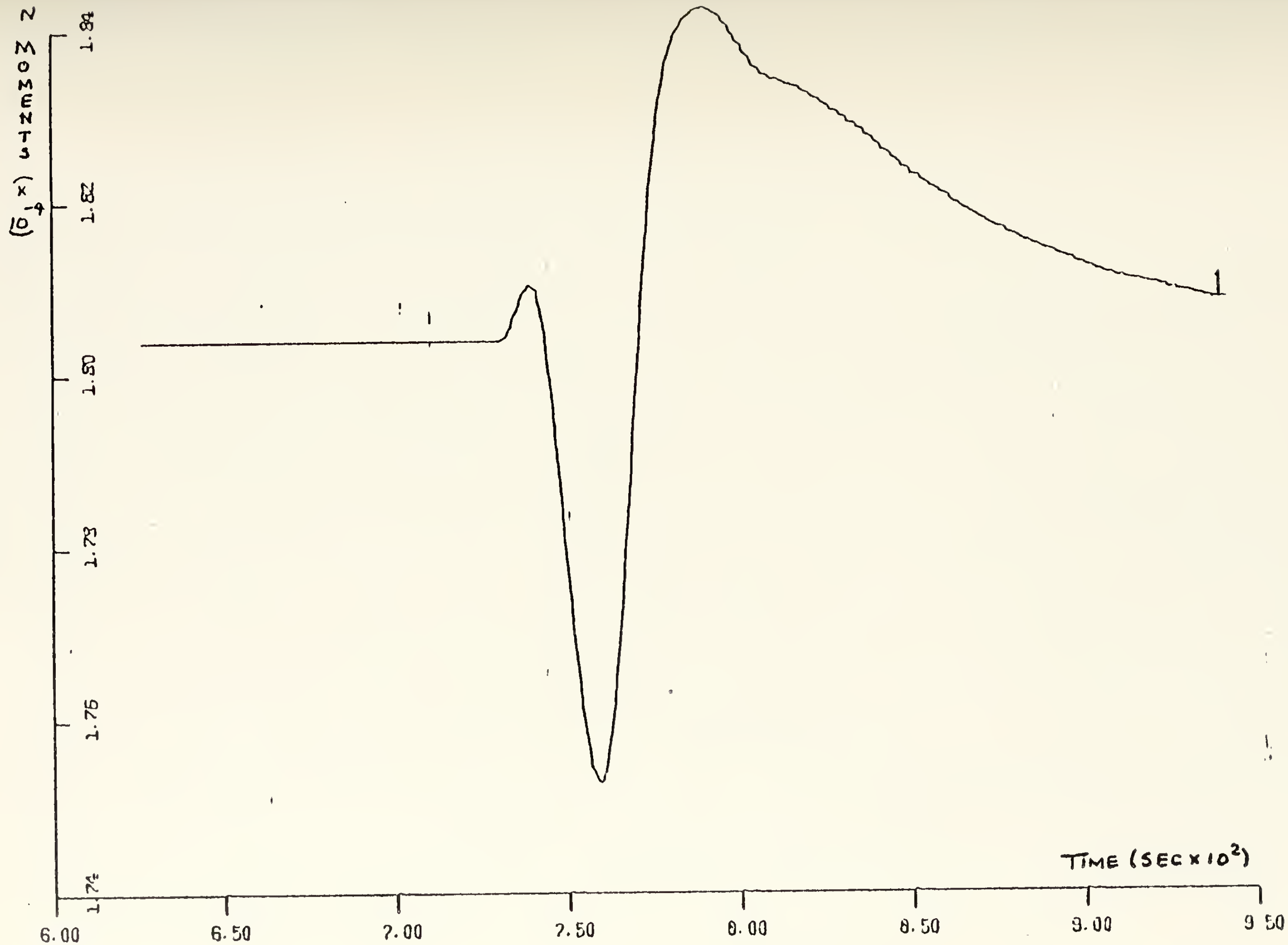


Figure C-7

Turn Phase Curve Fitted N Moments

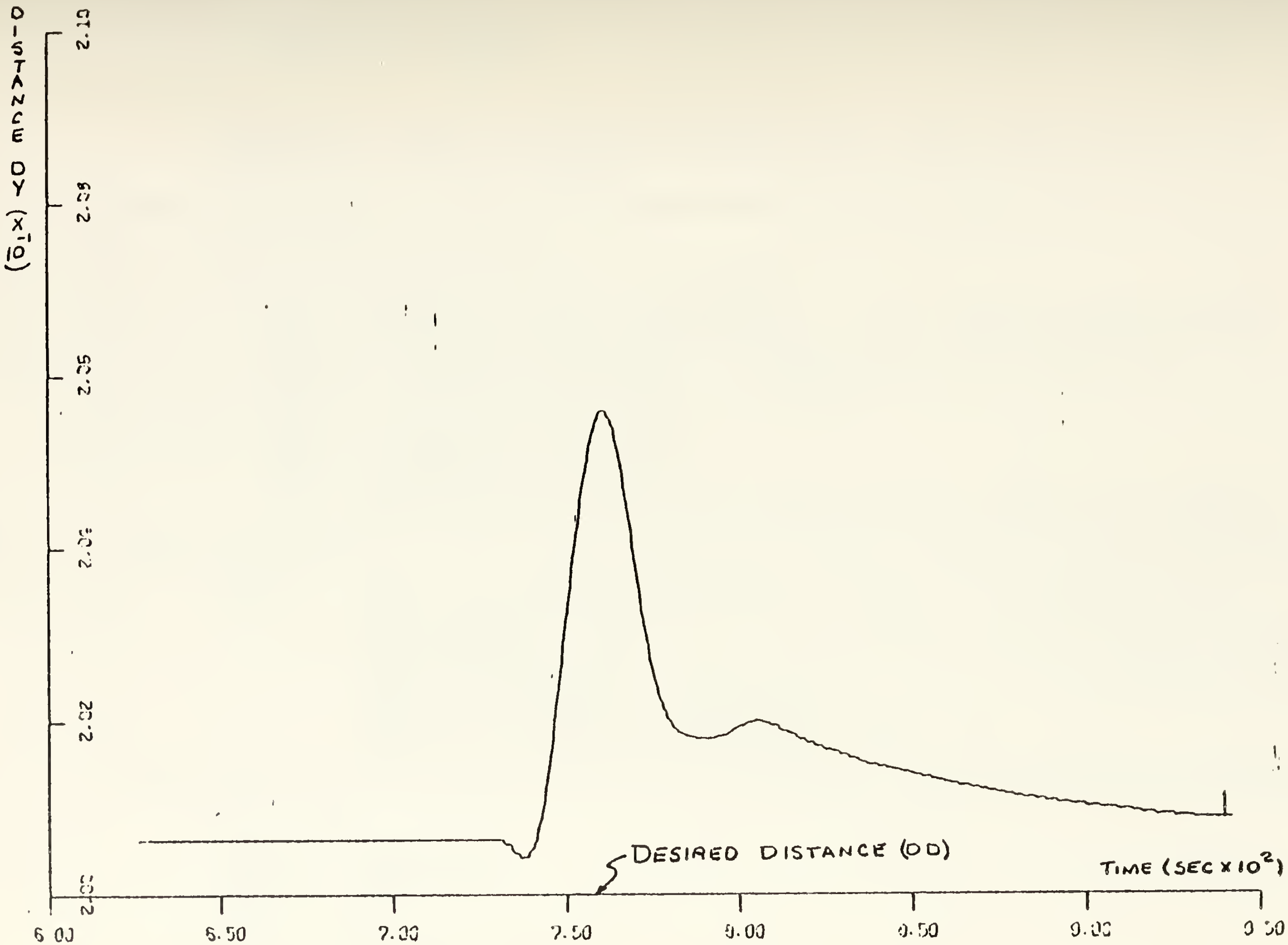


Figure C-8

Turn Phase Lateral Distance DY From Modified Interactive Effects

```
      SUBROUTINE FAMIC (XL1,XL2,ADX,ADY,CDOT1,CDOT2,YY1,YY2,YN1,YN2)
      DIMENSION YY(12), YN(12)
C
C*********************************************************************
C
C     DEFINITION OF TERMS:
C
C        XL1 = LENGTH OF SHIP #1 IN FEET
C        XL2 = LENGTH OF SHIP #2 IN FEET
C        ADX = LONGITUDINAL SEPARATION OF SHIPS (NONDIMENSIONALIZED)
C        ADY = LATERAL SEPARATION OF SHIPS (NONDIMENSIONALIZED)
C        CDOT1 = NORMALIZED SPEED OF SHIP #1
C        CDOT2 = NORMALIZED SPEED OF SHIP #2
C        YY1 = Y FORCE ON SHIP #1
C        YY2 = Y FORCE ON SHIP #2
C        YN1 = N MOMENT ON SHIP #1
C        YN2 = N MOMENT ON SHIP #2
C        X = LONGITUDINAL SEPARATION NORMALIZED TO CURVE FIT FACTOR
C
C*********************************************************************
C
C     NOTE:    FOR STBD SIDE TO APPROACH OF SHIP #2 ON SHIP #1, ADY
C              SHOULD BE NEGATIVE.
C              FOR APPROACH WHEN SHIP #2 IS ASTERN OF SHIP #1 ADX SHOULD
C              BE NEGATIVE.
C              CONVERSES ARE ALSO TRUE - PORT SIDE TO = POSITIVE ADY AND
C              FWD OF ALONGSIDE = POSITIVE.
C              ADX AND ADY ARE REFERENCED TO SHIP #2.
C              NORMALIZED SPEEDS BASED ON 15 KTS.
C
C*********************************************************************
C
      XL = XL2
      SPDP1 = CDOT1**2
      SPDP2 = CDOT2**2
      XLP1 = XL1/XL2
      XLP2 = 1.0
      N = 1
    1 ADXL = XL*ADX
      X = XL*ADX/250.0
      IF (ABS(ADXL).GT.550.0) GO TO 6
C
C     POLYNOMIAL POWER PRE-COMPUTATION
```

```
C
      X1 = X
      X2 = X**2
      X3 = X**3
      X4 = X**4
      X5 = X**5
      X6 = X**6
      X7 = X**7
      X8 = X**8
      X9 = X**9
      X10 = X**10
      X11 = X**11
      X12 = X**12
      X13 = X**13
      X14 = X**14
      X15 = X**15
      X16 = X**16
      X17 = X**17
      X18 = X**18
C
C     EQUATIONS OF Y FORCES
C
      YY(1) = 84.324+0.364580E+02*X1-0.118950E+03*X2-0.197311E+02*X3+0.5
     113437E+02*X4+0.503561E+01*X5-0.100337E+02*X6-0.484175E+00*X7+0.750
     2949E+00*X8
      YY(2) = 75.260+0.384289E+02*X1-0.116865E+03*X2-0.504085E+02*X3+0.6
     184302E+02*X4+0.466749E+02*X5-0.194633E+02*X6-0.238722E+02*X7-0.398
     2443E+00*X8+0.655075E+01*X9+0.181994E+01*X10-0.916189E+00*X11-0.428
     3170E+00*X12+0.514317E-01*X13+0.316613E-01*X14
      YY(3) = 67.668+0.305160E+02*X1-0.103683E+03*X2-0.265717E+02*X3+0.7
     119316E+02*X4+0.158656E+02*X5-0.344841E+02*X6-0.532674E+01*X7+0.101
     2864E+02*X8+0.877360E+00*X9-0.158465E+01*X10-0.556587E-01*X11+0.984
     3583E-01*X12
      YY(4) = 59.223+0.221400E+02*X1-0.944770E+02*X2-0.710681E+01*X3+0.1
     103147E+03*X4-0.286063E+01*X5-0.106332E+03*X6-0.304098E+00*X7+0.688
     2338E+02*X8+0.456701E+01*X9-0.221987E+02*X10-0.347210E+01*X11+0.160
     3105E+01*X12+0.114485E+01*X13+0.965145E+00*X14-0.179844E+00*X15-0.2
     456001E+00*X16+0.109745E-01*X17+0.189875E-01*X18
      YY(5) = 53.449+0.174604E+02*X1-0.740983E+02*X2-0.885819E+00*X3+0.4
     167865E+02*X4-0.749374E+01*X5-0.206166E+02*X6+0.488631E+01*X7+0.584
     2863E+01*X8-0.136413E+01*X9-0.941379E+00*X10+0.178935E+00*X11+0.729
     3225E-01*X12-0.901537E-02*X13-0.176277E-02*X14
      YY(6) = 47.441+0.115702E+02*X1-0.587039E+02*X2+0.165712E+02*X3+0.2
     142239E+02*X4-0.397052E+02*X5-0.164427E+01*X6+0.345891E+02*X7+0.677
     2233E+00*X8-0.158999E+02*X9-0.255071E+01*X10+0.405863E+01*X11+0.131
     3265E+01*X12-0.543239E+00*X13-0.256741E+00*X14+0.296983E-01*X15+0.1
     477684E-01*X16
      YY(7) = 40.423+0.142822E+02*X1-0.533075E+02*X2-0.103692E+02*X3+0.3
```

```
 189319E+02*X4+0.875165E+01*X5-0.200712E+02*X6-0.502212E+01*X7+0.615FAMI 920
 2393E+01*X8+0.154525E+01*X9-0.972521E+00*X10-0.237633E+00*X11+0.607FAMI 930
 3427E-01*X12+0.144010E-01*X13                                      FAMI 940
  YY(8) = 35.484+0.146142E+02*X1-0.458814E+02*X2-0.158329E+02*X3+0.3FAMI 950
 198457E+02*X4+0.163639E+02*X5-0.242537E+02*X6-0.953711E+01*X7+0.820FAMI 960
 2245E+01*X8+0.291771E+01*X9-0.137319E+01*X10-0.447954E+00*X11+0.890FAMI 970
 3809E-01*X12+0.272508E-01*X13                                      FAMI 980
  YY(9) = 30.606+0.138458E+02*X1-0.425663E+02*X2-0.139724E+02*X3+0.6FAMI 990
 161640E+02*X4+0.863816E+01*X5-0.800393E+02*X6-0.274913E+01*X7+0.533FAMI1000
 2407E+02*X8+0.211749E+01*X9-0.175113E+02*X10-0.190223E+01*X11+0.163FAMI1010
 3223E+01*X12+0.777504E+00*X13+0.560018E+00*X14-0.141969E+00*X15-0.1FAMI1020
 460683E+00*X16+0.960782E-02*X17+0.120067E-01*X18                   FAMI1030
  YY(10) = 26.063+0.120761E+02*X1-0.341076E+02*X2-0.150249E+02*X3+0.FAMI1040
 1574245E+02*X4+0.122510E+02*X5-0.702577E+02*X6-0.333288E+01*X7+0.48FAMI1050
 28103E+02*X8-0.946233E+00*X9-0.195409E+02*X10+0.766196E+00*X11+0.44FAMI1060
 38663E+01*X12-0.162184E+00*X13-0.548550E+00*X14+0.116108E-01*X15+0.FAMI1070
 4276591E-01*X16                                                    FAMI1080
  YY(11) = 22.876+0.108418E+02*X1-0.196219E+02*X2-0.127593E+02*X3+0.FAMI1090
 1133189E+02*X4+0.974262E+01*X5-0.695905E+01*X6-0.335866E+01*X7+0.21FAMI1100
 24837E+01*X8+0.453764E+00*X9-0.339142E+00*X10+0.407256E-03*X11+0.21FAMI1110
 31973E-01*X12-0.347399E-02*X13                                     FAMI1120
C                                                                   FAMI1130
C  EQUATIONS OF N MOMENTS                                           FAMI1140
C                                                                   FAMI1150
  YN(1) = -37.329+0.396089E+02*X1+0.402860E+02*X2-0.326885E+02*X3+0.FAMI1160
 1397695E+01*X4+0.847350E+01*X5-0.188304E+02*X6+0.334908E+00*X7+0.10FAMI1170
 23131E+02*X8-0.136166E+01*X9-0.222623E+01*X10+0.647133E+00*X11+0.33FAMI1180
 38223E-01*X12-0.132735E+00*X13+0.553291E-01*X14+0.989063E-02*X15-0.FAMI1190
 4574936E-02*X16                                                    FAMI1200
  YN(2) = -32.887+0.326330E+02*X1+0.362146E+02*X2-0.256484E+02*X3+0.FAMI1210
 1348686E+01*X4+0.747495E+01*X5-0.174840E+02*X6-0.236966E+01*X7+0.10FAMI1220
 22702E+02*X8+0.992354E+00*X9-0.287062E+01*X10-0.217277E+00*X11+0.39FAMI1230
 37733E+00*X12+0.168090E-01*X13-0.218380E-01*X14                    FAMI1240
  YN(3) = -29.637+0.242094E+02*X1+0.371281E+02*X2-0.711639E+01*X3-0.FAMI1250
 1705366E+01*X4-0.121721E+02*X5-0.782624E+01*X6+0.894936E+01*X7+0.65FAMI1260
 27382E+01*X8-0.251129E+01*X9-0.224627E+01*X10+0.326965E+00*X11+0.36FAMI1270
 36197E+00*X12-0.164232E-01*X13-0.231377E-01*X14                    FAMI1280
  YN(4) = -25.559+0.188936E+02*X1+0.304472E+02*X2-0.125164E+01*X3-0.FAMI1290
 1599694E+00*X4-0.141799E+02*X5-0.133875E+02*X6+0.915469E+01*X7+0.94FAMI1300
 23453E+01*X8-0.251460E+01*X9-0.303990E+01*X10+0.332880E+00*X11+0.47FAMI1310
 38124E+00*X12-0.174528E-01*X13-0.294675E-01*X14                    FAMI1320
  YN(5) = -23.064+0.173015E+02*X1+0.328997E+02*X2-0.559773E+01*X3-0.FAMI1330
 1167065E+02*X4-0.505714E+01*X5+0.626911E+01*X6+0.308497E+01*X7-0.17FAMI1340
 29228E+01*X8-0.605389E+00*X9+0.301083E+00*X10+0.410884E-01*X11-0.20FAMI1350
 36993E-01*X12                                                      FAMI1360
  YN(6) = -19.946+0.135031E+02*X1+0.311296E+02*X2-0.199635E+01*X3-0.FAMI1370
 1200747E+02*X4-0.613206E+01*X5+0.938014E+01*X6+0.324620E+01*X7-0.27FAMI1380
 29550E+01*X8-0.630851E+00*X9+0.443308E+00*X10+0.438239E-01*X11-0.28FAMI1390
```

```
31713E-01*X12
 YN(7) = -16.682+0.101849E+02*X1+0.223191E+02*X2-0.108658E+00*X3-0.
1296954E+01*X4-0.587218E+01*X5-0.100911E+02*X6+0.333381E+01*X7+0.83
20659E+01*X8-0.891852E+00*X9-0.281000E+01*X10+0.124323E+00*X11+0.44
33430E+00*X12-0.718710E-02*X13-0.268301E-01*X14
 YN(8) = -13.636+0.103644E+02*X1+0.191665E+02*X2-0.867847E+01*X3-0.
1102589E+02*X4+0.793490E+01*X5+0.357415E+01*X6-0.610692E+01*X7-0.69
27142E+00*X8+0.235015E+01*X9+0.575924E-01*X10-0.420471E+00*X11-0.83
35171E-03*X12+0.283053E-01*X13
 YN(9) = -10.637+0.800782E+01*X1+0.104719E+02*X2-0.587329E+01*X3+0.
1423064E+01*X4+0.508267E+01*X5-0.106235E+02*X6-0.388765E+01*X7+0.70
22568E+01*X8+0.146642E+01*X9-0.225402E+01*X10-0.258120E+00*X11+0.35
34734E+00*X12+0.172570E-01*X13-0.218733E-01*X14
 YN(10) = -8.880+0.603195E+01*X1+0.106262E+02*X2-0.222096E+01*X3-0.
1348743E+01*X4-0.205426E+00*X5+0.518022E+00*X6+0.102580E+00*X7-0.19
28157E-01*X8-0.592635E-02*X9-0.269964E-02*X10
 YN(11) = -6.955+0.502563E+01*X1+0.784026E+01*X2-0.261221E+01*X3-0.
1211257E+01*X4+0.551911E+00*X5+0.292597E+00*X6-0.213673E+00*X7-0.77
29550E-01*X8+0.513869E-01*X9+0.209270E-01*X10-0.396563E-02*X11-0.21
31007E-02*X12
C
  ADYA = ABS(ADY)
  ADYL = XL2*ADYA
  XII = (ADYL-40.0)/10.0
  I = XII
  XI = I
  IF (XII.LT.1.0.OR.I.LT.1) GO TO 4
  IF (XII.GT.11.0.OR.I.GT.11) GO TO 5
  IF (XI.LT.XII) GO TO 2
  IF (XI.GT.XII) GO TO 3
  YYI = YY(I)
  YNI = YN(I)
  GO TO 7
2 YYI = YY(I)+(YY(I+1)-YY(I))*(XII-XI)
  YNI = YN(I)+(YN(I+1)-YN(I))*(XII-XI)
  GO TO 7
3 YYI = YY(I)-(YY(I)-YY(I-1))*(XI-XII)
  YNI = YN(I)-(YN(I)-YN(I-1))*(XI-XII)
  GO TO 7
4 YYI = YY(1)
  YNI = YN(1)
  GO TO 7
5 IF (ADYL.GT.200.0) GO TO 6
  YYI = YY(11)*(1.0+(150.0-ADYL)/50.0)
  YNI = YN(11)*(1.0+(150.0-ADYL)/50.0)
  GO TO 7
6 YY1 = 0.0
  YY2 = 0.0
```

```
      YN1 = 0.0                                                         FAMI1880
      YN2 = 0.0                                                         FAMI1890
      RETURN                                                            FAMI1900
    7 IF (N.EQ.2) GO TO 8                                               FAMI1910
      YY2 = -YYI*SPDP2*XLP2*1.0E-05                                     FAMI1920
      YN2 = -YNI*SPDP2*XLP2*1.0E-05                                     FAMI1930
      N = N+1                                                           FAMI1940
      XL = -XL2                                                         FAMI1950
      GO TO 1                                                           FAMI1960
    8 YY1 = YYI*SPDP1*XLP1*1.0E-05                                      FAMI1970
      YN1 = YNI*SPDP1*XLP1*1.0E-05                                      FAMI1980
      IF (ADY.LT.0.0) GO TO 9                                           FAMI1990
      RETURN                                                            FAMI2000
    9 YY1 = -YY1                                                        FAMI2010
      YY2 = -YY2                                                        FAMI2020
      YN1 = -YN1                                                        FAMI2030
      YN2 = -YN2                                                        FAMI2040
      RETURN                                                            FAMI2050
      END                                                               FAMI2060
```

COMPUTER PROGRAM #1

This program incorporates the ship dynamics of two identical Mariner hulls. These hulls are superimposed in space to allow for comparison of the effects contributed to rudder modeling differences. In this particular run a step and ramp rudder were compared in chapter II.

Another benifit of this program is to set up the two identical ships required for the RAS simulations in chapter III. Basically, verification of the models in three degrees of freedom is accomplished for the Mariner hull chosen.

The plots produced in this run are shown in figures II-2 and II-3.

COMPUTER PROGRAM #1

```
//UHRINTF1 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=2
// EXEC DSL
//DSL.INPUT DD *
*      LINEAR RESPONSE OF THE MARINER - RAMP VS STEP RUDDER COMPARISON
TITLE LINEAR RESPONSE OF THE MARINER - RAMP VS STEP RUDDER COMPARISON
INTEG TRAPZ
INTGER NPLOT
CONST NPLOT=1
*      HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,IZNRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XJ=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELR=-0.00126,XDELR=0.0
*      INITIAL CONDITIONS
INCON X01=0.0,Y01=0.0,X02=0.0,Y02=0.0
INITIAL
*      CALCULATION OF THE COEFFICIENTS
      D1=0.0
      D2=0.0
      NC1=-XU
      NC2=-XU
      A11=MYVD
      B11=-YV
      A21=-YRD
      B21=MYR
      A12=-NVD
      B12=-NV
      A22=IZNRD
      B22=-NR
      A33=MXUD
      B33=-XU
      KA1=-YDELR
      KB1=NDELR
      KC1=XDELR
      D=A11*A22-A12*A21
      DELRM=41.6953
      RDC=180./3.1415926
      DRC=3.1415926/180.
      LUC=20.84765
DERIVATIVE
*      SIMULATION SHIP A
      IF11=KA1*D1
      IF21=KB1*D1
```

```
      IF31=KC1*D1+NC1
      I11=-B11*ADOT1-B21*BDOT1+IF11
      I21=-B12*ADOT1-B22*BDOT1+IF21
      I31=-B33*CDOT1+IF31
      ADDOT1=(I11*A22-I21*A21)/D
      BDDOT1=(I21*A11-I11*A12)/D
      CDDOT1=I31/A33
      ADOT1=INTGRL(0.,ADDOT1)
      BDOT1=INTGRL(0.,BDDOT1)
      CDOT1=INTGRL(1.0,CDDOT1)
      B1=INTGRL(0.,BDOT1)
      XDOT1=CDOT1*COS(B1)-ADOT1*SIN(B1)
      YDOT1=CDOT1*SIN(B1)+ADOT1*COS(B1)
      X1=INTGRL(X01,XDOT1)
      Y1=INTGRL(Y01,YDOT1)
      YAW1=B1
      SWAY1=Y1
      SURGE1=X1
*     SIMULATION SHIP B
      IF12=KA1*D2
      IF22=KB1*D2
      IF32=KC1*D2+NC2
      I12=-B11*ADOT2-B21*BDOT2+IF12
      I22=-B12*ADOT2-B22*BDOT2+IF22
      I32=-B33*CDOT2+IF32
      ADDOT2=(I12*A22-I22*A21)/D
      BDDOT2=(I22*A11-I12*A12)/D
      CDDOT2=I32/A33
      ADOT2=INTGRL(0.,ADDOT2)
      BDOT2=INTGRL(0.,BDDOT2)
      CDOT2=INTGRL(1.0,CDDOT2)
      B2=INTGRL(0.,BDOT2)
      XDOT2=CDOT2*COS(B2)-ADOT2*SIN(B2)
      YDOT2=CDOT2*SIN(B2)+ADOT2*COS(B2)
      X2=INTGRL(X02,XDOT2)
      Y2=INTGRL(Y02,YDOT2)
      YAW2=B2
      SWAY2=Y2
      SURGE2=X2
DYNAMIC REGION
*     RUDDER RESPONSE INPUT
      D1=0.2618*STEP(2.0)
      D2=RAMP(2.0)*DELRM*DRC
      IF(D2.GE.0.2618) D2=0.2618
      SWAYD=SWAY2-SWAY1
      SURGED=SURGE2-SURGE1
      YAWD1=YAW1*RDC
      YAWD2=YAW2*RDC
```

```
      YAWDD=YAWD1-YAWD2
      D1D=D1*RDC
      D2D=D2*RDC
      DDD=D1D-D2D
      ATIME=LUC*TIME
SAMPLE
PRINT 0.04,ATIME,YAWD1,D1D,YAWD2,D2D,YAWDD,DDD
CONTRL FINTIM=30.,DELT=0.04,DELS=0.04
PRPLOT ONLY
      CALL DRWG(1,1,SURGE1,SWAY1)
      CALL DRWG(1,2,SURGE2,SWAY2)
      CALL DRWG(2,1,ATIME,YAWDD)
      CALL DRWG(3,1,ATIME,DDD)
TERMINAL
      CALL ENDRW(NPLOT)
END
STOP
//PLOT.SYSIN DD *
                                                                          5
0.0        1.0        -5.0       1.0        8.0        6.0
                                                                          5
0.0        100.0      -6.0       1.0        8.0        6.0
                                                                          4
                                            7.0        5.0
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #2

This program models a practical rudder response for a mariner ship type. The rudder limits (stops) are set at ±30 degrees and the rate of response is limited to ±2 degrees/second. A scale factor (LUC) is introduced to modify the response to match real time of the mariner hull chosen.

Twelve passes thru the program are accomplished to conform to different sets of initial conditions and final desired rudder conditions. The plots produced in this run are shown in figures II-5 and II-6.

COMPUTER PROGRAM #2

```
//UHRINTF2 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=1
// EXEC DSL
//DSL.INPUT DD *
*      RAS RUDDER CONTROL RUN TF2 - PRACTICAL RUDDER RESPONSE
TITLE RAS RUDDER CONTROL RUN TF2 - PRACTICAL RUDDER RESPONSE
INTEG TRAPZ
INTGER NPLOT,CURVE
CONST NPLOT=2
CONST DLTDM=2.0,DLTEM=7.0
PARAM CURVE=1
PARAM D2DDES=30.0
PARAM D2DIC=-30.0
INITIAL
      KG=DLTDM/DLTEM
      LUC=20.84765
DERIVATIVE
      DLTS=LIMIT(-30.0,30.0,D2DDES)
      DLTE=DLTS-D2D
      DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
      D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
*     ACTUAL TIME CONVERSION
      ATIME=TIME*LUC
SAMPLE
CONTRL FINTIM=1.7,DELT=0.04,DELS=0.04
PRINT 0.04,ATIME,DLTS,DLTE,DLTBE,D2D,D2DDES,D2DIC
PRPLOT ONLY
      CALL DRWG(1,CURVE,ATIME,D2D)
TERMINAL
      WRITE(6,100)D2DIC,D2DDES
  100 FORMAT(//,' LAST RUN IS FOR INITIAL RUDDER=',F10.5,' DESIRED RUDDE
     1R=',F10.5)
      D2DDES=D2DDES-5.0
      D2DIC=D2DIC+5.0
      CURVE=CURVE+1
      IF(NPLOT.EQ.1) D2DIC=0.0
      IF(CURVE.EQ.7) GO TO 1
      GO TO 2
    1 CURVE=1
      D2DIC=0.0
      D2DDES=30.0
      CALL ENDRW(NPLOT)
    2 CALL RERUN
END
```

```
END
STOP
//PLOT.SYSIN DD *

0.0         5.0         -30.0       10.0        7.0         6.0                                5

0.0         5.0         0.0         5.0         7.0         6.0                                5
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #3

This program models a reduced order (first order) gas turbine propulsion plant for an input-output relationship. The program does not scale the plant to the mariner hull used. This was done when introduced into the main simulation program first listed as computer program #8.

The time delay (P) is assisted in initialization by a dual feed into the system; one thru the delay itself and one directly into SPDIN. The program can be modified to compare a family of curves by introducing the following sequence into the TERMINAL region:

```
          INTEGER NUMB
                IF (NCUR.EQ.NUMB) CALL ENDRW(NPLOT)
                IF (NCUR.NE.NUMB) CALL RERUN
                NCUR = NCUR + 1
```

where NUMB is the number of curves desired (less than or equal to 10) which is set with a PARAM statement. The comparison is done on the conditions set in the terminal region [i.e. decrement or increment the system gains (eg. G = G + 0.02)].

The plot produced by this run is shown as part of figure II-10.

COMPUTER PROGRAM #3

```
//UHRINTF3 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=2
// EXEC DSL
//DSL.INPUT DD *
TITLE  SPEED CONTROL - FIRST ORDER FIT
INTEG RKSFX
INCON UIC=20.0
INTGER NPLOT,NCUR
PARAM NPLOT=1
PARAM UF=21.73
PARAM A=22.0
PARAM P=0.0
PARAM NCUR=1
PARAM G=0.092
INITIAL
      K=UF/A
      WRITE(6,100) G,NCUR,K
  100 FORMAT(//,' THE FOLLOWING RUN FOR POLE=-',F10.5,/,23X,'NCUR=',
     1I3,/,23X,'K=',F10.5,//)
DERIVATIVE
      SPDDER=20.25
      SPDDES=1.75*STEP(10.0)
      SPDDEL=DELAY(7,P,SPDDES)
      SPDIN=K*SPDDEL+K*SPDDER
      SPDERR=(SPDIN-SPDOUT)*G
      SPDOUT=INTGRL(UIC,SPDERR)
DYNAMIC
      IF(TIME.GT.9.0) P=4.88
SAMPLE
CONTRL  FINTIM=320.0,DELT=0.8,DELS=0.8
PRINT 1.6,SPDDES,SPDDEL,SPDIN,SPDERR,SPDOUT,P
      CALL DRWG(1,NCUR,TIME,SPDOUT)
TERMINAL
      CALL ENDRW(NPLOT)
END
STOP
//PLCT.SYSIN DD *
                                                                   5
0.0       40.0      20.0      0.4       8.0       5.0
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #4

This program models a simplified wave simulation composed of two superimposed sinusoids (fundamental and second harmonic) and a small random wave. The model is inherently scaled to the mariner nondimensional characteristics. Introduction of these waves is accomplished in computer program #7. Subroutine DEGRAD is shown in appendix A.

The sea state force plots in the dimensions of the three degrees of freedom produced by this run is shown in figures II-17 thru II-22.

COMPUTER PROGRAM #4

```
//UHRINTF4 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=2
// EXEC DSL
//DSL.INPUT DD *
*      WAVE PERTURBATION SIMULATION
TITLE WAVE PERTURBATION SIMULATION
INTGER NPLOT
PARAM NPLOT=6
PARAM YAWDP2=0.0
PARAM CDOT2=1.5
PARAM LUC=20.84765
PARAM WS=5.0
PARAM WD=015.0
PARAM WL=0.5
PARAM WFMA=0.1137
      ATIME=LUC*TIME
      WFI = (WFMA/(0.1137*40.0))*RAMP(0.0)
      WF=LIMIT(-WFMA,WFMA,WFI)
      WRV=NORMAL(1975,0.0,WFMA/10.0)
      WV=WS/15.0
      EWDD=WD-YAWDP2
      EWD=DEGRAD(1,1,EWDD)
      WEF=2.0*3.1415926*(CDOT2+WV*COS(EWD))/WL
      WE=WEF*TIME
      WFY=WF*SIN(EWD)
      WFN=WF*SIN(2.0*EWD)
      WFX=WF*COS(EWD)
      WYF=WFY*SIN(WE)+(3.1415926*WFY**2/WL)*SIN(2.0*WE)
      WNF=WFN*SIN(WE)+(3.1415926*WFN**2/WL)*SIN(2.0*WE)
      WXF=WFX*SIN(WE)+(3.1415926*WFX**2/WL)*SIN(2.0*WE)
      WY=WYF+WRV*WFY*SIN(WE)
      WN=WNF+WRV*WFN*SIN(WE)
      WX=WXF+WRV*WFX*SIN(WE)
      WED=DEGRAD(0,1,WE)
SAMPLE
PRINT 0.04,ATIME,WY,WYF,WN,WNF,WX,WXF,WED,WRV
PRPLOT ONLY
CONTRL FINTIM=5.0,DELT=0.04,DELS=0.04
      CALL DRWG(1,1,ATIME,WX)
      CALL DRWG(1,2,ATIME,WY)
      CALL DRWG(1,3,ATIME,WN)
TERMINAL
      WRITE(6,100) WS,WD,WL,WFMA,CDOT2
  100 FORMAT(//,' LAST RUN FOR WS=',F10.5,/,14X,'WD=',F10.5,/,14X,'WL=',
```

```
     1F10.5,/,14X,'WFMA=',F10.5,/,14X,'CDOT2=',F10.5,///)
      CALL ENDRW(NPLOT)
END
PARAM WL=1.0
END
PARAM WL=1.5
END
PARAM WD=030.0
PARAM WL=0.5
END
PARAM WL=1.0
END
PARAM WL=1.5
END
STOP
FORTRAN
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
//PLOT.SYSIN DD *
                                                                         4
                                          7.0       5.0
                                                                         4
                                          7.0       5.0
                                                                         4
                                          7.0       5.0
                                                                         4
                                          7.0       5.0
                                                                         4
                                          7.0       5.0
                                                                         4
                                          7.0       5.0
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #5

This program uses the mariner hull model first introduced in computer program #1 and the control system designed in chapter III to simulate the approach phase of RAS. The subroutines and functions that are to be inserted from appendix A can also be done in object code by changing the word FORTRAN to OBJECT and placing pre-compiled decks in the same locations. In fact, due to the long length of subroutine SLOPES, this must be done to be able to run the simulation with the DSL default job control language (JCL) presently installed at the Naval Postgraduate School IBM 360/67.

The plots produced by this run are shown in figures III-7 thru III-13. By changing the gains and introducing the following code, the plots of figures III-14 thru III-19 are produced:

```
D1DES = 5.0*STEP(8.0)-5.0*STEP(9.0)
```

COMPUTER PROGRAM #5

```
//UHRINTF5 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=4
// EXEC DSL
//DSL.INPUT DD *
*       RAS RUDDER CONTROL - APPROACH PHASE
TITLE RAS RUDDER CONTROL - APPROACH PHASE
INTEG RKSFX
INTGER NPLOT
CONST NPLOT=1
INTGER N,IS
CONST N=1,RD=1.0
CONST IS=1,DD=0.2
*       INSERT APPROPRIATE GAIN SETTINGS HERE
PARAM RSENS=1.86642,WTSENS=2.3869,RGN=23.4185,VFBG=4.35162
*       HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,IZNRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELR=-0.00126,XDELR=0.0
CONST DLTDM=2.0,DLTEM=7.0
*       INITIAL SEPERATION
INCON X01=5.0,Y01=0.0,X02=0.0,Y02=0.4
*       INITIAL CONDITIONS
INCON YAW01=0.0
INCON YY1=0.,YY2=0.,YN1=0.,YN2=0.
INCON U01=1.0,U02=1.5
INITIAL
      DY0=Y02-Y01
      DX0=X02-X01
      CALL TRANS(YAW01,DX0,DY0,ADX,ADY)
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
*     CALCULATION OF THE COEFFICIENTS
      NC1=-XU
      NC2=-XU
      A11=MYVD
      B11=-YV
      A21=-YRD
      B21=MYR
      A12=-NVD
      B12=-NV
      A22=IZNRD
      B22=-NR
      A33=MXUD
      B33=-XU
```

```
      KA1=-YDELR
      KB1=NDELR
      KC1=XDELR
      D=A11*A22-A12*A21
      DELRM=41.6953
      RDC=180./3.1415926
      CRC=3.1415926/180.
      LUC=20.84765
      KG=DLTDM/DLTEM
      D2D=0.0
DERIVATIVE
*     REFEREMCE SHIP RUDDER CONTROL
      D1DES=0.0
      DLTS1=LIMIT(-30.0,30.0,D1DES)
      DLTE1=DLTS1-D1D
      DLTBE1=LIMIT(-DLTEM,DLTEM,DLTE1)
      D1D=INTGRL(D1DIC,KG*DLTBE1*LUC)
      D1=DEGRAD(1,1,D1D)
      DX=X2-X1
      DY=Y2-Y1
*     SIMULATION SHIP A
      IF11=KA1*D1
      IF21=KB1*D1
      IF31=KC1*D1+NC1
      I11=-B11*ADOT1-B21*BDOT1+IF11
      I21=-B12*ADOT1-B22*BDOT1+IF21
      I31=-B33*CDOT1+IF31
      ADDOT1=(I11*A22-I21*A21)/D
      BDDOT1=(I21*A11-I11*A12)/D
      CDDOT1=I31/A33
      ADOT1=INTGRL(0.,ADDOT1)
      BDOT1=INTGRL(0.,BDDOT1)
      CDOT1=INTGRL(U01,CDDOT1)
      BY1=INTGRL(0.,BDOT1)
      XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
      YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
      X1=INTGRL(X01,XDOT1)
      Y1=INTGRL(Y01,YDOT1)
      YAW1=BY1
      SWAY1=Y1
      SURGE1=X1
*     SIMULATION SHIP B
      IF12=KA1*D2+YY2
      IF22=KB1*D2+YN2
      IF32=KC1*D2+NC2
      I12=-B11*ADOT2-B21*BDOT2+IF12
      I22=-B12*ADOT2-B22*BDOT2+IF22
      I32=-B33*CDOT2+IF32
```

```
      ADDOT2=(I12*A22-I22*A21)/D
      BDDOT2=(I22*A11-I12*A12)/D
      CDDOT2=I32/A33
      ADOT2=INTGRL(0.,ADDOT2)
      BDOT2=INTGRL(0.,BDDOT2)
      CDOT2=SPDCTR(ADX,U01,U02)
      BY2=INTGRL(BY02,BDOT2)
      XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
      YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
      X2=INTGRL(X02,XDOT2)
      Y2=INTGRL(Y02,YDOT2)
      YAW2=BY2
      SWAY2=Y2
      SURGE2=X2
NOSORT
      YAWD1=DEGRAD(0,0,YAW1)
      YAWDP1=DEGRAD(0,1,YAW1)
      YAWD2=DEGRAD(0,0,YAW2)
      YAWDP2=DEGRAD(0,1,YAW2)
*     RUDDER RESPONSE INPUT
      CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
      CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...
      PSIDED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)
      BDOT2D=DEGRAD(0,1,BDOT2)
      BDOTFB=VFBG*BDOT2D
      DDUMB=YAWD2-PSIDED+BDOTFB
      IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
      IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
      DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
      DLTE=DLTS-D2D
      DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
      D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
      D2=DEGRAD(1,1,D2D)
SORT
*     RUDDER PART OF OBJECT FUNCTION
      DTRAN=TIME*ABS(D2)
      ROBJ=INTGRL(0.0,DTRAN)
*     DISTANCE PART OF OBJECT FUNCTION
      DISTE=TIME*10.0*ABS(DD-ADY)
      DOBJ=INTGRL(0.0,DISTE)
*     OBJECT FUNCTION
      OBJ=ROBJ+DOBJ
DYNAMIC REGION
*     ACTUAL SEPARATION
      DX=X2-X1
      DY=Y2-Y1
      CALL TRANS(YAW1,DX,DY,ADX,ADY)
*     EXTERNAL FORCES ACTING BETWEEN SHIPS
```

```
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
      IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
  100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
*     ACTUAL TIME CONVERSION (SEC)
      ATIME=LUC*TIME
SAMPLE
CONTFL FINTIM=20.,DELT=0.04,DELS=0.04
PRINT 0.04,X1,X2,DA,A1D,PSIDFD,Y1,Y2,R1,B1D,PSIADD,YAWD1,YAWD2,R2,...
      B2D,PSIDED,ATIME,D2D,ADX,D1D,YY2,BDOTFB,DLTS,ADY,DLTS1,...
      YN2,CDOT1,CDOT2,OBJ
PRPLOT CNLY
      CALL DRWG(1,1,ATIME,YAWDP2)
      CALL DRWG(1,2,ATIME,YAWDP1)
      CALL DRWG(2,1,ATIME,YY2)
      CALL DRWG(3,1,ATIME,YN2)
      CALL DRWG(4,1,SURGE2,SWAY2)
      CALL DRWG(4,2,SURGE1,SWAY1)
      CALL DRWG(5,1,ATIME,ADY)
      CALL DRWG(6,1,ATIME,DLTS)
      CALL DRWG(6,2,ATIME,D2D)
TERMINAL
      IF(IS.EQ.1) WRITE(6,101)
  101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
      IF(IS.EC.0) WRITE(6,102)
  102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
      CALL ENDRW(NPLOT)
END
STOP
FORTRAN
INSERT FUNCTION SPDCTR FROM APPENDIX A HERE
INSERT FUNCTICN DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FRCM APPENDIX A HERE
//G.FT06F001 DD SYSOUT=O,SPACE=(CYL,(4,1))
//PLCT.SYSIN DD *
                                                                4
                                   7.0       5.0
                                                                4
                                   7.0       5.0
                                                                4
                                   7.0       5.0
                                                                7
```

```
0.0       0.1                          7.0       5.0
                                       7.0       5.0                        4
                                       7.0       5.0                        4
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #6

This program combines the approach and turn phases of computer program #5. The added subroutine is a result of simulation requirements to switch between adaptive gains.

This run produced the plots of figures III-22 thru III-34. By substituting the initial conditions of table III-3, this program produced the plots of figures III-35 thru III-64.

COMPUTER PROGRAM #6

```
//UHRINTF6 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
// EXEC DSL
//DSL.INPUT DD *
*      RAS RUDDER CONTROL - APPROACH PHASE FOLLOWED BY TURN PHASE
TITLE RAS RUDDER CONTROL - APPROACH PHASE FOLLOWED BY TURN PHASE
INTEG RKSFX
INTGER NFLOT
CONST NPLOT=2
INTGER N,IS
CONST N=1,RD=1.0
*      SET IS FOR SIDE OF APPROACH IS=1 PORT, IS=0 STBD
*      SET DD FOR DESIRED FINAL LATERAL SEPARATION DESIRED
CONST IS=1,DD=0.2
PARAM RSENS=1.86642
PARAM WTSENS=2.38692
PARAM RGN=23.41847
PARAM VFBG=4.35162
PARAM TSTP1=35.0,TSTP2=36.0
*      HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,IZNRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELR=-0.00126,XDELR=0.0
CONST DLTDM=2.0,DLTEM=7.0
*      INITIAL SEPERATION
*      SET IC FOR APPROACH TESTING
INCON X01=5.0,Y01=0.0,X02=0.0,Y02=0.4
*      INITIAL CONDITIONS
INCCN YAW01=0.0
INCON YY1=0.,YY2=0.,YN1=0.,YN2=0.
INCON U01=1.0,U02=1.5
INITIAL
      DY0=Y02-Y01
      DX0=X02-X01
      CALL TRANS(YAW01,DX0,DY0,ADX,ADY)
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
*      CALCULATION OF THE COEFFICIENTS
      NC1=-XU
      NC2=-XU
      A11=MYVD
      B11=-YV
      A21=-YRD
      B21=MYR
```

```
      A12=-NVD
      B12=-NV
      A22=IZNRD
      B22=-NR
      A33=MXUD
      B33=-XU
      KA1=-YDELR
      KB1=NDELR
      KC1=XDELR
      D=A11*A22-A12*A21
      DELRM=41.6953
      RDC=180./3.1415926
      DRC=3.1415926/180.
      LUC=20.84765
      KG=DLTDM/DLTEM
      D1DIC=0.0
      D2=DEGRAD(1,1,D2D)
      BY02=0.0
DERIVATIVE
      D1DES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
      DLTS1=LIMIT(-30.0,30.0,D1DES)
      DLTE1=DLTS1-D1D
      DLTBE1=LIMIT(-DLTEM,DLTEM,DLTE1)
      D1D=INTGRL(D1DIC,KG*DLTBE1*LUC)
      D1=DEGRAD(1,1,D1D)
      DX=X2-X1
      DY=Y2-Y1
*     SIMULATION SHIP A
      IF11=KA1*D1
      IF21=KB1*D1
      IF31=KC1*D1+NC1
      I11=-B11*ADOT1-B21*BDOT1+IF11
      I21=-B12*ADOT1-B22*BDOT1+IF21
      I31=-B33*CDOT1+IF31
      ADDOT1=(I11*A22-I21*A21)/D
      BDDOT1=(I21*A11-I11*A12)/D
      CDDOT1=I31/A33
      ADOT1=INTGRL(0.,ADDOT1)
      BDOT1=INTGRL(0.,BDDOT1)
      CDOT1=INTGRL(U01,CDDOT1)
      BY1=INTGRL(0.,BDOT1)
      XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
      YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
      X1=INTGRL(X01,XDOT1)
      Y1=INTGRL(Y01,YDOT1)
      YAW1=BY1
      SWAY1=Y1
      SURGE1=X1
```

```
*      SIMULATION SHIP B
       IF12=KA1*D2+YY2
       IF22=KB1*D2+YN2
       IF32=KC1*D2+NC2
       I12=-B11*ADOT2-B21*BDOT2+IF12
       I22=-B12*ADOT2-B22*BDOT2+IF22
       I32=-B33*CDOT2+IF32
       ADDOT2=(I12*A22-I22*A21)/D
       BDDOT2=(I22*A11-I12*A12)/D
       CDDOT2=I32/A33
       ADOT2=INTGRL(0.,ADDOT2)
       BDOT2=INTGRL(0.,BDDOT2)
       CDOT2=SPDCTR(ADX,U01,U02)
       BY2=INTGRL(BY02,BDOT2)
       XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
       YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
       X2=INTGRL(X02,XDOT2)
       Y2=INTGRL(Y02,YDOT2)
       YAW2=BY2
       SWAY2=Y2
       SURGE2=X2
NOSORT
       YAWD1=DEGRAD(0,0,YAW1)
       YAWDP1=DEGRAD(0,1,YAW1)
       YAWD2=DEGRAD(0,0,YAW2)
       YAWDP2=DEGRAD(0,1,YAW2)
*      RUDDER RESPONSE INPUT
       CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
       CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...
       PSIDED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)
       BDOT2D=DEGRAD(0,1,BDOT2)
       BDOTFB=VFBG*BDOT2D
       DDUMB=YAWD2-PSIDED+BDOTFB
       IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
       IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
       DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
       DLTE=DLTS-D2D
       DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
       D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
       D2=DEGRAD(1,1,D2D)
SORT
       DISTE=ABS(DD-ADY)
       OBJ=INTGRL(0.0,DISTE)
DYNAMIC REGION
*      ACTUAL SEPARATION
       DX=X2-X1
       DY=Y2-Y1
       CALL TRANS(YAW1,DX,DY,ADX,ADY)
```

```
*     EXTERNAL FORCES ACTING BETWEEN SHIPS
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
      IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
  100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
*     ACTUAL TIME CONVERSION (SEC)
      ATIME=LUC*TIME
      AA1=(BB1+BB2)/2.0
      CALL SWTCH(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D)
SAMPLE
CONTRL FINTIM=30.,DELT=0.04,DELS=0.04
PRINT 0.20,X1,X2,DA,A1D,PSIDFD,Y1,Y2,R1,B1D,PSIADD,YAWD1,YAWD2,R2,...
      B2D,PSIDED,ATIME,D2D,ADX,D1D,YY2,BDOTFB,DLTS,ADY,DLTS1,...
      YN2,CDOT1,CDOT2,OBJ,RSENS,VFBG
PRPLCT CNLY
      CALL DRWG(1,1,ATIME,YAWDP2)
      CALL DRWG(1,2,ATIME,YAWDP1)
      CALL DRWG(2,1,ATIME,YY2)
      CALL DRWG(3,1,ATIME,YN2)
      CALL DRWG(4,1,SURGE2,SWAY2)
      CALL DRWG(4,2,SURGE1,SWAY1)
      CALL DRWG(5,1,ATIME,ADY)
      CALL DRWG(6,1,ATIME,DLTS)
      CALL DRWG(6,2,ATIME,D2D)
TERMINAL
      IF(IS.EQ.1) WRITE(6,101)
  101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
      IF(IS.EQ.0) WRITE(6,102)
  102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
      CALL ENDRW(NPLOT)
      CALL CONTIN
      FINTIM=45.0
END
STOP
FORTRAN
INSERT SUBROUTINE SWTCH FROM APPENDIX A HERE
INSERT FUNCTION SPDCTR FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//FLCT.SYSIN DD *
                                                                  4
                                 7.0       5.0
                                                                  4
                                 7.0       5.0
```

```
                                        7.0      5.0                        4
0.0       0.1                           7.0      5.0                        7
                                        7.0      5.0                        4
                                        7.0      5.0                        4
                                        7.0      5.0                        4
                                        7.0      5.0                        4
                                        7.0      5.0                        4
-2.0      0.5                           7.0      5.0                        7
                                        7.0      5.0                        4
                                        7.0      5.0                        4
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #7

This program combines the calm sea simulation of computer program #6 with the wave simulation of computer program #4 to simulate the model and control system in a sea state. The waves are introduced thru the rudder nondimensionalized coefficients as shown in chapter II.

The plots produced are shown in figures III-66 thru III-73. Figure III-65 was produced with the same program by setting WI=1.5.

COMPUTER PROGRAM #7

```
//UHRINTF7 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
// EXEC DSL
//DSL.INPUT DD *
*       RAS RUDDER CONTROL - SIMULATION WITH WAVE PERTURBATIONS
TITLE RAS RUDDER CONTROL - SIMULATION WITH WAVE PERTURBATIONS
INTEG RKSFX
INTGER NPLOT
CONST NPLOT=2
INTGER N,IS
CONST N=1,RD=1.0
PARAM IS=1,DD=0.2
PARAM RSENS=1.86642
PARAM WTSENS=2.38692
PARAM RGN=23.41847
PARAM VFBG=4.35162
PARAM TSTP1=35.0,TSTP2=36.0
*       HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,IZNRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELR=-0.00126,XDELR=0.0
CONST DLTDM=2.0,DLTEM=7.0
*       INITIAL SEPERATION
INCON X01=5.0,Y01=0.0,X02=0.0,Y02=0.4
*       INITIAL CONDITIONS
INCON YAW01=0.0
INCON YY1=0.,YY2=0.,YN1=0.,YN2=0.
INCON U01=1.0,U02=1.5
PARAM WS=5.0
PARAM WD=-015.0
PARAM WL=1.0
PARAM WFMA=0.05685
INITIAL
      DY0=Y02-Y01
      DX0=X02-X01
      CALL TRANS(YAW01,DX0,DY0,ADX,ADY)
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
*     CALCULATION OF THE COEFFICIENTS
      NC1=-XU
      NC2=-XU
      A11=MYVD
      B11=-YV
      A21=-YRD
```

```
      B21=MYR
      A12=-NVD
      B12=-NV
      A22=IZNRD
      B22=-NR
      A33=MXUD
      B33=-XU
      KA1=-YDELR
      KB1=NDELR
      KC1=XDELR
      D=A11*A22-A12*A21
      DELRM=41.6953
      RDC=180./3.1415926
      DRC=3.1415926/180.
      LUC=20.84765
      KG=DLTDM/DLTEM
      D1DIC=0.0
      D2=DEGRAD(1,1,D2D)
      BYO2=0.0
DERIVATIVE
      WFI = (WFMA/(0.1137*40.0))*RAMP(0.0)
      WF=LIMIT(-WFMA,WFMA,WFI)
      WRV=NORMAL(1975,0.0,WFMA/10.0)
      WV=WS/15.0
      EWDD=WD-YAWDP2
      EWD=DEGRAD(1,1,EWDD)
      WEF=2.0*3.1415926*(CDOT2+WV*COS(EWD))/WL
      WE=WEF*TIME
      WFY=WF*SIN(EWD)
      WFN=WF*SIN(2.0*EWD)
      WFX=WF*COS(EWD)
      WYF=WFY*SIN(WE)+(3.1415926*WFY**2/WL)*SIN(2.0*WE)
      WNF=WFN*SIN(WE)+(3.1415926*WFN**2/WL)*SIN(2.0*WE)
      WXF=WFX*SIN(WE)+(3.1415926*WFX**2/WL)*SIN(2.0*WE)
      WY=WYF+WRV*WFY*SIN(WE)
      WN=WNF+WRV*WFN*SIN(WE)
      WX=WXF+WRV*WFX*SIN(WE)
      D1DES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
      DLTS1=LIMIT(-30.0,30.0,D1DES)
      DLTE1=DLTS1-D1D
      DLTBE1=LIMIT(-DLTEM,DLTEM,DLTE1)
      D1D=INTGRL(D1DIC,KG*DLTBE1*LUC)
      D1=DEGRAD(1,1,D1D)
      DX=X2-X1
      DY=Y2-Y1
*     SIMULATION SHIP A
      IF11=KA1*D1
      IF21=KB1*D1
```

```
      IF31=KC1*D1+NC1
      I11=-B11*ADOT1-B21*BDOT1+IF11
      I21=-B12*ADOT1-B22*BDOT1+IF21
      I31=-B33*CDOT1+IF31
      ADDOT1=(I11*A22-I21*A21)/D
      BDDOT1=(I21*A11-I11*A12)/D
      CDDOT1=I31/A33
      ADOT1=INTGRL(O.,ADDOT1)
      BDOT1=INTGRL(O.,BDDOT1)
      CDOT1=INTGRL(U01,CDDOT1)
      BY1=INTGRL(O.,BDOT1)
      XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
      YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
      X1=INTGRL(X01,XDOT1)
      Y1=INTGRL(Y01,YDOT1)
      YAW1=BY1
      SWAY1=Y1
      SURGE1=X1
*     SIMULATION SHIP B
      IF12=KA1*D2+YY2+KA1*WY
      IF22=KB1*D2+YN2+KB1*WN
      IF32=KC1*D2+NC2+KC1*WX
      I12=-B11*ADOT2-B21*BDOT2+IF12
      I22=-B12*ADOT2-B22*BDOT2+IF22
      I32=-B33*CDOT2+IF32
      ADDOT2=(I12*A22-I22*A21)/D
      BDDOT2=(I22*A11-I12*A12)/D
      CDDOT2=I32/A33
      ADOT2=INTGRL(O.,ADDOT2)
      BDOT2=INTGRL(O.,BDDOT2)
      CDOT2=SFDCTR(ADX,U01,U02)
      BY2=INTGRL(BY02,BDOT2)
      XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
      YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
      X2=INTGRL(X02,XDOT2)
      Y2=INTGRL(Y02,YDOT2)
      YAW2=BY2
      SWAY2=Y2
      SURGE2=X2
NOSORT
      YAWD1=DEGRAD(0,0,YAW1)
      YAWDP1=DEGRAD(0,1,YAW1)
      YAWD2=DEGRAD(0,0,YAW2)
      YAWDP2=DEGRAD(0,1,YAW2)
      YAWDPD=YAWDP1-YAWDP2
*     RUDDER RESPONSE INPUT
      CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
      CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...
```

```
       PSIDED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)
       BDOT2D=DEGRAD(0,1,BDOT2)
       BCOTFB=VFBG*BDOT2D
       DDUMB=YAWD2-PSIDED+BDOTFB
       IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
       IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
       DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
       DLTE=DLTS-D2D
       DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
       D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
       D2=DEGRAD(1,1,D2D)
SORT
       DISTE=ABS(DD-ADY)
       CBJ=INTGRL(0.0,DISTE)
DYNAMIC REGION
*      ACTUAL SEPARATION
       DX=X2-X1
       DY=Y2-Y1
       CALL TRANS(YAW1,DX,DY,ADX,ADY)
*      EXTERNAL FORCES ACTING BETWEEN SHIPS
       CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
       IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
  100  FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
*      ACTUAL TIME CONVERSION (SEC)
       ATIME=LUC*TIME
       AA1=(BB1+BB2)/2.0
       CALL SWTCH(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D)
SAMPLE
CONTRL FINTIM=30.,DELT=0.04,DELS=0.04
PRINT  0.20,X1,X2,DA,A1D,PSIDFD,Y1,Y2,R1,B1D,PSIADD,YAWD1,YAWD2,R2,...
       B2D,PSIDED,ATIME,D2D,ADX,D1D,YY2,BDOTFB,DLTS,ADY,DLTS1,...
       YN2,CDOT1,CDOT2,OBJ,RSENS,VFBG,WYF,WNF,WXF,WRV,EWDD,WY,WN,...
       WX,WE,WF
PRPLOT CNLY
       CALL DRWG(1,1,ATIME,YAWDP2)
       CALL DRWG(1,2,ATIME,YAWDP1)
       CALL DRWG(2,1,ATIME,ADY)
       CALL DRWG(3,1,ATIME,DLTS)
       CALL DRWG(3,2,ATIME,D2D)
       CALL DRWG(4,1,ATIME,YAWDPD)
       CALL DRWG(5,1,ATIME,WX)
       CALL DRWG(5,2,ATIME,WY)
       CALL DRWG(5,3,ATIME,WN)
TERMINAL
       IF(IS.EQ.1) WRITE(6,101)
  101  FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
       IF(IS.EQ.0) WRITE(6,102)
  102  FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
```

```
      CALL ENDRW(NPLOT)
      CALL CONTIN
      FINTIM=45.0
END
STOP
FORTRAN
INSERT SUBROUTINE SWTCH FROM APPENDIX A HERE
INSERT FUNCTION SPDCTR FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN DD *
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
                                                              4
                              7.0        5.0
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #8

This program incorporates a fifth order polynomial curve fit speed control switching function to give optimal longitudinal positioning. The scenario is the same that was used in the design of the heading control development. The low order model of the gas turbine propulsion plant was used.

The plots produced are shown in figures III-80 thru III-83.

COMPUTER PROGRAM #8

```
//UHRINTF8 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
// EXEC DSL
//DSL.INPUT DD *
*      RAS SPEED CONTROL - CONTROL TESTING
TITLE RAS SPEED CONTROL - CONTROL TESTING
INTEG RKSFX
INTGER NPLOT
CONST NPLOT=2
INTGER N,IS
CONST N=1,RD=1.0
PARAM IS=1,DD=0.2
PARAM RSENS=1.86642
PARAM WTSENS=2.38692
PARAM RGN=23.41847
PARAM VFBG=4.35162
PARAM TSTP1=35.0,TSTP2=36.0
*      HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,IZNRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELR=-0.00126,XDELR=0.0
CONST DLTDM=2.0,DLTEM=7.0
*      INITIAL SEPERATION
INCON XO1=5.0,YO1=0.0,XO2=0.0,YO2=0.4
*      INITIAL CONDITIONS
INCON YAWO1=0.0
INCON YY1=0.,YY2=0.,YN1=0.,YN2=0.
INCON UO1=1.0,UO2=1.5
PARAM UF=21.73,A=22.0,      G=0.092
INITIAL
      DYO=YO2-YO1
      DXO=XO2-XO1
      CALL TRANS(YAWO1,DXO,DYO,ADX,ADY)
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
*      CALCULATION OF THE COEFFICIENTS
      NC1=-XU
      NC2=-XU
      A11=MYVD
      B11=-YV
      A21=-YRD
      B21=MYR
      A12=-NVD
      B12=-NV
```

```
         A22=IZNRD
         B22=-NR
         A33=MXUD
         B33=-XU
         KA1=-YDELR
         KB1=NDELR
         KC1=XDELR
         D=A11*A22-A12*A21
         DELRM=41.6953
         RDC=180./3.1415926
         DRC=3.1415926/180.
         LUC=20.84765
         KG=DLTDM/DLTEM
         D1DIC=0.0
         D2=DEGRAD(1,1,D2D)
         BY02=0.0
         K=UF/A
         P=4.88/LUC
         SPDDER=U01/K
DERIVATIVE
         D1DES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
         DLTS1=LIMIT(-30.0,30.0,D1DES)
         DLTE1=DLTS1-D1D
         DLTBE1=LIMIT(-DLTEM,DLTEM,DLTE1)
         D1D=INTGRL(D1DIC,KG*DLTBE1*LUC)
         D1=DEGRAD(1,1,D1D)
         DX=X2-X1
         DY=Y2-Y1
*        SIMULATION SHIP A
         IF11=KA1*D1
         IF21=KB1*D1
         IF31=KC1*D1+NC1
         I11=-B11*ADOT1-B21*BDOT1+IF11
         I21=-B12*ADOT1-B22*BDOT1+IF21
         I31=-B33*CDOT1+IF31
         ADDOT1=(I11*A22-I21*A21)/D
         BDDOT1=(I21*A11-I11*A12)/D
         CDDOT1=I31/A33
         ADOT1=INTGRL(0.,ADDOT1)
         BDOT1=INTGRL(0.,BDDOT1)
         CDOT1=INTGRL(U01,CDDOT1)
         BY1=INTGRL(0.,BDOT1)
         XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
         YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
         X1=INTGRL(X01,XDOT1)
         Y1=INTGRL(Y01,YDOT1)
         YAW1=BY1
         SWAY1=Y1
```

```
      SURGE1=X1
*     SIMULATION SHIP B
      IF12=KA1*D2+YY2
      IF22=KB1*D2+YN2
      IF32=KC1*D2+NC2
      I12=-B11*ADOT2-B21*BDOT2+IF12
      I22=-B12*ADOT2-B22*BDOT2+IF22
      I32=-B33*CDOT2+IF32
      ADDOT2=(I12*A22-I22*A21)/D
      BDDOT2=(I22*A11-I12*A12)/D
      CDDOT2=I32/A33
      ADOT2=INTGRL(0.,ADDOT2)
      BDOT2=INTGRL(0.,BDDOT2)
      SW=SWCL(UO1,UO2)
      SPDDES=SPDREC(ADX,UO1,UO2,SW)
      SPDDEL=DELAY(7,P,(SPDDES/K-SPDDER))
      SPDEL=SPINIT(SPDDEL,TIME,(SPDDES/K-SPDDER))
      SPDIN=K*(SPDEL+SPDDER)
      SPDERR=(SPDIN-CDOT2)*G
      CDOT2=INTGRL(UO2,SPDERR*LUC)
      BY2=INTGRL(BYO2,BDOT2)
      XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
      YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
      X2=INTGRL(XO2,XDOT2)
      Y2=INTGRL(YO2,YDOT2)
      YAW2=BY2
      SWAY2=Y2
      SURGE2=X2
NOSORT
      YAWD1=DEGRAD(0,0,YAW1)
      YAWDP1=DEGRAD(0,1,YAW1)
      YAWD2=DEGRAD(0,0,YAW2)
      YAWDP2=DEGRAD(0,1,YAW2)
      YAWDPD=YAWDP1-YAWDP2
*     RUDDER RESPONSE INPUT
      CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
      CALL FDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...
      PSIDED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)
      BDOT2D=DEGRAD(0,1,BDOT2)
      BDOTFB=VFBG*BDOT2D
      DDUMB=YAWD2-PSIDED+BDOTFB
      IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
      IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
      DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
      DLTE=DLTS-D2D
      DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
      D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
      D2=DEGRAD(1,1,D2D)
```

```
SORT
      DISTE=ABS(DD-ADY)
      OBJ=INTGRL(0.0,DISTE)
      DISTES=ABS(ADX)
      OBJS=INTGRL(0.0,DISTES/25.0)
DYNAMIC REGION
*     ACTUAL SEPARATION
      DX=X2-X1
      DY=Y2-Y1
      CALL TRANS(YAW1,DX,DY,ADX,ADY)
*     EXTERNAL FORCES ACTING BETWEEN SHIPS
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
      IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
  100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
*     ACTUAL TIME CONVERSION (SEC)
      ATIME=LUC*TIME
      AA1=(BB1+BB2)/2.0
      CALL SWTCH(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D)
SAMPLE
CONTRL FINTIM=30.,DELT=0.04,DELS=0.04
PRINT 0.20,ATIME,ADX,ADY,YAWD1,YAWD2,YAWDPD,SPDDES,CDOT2
PRPLOT ONLY
      CALL DRWG(1,1,ATIME,ADX)
      CALL DRWG(2,1,ATIME,SPDDES)
      CALL DRWG(2,2,ATIME,CDOT2)
TERMINAL
      IF(IS.EQ.1) WRITE(6,101)
  101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
      IF(IS.EQ.0) WRITE(6,102)
  102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
      CALL ENDRW(NPLOT)
      CALL CONTIN
      FINTIM=45.0
END
STOP
FORTRAN
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE SWTCH FROM APPENDIX A HERE
INSERT FUNCTION SWCL FROM APPENDIX A HERE
INSERT FUNCTION SPDREC FROM APPENDIX A HERE
INSERT FUNCTION SPINIT FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN DD *
```

4

```
                                         7.0     5.0
                                                                 4
                                         7.0     5.0
                                                                 4
                                         7.0     5.0
                                                                 4
                                         7.0     5.0
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #9

This program introduces a longitudinal position offset capability. The method takes the control ship to the alongside position until 450 seconds into the run. After that time, with the ship steadied, the offset position desired (XOFSD) is switched to the desired offset. This method negates some of the transient oscillations which cause unstable conditions in the approach phase. A secondary change is the use of subroutine SWTCHF instead of subroutine SWTCH developed in the heading control section. This new subroutine relaxes the heading velocity feedback gain (VFBG) to allow turn stability in the turn phase.

The plots produced by this program are shown in figures III-84 thru III-101.

COMPUTER PROGRAM #9

```
//UHRINTF9 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
// EXEC DSL
//DSL.INPUT DD *
*      RAS SPEED CONTROL - OFFSET TESTING
TITLE RAS SPEED CONTROL - OFFSET TESTING
INTEG RKSFX
INTGER NPLOT
CONST NPLOT=2
INTGER N,IS
CONST N=1,RD=1.0
PARAM IS=1,DD=0.2
PARAM RSENS=1.86642
PARAM WTSENS=2.38692
PARAM RGN=23.41847
PARAM VFBG=4.35162
PARAM TSTP1=35.0,TSTP2=36.0
*      HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,IZNRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELR=-0.00126,XDELR=0.0
CONST DLTDM=2.0,DLTEM=7.0
*      INITIAL SEPERATION
INCON X01=5.0,Y01=0.0,X02=0.0,Y02=0.4
*      INITIAL CONDITIONS
INCON YAW01=0.0
INCON YY1=0.,YY2=0.,YN1=0.,YN2=0.
INCON U01=1.0,U02=1.5
*      XOFS IS THE DESIRED NORMALIZED X POSITION / 0.0 IS ALONGSIDE /
*         NEG FOR ASTERN OF ALONGSIDE / POS FOR FWD OF ALONGSIDE
*      NOTE - XOFS CAN BE USED FOR VEHICLE OF APPRCACH AND EXIT FROM
*         STATION BY SETTING SOME DESIRED POSITION EG. -5.0 , 5.0
*         THIS WOULD CAUSE APPROACH OR BREAK AWAY STATION ATTAINMENT
*         WHICH CAN BE PRECEEDED BY OR FOLLOWED BY NON RAS POSITION
*         CONTROL
PARAM XOFS=0.0
PARAM XCFSD=0.0
PARAM UF=21.73,A=22.0,      G=0.092
INITIAL
      DY0=Y02-Y01
      DX0=X02-X01
      CALL TRANS(YAW01,DX0,DY0,ADX,ADY)
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
```

```
*       CALCULATION OF THE COEFFICIENTS
        NC1=-XU
        NC2=-XU
        A11=MYVD
        B11=-YV
        A21=-YRD
        B21=MYR
        A12=-NVD
        B12=-NV
        A22=IZNRD
        B22=-NR
        A33=MXUD
        B33=-XU
        KA1=-YDELR
        KB1=NDELR
        KC1=XDELR
        D=A11*A22-A12*A21
        DELRM=41.6953
        RDC=180./3.1415926
        DRC=3.1415926/180.
        LUC=20.84765
        KG=DLTDM/DLTEM
        D1DIC=0.0
        D2=DEGRAD(1,1,D2D)
        BY02=0.0
        K=UF/A
        P=4.88/LUC
        SPD01=1.0
        SPD02=1.5
        SPDDER=SPD01/K
DERIVATIVE
        D1DES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
        DLTS1=LIMIT(-30.0,30.0,D1DES)
        DLTE1=DLTS1-D1D
        DLTBE1=LIMIT(-DLTEM,DLTEM,DLTE1)
        D1D=INTGRL(D1DIC,KG*DLTBE1*LUC)
        D1=DEGRAD(1,1,D1D)
        DX=X2-X1
        DY=Y2-Y1
*       SIMULATION SHIP A
        IF11=KA1*D1
        IF21=KB1*D1
        IF31=KC1*D1+NC1
        I11=-B11*ADOT1-B21*BDOT1+IF11
        I21=-B12*ADOT1-B22*BDOT1+IF21
        I31=-B33*CDOT1+IF31
        ADDOT1=(I11*A22-I21*A21)/D
        BDDOT1=(I21*A11-I11*A12)/D
```

```
      CDDOT1=I31/A33
      ADOT1=INTGRL(0.,ADDOT1)
      BDOT1=INTGRL(0.,BDDOT1)
      CDOT1=INTGRL(U01,CDDOT1)
      BY1=INTGRL(0.,BDOT1)
      XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
      YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
      X1=INTGRL(X01,XDOT1)
      Y1=INTGRL(Y01,YDOT1)
      YAW1=BY1
      SWAY1=Y1
      SURGE1=X1
*     SIMULATION SHIP B
      IF12=KA1*D2+YY2
      IF22=KB1*D2+YN2
      IF32=KC1*D2+NC2
      I12=-B11*ADOT2-B21*BDOT2+IF12
      I22=-B12*ADOT2-B22*BDOT2+IF22
      I32=-B33*CDOT2+IF32
      ADDOT2=(I12*A22-I22*A21)/D
      BDDOT2=(I22*A11-I12*A12)/D
      CDDOT2=I32/A33
      ADOT2=INTGRL(0.,ADDOT2)
      BDOT2=INTGRL(0.,BDDOT2)
      SW=SWCL(SPD01,SPD02)
      SPDDES = SPDOFC(ADX,SPD01,SPD02,SW,XOFS)
      SPDDEL=DELAY(7,P,(SPDDES/K-SPDDER))
      SPDEL=SPINIT(SPDDEL,TIME,(SPDDES/K-SPDDER))
      SPDIN=K*(SPDEL+SPDDER)
      SPDERR=(SPDIN-CDOT2)*G
      CDOT2=INTGRL(U02,SPDERR*LUC)
      BY2=INTGRL(BY02,BDOT2)
      XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
      YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
      X2=INTGRL(X02,XDOT2)
      Y2=INTGRL(Y02,YDOT2)
      YAW2=BY2
      SWAY2=Y2
      SURGE2=X2
NOSORT
      YAWD1=DEGRAD(0,0,YAW1)
      YAWDP1=DEGRAD(0,1,YAW1)
      YAWD2=DEGRAD(0,0,YAW2)
      YAWDP2=DEGRAD(0,1,YAW2)
      YAWDPD=YAWDP1-YAWDP2
*     RUDDER RESPONSE INPUT
      CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
      CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...
```

```
      PSIDED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)
      BDOT2D=DEGRAD(0,1,BDOT2)
      BDOTFB=VFBG*BDOT2D
      DDUMB=YAWD2-PSIDED+BDOTFB
      IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
      IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
      DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
      DLTE=DLTS-D2D
      DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
      D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
      D2=DEGRAD(1,1,D2D)
SORT
      DISTE=ABS(DD-ADY)
      OBJ=INTGRL(0.0,DISTE)
      DISTES=ABS(ADX)
      OBJS=INTGRL(0.0,DISTES/25.0)
DYNAMIC REGION
*     ACTUAL SEPARATION
      DX=X2-X1
      DY=Y2-Y1
      CALL TRANS(YAW1,DX,DY,ADX,ADY)
*     EXTERNAL FORCES ACTING BETWEEN SHIPS
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
      IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
  100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
*     ACTUAL TIME CONVERSION (SEC)
      ATIME=LUC*TIME
      AA1=(BB1+BB2)/2.0
      CALL SWTCHF(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D,XOFS)
      IF (ATIME.GT.450.0) XOFS=XOFSD
SAMPLE
CONTRL FINTIM=30.,DELT=0.04,DELS=0.04
PRINT 0.20,ATIME,ADX,ADY,YAWD1,YAWD2,YAWDPD,SPDDES,CDOT2
PRPLOT ONLY
      CALL DRWG(1,1,ATIME,ADY)
      CALL DRWG(2,1,ATIME,YAWDPD)
      CALL DRWG(3,1,ATIME,SPDDES)
      CALL DRWG(3,2,ATIME,CDOT2)
      CALL DRWG(4,1,ATIME,ADX)
TERMINAL
      IF(IS.EQ.1) WRITE(6,101)
  101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
      IF(IS.EQ.0) WRITE(6,102)
  102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
      CALL ENDRW(NPLOT)
      CALL CONTIN
      FINTIM=45.0
END
```

```
STOP
FORTRAN
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE SWTCHF FROM APPENDIX A HERE
INSERT FUNCTION SWCL FROM APPENDIX A HERE
INSERT FUNCTION SPDOFC FROM APPENDIX A HERE
INSERT FUNCTION SPINIT FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN DD *
                                                                      4
                                         7.0       5.0
                                                                      4
                                         7.0       5.0
                                                                      4
                                         7.0       5.0
                                                                      4
                                         7.0       5.0
                                                                      4
                                         7.0       5.0
                                                                      4
                                         7.0       5.0
                                                                      4
                                         7.0       5.0
                                                                      4
                                         7.0       5.0
INSERT TWO /* CARDS HERE
```

COMPUTER PROGRAM #10

This program incorporates the sea state first programmed in computer programs #4 and #7. The WX wave force, however, is introduced at the end of the propulsion loop to allow more realistic perturbations. This is the final form of the complete heading and speed control systems. To run this without a sea state, set WFMA to 0.0.

This program produced the plots shown in figures III-103 thru III-105.

COMPUTER PROGRAM #10

```
//UHRINTFO JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=5
// EXEC DSL
//DSL.INPUT DD *
*      RAS SPEED CONTROL - WAVE INTERACTION
TITLE RAS SPEED CONTROL - WAVE INTERACTION
INTEG RKSFX
INTGER NPLOT
CONST NPLOT=2
INTGER N,IS
CONST N=1,RD=1.0
PARAM IS=1,DD=0.2
PARAM RSENS=1.86642
PARAM WTSENS=2.38692
PARAM RGN=23.41847
PARAM VFBG=4.35162
PARAM TSTP1=35.0,TSTP2=36.0
*      HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,IZNRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELR=-0.00126,XDELR=0.0
CONST DLTDM=2.0,DLTEM=7.0
*      INITIAL SEPERATION
INCON XO1=5.0,YO1=0.0,XO2=0.0,YO2=0.4
*      INITIAL CONDITIONS
INCON YAWO1=0.0
INCON YY1=0.,YY2=0.,YN1=0.,YN2=0.
INCON UO1=1.0,UO2=1.5
*      XOFS IS THE DESIRED NORMALIZED X POSITION / 0.0 IS ALONGSIDE /
*         NEG FOR ASTERN OF ALONGSIDE / POS FOR FWD OF ALONGSIDE
*      NOTE - XOFS CAN BE USED FOR VEHICLE OF APPROACH AND EXIT FROM
*         STATION BY SETTING SOME DESIRED POSITION EG. -5.0 , 5.0
*         THIS WOULD CAUSE APPROACH OR BREAK AWAY STATION ATTAINMENT
*         WHICH CAN BE PRECEEDED BY OR FOLLOWED BY NON RAS POSITION
*         CONTROL
PARAM XOFS=0.0
PARAM XOFSD=0.0
PARAM UF=21.73,A=22.0,      G=0.092
PARAM WS=5.0
PARAM WD=-015.0
PARAM WL=1.0
PARAM WFMA=0.05685
PARAM KS2=-1.0
```

```
INITIAL
      DYO=YO2-YO1
      DXO=XO2-XO1
      CALL TRANS(YAWO1,DXO,DYO,ADX,ADY)
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
*     CALCULATION OF THE COEFFICIENTS
      NC1=-XU
      NC2=-XU
      A11=MYVD
      B11=-YV
      A21=-YRD
      B21=MYR
      A12=-NVD
      B12=-NV
      A22=IZNRD
      B22=-NR
      A33=MXUD
      B33=-XU
      KA1=-YDELR
      KB1=NDELR
      KC1=XDELR
      D=A11*A22-A12*A21
      DELRM=41.6953
      RDC=180./3.1415926
      DRC=3.1415926/180.
      LUC=20.84765
      KG=DLTDM/DLTEM
      D1DIC=0.0
      D2=DEGRAD(1,1,D2D)
      BYO2=0.0
      K=UF/A
      P=4.88/LUC
      SPDO1=1.0
      SPDO2=1.5
      SPDDER=SPDO1/K
DERIVATIVE
      WFI = (WFMA/(0.1137*40.0))*RAMP(0.0)
      WF=LIMIT(-WFMA,WFMA,WFI)
      WRV=NORMAL(1975,0.0,WFMA/10.0)
      WV=WS/15.0
      EWDD=WD-YAWDP2
      EWD=DEGRAD(1,1,EWDD)
      WEF=2.0*3.1415926*(CDOT2+WV*COS(EWD))/WL
      WE=WEF*TIME
      WFY=WF*SIN(EWD)
      WFN=WF*SIN(2.0*EWD)
      WFX=WF*COS(EWD)
      WYF=WFY*SIN(WE)+(3.1415926*WFY**2/WL)*SIN(2.0*WE)
```

```
      WNF=WFN*SIN(WE)+(3.1415926*WFN**2/WL)*SIN(2.0*WE)
      WXF=WFX*SIN(WE)+(3.1415926*WFX**2/WL)*SIN(2.0*WE)
      WY=WYF+WRV*WFY*SIN(WE)
      WN=WNF+WRV*WFN*SIN(WE)
      WX=WXF+WRV*WFX*SIN(WE)
      D1DES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
      DLTS1=LIMIT(-30.0,30.0,D1DES)
      DLTE1=DLTS1-D1D
      DLTBE1=LIMIT(-DLTEM,DLTEM,DLTE1)
      D1D=INTGRL(D1DIC,KG*DLTBE1*LUC)
      D1=DEGRAD(1,1,D1D)
      DX=X2-X1
      DY=Y2-Y1
*     SIMULATION SHIP A
      IF11=KA1*D1
      IF21=KB1*D1
      IF31 = KC1*ABS(D1)+NC1
      I11=-B11*ADOT1-B21*BDOT1+IF11
      I21=-B12*ADOT1-B22*BDOT1+IF21
      I31=-B33*CDOT1+IF31
      ADDOT1=(I11*A22-I21*A21)/D
      BDDOT1=(I21*A11-I11*A12)/D
      CDDOT1=I31/A33
      ADOT1=INTGRL(0.,ADDOT1)
      BDOT1=INTGRL(0.,BDDOT1)
      CDOT1=INTGRL(U01,CDDOT1)
      BY1=INTGRL(0.,BDOT1)
      XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
      YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
      X1=INTGRL(X01,XDOT1)
      Y1=INTGRL(Y01,YDOT1)
      YAW1=BY1
      SWAY1=Y1
      SURGE1=X1
*     SIMULATION SHIP B
      IF12=KA1*D2+YY2+KA1*WY
      IF22=KB1*D2+YN2+KB1*WN
      IF32 = KC1*ABS(D2)+NC2
      I12=-B11*ADOT2-B21*BDOT2+IF12
      I22=-B12*ADOT2-B22*BDOT2+IF22
      ADDOT2=(I12*A22-I22*A21)/D
      BDDOT2=(I22*A11-I12*A12)/D
      ADOT2=INTGRL(0.,ADDOT2)
      BDOT2=INTGRL(0.,BDDOT2)
      SW=SWCL(SPD01,SPD02)
      SPDDES = SPDOFC(ADX,SPD01,SPD02,SW,XOFS)
      SPDDEL=DELAY(7,P,(SPDDES/K-SPDDER))
      SPDEL=SPINIT(SPDDEL,TIME,(SPDDES/K-SPDDER))
```

```
      SPDIN=K*(SPDEL+SPDDER)
      SPDERR=(SPDIN-CDOT2)*G
      CDOT2 = INTGRL(U02,SPDERR*LUC)+KS2*WX
      BY2=INTGRL(BY02,BDOT2)
      XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
      YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
      X2=INTGRL(X02,XDOT2)
      Y2=INTGRL(Y02,YDOT2)
      YAW2=BY2
      SWAY2=Y2
      SURGE2=X2
NOSORT
      YAWD1=DEGRAD(0,0,YAW1)
      YAWDP1=DEGRAD(0,1,YAW1)
      YAWD2=DEGRAD(0,0,YAW2)
      YAWDP2=DEGRAD(0,1,YAW2)
      YAWDPD=YAWDP1-YAWDP2
*     RUDDER RESPONSE INPUT
      CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
      CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...
      PSIDED,WT,DA,A1D,B1D,B2D,WTSENS,DD,RD)
      BDOT2D=DEGRAD(0,1,BDOT2)
      BDOTFB=VFBG*BDOT2D
      DDUMB=YAWD2-PSIDED+BDOTFB
      IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
      IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
      DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
      DLTE=DLTS-D2D
      DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
      D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
      D2=DEGRAD(1,1,D2D)
SORT
      DISTE=ABS(DD-ADY)
      OBJ=INTGRL(0.0,DISTE)
      DISTES=ABS(ADX)
      OBJS=INTGRL(0.0,DISTES/25.0)
DYNAMIC REGION
*     ACTUAL SEPARATION
      DX=X2-X1
      DY=Y2-Y1
      CALL TRANS(YAW1,DX,DY,ADX,ADY)
*     EXTERNAL FORCES ACTING BETWEEN SHIPS
      CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
      IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
  100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
*     ACTUAL TIME CONVERSION (SEC)
      ATIME=LUC*TIME
      AA1=(BB1+BB2)/2.0
```

```
      CALL SWTCHF(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D,XOFS)
      IF (ATIME.GT.450.0) XOFS=XOFSD
SAMPLE
CONTRL FINTIM=30.,DELT=0.04,DELS=0.04
PRINT 0.20,ATIME,YAWD1,D1D,DLTS1,CDOT1,OBJ,YAWD2,D2D,DLTS,CDOT2,...
      OBJS,YAWDPD,ADY,ADX,SPDDES,EMDD,WEF,WY,WN,WX,RSENS,WTSENS,...
      RGN,VFBG,SW
PRPLOT ONLY
      CALL DRWG(1,1,ATIME,CDOT2)
      CALL DRWG(1,2,ATIME,SPDDES)
      CALL DRWG(2,1,ATIME,ADX)
TERMINAL
      IF(IS.EQ.1) WRITE(6,101)
  101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
      IF(IS.EQ.0) WRITE(6,102)
  102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
      CALL ENDRW(NPLOT)
      CALL CONTIN
      FINTIM=45.0
END
STOP
FORTRAN
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE SWTCHF FROM APPENDIX A HERE
INSERT FUNCTION SWCL FROM APPENDIX A HERE
INSERT FUNCTION SPDOFC FROM APPENDIX A HERE
INSERT FUNCTION SPINIT FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN DD *
                                                                  4
                                          7.0       5.0
                                                                  4
                                          7.0       5.0
                                                                  4
                                          7.0       5.0
                                                                  4
                                          7.0       5.0
INSERT TWO /* CARDS HERE
```

BIBLIOGRAPHY

1. Calvano, C. N., An Investigation of the Stability of a System of Two Ships Employing Automatic Control while on Parallel Courses in Close Proximity, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, 1970.

2. Principles of Naval Architecture, revised ed., p.466-470, The Society of Naval Architects and Marine Engineers, 1967.

3. Principles of Naval Architecture, revised ed., p.477-486, The Society of Naval Architects and Marine Engineers, 1967.

4. Naval Ship Research and Development Center Report MEL 202/67, Development of a Simulation Model of the Steering-Control System for a Destroyer Class Ship, by C.L. Patterson Jr., March 1969.

5. Van, T. V., Simulation of Propulsion Plant Dynamics and Their Effect on Speed Control, M.S. Thesis, Naval Postgraduate School, Monterey, 1974.

6. Naval Ship Research and Development Center Report 27-745, Simulation of Maneuvering Characteristics of a Destroyer Study Ship Using a Modified Nonlinear Model, by S. H. Brown and R. Alvestad, August 1974.

7. Thaler, G. J., "Computer Determination of Low Order Models for High Order Systems," Computers and Electrical Engineering, v. 2, p. 117-123, 1975.

8. Massachusetts Institute of Technology Department of Naval Architecture and Marine Engineering Report no. 66-5, The Analysis and Modeling of Irregular Waves, by F. H. Sellars and T. A. Loukakis, July 1966.

9. Naval Ship Research and Development Center Report 6-106/70, Digital Computer Simulation of Sea Wave Height / Time Series, by A. S. Fields, December 1970.

10. Lima, C. G., Multivariable Systems Design: A Two Ship Controller for Replenishment at Sea, M.S. Thesis, Naval Postgraduate School, Monterey, 1974.

11. Astorquiza Vivar, G. M., Automatic Control System for Replenishment at Sea, M.S. Thesis, Naval Postgraduate School, Monterey, 1975.

12. Sarzetakis, T., Maneuvering Control of Replenishment at Sea, M.S. Thesis, Naval Postgraduate School, Monterey, 1972.

13. Aguayo, F., Course Keeping with Automatic Control, M.S. Thesis, Naval Postgraduate School, Monterey, 1973.

14. Mather, T. S., Precision Surface Ship Separation Measurement and Maneuvering Control Utilizing Radar Techniques, paper presented at Ship Control Systems Symposium, 2nd, Annapolis, Maryland, 4-5-6 November 1969.

15. Aseltine, J. A., Mancini, A. R., and Sarture, C. W., "A

Survey of Adaptive Control Systems," IRE Transactions on Automatic Control, v. PGAC-6, p. 102-108, December 1958.

16. Leondes, C. T., Modern Control Systems Theory, McGraw-Hill, 1965.

17. National Aeronautics and Space Administration NASA CR-669, Theoretical and Experimental Research on Digital Adaptive Control System, by J. Zaborszky, R. G. Marsh, R. E. Janitch, M. R. Chidambara, and E. E. Ruder, January 1967.

18. National Aeronautics and Space Administration NASA CR-810, Continuation of Theoretical and Experimental Research on Digital Adaptive Control Systems, by J. Zaborszky, R. G. Marsh, R. E. Janitch and M. R. Chidambara, July 1976.

19. Naval Electronics Laboratory NEL Report 1216, Adaptive Plant Identification Digital Control System, by W. V. Kershow, 30 March 1964.

20. Defense Documentation Center TR No. ASD-TR-61-27 Volume I, Fundamental Study of Adaptive Control Systems, by R. E. Kalman, T. S. Englar, and R. S. Bucy, April 1962.

21. National Aeronautics and Space Administration NASA CR-715, Study of Optimal and Adaptive Control Theory, by C. D. Johnson, April 1967.

22. Mishkin, E. and Braun Jr., L., Adaptive Control Systems, McGraw-Hill, 1961.

23. Cadzow, J. A., Martens, H. R., Discrete-Time and Computer Control Systems, Prentice-Hall, Inc., 1973.

24. Aguayo, E., Course-Keeping with Automatic Control, M.S. Thesis, Naval Postgraduate School, Monterey, 1973.

25. Hozos, A. G., Maneuvering Characteristics of Automatically Controlled Ships with Directionally Unstable Hulls, M.S. Thesis, Naval Postgraduate School, Monterey, 1974.

26. Navy Marine Engineering Laboratory MEL Report 333/65, Evaluation of Functional Performance of an Integrated Ship Control Conning Console by Operator Personnel, by J. T. McLane, W. J. Weingartner, and J. C. Townsend, May 1966.

27. Hess, D., "Shipboard Manning Reduction," All Hands, Nr. 707, p. 14-21, December 1975.

28. "Navy Commands Report 4 More Ship Collisions," Navy Times, p. 4, 31 December 1975.

29. Naval Research and Development Center Report 76-0040, Automatic Control of Underway Replenishment Maneuvers in Random Seas, by R. Alvestad, April 1976.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Professor George J. Thaler, Code 52Tr Naval Postgraduate School Monterey, California 93940	5
5.	Lt. John J. Uhrin III, USN 456 Lineberry Road Virginia Beach, Virginia 23452	3
6.	Reidar Alvestad NSRDC Annapolis Lab Annapolis, Maryland 21402	1
7.	Assoc. Professor Alex Gerba, Jr., Code 52Gz Naval Postgraduate School Monterey, California 93940	2
8.	Samuel E. Brown NSRDC Annapolis Lab Annapolis, Maryland 21402	1

INITIAL DISTRIBUTION LIST (cont.)

		No. Copies
9.	J. K. Zuidweg Royal Netherlands Naval College Den Helder, The Netherlands	1
10.	RADM R. L. Walters Project Manager Surface Ship Project Office PM18 Room 9SC8 National Center 3 Department of the Navy Washington, DC 20362 Attention PM18T	1
11.	Marvin Denicoff Office of Naval Research Code 437 Arlington, Virginia 22217	1