

AD-A098 487

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

F/G 13/10

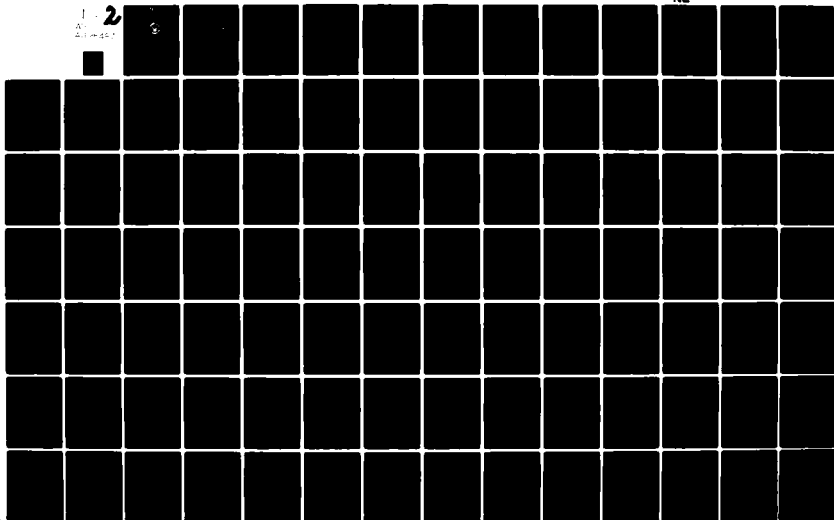
REAL TIME SIMULATION AND CONTROL 3000 TON SURFACE EFFECT SHIP W--ETC(U)

DEC 80 L L OLIPHANT

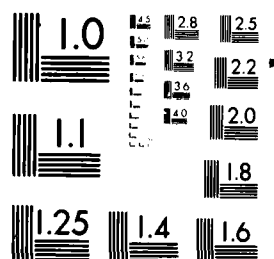
UNCLASSIFIED

NL

1-2
A1-147



1 OF 2
AD-
A098487



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL

②

AD A 098 487

NAVAL POSTGRADUATE SCHOOL
Monterey, California



DTIC
SELECTED
MAY 5 1981

THESIS

REAL TIME SIMULATION AND CONTROL
3000 TON SURFACE EFFECT SHIP
WITH NEGATIVE DRAG CHARACTERISTICS
IN SEA STATE

by

Lee Lewis Oliphant

December 1980

Thesis Advisor:

A. Gerba

Approved for public release; distribution unlimited

DTIC FILE COPY

81 5 04 078

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
	AD-A098	487
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED
Real Time Simulation and Control 3000 Ton Surface Effect Ship With Negative Drag Characteristics In Sea State.		Master's Thesis December 1980
6. PERFORMING ORG. REPORT NUMBER		
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
Lee Lewis Oliphant		
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Naval Postgraduate School Monterey, California 93940		
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
Naval Postgraduate School Monterey, California 93940		Dec 1980
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES
		117
		15. SECURITY CLASS. (of this report)
		UNCLASSIFIED
		16. 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)		
Surface Effect Ship; Real Time Simulation; Negative Drag Characteristic; Speed Control; Sea State Effect		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
The model of a Surface Effect Ship was refined to include simplified propulsion dynamics, negative drag characteristics, sea state effects and an autopilot for speed control. These design modifications were introduced into a real time, man controlled simulation of a 3000 ton Surface Effect Ship (3K-SES) in 5 degrees of freedom (RTS5D) and results were compared with a Data Base Program (DBSIM5D) based on towing tank data scaled up to model the ship.		

Hardware and Software design changes were incorporated into the RTS5D model to provide a more accurate approximation of real time, a faster computer iteration time, and a broach condition warning if the operator exceeded certain thrust vectoring limits.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	

Approved for public release; distribution unlimited

Real Time Simulation and Control
3000 Ton Surface Effect Ship
With Negative Drag Characteristics
In Sea State

by

Lee Lewis Oliphant
Lieutenant, United States Navy
B.S., University of Texas, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

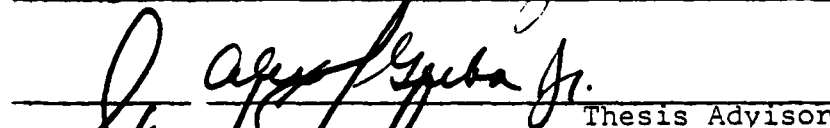
from the

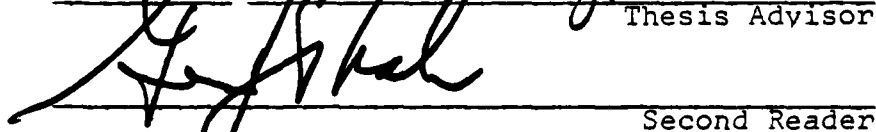
NAVAL POSTGRADUATE SCHOOL
December 1980

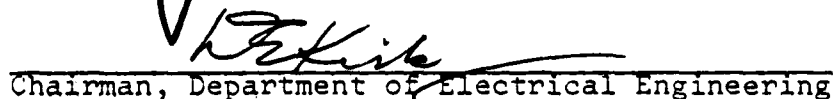
Author



Approved by:

 Thesis Advisor

 Second Reader


Chairman, Department of Electrical Engineering


Dean of Science and Engineering

ABSTRACT

The model of a Surface Effect Ship was refined to include simplified propulsion dynamics, negative drag characteristics, sea state effects and an autopilot for speed control. These design modifications were introduced into a real time, man controlled simulation of a 3000 ton Surface Effect Ship (3K-SES) in 5 degrees of freedom (RTSSD) and results were compared with a Data Base Program (DBSIMSD) based on towing tank data scaled up to model the ship.

Hardware and Software design changes were incorporated into the RTSSD model to provide a more accurate approximation of real time, a faster computer iteration time, and a broach condition warning if the operator exceeded certain thrust vectoring limits.

TABLE OF CONTENTS

I.	INTRODUCTION- - - - -	13
II.	EQUATIONS OF MOTION - - - - -	15
	A. COORDINATE SYSTEMS AND ASSUMPTIONS- - - - -	15
	B. FORCES AND MOMENTS- - - - -	23
	1. Surge Forces- - - - -	23
	2. Sway Forces - - - - -	23
	3. Yaw Moments - - - - -	24
	4. Pitch Moments - - - - -	24
	5. Roll Moments- - - - -	26
	C. PARAMETER IDENTIFICATION- - - - -	28
III.	RTS5D VALIDATION- - - - -	31
	A. DBSIM5D BENCHMARK - - - - -	31
	B. VALIDATION RESULTS- - - - -	31
IV.	MODEL DEVELOPMENT - - - - -	36
	A. DRAG CHARACTERISTICS- - - - -	36
	1. Surge Equation Model- - - - -	36
	2. Implementation- - - - -	36
	3. Identification of Surge Drag Coefficients- - - - -	38
	B. LINEARIZED MODEL ANALYSIS - - - - -	38
	C. SIMPLIFIED PROPULSION DYNAMICS- - - - -	42
	D. ANALYSIS OF LINEARIZED MODEL WITH SPEED CONTROLLER- - - - -	43
	E. EFFECTIVE THRUST EFFECTOR ANGLE - - - - -	47
	F. BROACH CONDITION FLAG - - - - -	47

V.	IMPLEMENTATION-	54
A.	INTRODUCTION-	54
B.	REQUIREMENTS-	54
C.	HARDWARE DESCRIPTION-	55
D.	SOFTWARE DESCRIPTION-	55
VI.	RTSSD MODS I AND II RESPONSE CHARACTERISTICS-	61
VII.	CONCLUSIONS -	69
VIII.	RECOMMENDATIONS -	71
APPENDIX A -	REAL TIME ANALYSIS-	72
APPENDIX B -	RTSSD MODIFIED PROGRAM NOMENCLATURE-	76
APPENDIX C -	RTSSD MODIFIED COMPUTER PROGRAM LISTING -	89
APPENDIX D -	RTSSD MODIFIED WIRING DIAGRAM -	115
LIST OF REFERENCES -		116
INITIAL DISTRIBUTION LIST-		117

LIST OF FIGURES

1.	Definition of Coordinate System (Part I) - - - - -	18
2.	Definition of Coordinate System (Part II)- - - - -	19
3.	Surface Effect Ship (Top View) - - - - -	20
4A.	Surface Effect Ship (Stern View) - - - - -	21
4B.	Surface Effect Ship (Side View)- - - - -	22
5.	Full Scale Drag Curves - - - - -	37
6.	Linearized Surge Model - - - - -	40
7.	Linearized Surge Root Map- - - - -	41
8.	Simplified Propulsion System - - - - -	43
9.	Linearized Surge Model with Propulsion System and Speed Controller- - - - -	44
10.	Maximum Thrust Effector Angle as a Function of Surge Velocity in Order To Avoid Broaching - - - - -	48
11.	Inlet Broaching Boundaries at 40 Knots - - - - -	49
12.	Inlet Broaching Boundaries at 60 Knots - - - - -	51
13A.	Pilot Graphic Display (Normal) - - - - -	52
13B.	Pilot Graphic Display (Broach Condition Exists)- -	53
14.	RTS5D Mod Block Diagram- - - - -	57
15.	Thruster Console - - - - -	58
16.	RTS5D Mod Program Flow Chart - - - - -	59
17.	RTS5D Mod Display Multiplex Algorithm- - - - -	60
18.	360° Turn Comparison of Four Models- - - - -	65
19.	Response Time Comparison at 40 knots - - - - -	66
20.	Response Time Comparison at 50 knots - - - - -	67
21.	Response Time Comparison at 60 knots - - - - -	68

LIST OF TABLES

Table I	RTSSD and DBSIM5D Performance Test at 40 Knots- - - - -	33
Table II	RTSSD and DBSIM5D Performance Test at 50 Knots- - - - -	34
Table III	RTSSD and DBSIM5D Performance Test at 60 Knots- - - - -	35
Table IV	RTSSD Mod I and Mod II Performance Test at 40 Knots- - - - -	62
Table V	RTSSD Mod I and Mod II Performance Test at 50 Knots- - - - -	63
Table VI	RTSSD Mod I and Mod II Performance Test at 60 Knots- - - - -	64

NOMENCLATURE

A_{ws}	Average sidewall wetted area, starboard side	ft^2
A_{wp}	Average sidewall wetted area, port side	ft^2
A_{31}	Added mass coefficient in roll force equation	ft-slug
A_{33}	Added mass coefficient in pitch force equation	ft-slug
A_{w2}	Average wetted sidewall area of the bow	ft^2
A_{w1}	Average wetted sidewall area of the stern	ft^2
A_{22}	Added mass coefficient in yaw force equation	$slug \frac{s^2}{ft} lb_f$
β	Sideslip angle	rad
$C_{DX(K)}$	Coefficients drag in x-direction	$lb_f s^2 lb_m / ft^2$
C_{DY}	Coefficient drag in y-direction	$lb_f s^2 lb_m / ft^2$
C_{DZP}	Sidewall roll moment lumped parameter coefficient	$lb_f s^2 / ft^2$
C_{DP}	Bow pitch force lumped parameter coefficient	non-dimensional
F_{sw}	Sidewall roll force	lbs
F_{ss}	Sidewall starboard buoyancy force	lbs
F_{sp}	Sidewall port buoyancy force	lbs
F_1	Stern buoyancy force	lbs
F_2	Bow seal pitching force	lbs
F_3	Bow buoyancy force	lbs
g	Gravitational acceleration	ft/s^2

I_X	Moment of inertia about x-axis	slug ft ²
I_Y	Moment of inertia about y-axis	slug ft ²
I_Z	Moment of inertia about z-axis	slug ft ²
K	Summation of moments about x-axis	lbs-ft
l_{sw}	Length of sidewall	ft
l_{dp}	Actual draft of port sidewall	ft
l_w	x-direction displacement of hull drag centroid	ft
l_{d1}	Average draft of SES sidewall	ft
l_{s1}	Average draft of bow seal	ft
l_x	Length from center of gravity to stern	
l_{x1}	Average draft of widewall	ft
l_{31}	Pitch moment lever arm for bow sidewall buoyance force	ft
l_3	Pitch moment lever arm for bow seal force	ft
m	Mass of the rigid ship	ft
M	Summation of moments about y-axis	lbs-ft
N	Summation of moments about z-axis	lbs-ft
P_b	Plenum pressure	lbs-ft ²
p	Lumped drag centroid point	non-dimensional
p'	Lumped drag centroid point	non-dimensional
$\ddot{p}$	Roll acceleration	rad/sec ²

q	Pitch rate	rad/sec
$\dot{q}$	Pitch acceleration	rad/sec ²
r	Yaw rate	rad/sec
$\dot{r}$	Yaw acceleration	rad/sec ²
s_1	Turning moment lever arm of no. 1 engine	ft
s_2	Turning moment lever arm of no. 2 engine	ft
s_3	Turning moment lever arm of no. 3 engine	ft
s_4	Turning moment lever arm of no. 4 engine	ft
T_7	Total thrust magnitude on no. 1 engine	lbs
T_8	Total thrust magnitude on no. 2 engine	lbs
T_9	Total thrust magnitude on no. 3 engine	lbs
T_{10}	Total thrust magnitude on no. 4 engine	lbs
T_{forw}	Total forward thrust vector of effectors	lbs
T_{side}	Total side thrust vector of effectors	lbs
T_{yaw}	Total turning moment generated by effectors	lbs ft
u	Velocity in x-direction (surge)	ft/sec
$\dot{u}$	Acceleration in x-direction	ft/sec ²
v	Velocity in y-direction (sway)	ft/sec
$\dot{v}$	Acceleration in y-direction	ft/sec ²
v_s	Total velocity	ft/sec

w_e	Width of bow seal	ft
X	Summation of forces in x-direction	lbs
X_o	X_{NAV} coordinate of SES	ft
$\dot{X}_o$	X_{NAV} velocity of SES	ft/sec
Y	Summation of forces in y-direction	lbs
Y_o	Y_{NAV} coordinates of SES	ft
$\dot{Y}_o$	Y_{NAV} velocity of SES	ft/sec
δ	Effector angle commanded	rad
δ_7	Effector angle of no. 1 nozzle	rad
δ_8	Effector angle of no. 2 nozzle	rad
δ_9	Effector angle of no. 3 nozzle	rad
δ_{10}	Effector angle of no. 4 nozzle	rad
ψ	Heading angle of SES	rad
ϕ	Roll angle of SES	rad
θ	Pitch angle of SES	rad
ρ	Density of sea water	lbm/ft ³

I. INTRODUCTION

Rising fuel costs, critical manpower shortages and an increasing need for rapid deployment of forces in defense of United States' interests throughout the world mandate the development of fuel efficient, high speed, minimum manned ships for the Navy. The Captured Air Bu^{bb}dle (CAB) Surface Effect Ship (SES) holds great promise in providing an answer to these problems.

A ship with minimum manning traveling at the high speeds envisioned for the SES would require highly skilled operator personnel. It would be necessary to provide these personnel with real time operational training similar to that which pilots receive in aircraft simulators. This would test the man in responding to failure situations that may be encountered during actual operation and train personnel in control techniques that are unique to SES type vehicles. A Real Time Simulator could also be used as a means to test hardware used in SES craft and provide a design tool for the development of future modifications.

A simplified, non-linear five degree of freedom Real Time Simulator (RTS5D) for the 3K-SES has been developed by T. S. Nelson [Ref. 1] such that a continuously observable real time solution is interfaced with man-generated control. Certain "man-in-the-loop" experiments have been conducted using the

RTSSD that have demonstrated limitations in that model. It was necessary to: 1) refine ship dynamics; 2) introduce simplified propulsion dynamics; 3) constrain thrust angle in order to remain within the bounds of programmed dynamic response (avoid broaching); 4) refine the software to provide faster computer iteration time; and 5) change the hardware to provide the operator with speed control as well as turning control.

Accordingly, changes have been made in the RTSSD model to more accurately reflect the nonlinear forward drag characteristics of the SES. A speed controller has been designed to allow operation of the 3K-SES within a speed range of 40-60 knots and limits have been placed on the thruster deflection angle in order to prevent broaching, a condition which is not allowed under the existing dynamic model. Software refinements have been incorporated to improve computer iteration time.

II. EQUATIONS OF MOTION

A. COORDINATE SYSTEMS AND ASSUMPTIONS

The RTSSD model of the 3K-SES used simplified equations of motion. For completeness of comparison the development of these equations is repeated here. The rationale for requiring simplification was to start with differential equations of motion that could be utilized effectively in a real time simulation. To satisfy this requirement a model was selected that would use point source of drag forces consisting of only a few lumped-coefficient terms. The equations used are based upon those developed in Ref. 2 which follow this concept. Additional assumptions used in the simplified 5 DOF equations were as follows:

- 1) All accelerations are measured at the center of gravity.
- 2) Vehicle is "free-to-heave".
- 3) All cross coupled moments of inertia are zero.
- 4) All cross products of angular velocities in force equations are zero.
- 5) All roll and pitch angles used in force equations are subject to small angle approximations.
- 6) Calm water conditions.
- 7) Mass and mass distribution of vehicle are constant.
- 8) Effect of changes of aerodynamic force are neglected.

Utilizing the above assumptions, the simplified 5 DOF equations of motion are developed using the coordinate systems shown in Fig. 1 and Fig. 2, the force vector diagrams shown in Fig. 3, Fig. 4A, Fig. 4B, and the following equations from Newton's laws of motion:

$$\begin{array}{lll} \text{SURGE} & m(\dot{u} - vr) & = X \\ \text{SWAY} & m(\dot{v} - ur) & = Y \\ \text{YAW} & I_z \dot{r} & = N \\ \text{PITCH} & I_y \dot{q} & = M \\ \text{ROLL} & I_x \dot{p} & = K \end{array}$$

where:

m	= mass of the rigid ship	' slugs
u	= velocity in x-direction (SURGE)	' ft/sec
v	= velocity in y-direction (SWAY)	' ft/sec
r	= angular velocity about the z-axis	' deg/sec
p	= angular velocity about the x-axis	' deg/sec
q	= angular velocity about the y-axis	' deg/sec
I_z	= moment of inertia about the z-axis	' $\text{lb}_m\text{-ft}^2$
I_y	= moment of inertia about the y-axis	' $\text{lb}_m\text{-ft}^2$
I_x	= moment of inertia about the x-axis	' $\text{lb}_m\text{-ft}^2$
X	= summation of forces in x-direction	' lb_f
Y	= summation of forces in y-direction	' lb_f
N	= summation of moments about the z-axis	' ft-lb_f
M	= summation of moments about the y-axis	' ft-lb_f
K	= summation of moments about the x-axis	' ft-lb_f

Additionally the following navigation relationships are utilized (see Fig. 1).

$$\dot{X}_O = u \cos \psi - v \sin \psi$$

$$\dot{Y}_O = u \sin \psi + v \cos \psi$$

$$v_s^2 = u^2 + v^2$$

$$\beta = \tan^{-1} (-v/u)$$

where:

ψ = heading angle

β = drift angle

δ = thrust vector angle

Also,

$$X_O = \int \dot{X}_O dt$$

$$Y_O = \int \dot{Y}_O dt$$

$$r = \int \dot{r} dt$$

$$p = \int \dot{p} dt$$

$$q = \int \dot{q} dt$$

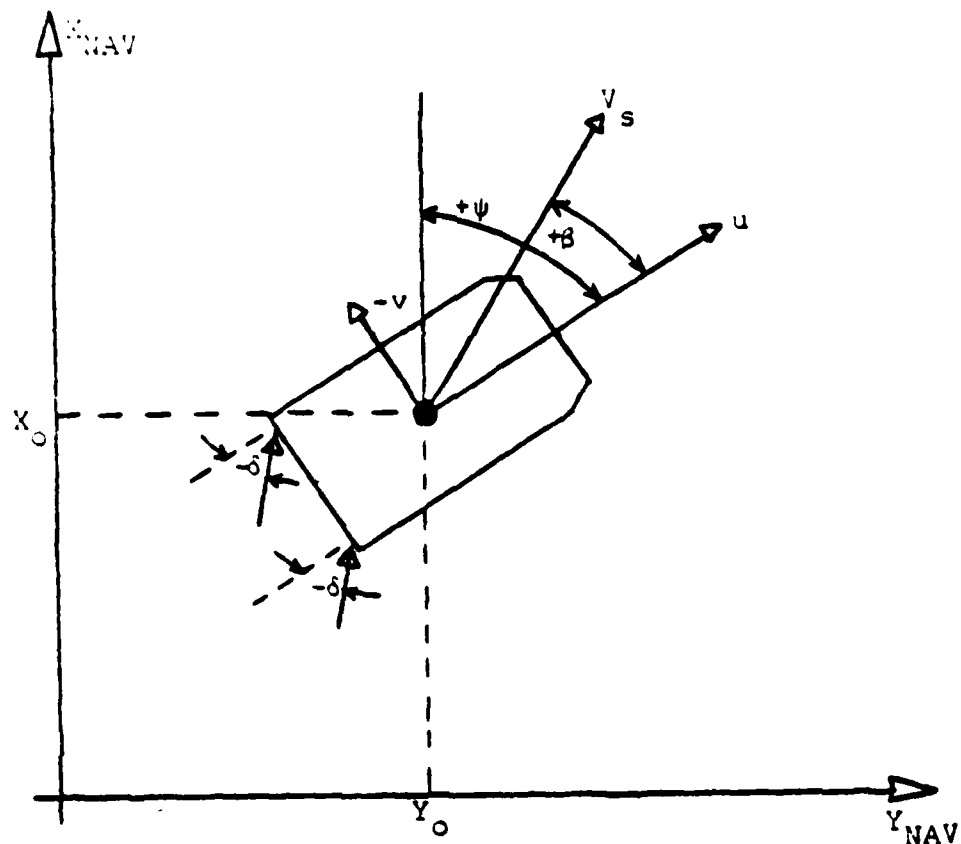
$$u = \int \dot{u} dt$$

$$v = \int \dot{v} dt$$

$$\psi = \int \dot{\psi} dt$$

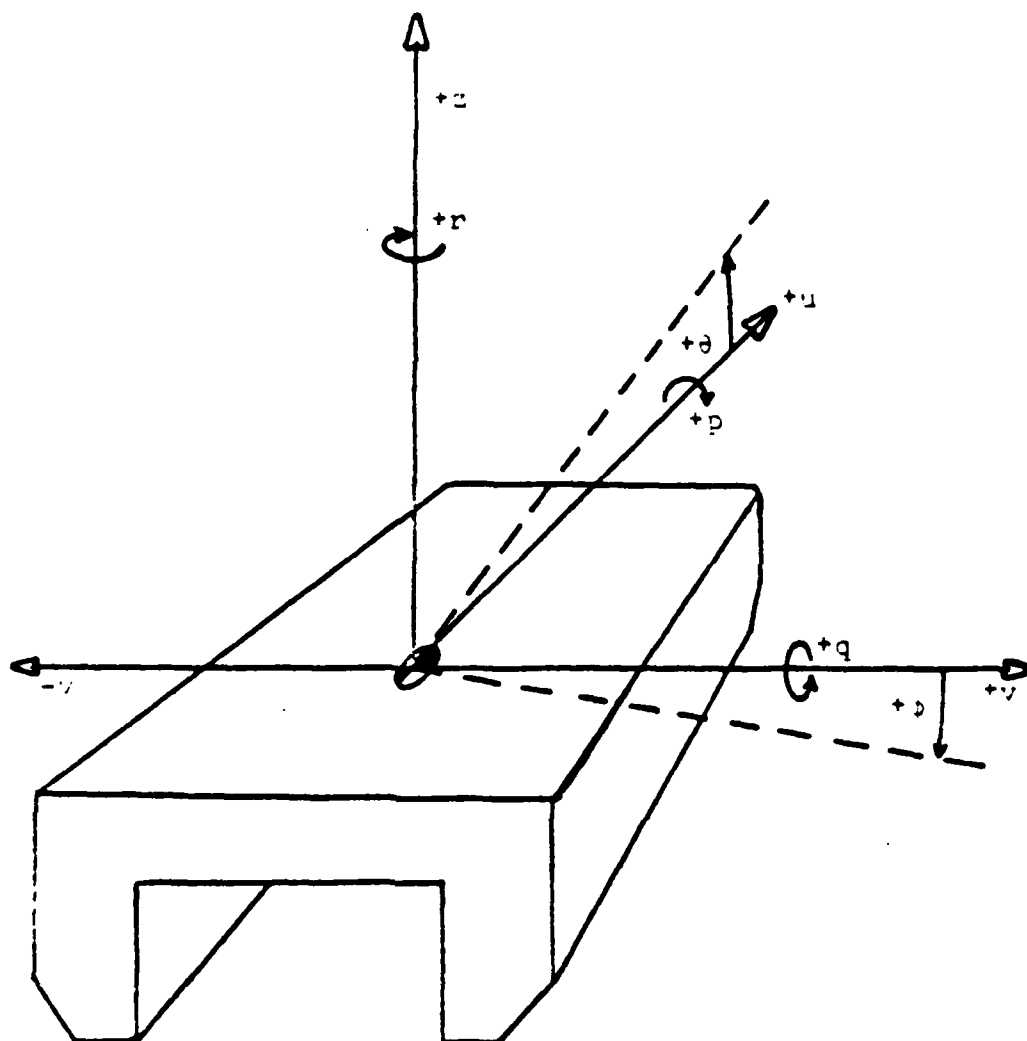
$$\phi = \int \dot{\phi} dt$$

$$\theta = \int \dot{\theta} dt$$



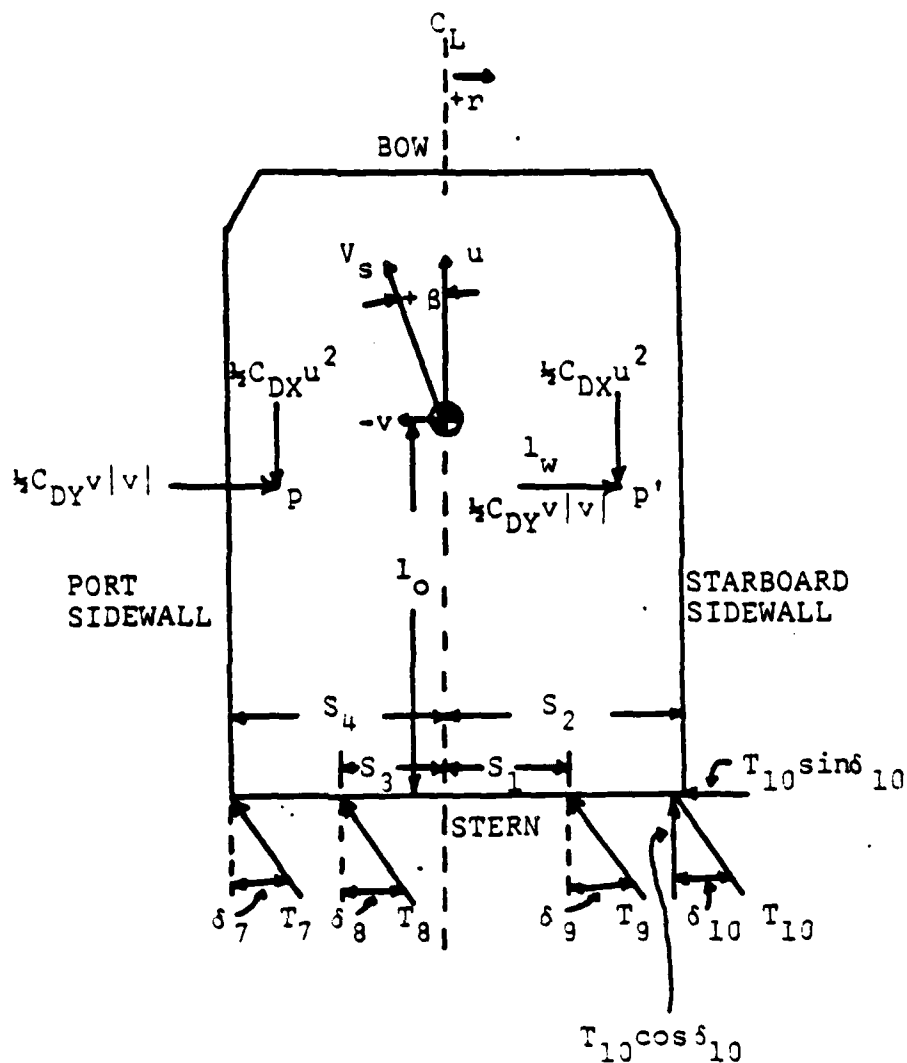
- Note: 1) $\psi=0$ when vehicle heads parallel to $+ X_{NAV}$ axis.
- 2) Vehicle in figure above shown with negative thrust vector angle δ , positive ψ , negative β , negative sway velocity v , i.e., in a right turn

Figure 1
Definition Of Coordinate System (Part I)



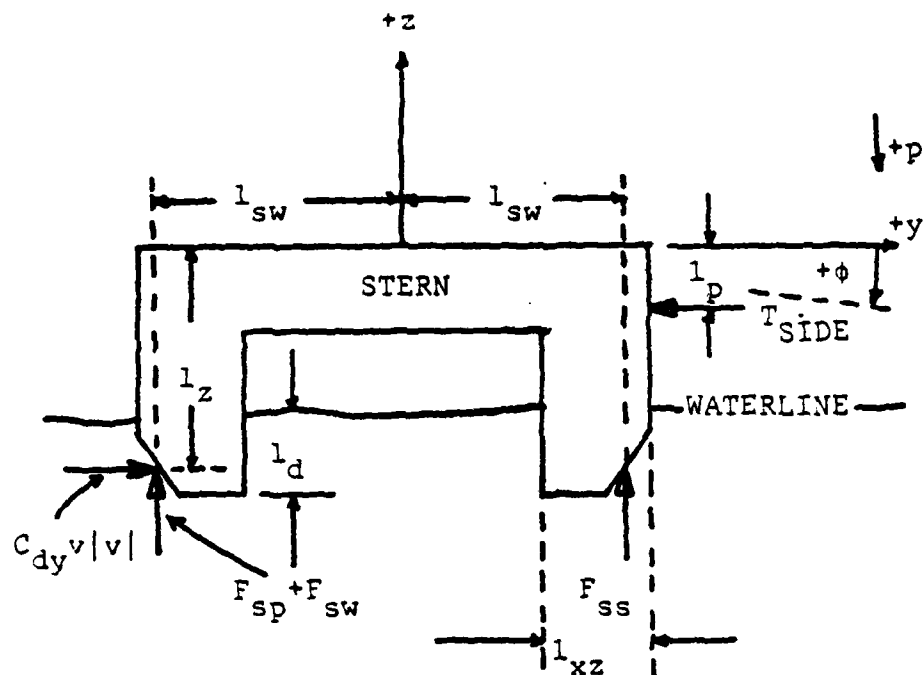
Note: Direction for positive velocity and angular rates are shown

Figure 2
Definition Of Coordinate System (Part II)



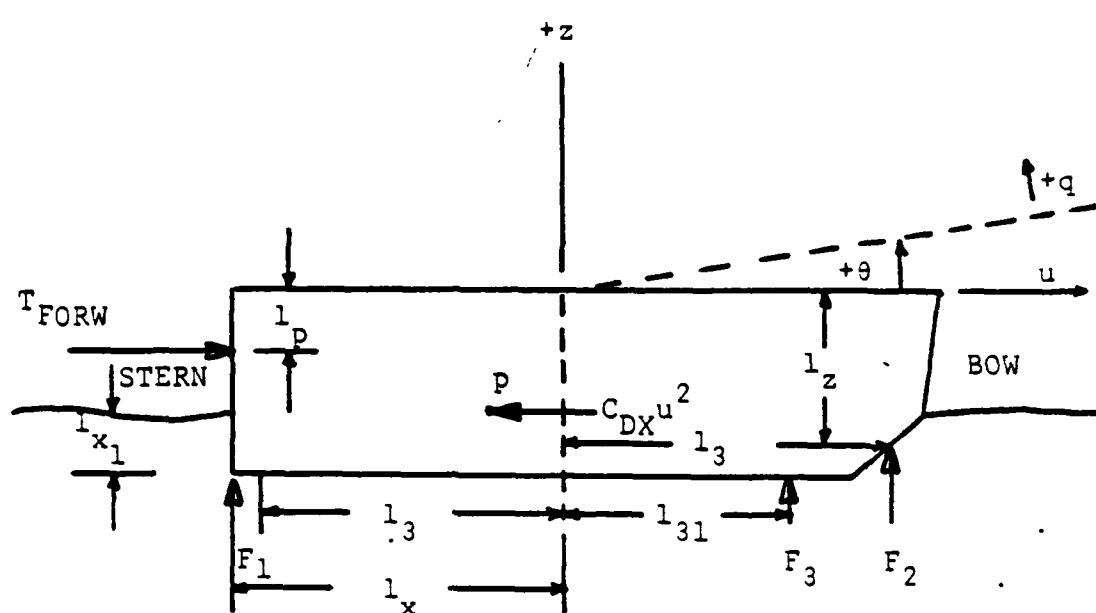
- Note:
- 1) p and p' are equivalent point force centroids.
 - 2) $\delta_7, \delta_8, \delta_9, \delta_{10}$ are thrust vector angles and are not required to be equal.
 - 3) T_7, T_8, T_9, T_{10} are thrust magnitudes and are not required to be equal.
 - 4) Force directions shown are for a right turn

Figure 3
Surface Effect Ship (Top View)



- Note: 1) Right Turn Forces Acting, i.e., δ_7 , δ_8 , δ_9 , δ_{10} are negative
- 2) $T_{SIDE} = T_7 \sin \delta_7 + T_8 \sin \delta_8 + T_9 \sin \delta_9 + T_{10} \sin \delta_{10}$

Figure 4A
Surface Effect Ship (Stern View)



Note: $T_{FORW} = T_7 \cos \delta_7 + T_8 \cos \delta_8 + T_9 \cos \delta_9 + T_{10} \cos \delta_{10}$
 (see Fig. 3)

Figure 4B
 Surface Effect Ship (Side View)

B. FORCES AND MOMENTS

1. Surge Forces (see Fig. 4B)

The forward acceleration was determined to be generated by the summation of thrust vectored in the forward direction T_{forw} . The resultant acceleration of the SES was assumed to be opposed by a retarding force exhibiting a "velocity-squared" characteristic. Additionally, Newton's laws of motion specified a "v.r" product that contributed to the drag component of the equation when the vehicle was in a turn (contrifugal force reaction term). The simplified equation of motion for the surge acceleration follows from a summation of forces in the u direction as depicted in Fig. 4B.

$$\dot{u} = (T_{\text{forw}}/m) - (C_{DX}u^2/m) + vr$$

2. Sway Forces (see Fig. 4A)

The side acceleration of the vehicle was analyzed to be the result of the summation of the thrust vectored parallel to the vehicle's y-axis and a retardation "velocity squared" phenomena such as that produced by the cross flow drag term of the sidewall as described in Ref. 3. This retardation force is augmented by a "u.r" product as specified by Newton's Law.

$$\dot{v} = T_{\text{side}}/m - C_{DY} v|v|/m - ur$$

Note the requirement to force the $C_{DY}v^2/m$ term to maintain the sign of the sway velocity such that both left and right turns may be computed.

3. Yaw Moments (see Fig. 3)

The yaw acceleration was a turning moment summation which was defined to be influenced by three primary forces operating over moment arms. It is of interest to note that the third term is an "added mass" term which is considered to operate on the same pressure point as the $C_{DY}v|v|$ term. The term T_{yaw} is a summation of moments generated by the four thrusters as shown in Fig. 3.

$$\begin{aligned} T_{yaw} = & s_4 T_7 \cos \delta_7 + s_3 T_8 \cos \delta_8 - s_1 T_9 \cos \delta_9 \\ & - s_2 T_{10} \cos \delta_{10} - \lambda_0 T_7 \sin \delta_7 - \lambda_0 T_8 \sin \delta_8 \\ & - \lambda_0 T_9 \sin \delta_9 - \lambda_0 T_{10} \sin \delta_{10} \end{aligned}$$

$$\dot{r} = C_{DY} \lambda_w v|v|/I_z + A_{22}uv\lambda_w/I_z + T_{YAW}/I_z$$

4. Pitch Moments (see Fig. 4B)

The pitch acceleration was assumed to be the summation of moments generated by forward thrust, T_{forw} , the buoyancy of the sidewalls of the SES, F_1 and F_2 , and a vertical force generated at the bow of the vehicle, F_3 . This vertical bow force was defined to be a lumped parameter term which modeled the reaction force due to plenum pressure acting against the bow seal.

l_d = average draft of bow seal

$A_{w1} = l_x l_{x1} + ((l_x \tan \theta)/2) l_x$

$A_{w2} = l_x l_{x1} - ((l_x \tan \theta)/2) l_x$

$F_1 = A_{w1} l_{x2} \rho g$

$F_2 = A_{w2} l_{x2} \rho g$

$F_3 = C_{DP} \bar{p}_b w_e (l_d - l_{31} \tan \theta)$

l_{x1} = average draft of sidewall

p_b = plenum pressure

A_{w1} = average wetted sidewall area of the stern

l_{x2} = width of one sidewall

w_e = width of bow seal

A_{w2} = average wetted sidewall area of the bow

A_{33} = added mass coefficient

ρ = water density

g = 32.3 ft/s²

l_{31} = lever arm of bow seal

l_3 = lever arm of buoyancy force

$$\dot{q} = (T_{\text{forw}} l_p + F_3 l_{31} + F_2 l_3 - C_{DX} u^2 l_z$$

$$- F_1 l_3 - A_{33} u \dot{q}) / I_Y$$

Note the added mass term $A_{33}u\dot{u}$ which was required to provide damping to the pitch moment and is specified in Ref. 2.

It was found that the response of the SES to a pitch perturbation without the added mass component was undamped and approximately sinusoidal. Unlike the yaw equation where the added mass term was a small contributor to damping, the pitch added mass term was found to be of significant importance in modeling the known pitch motion.

5. Roll Moments (see Fig. 4A)

The roll acceleration equation was analyzed to be a summation of moments generated by buoyancy forces of the port and starboard sidewalls, F_{sp} and F_{ss} , the thrust vectored parallel to the y-axis of the vehicle, T_{side} , a side force $C_{DY} v|v|$, and a lumped coefficient vertical force, F_{sw} . The vertical force F_{sw} was defined as the force generated in a turn due to the sidewall curvature (dead rise angle) acting against the cross flow of water.

$$F_{sp} = \rho g A_{wp} l_{dp}$$

$$F_{ss} = \rho g A_{ws} l_{ds}$$

$$F_{sw} = C_{DZP} v|v|$$

$$l_{d1} = \text{average draft of SES sidewall}$$

$$l_{dp} = l_{d1} - l_{sw} \tan \phi$$

$$l_{ds} = l_{d1} + l_{sw} \tan \phi$$

A_{wp} = average wetted area port

A_{ws} = average wetted area starboard

$$\begin{aligned} \dot{p} = & ((F_{sp} - F_{ss})\ell_{sw} - T_{side}\ell_p + C_{DY} v|v|\ell_z \\ & - F_{sw}\ell_{sw} - A_{31} up)/I_X \end{aligned}$$

Again note the added mass term $A_{33}up/I_X$ which was found to be essential in modeling the damping phenomena of the roll motion.

In summary, the simplified 5 DOF equations of motion used in RTS5D for the 3K TON SES are:

$$\text{SURGE} \quad \dot{u} = T_{forw}/m - C_{DX} u^2/m + vr \quad (\text{Fig. 4B})$$

$$\text{SWAY} \quad \dot{v} = T_{side}/m - C_{DY} v|v|/m = ur \quad (\text{Fig. 4A})$$

$$\text{YAW} \quad \dot{r} = T_{yaw}/I_Z + C_{DY} v|v|\ell_w/I_Z + A_{22}uv\ell_w/I_Z \quad (\text{Fig. 3})$$

$$\begin{aligned} \text{PITCH} \quad \dot{q} = & (T_{forw}\ell_p + F_3\ell_{31} + F_2\ell_3 - C_{DX} u^2\ell_z \\ & - F_1\ell_3 - A_{34}uq)/I_Y \quad (\text{Fig. 4B}) \end{aligned}$$

$$\begin{aligned} \text{ROLL} \quad \dot{p} = & ((F_{sp} - F_{ss})\ell_{sw} - T_{side}\ell_p + C_{DY} v|v|\ell_z \\ & - F_{sw}\ell_{sw} - A_{33} up)/I_X \quad (\text{Fig. 4A}) \end{aligned}$$

C. PARAMETER IDENTIFICATION

These equations are an extension of the 3 DOF flat turn SES model developed by Gerba and Thaler in Ref. 4. The identification of craft parameters C_{DX} , C_{DY} , and ℓ_w is described in Ref. 4 and repeated here for completeness.

The surge drag coefficient C_{DX} is determined by selecting a steady state turn condition, where T_{forw} , u , v , r , and m are known.

$$\dot{u} = 0 = \frac{T_{forw}}{m} - \frac{C_{DX}u^2}{m} + vr$$

from which

$$C_{DX} = \frac{T_{forw} + mvr}{u^2}$$

The sway drag coefficient C_{DY} is determined by the same steady state turn condition which requires

$$\dot{v} = 0 = \frac{T_{side}}{m} - \frac{C_{DY}}{m} v|v| - ur$$

where

$$C_{DY} = \frac{T_{side} - mur}{v|v|}$$

The sway drag moment arm follows utilizing the yaw acceleration equation

$$\dot{r} = 0 = \frac{T_{yaw}}{I_z} + \frac{C_{DY}v|v|\ell_w}{I_z} + \frac{A_{22}uv\ell_w}{I_z}$$

which yields

$$w = \frac{-T_{\text{yaw}}}{C_{DY} v|v| + A_{22}uv}$$

The addition of the pitch and roll equations introduces additional lumped coefficients C_{DP} and C_{DZP} . These are easily solved using known constants ℓ_w , ℓ_p , C_{DY} , ℓ_z , ℓ_{sw} , w_e , p_b , ℓ_d and steady state values of F_{sp} , F_{ss} , F_1 , F_2 , T_{forw} , T_{side} , u , v , and r . It is significant to note that in a steady state condition the angular roll rate, p , and angular pitch rate, q , are both required to equal zero. This is in contrast to the yaw rate, r , where a finite steady state value is desired in a turn. Thus the added mass terms of $A_{33}uq$ and A_{31} up in the pitch and roll equations are not utilized in the determination of craft parameters; their function is strictly confined to damping of their respective accelerations. Therefore, utilizing the pitch equation with known steady state turn values.

$$\begin{aligned} \dot{q} = 0 = & (T_{\text{forw}}\ell_p + C_{DP}p_b w_e(\ell_d - \ell_{31} \tan \theta)\ell_{31} \\ & + F_2\ell_3 - C_{DX}u^2\ell_2 - F_1\ell_3)/I_Y \end{aligned}$$

yields

$$C_{DP} = \frac{C_{DX}u^2\ell_z + F_1\ell_3 - F_2\ell_3 - T_{\text{forw}}\ell_p}{p_b w_e(\ell_d - \ell_{31} \tan \theta)\ell_{31}}$$

The roll equation under steady state turn conditions yields

$$\dot{p} = 0 = ((F_{sp} - F_{ss})l_w - T_{side} l_p + C_{DY} v|v| l_z - C_{DZP} v|v| l_{sw}) I_z$$

from which

$$C_{DZP} = ((F_{sp} - F_{ss})l_w - T_{side} l_p + C_{DY} v_{ss}|v_{ss}| l_z) / (v_{ss}|v_{ss}| l_{sw})$$

where

- F_{sp} = port sidewall buoyancy in pounds (Fig. 4A)
- F_{ss} = starboard sidewall buoyancy in pounds (Fig. 4A)
- T_{side} = thrust vectored parallel to vehicles y-axis in pounds (Fig. 3)
- v_{ss} = sway steady state velocity in feet/sec
- C_{DY} = sway drag coefficient for steady state condition
- l_w = sway drag moment in feet (Fig. 3)
- l_p = effector thrust moment arm in feet (Fig. 4A)
- l_z = surge drag moment arm in feet (Fig. 4A)
- l_{sw} = sidewall buoyancy force moment arm in feet (Fig. 4A)

III. RTS5D VALIDATION

A. DBSIM5D BENCHMARK

Validation of the RTS5D was attempted using the data base program (DBSIM5D) [Ref. 5] as a benchmark. However, new guidance in the proper use of the data base program contained in Ref. 6 revealed the following errors in the use of the DBSIM5D. 1) Initial conditions had been improperly set and 2) Data used for validation for the 56 knot run at effector deflection angles greater than 10° was not correct because of the thrust inlet broach conditions which existed using the DBSIM5D. These conditions were not accounted for in the RTS5D model. The noted discrepancies in the operation of the DBSIM5D necessitated a revalidation of the RTS5D.

B. VALIDATION RESULTS

Using the information contained in Ref. 6, the DBSIM5D program was again used as a benchmark to validate the RTS5D simulation. A DBSIM5D sequence of runs with initial forward velocities equal to 40 knots, 50 knots and 60 knots with step effector angles of 5° and 10° (at 40 knots), 5° , 10° and 15° (at 50 knots), and 5° (at 60 knots) were compared to identical runs of the RTS5D. Thrust effector angles were kept below a maximum that would have caused a broach

condition. Broach conditions are explained in Chapter IV. The first peak overshoot value and "Quasi Steady State" values for the variables u (surge), v (sway), r (yaw rate), ϕ (roll), and θ (pitch) were used as a basis for comparison. The results are shown in Tables I-III. Significant error in forward velocity occurred in the 50 knot test. Also, it was noted that in 40 knot test with 5° thruster deflection angle that the DBSIMSD actually reached a "Quasi Steady State" forward velocity greater than the initial value upon entering the turn. These conditions led to the development of the model of the RTS5D with negative drag characteristics in sea state. Differences in the other measured parameters were addressed by parameter adjustment on the new model in order to more closely match values obtained with the DBSIMSD.

TABLE I

RTS5D and DBSIM5D Performance Test at 40 Knots

effector angle	<u>DBSIM5D</u>		<u>RTS5D</u>	
	ϕ	in degrees	ϕ	
	1st pk	Qss	1st pk	Qss
5°	1.01	.742	.17	.12
10°	1.46	1.39	.31	.24
	ϕ	in degrees	ϕ	
	1st pk	Qss	1st pk	Qss
5°	1.44	1.40	1.27	1.21
10°	1.48	1.48	1.28	1.20
	u	in ft/sec	u	
	1st pk	Qss	1st pk	Qss
5°	67.4	70.9	n/a	67.03
10°	n/a	66.3	n/a	65.98
	v	in ft/sec	v	
	1st pk	Qss	1st pk	Qss
5°	1.42	1.37	2.84	2.42
10°	3.20	3.26	3.83	3.41
	r	in degrees/sec	r	
	1st pk	Qss	1st pk	Qss
5°	.396	.403	.58	.33
10°	.953	.959	.98	.66

TABLE II

RTS5D and DBSIM5D Performance Test at 50 Knots

effector angle	<u>DBSIM5D</u>		<u>RTS5D</u>	
	ϕ	in degrees	ϕ	
	1st pk	Qss	1st pk	Qss
5°	.424	.446	.27	.19
10°	.757	.886	.49	.38
15°	1.02	1.4	.72	.57
	θ	in degrees	θ	
	1st pk	Qss	1st pk	Qss
5°	.862	.892	1.16	1.15
10°	.943	1.13	1.15	1.16
15°	1.12	1.35	1.22	1.17
	u	in ft/sec	u	
	1st pk	Qss	1st pk	Qss
5°	n/a	83.6	n/a	83.8
10°	n/a	78.7	n/a	82.51
15°	n/a	69.2	n/a	80.58
	v	in ft/sec	v	
	1st pk	Qss	1st pk	Qss
5°	1.31	1.30	3.55	3.02
10°	2.72	2.70	4.79	4.26
15°	4.05	4.33	5.69	5.207
	r	in degrees/sec	r	
	1st pk	Qss	1st pk	Qss
5°	.368	.365	.73	.41
10°	.864	.870	1.22	.82
15°	1.35	1.44	1.66	1.26

TABLE III

RTS5D and DBSIM5D Performance Test at 60 Knots

effector angle	<u>DBSIM5D</u>		<u>RTS5D</u>	
	ϕ	in degrees	ϕ	
3°	1st pk	Qss	1st pk	Qss
	.141	.200	.39	.274
5°	θ	in degrees	θ	
	1st pk	Qss	1st pk	Qss
	.521	.583	1.14	1.07
5°	u	in ft/sec	u	
	1st pk	Qss	1st pk	Qss
	n/a	97.8	n/a	100.29
5°	v	in ft/sec	v	
	1st pk	Qss	1st pk	Qss
	2.03	1.84	4.26	3.62
5°	r	in degrees/sec	r	
	1st pk	Qss	1st pk	Qss
	.532	.494	.87	.487

IV. MODEL DEVELOPMENT

A. DRAG CHARACTERISTICS

1. Surge Equation Model

The RTS5D model assumed a velocity squared drag retarding characteristic for the 3K-SES which led to the equation for surge acceleration that was presented in Chapter II. The drag characteristics from Ref. 7 for the 3K-SES that is simulated by the DBSIM5D are shown in Fig. 5. The data base program was run using parameters that caused it to exhibit a retarding force corresponding to the curve at sea state four, therefore this curve was used to remodel the RTS5D. The new simplified equation of motion for surge acceleration became

$$\dot{u} = (T_{\text{forw}}/m) - (FSD/m) + v \cdot r$$

where FSD is full scale drag which is defined by the curve in Fig. 5. T_{FORW} is the summation of the four thrusts assumed in this model to be applied at the same point along the center line of the craft.

2. Implementation

FSD was implemented into the RTS5D model by dividing the curve into four separate regions and defining FSD within each region. In region I the retarding force was approximated by a normalized velocity cubed term. In regions II

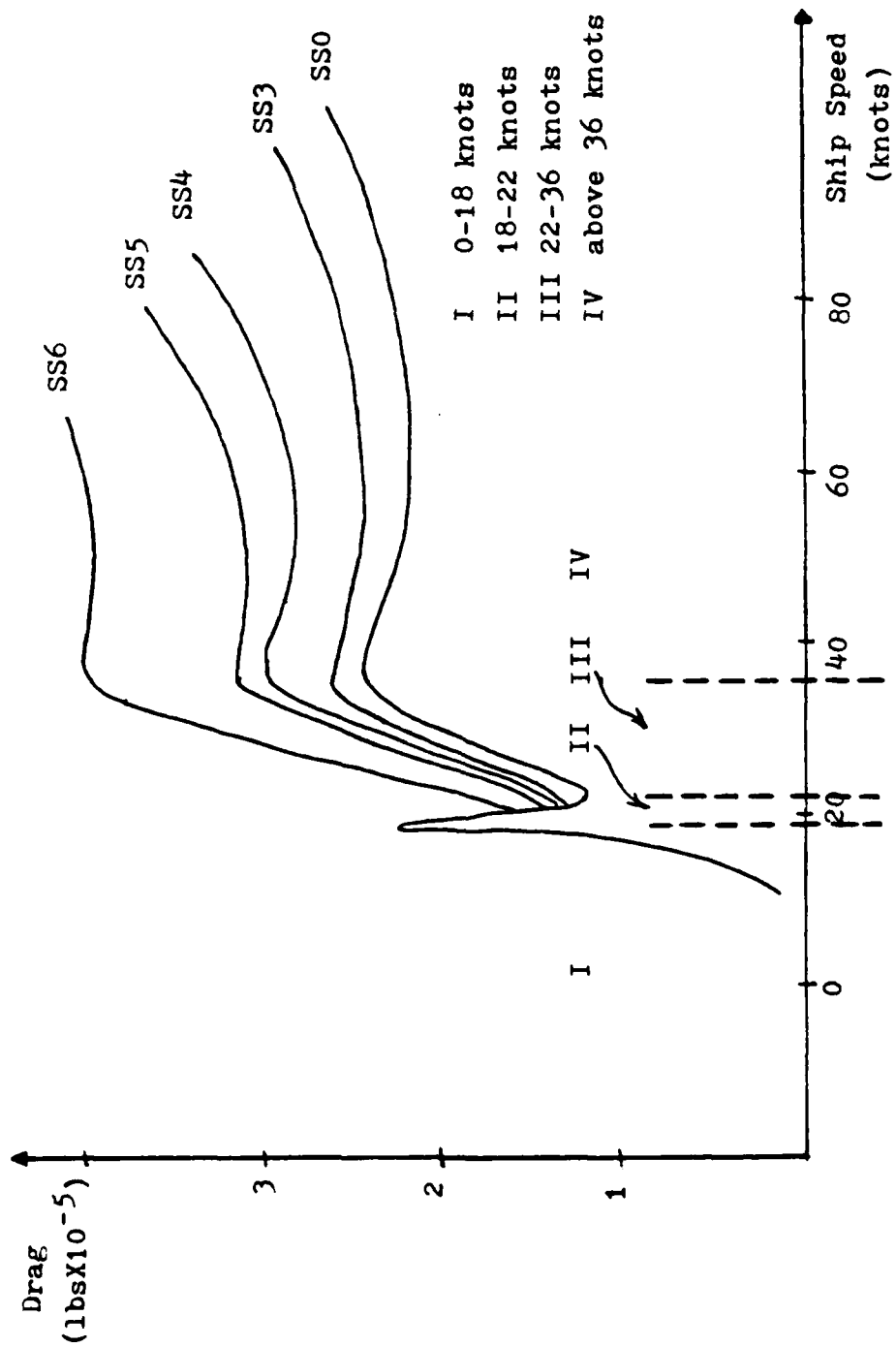


Figure 5
Full Scale Drag Curves

and III straight line approximations were used. In region IV a polynomial curve fit was used. The equations defined for each region are listed below.

Region I - (0-18 knots)

$$FSD = C_{DX1} (u / u_{max})^3$$

Region II - (18-22 knots)

$$FSD = CONSTANT_1 - C_{DX5} u$$

Region III - (22-36 knots)

$$FSD = C_{DX2} u - CONSTANT_2$$

Region IV - (above 36 knots)

$$FSD = C_{DX3} u^2 + C_{DX4} / u^{1.5}$$

3. Identification of Surge Drag Coefficients

In region I, C_{DX1} is equal to the first peak value of drag. In regions II and III, C_{DX5} and C_{DX2} are equal to the slopes of the drag curve and the constants were solved for by applying the boundary conditions at the intersections of the adjacent regions. The values of C_{DX3} and C_{DX4} were computed by taking FSD at two different known values of u and solving for C_{DX3} and C_{DX4} simultaneously.

B. LINEARIZED MODEL ANALYSIS

Since the surge model of the RTS5D is extremely non-linear over the complete speed range, analysis is very complex. Therefore the model was linearized and it's performance

analyzed at various points on the FSD curve. Linearization of the model was accomplished by computing the linear coefficients through partial differentiation with respect to surge in each of the four region equations and evaluating these terms at various operating point surge velocities (u_o). The linearized model is shown in Fig. 6 and the linear coefficients for the surge drag force FSD are shown below.

Region

$$\text{I} \quad \text{FSD}'_{(u_o)} = 3 C_{DX1} (u_o / u_{\max})^2$$

$$\text{II} \quad \text{FSD}'_{(u_o)} = - C_{DX5}$$

$$\text{III} \quad \text{FSD}'_{(u_o)} = C_{DX2}$$

$$\text{IV} \quad \text{FSD}'_{(u_o)} = 2 C_{DX3} u_o - 1.5 C_{DX4} / u_o^{2.5}$$

A map of the linearized surge directional roots are shown in Fig. 7. The characteristic roots for each operating region are shown coded (see legend). Note that only Region I and Region IV roots are u_o dependent and therefore move as u_o changes. The direction of change along the real axis is noted by the arrows shown. Region II and III roots are fixed in locations and independent of u_o .

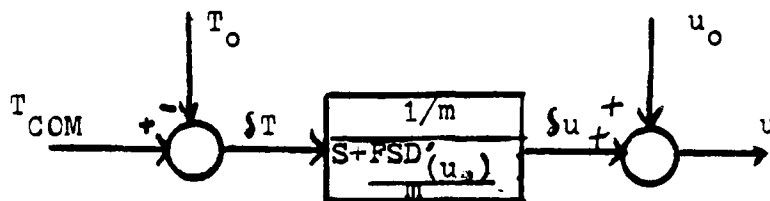


Figure 6
Linearized Surge Model

Region IV of Fig. 7 is of particular interest since the ability to operate in this speed range is a primary consideration in the development of the SES. It can be seen from the map of the roots that within this region, at speeds from 36 to almost 60 knots, poles exist in the RHP. This is a result of the negative drag slope where the retarding force actually decreases with an increase in speed, the magnitude of variation being dependent on sea state. Region II also exhibits this negative drag characteristic but since it is of such a narrow speed range, operation in this region was not considered in this design. Further modeling changes for the other degrees of freedom were based on Region IV response. In Regions I, III and in Region IV above 60 knots, the system will be stable at the speed u_o where FSD is equal to commanded thrust.

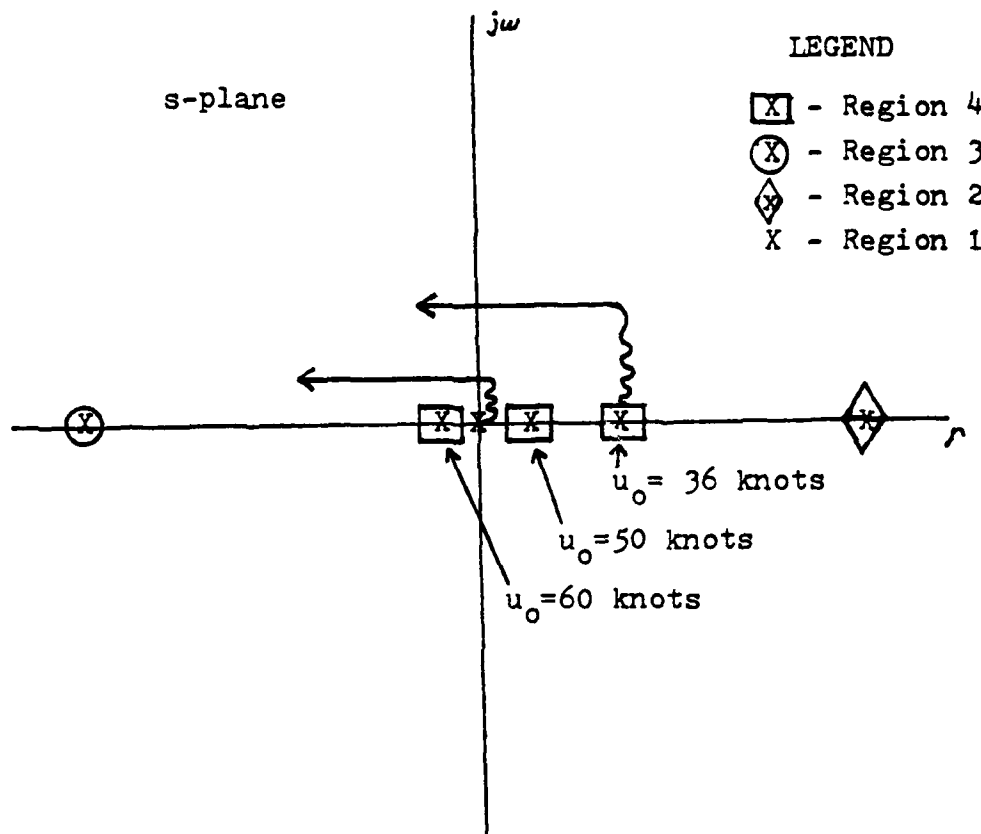


Figure 7
Linearized Surge Root Map

If it were necessary to operate at a speed within the negative drag slope characteristic, it would be necessary to actively modulate the thrust command in order to maintain a nearly constant speed, but this would require a great deal of operator attention since the negative drag slope in Region IV is one of slowly divergent instability. For this reason a closed loop speed controller has been included in the data base model and therefore was also added to the RTS5D model.

C. SIMPLIFIED PROPULSION DYNAMICS

Before adding a speed autopilot to the model one additional refinement was included in the design. Thrust command changes in the RTS5D as reported in Ref. 1 were instantaneously entered into the equations of motion as thrust changes. This was not the case for the data base simulation program. Specific thrust levels are generated by specific water flow rates through water jet nozzles. These flow rates are nearly proportional to the pump speed that produces them. Since the pumps are driven directly by gas turbine engines, a change in thrust occurs with a change in turbine speed. A thrust command translates to a power turbine speed command which as an output has power turbine speed achieved. This output is translated into thrust achieved. It was specified in Ref. 7 that one of the goals in the base line design was to have power turbine speed control response characteristics which are relatively independent of the power setting. Accordingly, in the design

of the auto pilot of Ref. 7, the simplified propulsion block shown in Fig. 8 was used. The magnitude of the propulsion system time constant, τ was not specified. For the purpose of this analysis τ was chosen to be 1 second.

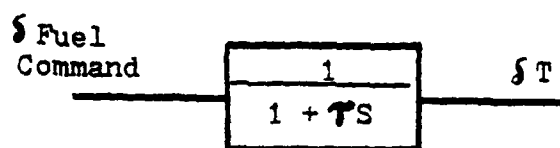


Figure 8
Simplified Propulsion System

D. ANALYSIS OF LINEARIZED MODEL WITH SPEED CONTROLLER

The complete linearized model with speed controller that is included in the model for the purposes of this study is shown in Fig. 9. The saturation effects of fuel control are ignored and the linear representation is shown as K_f which is assumed to be unity. The controller includes proportional plus integral control. The proportional control was included to keep the ship speed error low while the integral control was included in order to zero out long term control error.

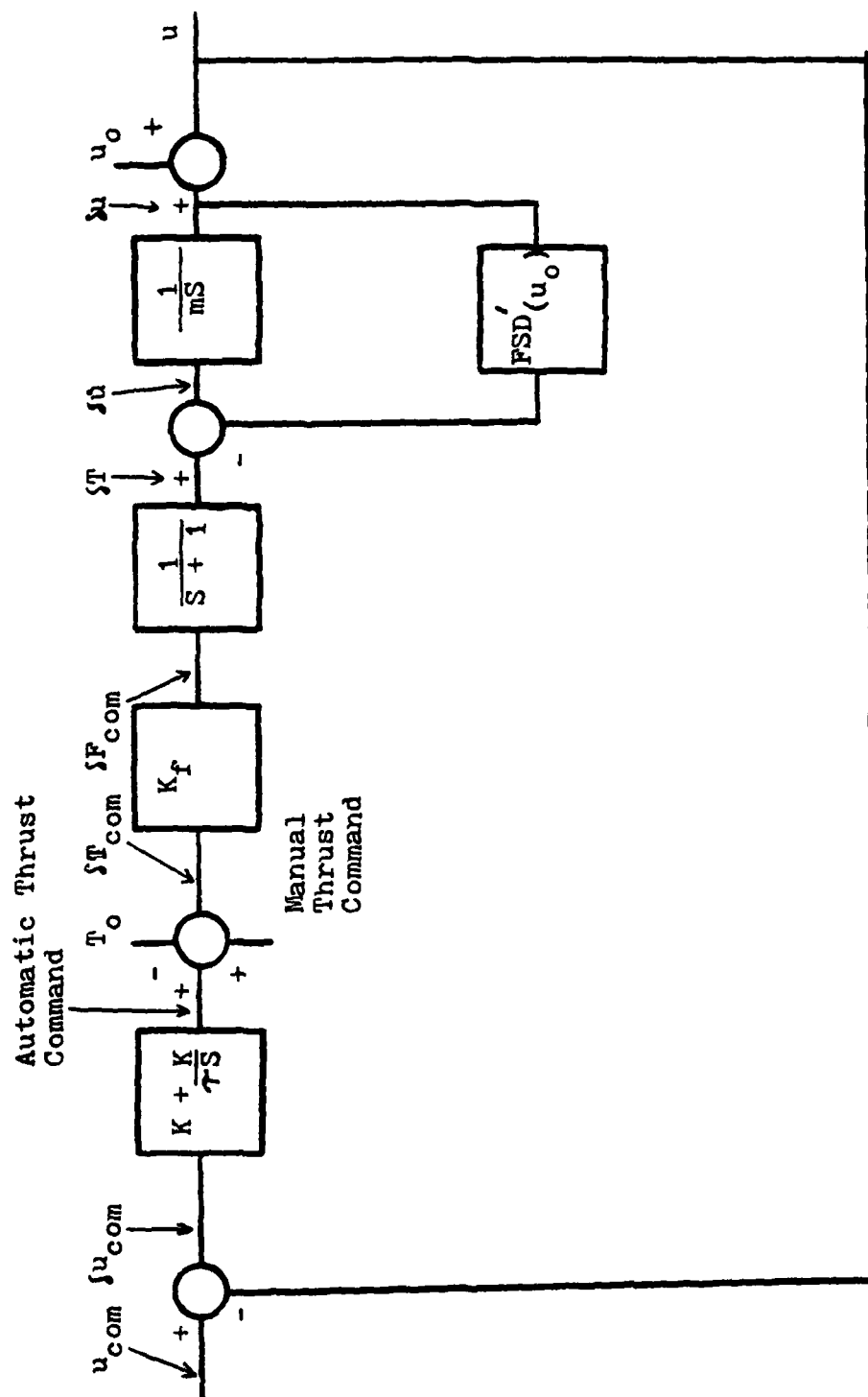


Figure 9
Linearized Surge Model With
Propulsion System and Speed Controller

The characteristic equation for the system in Fig. 9 is

$$s^3 + (1 + \text{FSD}'_{(u_o)}/m) s^2 + 1/m (\text{FSD}'_{(u_o)} + K) s + K/m\tau = 0$$

Using the Routh Hurwitz Criterion for stability

s^3	1	$1/m (\text{FSD}'_{(u_o)} + K)$
s^2	$(1 + \text{FSD}'_{(u_o)}/m)$	$K/m\tau$
s^1	$1/m (1 + \text{FSD}'_{(u_o)}/m) (\text{FSD}'_{(u_o)} + K) - K/m\tau$	0
s^0	$K/m\tau$	0

For Stability K must be greater than 0 and greater than

$$\frac{\text{FSD}'_{(u_o)}{}^2 \tau + m\tau \text{FSD}'_{(u_o)}}{\text{FSD}'_{(u_o)} \tau + m\tau - m}$$

The values of controller parameters K and τ can be initialized by the operator for either loosely controlled speed or tightly controlled speed for such situations as station keeping, UNREPS, or test operations. For comparison to the data base, these parameters were selected to achieve closest agreement between the values of surge achieved by the RTS5D and DBSIM5D models after the completion of a 360° turn with 50 knot initial surge velocity and a thrust effector angle of 15°. This test

condition for the nonlinear models was chosen for two reasons. First, given fuel conservation as a necessity, 50 knots is a likely high speed initial condition because it is close to minimum drag. Second, this initial condition provided an opportunity to test the operation of the speed controller in the negative drag region of FSD since the decrease in speed going into the turn with $u_0=50$ knots caused increased drag. The values of controller parameters that caused closest agreement between the two nonlinear models, when tested in the linearized model showed instability in the linear sense. This is desirable in a turning mode for the nonlinear model because it allows speed to decrease and the turn to be completed more quickly. Stiff speed control could be achieved by increasing the controller gains, but would normally be used only in straight ahead runs.

Coupling of the equations of motion developed in Chapter II necessitated a repetition of the parameter identification procedure resulting in new values for C_{DY} , Δw , C_{DP} and C_{DZP} . These values were adjusted for the best curve fit compared to the data base using the linearized model for the RTS5D in a 360° turn using 15° thrust effector angle with initial surge velocity 50 knots. An identical series of tests to those in Chapter III were conducted comparing the new model of the RTS5D to the DBSIM5D. Complete validation results are tabulated in Chapter VI.

E. EFFECTIVE THRUST EFFECTOR ANGLE

As a result of the negative drag characteristics imposed on the model, RTS5D yaw rate (r) was high in the low speed test and low in the high speed test. This opposite error at both ends of the test speed spectrum was reduced by introducing a correction factor to the thrust effector angle proportional to yaw rate

$$Z_{\text{eff}} = Z - \text{SLIP} \cdot r$$

and adjusting Δw to increase r . This had the desired effect of reducing error at both ends of the test spectrum since the reduction of the thrust effector angle was greater at the high end.

F. BROACH CONDITION FLAG

As stated in Chapter I, broach response characteristics are not included in this model, therefore a broach condition warning flag was included in the design in order to alert the operator who could then take corrective action to avoid broaching.

Broaching is caused by a water intake being exposed to air and is a function of surge speed, roll angle, pitch angle, drift angle and sea state. For the data base 3K-SES with broach boundaries defined at 10% air by volume, the broaching boundaries in calm water in terms of roll angle, pitch angle and drift angle are shown in Fig. 11 for 40 knots

and Fig. 12 for 60 knots. Reference 6 listed maximum thrust effector angles as a function of surge speed and air flow in order to avoid broaching. This condition is shown plotted in Fig. 10 for fixed plenum air flow rate.

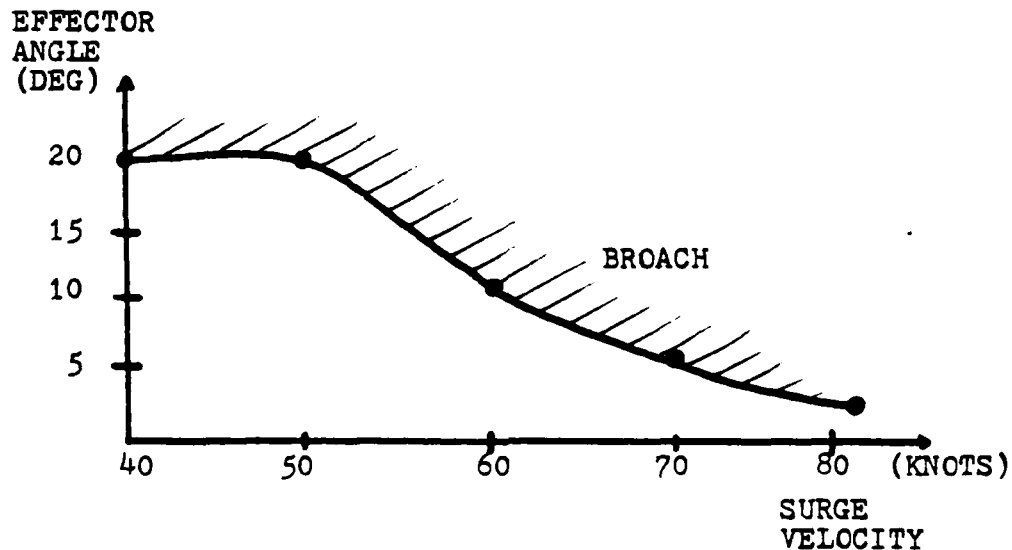


Figure 10

Maximum Thrust Effector Angle As A Function
Of Surge Velocity In Order To Avoid Broaching

As stated in Chapter III the model developed in Ref. 1 did not flag the broaching condition. To implement the broach warning flag and display the other new parameters for the modified RTS5D the pilot graphic display described in Ref. 1 was changed to incorporate the necessary data. Figure 13A depicts typical information displayed to the pilot under normal operating conditions. The new information to the

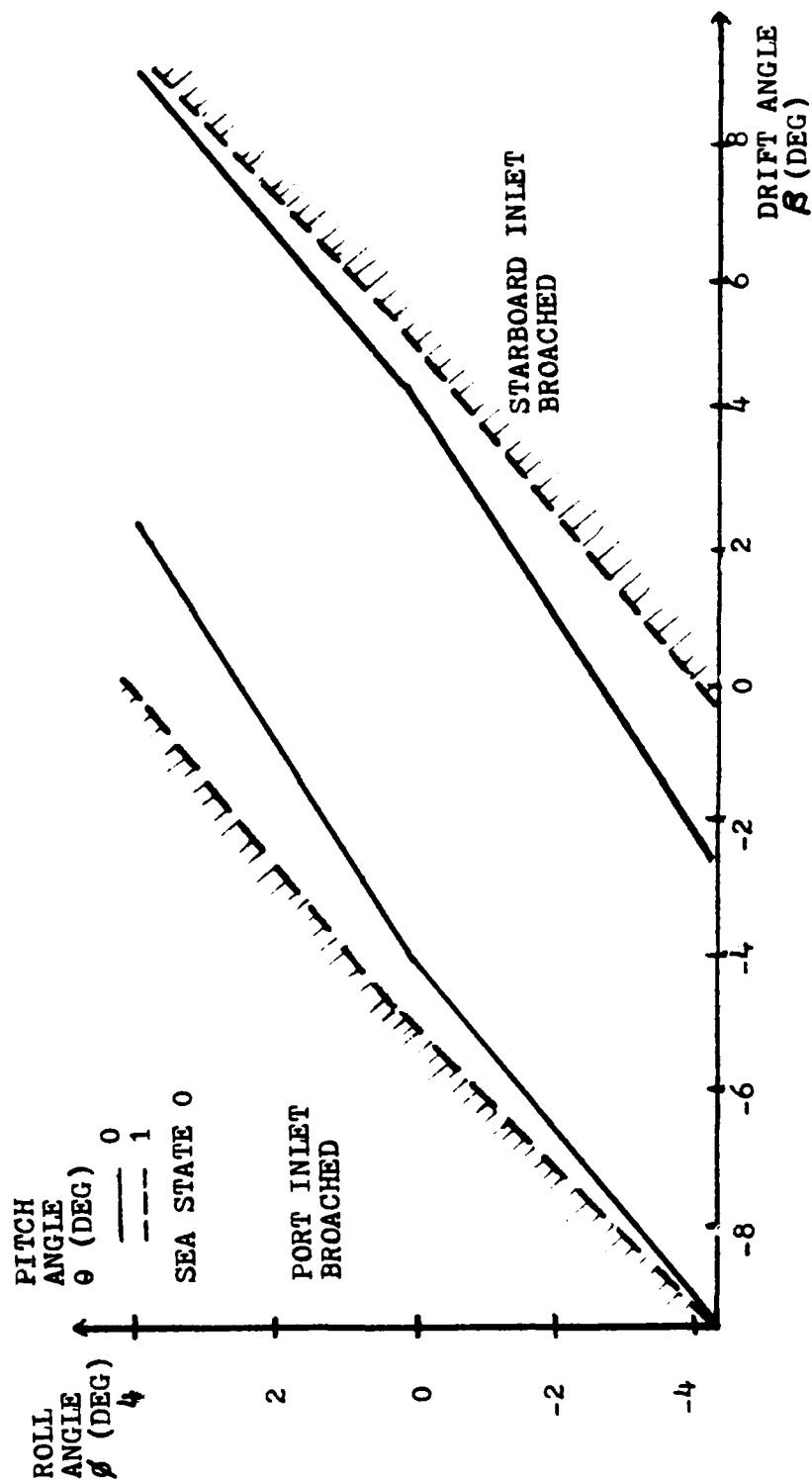


Figure 11
Inlet Broaching Boundaries at 40 Knots

operator shown in the lower right of Fig. 13A is sea state (SEAST), computer assist (CASST, on=1 off=0), controller gains (K, KK) and speed commanded (SPCOM). If the operator exceeds the thrust effector limits (ZMAX) to prevent broaching (see Fig. 10), a flashing warning is displayed in place of the bottom line of information on the screen. This is shown in Fig. 13B.

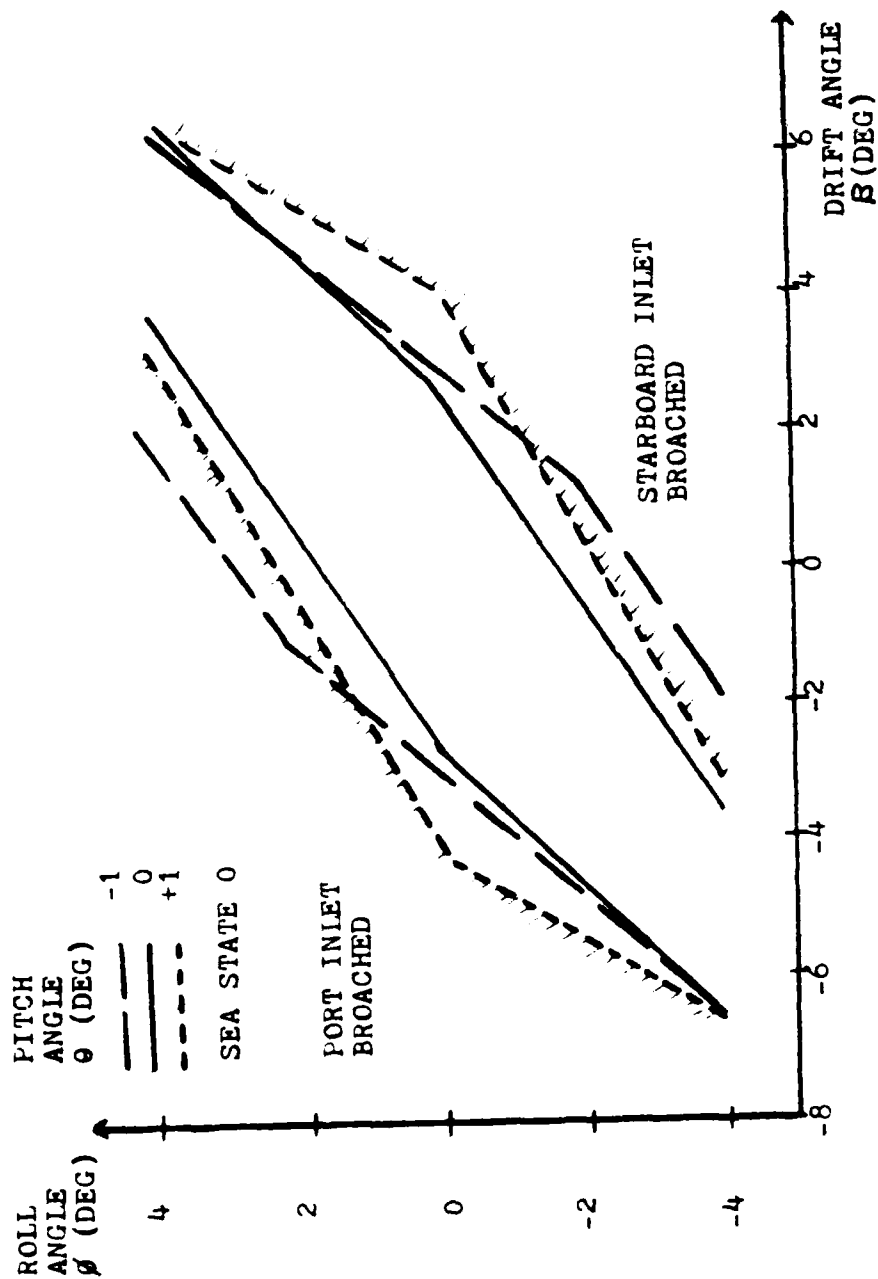
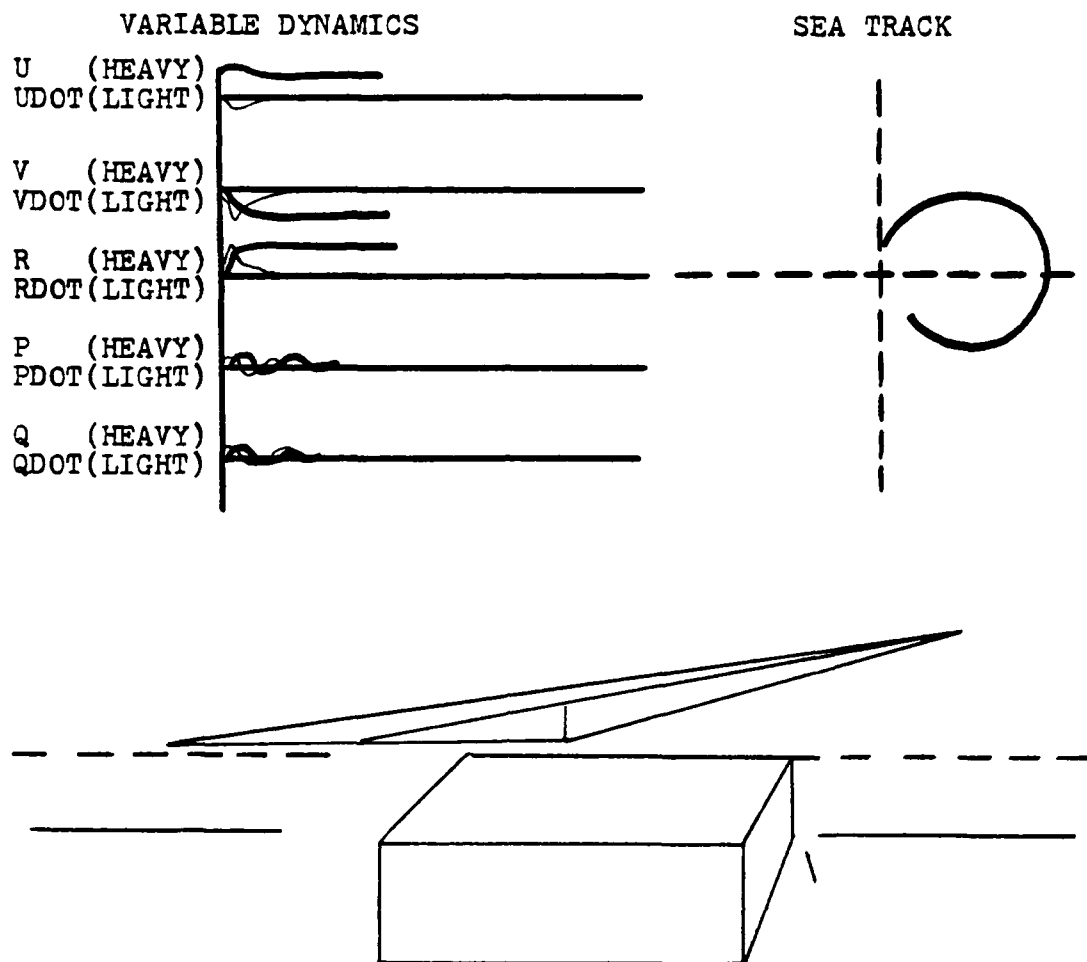
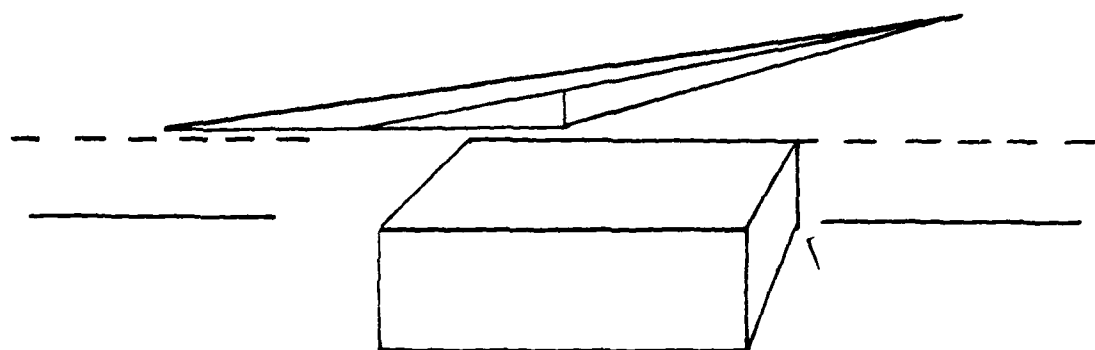
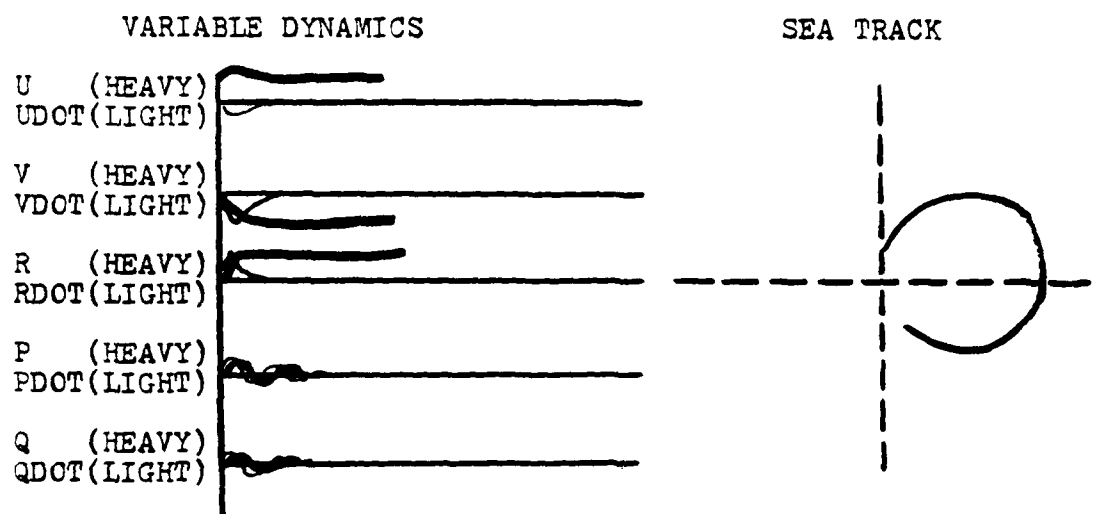


Figure 12
Inlet Broaching Boundaries at 60 Knots



CONTROL INPUTS			TIME		NAVIGATIONAL DATA				
THRUST	RUDDER	DELT	ELAPSED	PITCH	HEADING/RATE	DRIFT	ROLL	SPEED	
300000	-.20	.084	22.2	.52	325/2.1	-.28	.8	50.1	
T7	T8	T9	T10	SEAST	CASST	K	KK	SPCOM	
75000	75000	75000	750000	4	1	2900	3	50	

Figure 13A
Pilot Graphic Display (Normal)



CONTROL INPUTS			TIME	NAVIGATIONAL DATA				
THRUST	RUDDER	DELT	ELAPSED	PITCH	HEADING/RATE	DRIFT	ROLL	SPEED
3000000	-.52	.084	22.2	.52	325/2.1	-.28	.8	50.1
T7	T8	T9	T10	SEAST	CASST	K	KK	SPCOM
≡ B R O A C H ≡								

Figure 13B

Pilot Graphic Display (Broach Condition Exists)

V. IMPLEMENTATION

A. INTRODUCTION

Given the existence of a Real Time Simulation of a SES, a decision was made to refine that model using more accurate equations of motion, improved computer iteration time and operator hardware modifications. A complete description of the RTS5D is contained in Ref. 1, therefore only changes are discussed here, and included are the introduction of negative drag in a sea state, propulsion dynamics, speed control and a broach condition warning in software. Hardware modification included changing the thruster control box to make it more accessible to an operator.

B. REQUIREMENTS

The following criteria were required of the remodeled RTS5D.

- 1) The refined simplified equations of motion would be solved and the results output on a real time basis.

- 2) The solution would be subject to real time control efforts generated by an operator observing the output.

- 3) Computer iteration time was required to be less than 100 ms.

A more complex discussion of condition 3) is presented in Appendix A.

C. HARDWARE DESCRIPTION

The hardware arrangement in block diagram form is shown in Fig. 14. The only change to the RTSSD was the redesign and relocation of the thruster console to the pilots seat where it can be more conveniently utilized by the operator. The thruster console uses "linear movement" potentiometers in place of previously used "dial" potentiometers in order to more realistically simulate thrust controls for the SES. The thruster console is shown in Fig. 15.

D. SOFTWARE DESCRIPTION

The remodeled software package was a FORTRAN IV digital program which consisted of a main graphics program and three major subroutines. It was necessary to make a third subroutine out of the equations of motion computation that had previously resided in the main program prior to making any changes to these equations. It was discovered when the first program modifications were attempted that the compiler for the XDS-9300 computer was operating at its limit with the program as it then existed, therefore, before changes to the equations could be implemented, the program structure had to be modified. The failure mode was eliminated from the new model because of initial assumptions that the thrusts would be all applied at the same point. Thrust failures should not be considered until the model includes broaching dynamics which is a major source of thrust loss. The flow

chart for the remodeled RTS5D is shown in Fig. 16. The complete multiplexing algorithm is shown in a flow chart in Fig. 17.

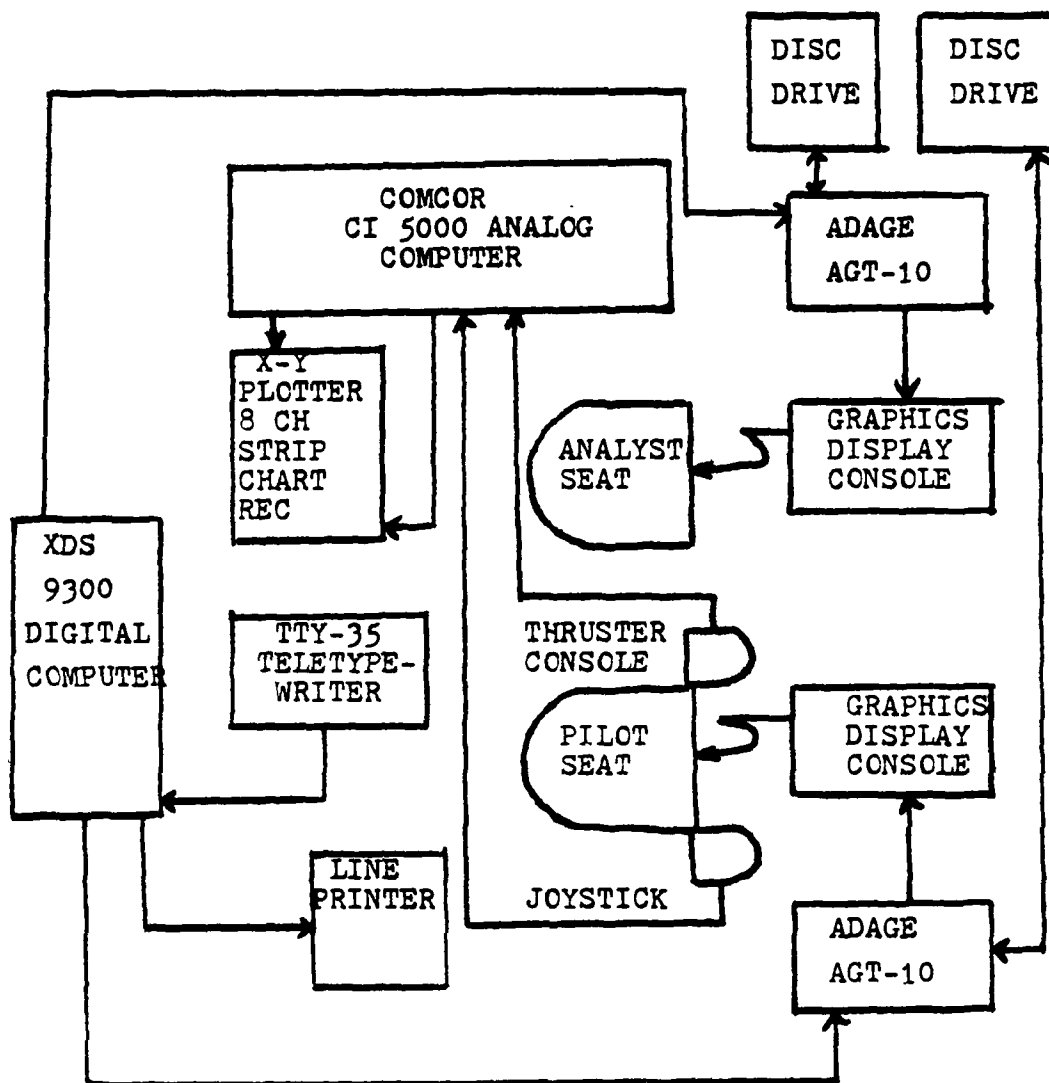
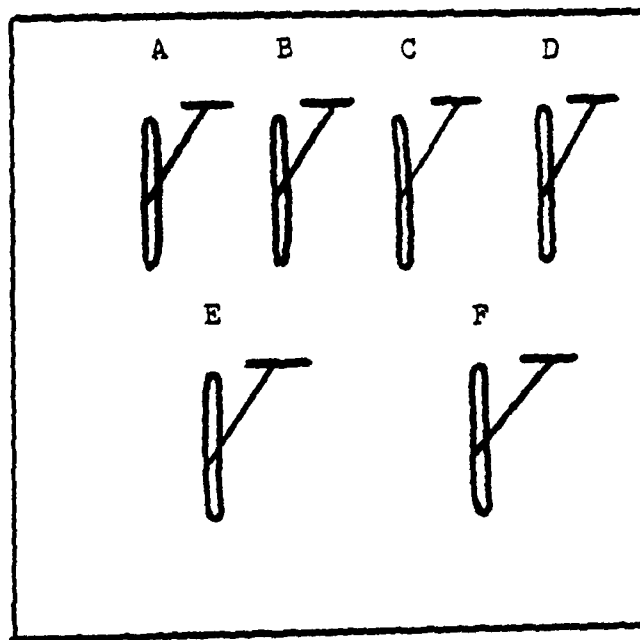


Figure 14
RTS5D Mod Block Diagram



- A - Thruster T7
- B - Thruster T8
- C - Thruster T9
- D - Thruster T10
- E - Speed Command Control
- F - Run, Hold, Restart

Figure 15

Thruster Console

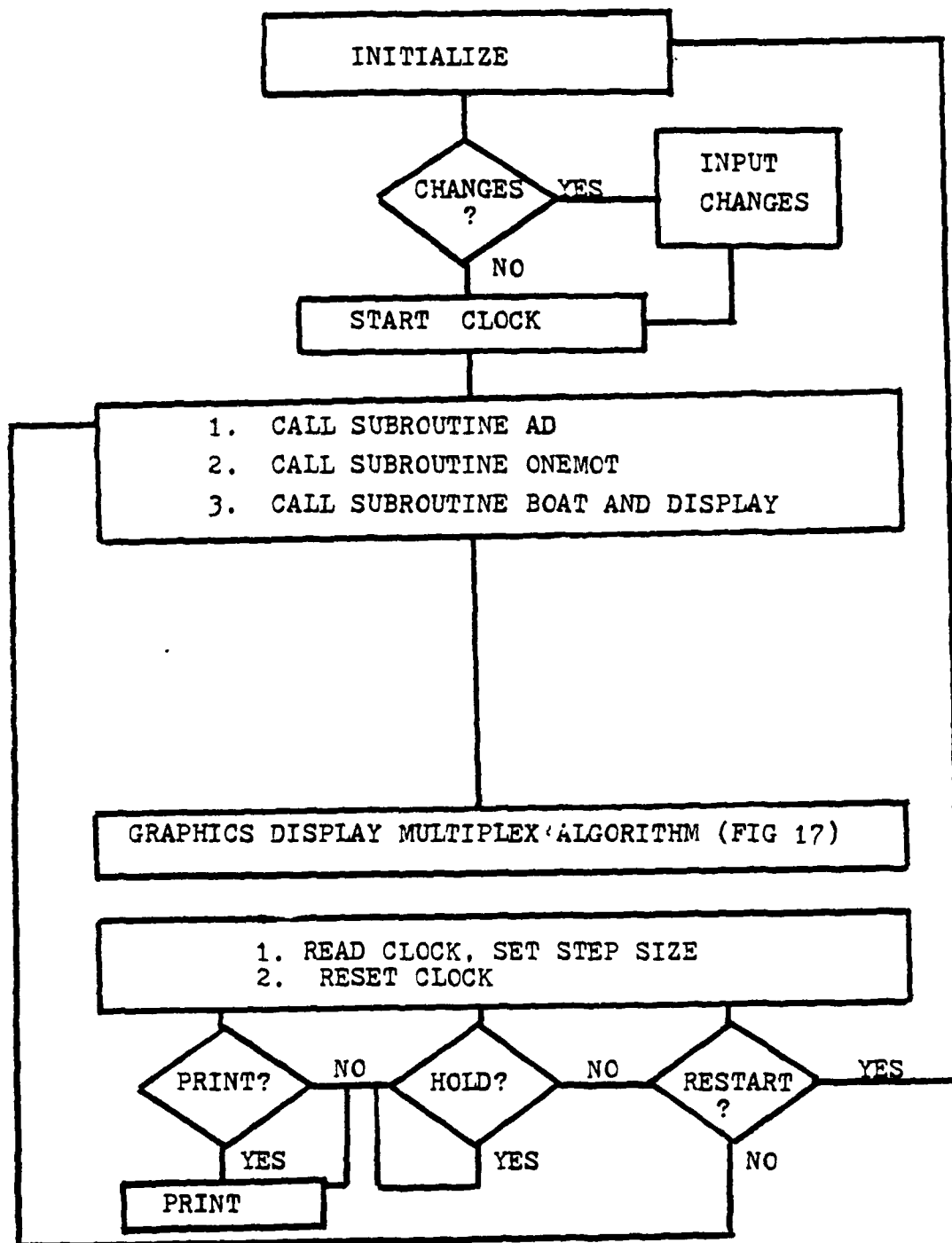
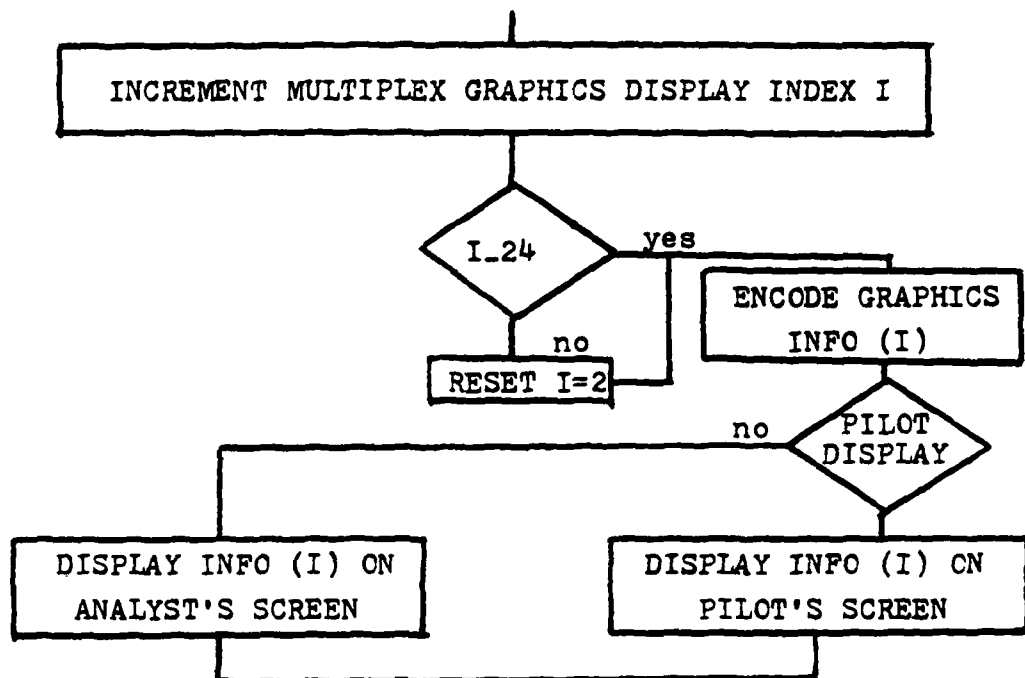


Figure 16

RTS5D Mod Program Flow Chart



DESCRIPTION OF INFO(I)

INFO(1) u DYNAMICS	INFO(13) VELOCITY MAX VALUES
INFO(2) $\dot{u}$ DYNAMICS	INFO(14) VELOCITY MIN VALUES
INFO(3) v DYNAMICS	INFO(15) ACCELERATION MAX VALUES
INFO(4) $\dot{v}$ DYNAMICS	INFO(16) ACCELERATION MIN VALUES
INFO(5) r DYNAMICS	INFO(17) NAVIGATION DATA
INFO(6) $\dot{r}$ DYNAMICS	INFO(18) EFFECTOR/THRUSTER DATA
INFO(7) p DYNAMICS	INFO(19) SEA TRACK NAV PLOT
INFO(8) $\dot{p}$ DYNAMICS	INFO(20) SEA TRACK NAV PLOT
INFO(9) q DYNAMICS	INFO(21) PRESENT POSITION VARIABLES
INFO(10) $\dot{q}$ DYNAMICS	INFO(22) PRESENT VELOCITY VAREABLES
INFO(11) POSITION MAX VALUES	INFO(23) PRESENT ACCELERATION VAR.
INFO(12) POSITION MIN VALUES	

Figure 17

RTS5D MOD DISPLAY MULTIPLEX ALGORITHM

VI. RTS5D MODS I AND II RESPONSE CHARACTERISTICS

RTS5D MOD I includes negative drag effects in a sea state, simplified propulsion dynamics, speed control and a broach condition warning flag. RTS5D MOD II is identical to MOD I except that it includes the correction factor for thrust effector angle developed in Chapter IV, Section E.

The validation of the modified RTS5D was accomplished in the same manner as the RTS5D. (See Chapter III). The results are displayed in the following tables. A complete 360° turn comparison of the DBSIM5D, the RTS5D and the modified RTS5D at 50 knots, 15° thruster angle is shown in Fig. 18. Additionally, a comparison of transient response was undertaken and the time of first peak for the variable v , r , $\dot{\phi}$ and θ for each model was graphed in Figs. 19-21 for speeds of 40, 50 and 60 knots. Transient response characteristics are an important consideration in the development of a real time model and are discussed in Appendix A. The transient response of the RTS5D models was faster than that of the data base model with the exception of roll angle (ϕ). This impacted on the validity of a real time solution of the equations of motion and the display of the results.

TABLE IV

RTS5D Mod I and Mod II Performance Test at 40 Knots

effector angle		<u>RTS5D Mod I</u>		<u>RTS5D Mod II</u>	
		ϕ	in degrees	ϕ	
	1st pk		Qss	1st pk	Qss
5°	.52		.45	.40	.38
10°	.96		.92	.79	.76
		θ	in degrees	θ	
	1st pk		Qss	1st pk	Qss
5°	1.21		1.17	1.20	1.18
10°	1.20		1.18	1.20	1.18
		u	in ft/sec	u	
	1st pk		Qss	1st pk	Qss
5°	62.56		62.64	62.58	67.65
10°	n/a		61.37	n/a	61.71
		v	in ft/sec	v	
	1st pk		Qss	1st pk	Qss
5°	2.62		2.47	2.33	2.27
10°	3.53		3.52	3.23	3.20
		r	in degrees/sec	r	
	1st pk		Qss	1st pk	Qss
5°	.69		.59	.57	.54
10°	1.18		1.22	1.04	1.09

TABLE V

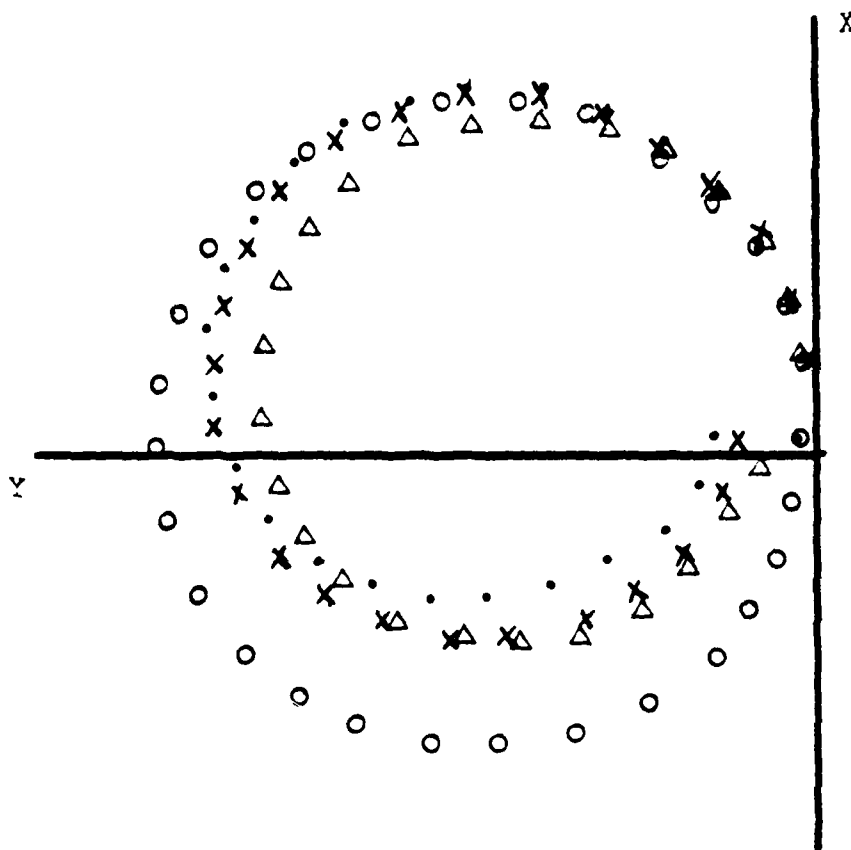
RTS5D Mod I and Mod II Performance Test at 50 Knots

effector angle	<u>RTS5D Mod I</u>		<u>RTS5D Mod II</u>	
	ϕ	in degrees	ϕ	
	1st pk	Qss	1st pk	Qss
5°	.50	.39	.43	.39
10°	.90	.82	.83	.78
15°	1.31	1.36	1.24	1.19
	θ	in degrees	θ	
	1st pk	Qss	1st pk	Qss
5°	1.23	1.19	1.23	1.20
10°	1.24	1.19	1.23	1.20
15°	1.24	1.18	1.23	1.19
	u	in ft/sec	u	
	1st pk	Qss	1st pk	Qss
5°	n/a	82.73	n/a	82.98
10°	n/a	78.68	n/a	79.55
15°	n/a	68.30	n/a	72.75
	v	in ft/sec	v	
	1st pk	Qss	1st pk	Qss
5°	2.57	2.30	2.39	2.29
10°	3.48	3.33	3.25	3.25
15°	4.13	4.29	3.98	4.00
	r	in degrees/sec	r	
	1st pk	Qss	1st pk	Qss
5°	.57	.39	.49	.41
10°	.96	.85	.88	.86
15°	1.31	1.62	1.25	1.41

TABLE VI

RTS5D Mod I and Mod II Performance Test at 60 Knots

effector angle	<u>RTS5D Mod I</u>		<u>RTS5D Mod II</u>	
	ϕ	in degrees	ϕ	
5°	1st pk .53	Qss .39	1st pk .46	Qss .41
5°	1st pk 1.23	θ Qss 1.20	1st pk 1.24	θ Qss 1.20
5°	1st pk n/a	u Qss 100.21	1st pk n/a	u Qss 99.59
5°	1st pk 2.64	v Qss 2.29	1st pk 2.49	v Qss 2.35
5°	1st pk .52	r Qss .32	1st pk .46	r Qss .36



$u_0 = 50$ knots
 $z = 15^\circ$ (effector angle)
 10 second intervals

○ RTS5D
 △ DBSIM5D
 • RTS5D MOD I
 × RTS5D MOD II

Figure 18
 360° Turn Comparison of Four Models

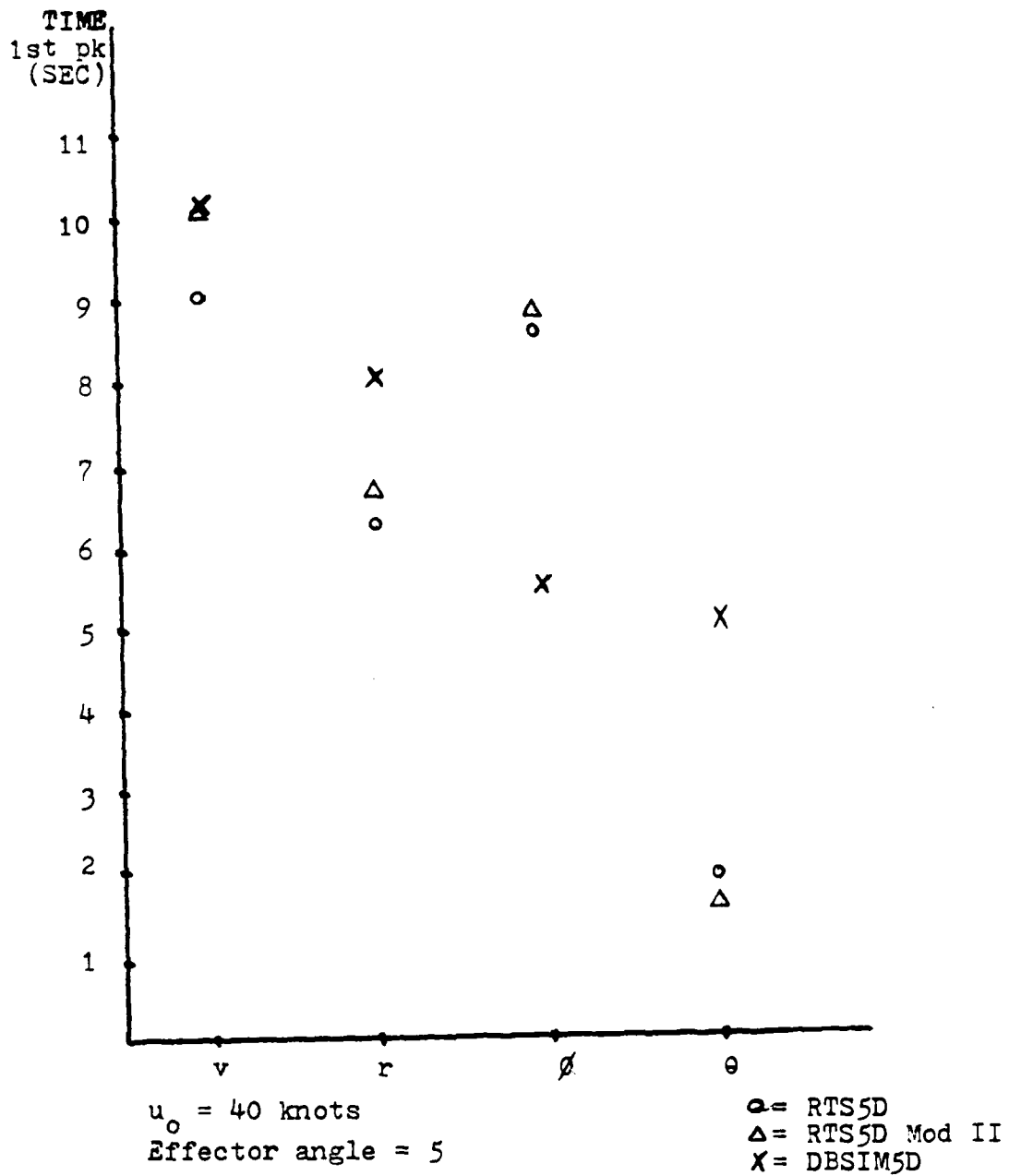


Figure 19
Response Time Comparison at 40 Knots

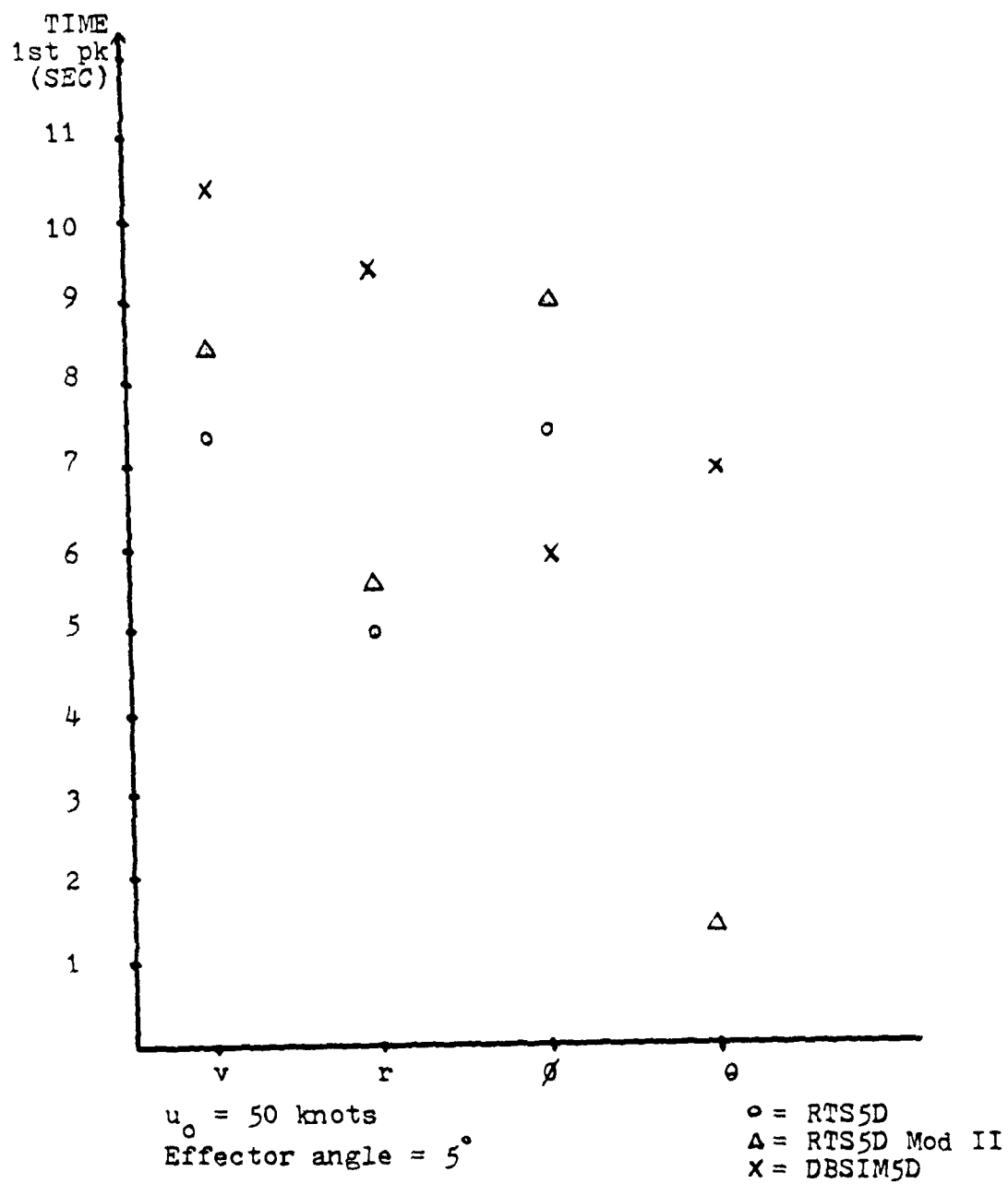


Figure 20
Response Time Comparison at 50 Knots

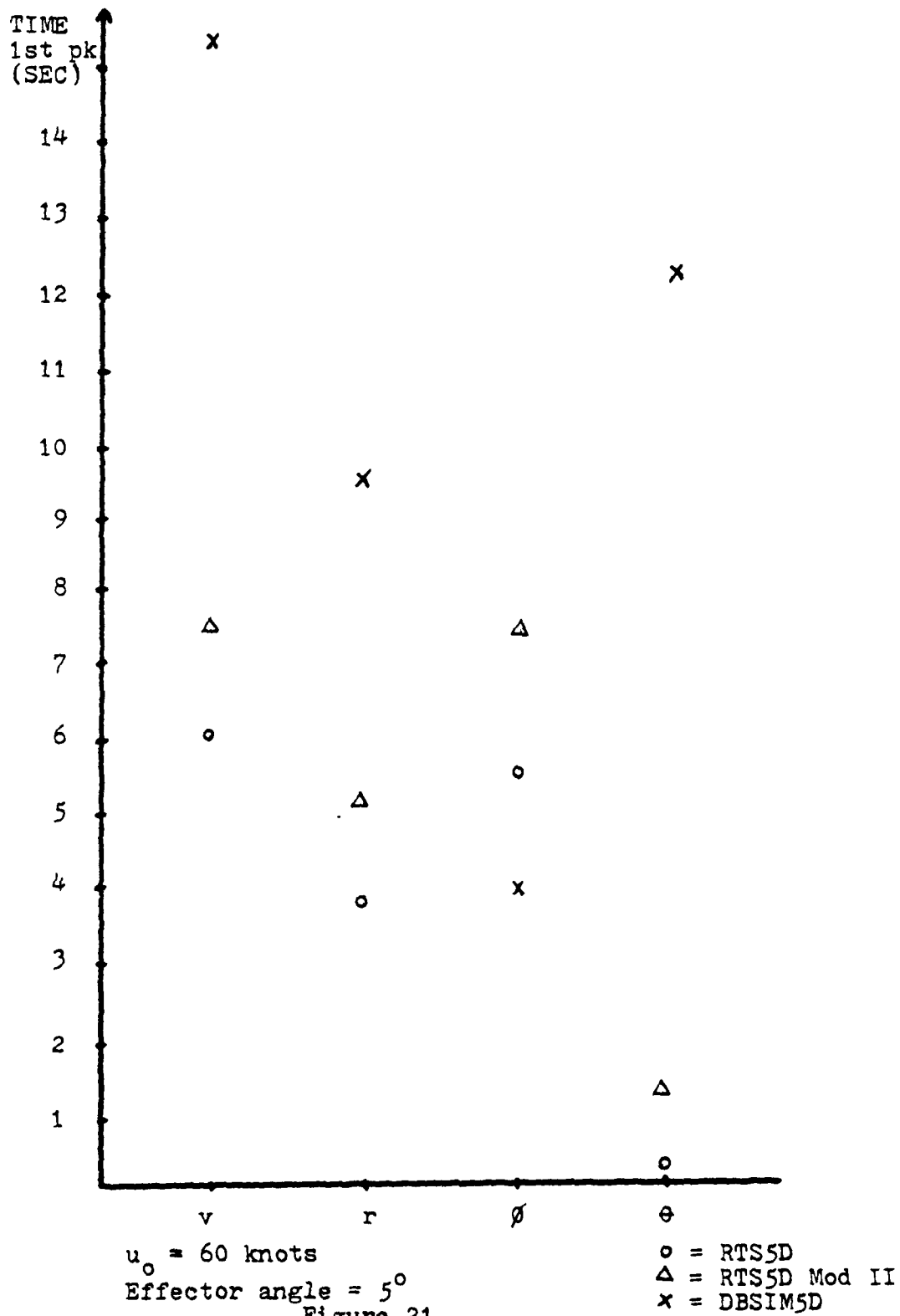


Figure 21
Response Time Comparison at 60 Knots

VII. CONCLUSIONS

The RTS5 D has been shown in Section VI to be a viable input/output model of a 3K-SES by closely matching its output characteristics for a given input to those of the data base model.

The refinement of the right side of the simplified equations of motion, especially the inclusion of the negative drag characteristic in a sea state has made the model more accurate over a wide speed range and gives the operator, upon program initialization, the ability to choose different drag characteristics caused by sea state.

The model was tested in a negative drag region with automatic speed control and simplified propulsion dynamics and it was demonstrated that by properly adjusting controller gains that the output characteristics of the DBSIM5 D were closely matched.

Previously conducted thrust failure analysis with man generated corrective action has been shown to be invalid until broach dynamics are included into the model. Previous tests allowed maximum thrust effector angle correction regardless of this consideration. The modified version of the RTS5 D does not include complete broach dynamics but does warn the operator if he should exceed a maximum thrust effector angle for a given forward speed.

The 3K-SES may be operated in the negative drag slope speed region approximately (36-58) knots without the aid of a speed controller but would require constant manual modulation of the thrusters. Since minimum manning is one of the criteria in 3K-SES design, a speed controller is necessary to free operator attention to other duties when operating in this speed range.

Another consideration of the implementation of the non-linear drag curves is that for a given sea state, two values of minimum drag can be determined, one in Region IV and one between Region II and Region III. Knowledge that drag can be lower in a particular operating region for fuel conservation is of prime importance.

It is recognized that a time delay exists in the solution of the equations of motion and a time delay exists in the graphical display of the solutions to the operator. Real time interaction between the man in the loop and the computer is achieved by setting the integration step size equal to the computer loop iteration time. Real time considerations are further discussed in Appendix A.

VIII. RECOMMENDATIONS

The following recommendations are made concerning future improvements in the RTS5D model.

1. Further refinement of the right hand side of the equations of motion is necessary in order to more closely match the characteristics of the data base model.
2. Implement software broach dynamics in the model with thrust applied differentially at different lever arms (see Fig. 3) so that thrust failures and effectiveness of corrective action may be evaluated.
3. Rewrite the program such that the AGT-10's would provide asynchronous parallel graphics (existing equipment) or implement the model on a more modern system which uses parallel processing and improved graphics.
4. Introduce the sixth degree of freedom (heave) as a parallel operation. Since heave time constant is much faster than the rest of the system dynamics, real time heave analysis could be conducted simultaneously if cross coupling is neglected.
5. The system is designed for man in the loop interface, so further operator testing is recommended as a valuable source of information for future refinement of the model.

APPENDIX A
REAL TIME ANALYSIS

T. S. Nelson, in Ref. 1, accomplished real time solution of SES equations of motion with the use of a 1000 Hz clock to measure computer iteration time and by using that time as an integration step size for rectangular integration to solve the simplified dynamic equations. He concluded, by comparing the solutions of these equations to that obtained by using a Runga Kutta variable integration step size, that the computer iteration loop time must stay below 100 ms in order to maintain at least 3% accuracy of solution. He further developed a multiplexing algorithm in order to output these solutions to an operator in the form of graphics on a real time basis. This model can be invalidated as a real time model in two ways. First, in order to have real time, the integration step size used must be equal to the computer iteration time and in this model, because of the multiplexing algorithm, each loop time is different. The integration step size, Δt determined by measuring loop N is used to solve the equations in loop N+1, which has a different loop time, therefore the solutions are not exactly real time. Secondly, the multiplexing scheme makes the model into a sampled data system with sample time (T_s) approximately 2 seconds. Even if the first condition did not apply and real time solutions to the equations could be assumed, then the output to the

operator would be sampled real time data. Given the fastest time constant for the RTS5D simplified equations (approximately 1.3 seconds for pitch rotational motion), it can be seen that the Nyquist Criteria is clearly violated. Yet even with these obvious limitations, it has been demonstrated that the RTS5D can closely reproduce the output characteristics of the DBSIM5D and therefore is a useful model. Surge speed as well as sway, yaw and roll (Figs. 21-23) have much longer effective time constants. It is recognized that there is a delay between the real time of any event and the observation of that event, therefore "Real Time" as used in this report is a matter of degree, or more specifically how much delay time between solution and display can be tolerated. Given the scale of graphics presented to the operator in this model and the speed with which the SES responds, the RTS5D is an acceptable model, however it would be desirable to reduce the sample time T_s and make the integration step size equal to the computer loop iteration time for each iteration.

An algorithm to set the integration step size equal to the computer iteration time of the current loop was implemented as a software design change to the RTS5D model. This was accomplished by defining the integration step size as an array DELT(N) with N equal to the total number of iterations necessary to display all of the graphical information to the operator. Each loop iteration time was measured and

stored in this array during the first pass through the multiplexing algorithm. $\Delta T(N)$ was initialized at zero so that time would not increment and integration would not be performed during the first pass through the multiplexing algorithm. On subsequent passes through the multiplexing algorithm the integration step size was equal to the loop time in which it was used.

This timing algorithm when tested in the RTS5D model was found to accurately represent real time but the deviation from previously used integration step size values caused instability in both the pitch and roll dynamics. This needs further investigation.

Sturgeon, in Ref. 8, demonstrated the value of using parallel array processors to reduce computational time in real time simulations. Such a model could be designed for the RTS5D using the existing equipment. The AGT-10's which are used for displaying the output can interface with the KDS-3300 main computer in two ways. The multiplexing scheme of the RTS5D uses a "hand shaking" method with a GRAPHO or TEXT0 subroutine call in each loop. It is precisely the inclusion of graphics in each loop that causes problems for real time presentation. The solution of the equations requires only approximately 35 ms. Each GRAPHO or TEXT0 call, with its associated "hand shaking" time, makes up the rest of the loop time, approximately 40 ms more. It can be seen that by introducing two such calls in one loop that the 100 ms time restriction would be exceeded.

An improvement to this method would be to use the second interface between the AGT-10's and the XDS-9300 which is that the AGT-10's have a direct connection to some memory locations in XDS-9300. By solving the dynamics for the SES and storing the data in these memory addresses in the form of arrays, the AGT-10's could be made to operate asynchronously in parallel as array processors by using a process known as "cycle stealing" to fetch the data from the XDS-9300 and to display this on the screen for the operator. This process would have the advantage of reducing and making more uniform the computer iteration time used as the step size for the rectangular integration in the solution of the SES equations of motion. The sample time would also be reduced because all of the time wasted in the "hand shaking" is eliminated. It's drawback is that the output is not truly real time because of the asynchronous operation of the AGT-10's. However, the reduced sampled data time and computer iteration time would make it a closer approximation of real time than presently exists.

APPENDIX E
RTS5D MODIFIED PROGRAM NOMENCLATURE

A11	A/D trunk 500 voltage
A21	A/D trunk 501 voltage
A22	Added mass coefficient in yaw moment equation
A31	A/D trunk 502 voltage
A33	Added mass coefficient in roll moment equation
A34	Added mass coefficient in pitch moment equation
A41	A/D trunk 503 voltage
ABSYD	$\dot{y}_0$
ABSYCD	$\ddot{y}_0$
AD	array of A/D lines
A1CCA	speed line index counter
AM	mass
APITCH	pitch angle in degrees
APRINT	# iterations between print execute commands
ARR	Real IARR (1)
ARATE	turn rate in degrees
AROLL	roll angle
ASTOP	performance index J_1 limit
AXST	initial condition X-coordinate on sea track
AYST	initial condition Y-coordinate on sea track
BETA	drift angle
BCAT	subroutine to generate perspective of SES

CASST	automatic speed control mode indicator
CDP	bow real forces lumped coefficient
CDX _(i)	drag forces lumped coefficients
CDY	sway forces lumped coefficient
CDZP	sidewall rolling moment lumped coefficient
DAL	digital to analog call
DELT	iteration loop time
DELTA	commanded iteration loop time
DFCCM	total change in fuel command from speed controller
DFCCM 1	change in fuel command due to proportional speed controller
DFCCM 2	change in fuel command due to proportional speed controller
DTIME	v time
DTIMPLT	scaled value for v
DUCOM	difference between ucom and u
DXPLOT	same as DTIMPLT
DYPLOT	scaled value for v
DYREPET	restart value for v
EFLG	error flag for TEXTC call
ETIME	r time
EYREPET	restart value for r
EXPLOT	scaled r time
EYPLOT	scaled r
F1	stern buoyancy force
F2	bow seal pitch force
F3	bow buoyance force

FCCM	fuel command
FSP	port sidewall buoyancy force
FSS	starboard sidewall buoyancy force
FTIME	$\dot{v}$ time
FTIMPLT	scaled $\dot{v}$ time
FXPLOT	same as FTIMPLT
FYPLOT	scaled $\dot{v}$
FYREPET	restart value for $\dot{v}$
G	gravitational acceleration
GRAPHC	subroutine to project image (non alpha-numeric)
GTIME	$\dot{R}$ time
GTIMPLT	scaled $\dot{R}$ time
GXPLOT	same as GTIMPLT
GYPLOT	scaled $\dot{R}$
GYREPET	restart $\dot{R}$ value
$H(i)$	dynamics maximum and minimum array
HEAD	craft heading
HOLD	subroutine to freeze display/program
HTIME	$\dot{u}$ time
HTIMPLT	scaled $\dot{u}$ time
HYREPET	analog restart y coordinate on sea track
HYESX	analog y position of craft
HYESY	analog x position of craft
IARR(1)	digitized mode selector
IARR(2)	digitized effector angle
IARR(3)	digitized T7

IARR(4)	digitized T8
IARR(5)	digitized T9
IARR(6)	digitized T10
IARR(7)	digitized surge velocity command
ICO	multiplex graphics index
ICCA	speed line index
ICRAFT	vehicle designator
IER	error flag
IIPR	time history limit
IIPRD	time history limit
IIPRE	time history limit
IIPRF	time history limit
IIPRG	time history limit
IIPRH	time history limit
IIPRC	time history limit
IIPRP	time history limit
IIPRQ	time history limit
IIPRR	time history limit
IIPRS	time history limit
IIX	print counter
IJ	craft perspective counter
IJA	craft perspective counter
IJAD	time history counter
IJAE	time history counter
IJAF	time history counter
IJAG	time history counter
IJAH	time history counter

IJAO	time history counter
IJAP	time history counter
IJAQ	time history counter
IJAR	time history counter
IJAS	time history counter
IJD	time history counter
IJE	time history counter
IJF	time history counter
IJG	time history counter
IJH	time history counter
IJO	time history counter
IJP	time history counter
IJQ	time history counter
IJR	time history counter
IJS	time history counter
IL	graphics console display array
ILA1	graphics console display array
ILA2	graphics console display array
ILA3	graphics console display array
ILA4	graphics console display array
ILA5	graphics console display array
ILA6	graphics console display array
ILA7	graphics console display array
ILA8	graphics console display array
ILA9	graphics console display array
ILA10	graphics console display array

ILA11	graphics console display array
ILA12	graphics console display array
ILA13	graphics console display array
ILA14	graphics console display array
ILA15	graphics console display array
ILA16	graphics console display array
ILA17	graphics console display array
ILA18	graphics console display array
ILA19	graphics console display array
ILA20	graphics console display array
ILA21	graphics console display array
ILA22	graphics console display array
ILA23	graphics console display array
ILA24	graphics console display array
ILA25	graphics console display array
ILA26	graphics console display array
ILA27	graphics console display array
ILA28	graphics console display array
ILA29	graphics console display array
ILAP	graphics console display array
IPACK	subroutine to load display arrays
IPLOT	graphics console display array
IPR	time history counter limit
IPRD	time history counter limit

IPRE	time history counter limit
IPRF	time history counter limit
IPRG	time history counter limit
IPRH	time history counter limit
IPRO	time history counter limit
IPRP	time history counter limit
IPRQ	time history counter limit
IPRR	time history counter limit
IPRS	time history counter limit
ISWA	display array
ITEXT	subroutine to display alpha-numeric data
IVEIWA	craft perspective/north road array
IVIEWAA	sea track array
IVIEWB	sea track axis array
IVIEWC	u time history array
IVIEWD	v time history array
IVIEWE	r time history array
IVIEWF	$\dot{v}$ time history array
IVIEWG	$\dot{r}$ time history array
IVIEWH	$\dot{u}$ time history array
IVIEWO	$\dot{p}$ time history array
IVIEWP	$\dot{q}$ time history array
IVIEWQ	p time history array
IVIEWR	q time history array
IVIEWS	analog sea track array
IVIEWZ	time history axis array

IX	x axis moment of inertia
J	time history counter
JD	time history counter
JE	time history counter
JF	time histroy counter
JG	time history counter
JH	time history counter
JO	time history counter
JP	time history counter
JQ	time history counter
JR	time history counter
JS	time history counter
K	proportional speed controller gain
KK	intregal speed controller gain
L3	lever arm for buoyancy force
LD	draft of sidewall
LP	side thrust lever arm
L2	lever arm for sway drag force
N	clock interrupt count
NAD	AD trunk line array
ONEMOT	subroutine to calculate SES dynamics
CO	lever arm for thrust side component
CTIME	$\dot{p}$ time
CTIMPLT	scaled $\dot{p}$ time
CKPLOT	same as CTIMPLT
CYFLOT	scaled $\dot{p}$
CYREPET	restart $\dot{p}$ value

P	roll rate
PE	plenum pressure
PDCT	roll acceleration
PDCTM	max roll acceleration
PHI	roll angle
PMAX	max roll rate
PSI	craft heading
PTIME	$\dot{q}$ time
PTIMPLT	scaled $\dot{q}$ time
PXPLOT	same as PTIMPLT
PYPLOT	scaled $\dot{q}$
PYREPET	restart $\dot{q}$ value
Q	pitch rate
QDCT	pitch acceleration
QDCTM	max $\dot{q}$
QMAX	max $\dot{q}$
QTIME	p time
QTIMPLT	scaled p time
QXPLOT	same as QTIMPLT
QYFLOT	scaled p value
QYREPET	restart p value
R	craft turn rate r
RDCT	$\dot{r}$
RDCTM	max $\dot{r}$
READCLOCK	subroutine to sample real time clock

RESET	subroutine to reset analog computer
RHC	density of water
RMAX	maximum yaw rate of craft
RTIME	q rate
RTIMPLT	scaled q time
RX	I_x
RXPLOT	same as RTIMPLT
RY	I_y
RYPLOT	scaled q value
RYREPET	restart q value
RZ	I_z
SEAST	sea state
SESX	scaled X_0
SESY	scaled Y_0
SF1	scale factor in time history arrays
SF14	scale factor in time history arrays
SLIP	Beta influence coefficient
SPDLIN	speedline lower screen limit
SPEED	Craft's velocity in knots
STARTCLOCK	subroutine to start clock
STIME	analog time
STIMPLT	scaled analog time
SXPLOT	HYSESX
SYPLOT	HYSESY
SKREPET	restart value for HYSESX
SYREPET	restart value for HYSESY

T	total thrust
T10	thruster number 4
T10MAX	max T10 value
T7	thruster number 1
T7MAS	max T7 value
T8	thruster number 2
T8MAX	max value of T8
T9	thruster number 3
T9MAX	max value of T9
TEXTC	subroutine to display alpha-numeric data
TDCT	T
THETA	pitch angle
TIM	u time
TIME	time
TIMPLT	scaled TIM
TINT	real clock interrupt frequency
TITLE	program name display array
TITLE0	program name display array
TITLE1	program name display array
TM	manual thrust command
TMAX	max T
TSIDE	sum of side thrust components
TYAW	sum of yaw moment
U	forward velocity (surge)
UCCOM	forward velocity (surge) command

UDCT	$\dot{u}$
UDCTM	$\max \dot{u}$
UMAX	$\max u$
UMUL	scale factor in speedline generation
V	side velocity (sway)
VCD	thruster console pot array
VDOT	$\dot{v}$
VDCTM	$\max \dot{v}$
VMAX	$\max v$
VS	total velocity
W	heave velocity
WDCT	heave acceleration
WE	width of bow seal
WRITECLOCK	subroutine to assign clock interrupts to variable
WW	lever arm for sway drag forces
X	y_o
XC	x_o
XCDOT	$\dot{x}_o$
XDx	boat perspective scale factor
XCX	boat perspective scale factor
XPLCT	display array
XREPET	reset value for SESX
XST	scale factor for analog plot
YC	y_o
YCDCT	$\dot{y}_o$

YHIGH	point D vertical position constant
YPLCT	display array
YREPET	reset value for SESY
YST	scale factor for analog plot
Z	operational effector angle
Z2	upper effector angle limitation
Z3	lower effector angle limitation
Z4	zero angle upper limit
Z5	zero angle lower limit
ZAB	scale factor for Z10
ZERC	effector broach limit angle
ZI	modified Z
ZZZZ	fixed step effector angle input

APPENDIX C

RTSSD MODIFIED COMPUTER PROGRAM LISTING

```

0001 DIMENSION ITEXT(29), ILA5(24), IVIEWA(75), IVIEWB(70), ISWA(40),
0002 CIL(25), IILEC(24), IARR(7), ILA2(24), IVIEWC(205), IVIEWZ(70),
0003 CIL4(24), ILA5(24), ILA6(24), ILA7(24), ILA8(24), IPLCT(29), ILA3(24),
0004 C11ILEC(24), I1ILE1(24), IVIEWD(205), IVIEWE(205), IVIEWF(205),
0005 C11ILEG(205), IVIEWH(205), ILA10(24), ILA11(24), ILA12(24), ILA13(24),
0006 C11ILA14(24), ILA15(24), IVIEWI(205), IVIEWJ(205),
0007 C11ILEH(205), IVIEWK(205), IVIEWL(205),
0008 C11ILA17(25), ILA18(24), ILA19(24), ILA20(24), ILA21(24),
0009 C11ILA23(24), ILA24(24), ILA25(24), ILA26(24), ILA27(24), ILA28(24),
0010 C11LA29(24), I(50), IIPR(20), IJ(20), IJ(20), IJA(20), XREPEI(20),
0011 CYREPEI(20), TIM(20), TIMPLI(20), XPLUI(20), YPLCT(20)
0012 INTEGER IILE, IILEC
0013 EQUIVALENCE (IILE, IILEC)
0014 C(IILA1, ILA5), (IILA1, ILA6), (IILA1, ILA7), (IILA1, ILA8), (IILA1, ILA9),
0015 C(IILA1, ILA10), (IILA1, ILA11), (IILA1, ILA12), (IILA1, ILA13), (IILA1, ILA14),
0016 C(IILA1, ILA15), (IILA1, ILA16), (IILA1, ILA17), (IILA1, ILA18), (IILA1, ILA19),
0017 C(IILA1, ILA20), (IILA1, ILA21), (IILA1, ILA22), (IILA1, ILA23), (
0018 C(IILA1, ILA24), (IILA1, ILA25), (IILA1, ILA26), (IILA1, ILA27),
0019 C(IILA1, ILA28), (IILA1, ILA29)
0020 COMMON A41, U, DELT, UGCT, V, VDUI, R, RDUI, P, PLOT, C, QDGI, IHETA, I, AM,
0021 CUP, LZ, AL, A21, A31, X, AXSI, AYSI,
0022 CZI, CDX, RZ, UD, CDY, W, A22, FDP, PHI, RDS, FSP, FSS, CDZP, A33, RX, FL, F2,
0023 CF3, LF, LL, PR, WE, RY, L3, A24, Z, BETA, HYSE SX, XSI, HYSE SY, YSI, UMAX, XGX,
0024 CXDX, CMPL, FEAD, VHIGH, SPCLIN, AICOA, YJ, ICCA, SEAST
0025 COMMON /DATA/IARR
0026 NAMELIST
0027 CALL AL(IARR)
0028 CONTINUE
0029 CALL RESET(IUJ)
0030 CALL DTINIT(1, ITEXT, 25, IER)
0031 CALL DCINIT(1, ISWA, 40, IER)
0032 CALL DTINIT(2, IPLCT, 29, IER)
0033 CALL DGINIT(2, I, 25, IER)
0034 ENCCCE(56, 1, IILEC)
0035 FORMAT(
0036 CALL ITEXT(1, IILEC(24), IC(1, 3, 3, IER)
0037 IF(IER.NE.0) OUTPUT(1, IER,
0038 ENCODE(1, 2, IILEC)
0039 FORMAT(
0040 CALL ITEXT(1, IILEC(24), IC(1, 3, 3, IER)
0041 IF(IER.NE.0) OUTPUT(1, IER,
0042 I=ICGCCC
0043 CALL DELAY
0044 V=0.
0045 W=0.
0046 R=0.
0047 P=0.
0048

```

C=0.
 XO=0.
 YO=0.
 ZZ=0.
 PSI=0.
 PHI=0.
 THETA=C.
 UDOT=C.
 VDOT=C.
 WDOT=C.
 RDOT=0.
 PDOT=0.
 QDOT=C.
 TIME=0.
 DO 3131 I=1,50
 3131 F(I)=C.
 3001 CLNTINCE
 SEAST=4
 UMAX=C95.
 VMAX=10.
 RMAX=C.125
 FMAX=.04
 GMAX=.02
 UDOTM=1.8
 VDOTM=3.6
 RDOTM=C.C40
 PDOTM=.1
 QDOTM=.02
 A32=54118.
 A34=588238.
 SLIP=0.
 LKR=50.
 LK=30.
 LZ=30.
 L3=100.
 LQ=3.
 WE=LCC.
 PB=240.
 LP=5.
 CUP=.174
 CDZ=100000.
 CDZP=30000.
 RX=18220000.
 RY=815859900.
 L=0.87
 IMAX=26000
 ITMAX=50000
 IDMAX=50000

0049
 0050
 0051
 0052
 0053
 0054
 0055
 0056
 0057
 0058
 0059
 0060
 0061
 0062
 0063
 0064
 0065
 0066
 0067
 0068
 0069
 0070
 0071
 0072
 0073
 0074
 0075
 0076
 0077
 0078
 0079
 0080
 0081
 0082
 0083
 0084
 0085
 0086
 0087
 0088
 0089
 0090
 0091
 0092
 0093
 0094
 0095
 0096

0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144

19MAX=90000
110MAX=90000
AM=170364.
CDY=22500.
AZZ=1.
CC=110.
RW=22.
RZ=55270000.
AXST=20000.
CONTINUE
ZZZZ=C.543
DELTA=C.04
SFZZX=-.1
Z1=2.
Z2=0.5243
Z3=-0.5263
Z4=C.02
Z5=-C.02
Z6=-.05
Z11=3.1416
Z12=0.75
XST=C.75
YST=C.75
XREPET(1)=XST
YREPET(1)=YST
SXREPET=XST
SYREPET=YST
V=C.
R=C.
W=C.
XU=C.
YU=C.
ZZ=C.
PHI=C.
TFETA=C.02
PSI=C.
LDDI=C.
VDCI=C.
MDLI=C.
MDQT=C.
PDCT=C.
GDDI=C.
SFI=12.
SFI4=C.5
JJ=0
JP=0
JQ=0

3002

JK=0
 JS=0
 LTIME=C.
 PTIME=C.
 RTIME=C.
 STIME=C.
 CYRFPET=.3
 PYRFPET=.1
 QYRFPET=.3
 RYRFPET=.1
 JD=0
 JE=0
 JF=0
 JG=0
 JH=0
 LTIME=C.C
 PTIME=C.C
 RTIME=C.C
 STIME=C.C
 CYRFPET=C.7
 PYRFPET=C.5
 QYRFPET=C.7
 RYRFPET=C.5
 UMUL=4.C
 IPR(1)=23C
 IPR(2)=110
 IPRO=110
 IPRE=110
 IPRF=110
 IPRG=110
 IPRH=110
 IPRC=110
 IPRP=110
 IPRQ=110
 IPRR=110
 IPRS=110
 ICCA=0
 ICC=2
 AICCA=10.
 X=-100.
 TIME=C.C
 P=C.C
 C=0.C
 RHU=2.C
 C=32.2

0145
 0146
 0147
 0148
 0149
 0150
 0151
 0152
 0153
 0154
 0155
 0156
 0157
 0158
 0159
 0160
 0161
 0162
 0163
 0164
 0165
 0166
 0167
 0168
 0169
 0170
 0171
 0172
 0173
 0174
 0175
 0176
 0177
 0178
 0179
 0180
 0181
 0182
 0183
 0184
 0185
 0186
 0187
 0188
 0189
 0190
 0191
 0192

0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240

```

AXU=25.
AYU=25.
TINT=1000.
XGX=24.
XDX=24.
ZAB=-3.1416
YHIG=C.1
IN=0
J(1)=C
J(2)=0
TIM(2)=C
YREPE(2)=C.9
APRINT=5.
K=2900.0
KK=3.0
DTCC#2=C.0
IF((10.0*IAKK(1)/(2**23).GT.--.8).AND.(10.0*IAKK(1)/(2**23)
C.LI.5)ICC TO 22
1112 CUIPUT(101) CHANGES,*,* (/R
      INPUT(101)
      IF(10.0*IAKK(1)/(2**23).GT.--.8).AND.(10.0*IAKK(1)/(2**23)
22  CONTINUE
65C CALL DTINII(1,ITEXT,29,IER)
CALL DTINII(1,ISWA,40,IER)
CALL DTINII(2,IPLGT,29,IER)
CALL DTINII(2,IL,29,IER)
IVIEWB(1)=IHEAD(1,IC)
IVIEWB(2)=IPACK(0.0,1.0,0)
IVIEWB(3)=IPACK(0.0,0.5,0)
IVIEWB(4)=IPACK(-1.0,0.0,0)
IVIEWB(5)=IPACK(-0.5,0.0,1)
IVIEWB(6)=IPACK(0.5,0.0,0)
IVIEWB(7)=IPACK(1.0,0.0,1)
IVIEWB(8)=IPACK(0.75,1.0,0)
IVIEWB(9)=IPACK(0.75,0.5,1)
IVIEWB(10)=IPACK(0.5,0.75,0)
IVIEWB(11)=IPACK(1.0,0.75,1)
IVIEWB(12)=C
CALL CM/PAC(1,IVIEWB,12,1,IER)
IVIEWZ(1)=IFAD(0,10)
IVIEWZ(2)=IPACK(-5.0,0.5,0)
IVIEWZ(3)=IPACK(-9.9,1)
IVIEWZ(4)=IPACK(-5.0,0.5,0)
IVIEWZ(5)=IPACK(0.0,0.9,1)
IVIEWZ(6)=IPACK(-9.9,1)
IVIEWZ(7)=IPACK(0.0,0.9,1)
IVIEWZ(8)=IPACK(-9.9,1)
IVIEWZ(9)=IPACK(0.0,0.9,1)

```


AD-A098 487

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

F/G 13/10

REAL TIME SIMULATION AND CONTROL 3000 TON SURFACE EFFECT SHIP W--ETC(U)

DEC 80 L L OLIPHANT

UNCLASSIFIED

NL

AD
AD-000000



END
DATE
FILMED
5 81
DTIC


```

3010 ENCGCE(56,3)IO,ILA21)
    FORMAT(1,F(HEAVY))
    CALL TEXIT(1,ILA21,24,16,1,1,3,IER)
3011 ENCGCE(56,3011,ILA22)
    FORMAT(1,FDC(LIGHT))
    CALL TEXIT(1,ILA22,24,17,1,1,3,IER)
3012 ENCGCE(56,3012,ILA23)
    FORMAT(1,C(HEAVY))
    CALL TEXIT(1,ILA23,24,15,1,1,3,IER)
3013 ENCGCE(56,3013,ILA24)
    FORMAT(1,CDC(LIGHT))
    CALL TEXIT(1,ILA24,24,20,1,1,3,IER)
      2 ENCGCE(56,3,ILA25)
    FORMAT(1,WATERLINE)
    CALL TEXIT(1,ILA25,24,24,1,1,3,IER)
3014 ENCGCE(56,1,F8.1,
    FORMAT(1,F8.1,F5.2)
      3 CALL WRTIECLOCK(0)
    CALL STARTCLOCK
    A41=IC.C+IARR(2)/2**23
    CALL CCMPUTE
    CCNT=INLE
    ICNT=ICG+1
    CCNT=INLE
    IF(SENSE SWITCH 6)5559,5598
5559 Z1=Z
    GO TO 752
5558 CONTINUE
    Z1=4**A41
    GO TO 744
744 CCNT=INLE
    IF(Z.C1.23)GO TO 79C
    IF(Z.C1.23)GO TO 791
    IF(Z.C1.24)GO TO 792
    IF(Z.C1.25)GO TO 752
    Z=0.0
    Z1=Z
    GO TO 752
790 Z=Z2
    Z1=Z
    GO TO 752
791 Z=Z3
    Z1=Z
    GO TO 752
792 CCNT=INLE
    IF(U.L1.7C.1)ZBK0=Z2

```

```

IF(U-GE-70.)AND.(U-LI-87.)ZBR0=.3491
IF(U-GE-87.)AND.(U-LI-104.)GO TO 4005
IF(U-GE-104.)AND.(U-LI-121.)GO TO 4006
IF(U-GE-121.)AND.(U-LI-138.)GO TO 4007
IF(U-GE-138.)ZBR0=.C345
GO TO 4008
4005 ZBR0=.3491*17/(U-70)
GO TO 4009
4006 ZBR0=.1745*17/(U-87)
GO TO 4008
4007 ZBR0=.C373*17/(U-104)
4008 CONTINUE
IF(ABS(Z)-LI-ZBR0)GO TO 4009
ENCODE(S6,4010,ILA9)
FORMAT(8R C A C H
CALL TEXIG(1,ILA9,24,37,1,1,3,1EK)
CONTINUE
4009 CONTINUE
I7=I7MAX*(.5+(5.0*IARR(3)/2**23))
I8=I8MAX*(.5+(5.0*IARR(4)/2**23))
I9=I9MAX*(.5+(5.0*IARR(5)/2**23))
I10=I9
I1=I7+I8+I9+I10
IF(SENSE SWITCH 3)4000,4001
CASST=C
GO TO 4002
4001 FCOM=TM
CASST=1
LCOM=200*(1.5+(5.0*IARR(6)/2**23))
LUCOM=LCOM-U
DFCCM1=K*ELCCM
DFCCM2=DFCCM2+DUCOM*DELT
FCCM=DFCCM1+KK*DFCCM2
FCCM=DFCCM+TM
IF(FCOM-CT-IMAX)GO TO 4003
IF(FCCM-LI-0.0)GO TO 4004
GO TO 4002
4003 FCOM=TMAX
GO TO 4002
4004 FCOM=0.0
GO TO 4002
4002 TUCT=FCCM-I
I=I+TUCT*CELT
IF(I-GI-IPAX)I=IMAX
IF(I-LI-C-CT)I=0.0
XI=PSI/6.283185
IX=XI
AIX=XI-IX

```

0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432

0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480

```

IF(AIX-11-C.OIGG IC 10
HEAD=AIX+360.
UG TC 11
AIX1=1.+AIX
LEAD=AIX1+360.
CONTINALE
ARATE=(360.*K)/6.282185
ARGUL=(360.*PHI)/6.282185
APITCH=(360.*THETA)/6.282185
SPELD=.771*U
SPCLP=.771*UCOM
IF(SENSE SWITCH 2)176,175
CALL CNEMCT
176 CONTINALE
177 CONTINALE
BETA=SLIP*K
VS=(U+L)+(V+V)**.5
PSI=PSI+DELI*R
XJCU=U+COS(PSI)-V*SIN(PSI)
YJCU=U+SIN(PSI)+V*CCS(PSI)
XU=XG+DELI*YJCU
YU=YG+DELI*YJCU
SESA=XST+(YQ/AXST)
SESY=YST+(YC/AYST)
IF(H(1)-CI.XO)GO TO 700
H(1)=XG
IF(H(2)-CI.YO)GO TO 701
H(2)=YG
IF(H(3)-GI.ZZ)GO TO 702
H(3)=ZZ
IF(H(4)-GI.PSI)GO TO 703
H(4)=PSI
IF(H(5)-CI.PHI)GO TO 704
H(5)=PHI
IF(H(6)-CI.TETA)GO TO 705
H(6)=TETA
IF(H(7)-CI.U)GO TO 706
H(7)=U
IF(H(8)-CI.V)GO TO 707
H(8)=V
IF(H(9)-CI.W)GO TO 708
H(9)=W
IF(H(10)-CI.R)GO TO 709
H(10)=R
IF(H(11)-CI.P)GO TO 710
H(11)=P
IF(H(12)-CI.Q)GO TO 711
H(12)=Q

```

711 IF(H(13).GT.UDDT)GC IC 712
712 IF(H(14).GT.VCOT)GO TO 713
713 IF(H(15).GT.WCOT)GO TO 714
714 IF(H(16).GT.ADDT)GC IC 715
715 IF(H(17).GT.PCOT)GC IC 716
716 IF(H(18).GT.CCOT)GO TO 718
718 IF(H(30).LT.XO)GO TO 719
719 IF(H(31).LT.YO)GO IC 720
720 IF(H(32).LT.ZZ)GO IC 721
721 IF(H(33).LT.PSI)GO TO 722
722 IF(H(34).LT.PFI)GO TO 723
723 IF(H(35).LT.THETA)GC IC 724
724 IF(H(36).LT.U)GC IC 725
725 IF(H(37).LT.V)GO IC 726
726 IF(H(38).LT.W)GO TO 727
727 IF(H(39).LT.R)GO TO 728
728 IF(H(40).LT.P)GO IC 729
729 IF(H(41).LT.Q)GO TO 730
730 IF(H(42).LT.UDDT)GO TO 731
731 IF(H(43).LT.VDDT)GC IC 732
732 IF(H(44).LT.WDCT)GC IC 733
733 IF(H(45).LT.RCOT)GO TO 734
734 IF(H(46).LT.PDCT)GO TO 735
735 IF(H(47).LT.QDCT)GC IC 736

0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528

```

736 CONT INLE SWITCH 111C2,105
1C2 CCNT INLE
    IIX=111+1
    IF(IIX.LT.APRINT)GO TO 7654
    IIX=C
    WRITE(6,100)TIME,U,V,R,PHI,THETA,XO,YO
    FLRMAI(6F12.5,2F8.1)
7654 CONT INLE
105 CCNT INLE
    IF(IICC.RE.3)GC TO 623
602 ENCODE(56,516,ILAP)1,Z,DELT,TIME,API(ICH,HEAL,AKATE,BETA,ARCLL,SPEE
CC
    CALL TEXIC(1,ILAP,24,34,1,1,3,IER)
623 CONT INLE
58 CCNT INLE
    CALL BCAL
2071 IF(110.0+IARR(1)/(2+23)).GT.--.8)GO TO 2080
2080 GO TC 2072
    IF(110.0+IARR(1)/(2+23)).GT.--.5)GO TO 1111
    IF(110.0+IARR(1)/(2+23)).LT.--.4)GO TO 23
    GO TC 1111
23 CALL WRITECLUCK(0)
    GO TC 2071
2072 CCNT INLE
    IF(110.NE.15)GO TO 624
    IIPR(1)=IIPR(1)-1
    J(1)=J(1)+1
    IJ(1)=4+J(1)
    IJA(1)=IJ(1)+1
    IVIEWAA(1)=IHEAD(C,IC)
    IVIEWAA(2)=IPACK(.0,.0,0)
    IVIEWAA(3)=IPACK(XREPET(1),YREPET(1),0)
    IVIEWAA(4)=IPACK(XREPET(1),YREPET(1),1)
    IVIEWAA(1J(1))=IPACK(SESX,SESY,1)
    DO 2015 I=1JA(1),IIPR(1)
2015 IVIEWAA(I)=0
    IF(IJA(1).LT.IIPR(1))CC TC 2017
2018 CCNT INLE
    I=1,IIPR(1)
2016 IVIEWAA(I)=0
    IVIEWAA(1)=0
    XREPET(1)=SESX
    YREPET(1)=SESY
    J(1)=C
    IJ(1)=0
    IJA(1)=C
2017 CCNT INLE
    CALL GRAFFC(1,IVIEWAA,IIPR(1),15,IER)

```

0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624

```

624 CCNT INLE
606 IF (IC-NE.10) GO TO 63C
    IIPR(2)=IPR(2)-1
    JJ(2)=JJ(2)+1
    IJ(2)=4+JJ(2)+1
    IJA(2)=IJ(2)+1
    TIM(2)=TIM(2)+DELT
    TIMPLT(2)=TIMPLT(2)
    XPLOT(2)=CL/(10.*UMAX))+.9
    IVIEWC(1)=IHEAD(0,1C)
    IVIEWC(2)=IPACK(-.9,0,0)
    IVIEWC(3)=IPACK(-.9,YREPET(2),1)
    IVIEWC(4)=IPACK(-.9,YREPET(2),1)
    IVIEWC(IJ(2))=IPACK(XPLCT(2),YPLCT(2),1)
    DO 202C I=1,JA(2),IPR(2)
202C IVIEWC(I)=0
    IF (IJA(2).LT.IIPR(2)) GO TO 2032
2038 CC 2044 I=1,IPR(2)
2044 IVIEWC(I)=C
    TIM(2)=C
    YREPET(2)=YPLCT(2)
    JJ(2)=C
    IJ(2)=0
    IJA(2)=C
2032 CCNT INLE
63C CALL GRAFFC(1,IVIEWC,IPR(2),10,IER)
    CCNT INLE
    IF (IC-NE.11) GO TO 621
    IIFRD=IFRD-1
    JD=JD+1
    IJC=4+JC
    IJAC=1JL+1
    DTIME=DTIME+DELT
    CTIMPLT=CTIMPLT+1
    DXPLCT=CTIMPLT
    DYPLCT=(V/(10.*VMAX))+C.7
    IVIEWC(1)=IHEAD(0,10)
    IVIEWC(2)=IPACK(-.9,0,0)
    IVIEWC(3)=IPACK(-.9,YREPET,0)
    IVIEWC(4)=IPACK(-.9,YREPET,1)
    IVIEWC(IJD)=IPACK(DXFLCT,DYFLCT,1)
    DO 2021 I=1,JAD,IPKD
2021 IVIEWC(I)=0
    IF (IJA(2).LT.IIPRD) GO TO 2033
2035 CCNT INLE

```

```

2045      DO 2045 I=1,IPRD
        IVIEWC(I)=0
        OTIME=C
        DYREPET=DYPLCT
        JC=0
        IJC=C
        IJAC=C
        CCNT INUE
        CALL GRAPHC(1,IVIEWD,IPRC,11,IER)
        CONT INLE
        IF(IICC.NE.12)GO TO 632
        IIPRE=IPRE-1
        JE=JE+1
        IJE=4+JE
        IJAE=1JE+1
        ETIME=ETIME+DELT
        ETIMPL=(ETIME/SF1)-SF14
        EYPLCT=(ETIMPLT-ETIMPL)*0.5
        IVIEWE(1)=IHEAD(0,1C)
        IVIEWE(2)=IPACK(-.9,.9)
        IVIEWE(3)=IPACK(-.9,.9)
        IVIEWE(4)=IPACK(-.9,.9)
        IVIEWE(1JE)=IPACK(ETIMPLT,EYPLCT,1)
        DO 2022 I=1,IPRE
        IVIEWE(I)=C
        IF(IJAE.LT.IIPRE)GO TO 2034
        CONT INLE
        CU 2046 I=1,IPRE
        IVIEWE(I)=0
        ETIME=C
        EYREPET=EYPLCT
        IJE=0
        IJAE=0
        IJAF=0
        CCNT INLE
        CALL GRAPHC(1,IVIEWE,IPRE,12,IER)
        CONT INUE
        IF(IICC.NE.6)GO TO 626
        IIPRF=IPRF-1
        JF=JF+1
        IJF=4+JF
        IJAF=1JF+1
        ETIME=ETIME+DELT
        ETIMPL=(ETIME/SF1)-SF14
        EYPLCT=(ETIMPLT-ETIMPL)*0.7
        IVIEWE(1)=IHEAD(0,3)

```

```

0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672

```

0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720

```

2023  IVIEWF(2)=IPACK(-.0,-.0,0)
      IVIEWF(3)=IPACK(-.9,FYREPEI,0)
      IVIEWF(4)=IPACK(-.9,FYREPEI,1)
      IVIEWF(1,JF)=IPACK(FXFLCT,FYFLCT,1)
      DO 2023 I=1,JAF,IPRF
      IVIEWF(I)=C
      IF(1JAF.LT.1IPRG)GO TO 2035
2041  CONTINUE
      DO 2047 I=1,IPRF
2047  IVIEWF(I)=0
      FTIME=C
      FYREPET=FYFLOT
      JF=0
      IJF=C
      IJAF=0
2035  CONTINUE
      CALL GRAPHG(1,IVIEWF,IPRF,6,IER)
626  CONTINUE
      IF(ICC.NE.7)GO TO 627
      IIPRG=IPRG-1
      JG=JG+1
      IJG=4+JG
      IJAG=1JG+1
      GTIME=CTIME+DELTA
      GTIMFLT=(GTIME/SF1)-SF14
      GXPLOT=(RCCT/(10.*RCCTM))+0.5
      IVIEWG(1)=IHEAD(0,3)
      IVIEWG(2)=IPACK(-.0,-.0,0)
      IVIEWG(3)=IPACK(-.9,CYREPEI,0)
      IVIEWG(4)=IPACK(-.9,CYREPEI,1)
      IVIEWG(IJG)=IPACK(GXFLCT,GYFLOT,1)
      DO 2024 I=1,IJAG,IPRG
2024  IVIEWG(I)=C
      IF(1JAG.LT.1IPRG)GO TO 2036
2042  CONTINUE
      DO 2048 I=1,IPRG
2048  IVIEWG(I)=0
      GTIME=C
      GYREPET=GYFLOT
      JG=0
      IJG=C
      IJAG=0
2036  CONTINUE
      CALL GRAPHG(1,IVIEWG,IPRG,7,IER)
627  CONTINUE
      IF(ICO.NE.5)GO TO 625
      IIPRF=IPRF-1

```

0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768

```

JH=JH+1
IJAH=IJAH+1
HTIME=TIME+DELT
HIMPLT=(HIME/SF1)-SF14
HXPLOT=HIMPLT
HYFLCT=(CCT/(10.*UCCTM))+0.9
IVIEWH(1)=IHEAD(0,3)
IVIEWH(2)=IPACK(-.5,CYREPET,0)
IVIEWH(3)=IPACK(-.5,CYREPET,1)
IVIEWH(4)=IPACK(-.5,CYREPET,1)
IVIEWH(IJH)=IPACK(HXPLOT,CYPLCT,1)
DO 2025 I=1,IJAH,IPRH
  IVIEWH(I)=0
  IF(IJAH.LT.IIPRH)GC TC 2037
2025 CONTINUE
  I=1,IPRH
  IVIEWH(I)=C
  HIME=C
  HYREPET=HYFLCT
  JH=0
  IJH=0
  IJAH=C
  CONTINUE
  CALL GRAFPO(1,IVIEWH,IPRT,5,IER)
  625 CONTINUE
  IF(IJCO.NE.8)GO TO 628
  IFRC=IFRC-1
  JO=JC+1
  IJO=4+JO
  IJAC=IJC+1
  CTIME=CTIME+DELT
  OTIMPLT=(CTIME/SF1)-SF14
  CXFLCT=OTIMPLT
  OYPLCT=(PDC/(10.*PDCM))+.3
  IVIEWH(1)=IHEAD(0,3)
  IVIEWH(2)=IPACK(-.5,CYREPET,0)
  IVIEWH(3)=IPACK(-.5,CYREPET,1)
  IVIEWH(4)=IPACK(-.5,CYREPET,1)
  IVIEWH(IJC)=IPACK(CXFLCT,CYPLCT,1)
  DO 2060 I=1,IJAC,IPRO
    IVIEWH(I)=0
  2060 IF(IJAC.LT.IIPRO)GC TC 2065
  CO 2050 I=1,IPRO
  2050 IVIEWH(I)=0
  CTIME=C
  CYREPET=CYPLCT
  JO=C

```

0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816

```

IJC=C
IJAC=C
CONT INDE
CALL GRAF+C(1,IVIEWC,IPRC,8,IER)
628 CONT INDE
IF(IJC.NE.9)GO TO 625
IIPRF=IPRF-1
JP=JP+1
IJP=4+JP
IJAP=IJF+1
PTIME=PTIME+DELT
PTIMPLT=(PTIME/SF1)-SF14
PXPLCT=PTIMPLT
PYPLCT=(COT/(10.*QDQIM))+.1
IVIEWP(1)=IHEAD(0,3)
IVIEWP(2)=IPACK(.0,C,C)
IVIEWP(3)=IPACK(-9,PYREFET,0)
IVIEWP(4)=IPACK(-9,FYREFET,1)
IVIEWP(IJP)=IPACK(PXPLOT,PYPLOT,1)
DO 2061 I=1,IJAP,IPRP
IVIEWP(I)=0
IF(IJAF.LT.IIPRP)GO TC 2066
DO 2091 I=1,IPRP
IVIEWP(I)=0
PTIME=C
PYREFET=PYPLOT
JP=0
IJP=C
IJAP=0
CONT INDE
CALL GRAF+C(1,IVIEWP,IPRF,9,IER) 10
629 CONT INDE
IF(IJC.NE.13)GO TO 633
IIPRC=IPRC-1
JQ=JQ+1
IJQ=4+JQ
IJAC=IJC+1
QTIME=QTIME+DELT
QTIMPLT=(QTIME/SF1)-SF14
QXPLOT=QTIMPLT
CYPLCT=(P/(10.*PMAK))+.3
IVIEWC(1)=IHEAD(0,1C)
IVIEWC(2)=IPACK(.C,C,C)
IVIEWC(3)=IPACK(-9,CYREFET,0)
IVIEWC(4)=IPACK(-9,FYREFET,1)
IVIEWC(IJQ)=IPACK(QXPLOT,CYPLOT,1)
DO 2062 I=1,IJAC,IPRC

```

```

2062 IVIEWQ(1)=C
      IF(1JAK-LT.11PRQ)GO TO 2067
      DO 2092 I=1,1PRQ
2092  IVIEWQ(I)=C
      QTIME=C.
      CYREFET=CYFLOT
      JQ=0
      IJQ=C
      IJAU=C
2067  CCNT INUE
      CALL GRAFFC(1,IVIEWC,IPRC,13,IER) 9-
      IF(1IER.NE.C)OUTPUT(1C1) IER,
622  CONTINUE
      IF(1ICC.NE.14)GO TO 634
      11PRR=1PRR-1
      JR=JK+1
      IJR=4+JJR
      IJAK=1JJR+1
      RTIME=RTIME+DELT
      RTIMPLT=RTIME/SF1)-SF14
      RXPLUT=RTIMPLT
      RYFLCT=((10.*QMAX))+.1
      IVIEWR(1)=IHEAD(0,1C)
      IVIEWR(2)=IPACK(.C,.C,C)
      IVIEWR(3)=IPACK(-.9,RVREFET,0)
      IVIEWR(4)=IPACK(-.9,RVREFET,1)
      IVIEWR(1JR)=IPACK(RXPLUT,RYFLOT,1)
      DO 2063 I=1JJR,1PRR
2063  IVIEWR(I)=0
      IF(1JAF-LT.11PRR)GO TO 2068
      DO 2072 I=1,1PRR
2072  IVIEWR(I)=C
      RTIME=C.
      RYREFET=RYFLOT
      JR=0
      IJR=C
      IJAK=C
2068  CCNT INUE
      CALL GRAFFH(1,IVIEWH,IPRH,14,IER) 8
634  CONTINUE
      IF(1ICC.NE.4)GO TO 635
      11PRS=1PRS-1
      JS=JS+1
      IJS=4+JJJ
      IJAS=1JJS+1
      STIME=STIME+DELT
      STIMPLI=(STIME/SF1)-SF14

```

0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864

0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912

```

SXPLCT=FYSESY
SYPLCT=FYSESY
IVIEWS(1)=IHEAD(0,1)
IVIEWS(2)=IPACK(0,0,0,0)
IVIEWS(3)=IPACK(SXREPET,SYREPET,0)
IVIEWS(4)=IPACK(SXREPET,SYREPET,1)
IVIEWS(1JS)=IPACK(SXFLCT,SYPLCT,1)
DU 2064 I=1JAS,IPRS
IVIEWS(1)=0
IF(1JAS.11,1IPRS)GC TC 2069
DO 2074 I=1,IPRS
IVIEWS(1)=0
STIME=C
SXREPET=FYSESY
SYREPET=FYSESY
JS=0
IJS=C
IJS=C
2064 IVIEWS(1)=0
IF(1JAS.11,1IPRS)GC TC 2069
DO 2074 I=1,IPRS
IVIEWS(1)=0
STIME=C
SXREPET=FYSESY
SYREPET=FYSESY
JS=0
IJS=C
IJS=C
2065 CONTINUE
CALL GRAFFC(1,IVIEWS,IPRS,4,IER) 5
IF(1IER.NE.C)OUTPUT(101) IER,
635 CONTINUE
IF(1ICC.NE.16)GO TO 636
653 ENCODE(56,501,ILAP)YC,XO,ZZ,FSI,PHI,THETA
600 CALL TEXT(2,ILAP,24,10,4,1,3,IER)
636 CONTINUE
IF(1ICC.NE.17)GO TO 637
110 ENCODE(56,526,ILAP)F(31),H(30),H(32),H(33),H(34),H(35)
CALL TEXT(2,ILAP,24,13,4,1,3,IER)
637 CONTINUE
IF(1ICC.NE.18)GO TO 638
108 ENCODE(56,525,ILAP)H(2),H(1),H(3),H(4),F(5),F(6)
CALL TEXT(2,ILAP,24,12,4,1,3,IER)
638 CONTINUE
IF(1ICC.NE.19)GO TO 639
601 ENCODE(56,507,ILAP)U,V,W,R,P,Q
CALL TEXT(2,ILAP,24,18,4,1,3,IER)
639 CONTINUE
IF(1ICC.NE.20)GO TO 640
115 ENCODE(56,528,ILAP)F(36),H(37),H(38),H(39),H(40),H(41)
CALL TEXT(2,ILAP,24,21,4,1,3,IER)
640 CONTINUE
IF(1ICC.NE.21)GO TO 641
113 ENCODE(56,527,ILAP)F(7),F(8),H(9),H(10),H(11),H(12)
CALL TEXT(2,ILAP,24,20,4,1,3,IER)
641 CONTINUE
IF(1ICC.NE.22)GO TO 642
604 ENCODE(56,511,ILAP)CDFI,VDOI,WDOI,RDOI,PDOI,QDOI

```

```

042 CALL TEXTIC(2,ILAP,24,26,4,1,3,IER)
    CONTINUE
120 IF(ICC.NE.23)GO TO 643
    ENCODE(56,530,ILAP)F(42),H(43),H(45),H(46),H(47)
    CALL TEXTIC(2,ILAP,24,29,4,1,3,IER)
643 CONTINUE
    IF(ICC.NE.24)GO TO 644
118 ENCODE(56,529,ILAP)F(13),H(14),H(15),H(16),F(17),H(18)
    CALL TEXTIC(2,ILAP,24,28,4,1,3,IER)
044 CONTINUE
    IF(ICC.NE.25)GO TO 645
    ENCODE(56,531,ILAP)F(17),I(8),T(9),T(10),SEAST,CASST,KK,SPCCM
    CALL TEXTIC(1,ILAP,24,37,1,1,3,IER)
645 CONTINUE
    CALL REACCLOCK(N)
    CALL WRITECLOCK(O)
    IF(SENSE SWITCH 5)74C,74I
740 DELT=DELTA
    GO TO 742
741 CONTINUE
    DELT=N/TINT
742 CONTINUE
    TIME=TIME+DELT
010 IF(ICC.LE.24)GO TO 101
    ICC=C
    GO TO 101
END
SUBROUTINE BUAT
    DIMENSION IVIEWA(75)
    COMMON A41,U,DELT,UCCCT,V,VOLT,K,RDUT,P,PDUT,Q,QDUT,THETA,T,AM,
    CCDP,L2,A11,A21,A31,X,AXS1,AYST
    CZ1,CLX,RZ,OO,COY,WW,A22,RDP,PHI,RDS,FSP,FSS,COZP,A33,RX,FL,F2,
    CF3,LP,LL,PB,WE,RY,L3,A34,Z,BETA,HYSESX,XST,FYSESY,YST,UMAX,XGX,
    CXDX,UMLL,FEAD,VHIGH,SFDLIN,AICCA,YO,ICCA,SEAST
    AUSVO=AES(YO)
    HUR3=XIXE-.13
    HUK5=XIXA+.13
    YAWPX=2.5#R
    YAWPY=TFETA
    XIX=.45*CGS(PHI)
    YLY=.45*SIN(PHI)
    XIXA=(2.5#R)+XIX
    XLI=2.5#F
    YLYA=TFETA-YLY
    XLYB=(2.5#R)-XIX
    YLYB=TFETA+YLY
    XLYC=(2.5#H)+(O.1*SIN(PHI))
    YLYC=TFETA+(O.1*CGS(PHI))

```


1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056

```

IF (RCAL2X.GT.1.)GO TO 36
IF (RCAL2X.LT.-1.)GO TO 37
GO TO 38
36 CCNTINLE
ROAD2X=1.
GO TO 38
37 RCAL2X=-1.
GO TO 38
38 CCNTINLE
RODHED1=18C.
RODHED2=FEAD+180.
IF (RCDFEC2.LT.360.)GO TO 42
RODHED2=RODHED2-36C.
CONTINLE
42 RODDIF=((RODFED1-RODFED2)*3.14)/180.
RODHED4=SIN(RODDIF)
ROAD3X=RODHED4+ROAD4X
IF (RCAL3X.GT.1.)GO TO 39
IF (RCAL3X.LT.-1.)GO TO 40
GO TO 41
39 CCNTINLE
ROAD3X=1.
GO TO 41
40 CONTINLE
ROAD3X=-1.
GO TO 41
41 CCNTINLE
YRCAC=AES(ROAD4X)
ROAD3Y=YHIGH-((YRCAC)*YHIGH)
ROAD1Y=C.C1
ROAD2Y=C.C1
ROAD4Y=C.C1
IF (AESYC.GT.1500.)GO TO 47
IF (RCDFEC2.LT.270.)GO TO 46
GO TO 47
46 IF (RCDFEC2.GT.090.)GO TO 45
GO TO 47
47 CCNTINLE
ROAD1X=-1.
ROAD2X=-1.
ROAD3X=-1.
ROAD4X=-1.
ROAD1Y=C.
ROAD2Y=C.
ROAD3Y=C.
ROAD4Y=C.
CCNTINLE
45
65

```

1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104

```

182 CONTINUE
IF(VIVE.GI.SPOLINIGC IG 79
ICCA=0
CONTINUE
CONTINUE
IVIEWA(1)=IHEAD(0,10) REAR2,0)
IVIEWA(2)=IPACK(XIXB,VIVB,1)
IVIEWA(3)=IPACK(XIXA,VIVA,1)
IVIEWA(4)=IPACK(XIXA,VIVA,1)
IVIEWA(5)=IPACK(XIXA,VIVA,1)
IVIEWA(6)=IPACK(XIXA,VIVA,1)
IVIEWA(7)=IPACK(XIXA,VIVA,1)
IVIEWA(8)=IPACK(XIXA,VIVA,1)
IVIEWA(9)=IPACK(XIXA,VIVA,1)
IFR.GI..C2)GO TO 5C
IFR.LI..C2)GO TO 51
IVIEWA(10)=IPACK(XIXC,VIVEE,0)
IVIEWA(11)=IPACK(XIXE,VIVE,0)
IVIEWA(12)=IPACK(XIXF,VIVEE,0)
IVIEWA(13)=IPACK(XIXC,VIVEE,1)
IVIEWA(14)=IPACK(XIXC,VIVEE,1)
IVIEWA(15)=IPACK(XIXC,VIVEE,1)
IVIEWA(16)=IPACK(XIXC,VIVEE,1)
IVIEWA(17)=IPACK(XIXC,VIVEE,1)
IVIEWA(18)=IPACK(XIXC,VIVEE,1)
GO TO 55
IVIEWA(10)=IPACK(XIXA,VIVA,C)
IVIEWA(11)=IPACK(XIXA,VIVA,C)
IVIEWA(12)=IPACK(XIXA,VIVA,C)
IVIEWA(13)=IPACK(XIXA,VIVA,C)
IVIEWA(14)=IPACK(XIXA,VIVA,C)
IVIEWA(15)=IPACK(XIXA,VIVA,C)
IVIEWA(16)=IPACK(XIXA,VIVA,C)
IVIEWA(17)=IPACK(XIXA,VIVA,C)
IVIEWA(18)=IPACK(XIXA,VIVA,C)
GO TO 55
IVIEWA(10)=IPACK(XIXB,VIVB,0)
IVIEWA(11)=IPACK(XIXB,VIVB,0)
IVIEWA(12)=IPACK(XIXB,VIVB,0)
IVIEWA(13)=IPACK(XIXB,VIVB,0)
IVIEWA(14)=IPACK(XIXB,VIVB,0)
IVIEWA(15)=IPACK(XIXB,VIVB,0)
IVIEWA(16)=IPACK(XIXB,VIVB,0)
IVIEWA(17)=IPACK(XIXB,VIVB,0)
IVIEWA(18)=IPACK(XIXB,VIVB,0)
CONTINUE
IVIEWA(19)=IPACK(YAMPX,YAMPY,0)
IVIEWA(20)=IPACK(YAMPX,YAMPY,0)

```

1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152

```

IVIEWA(21)=IPACK(YAMPX,YAMPY,1)
IVIEWA(22)=IPACK(YAMPX,YAMPY,1)
IVIEWA(23)=IPACK(RCADIX,FCADIV,1)
IVIEWA(24)=IPACK(RCADIX,FCADIV,1)
IVIEWA(25)=IPACK(RCADIX,FCADIV,1)
IVIEWA(26)=IPACK(RCADIX,FCADIV,1)
IVIEWA(27)=IPACK(RCADIX,FCADIV,1)
IVIEWA(28)=IPACK(RCADIX,FCADIV,1)
IVIEWA(29)=IPACK(RCADIX,FCADIV,1)
IVIEWA(30)=IPACK(RCADIX,FCADIV,1)
IVIEWA(31)=0
CALL GRAPPG(1,IVIEWA,31,3,IER)
3333 CONTINUE
RETURN
END
SUBROUTINE CNEMOT
DIMENSION A4(1),U,DELTA,ULCT,V,VDOIT,R,RDUT,P,PUOT,G,ODOT,THETA,T,AM,
COMMON L2,A1,A21,A31,X,AXS1,AYST,
LCDP,L2,A1,A21,A31,X,AXS1,AYST,
CF3,LF,LC,FB,WE,RY,L3,A34,Z,BETA,HYSESY,XSI,HYSESY,YST,UPAX,XGX,
CXDX,LP,LL,FEAD,YHIGH,SPDLIN,AICCA,YO,ICCA,SEAST
COMMON /CATA/IAKR
CDX1=230000.
CDX2=5575.31
CDX3=13.46
CDX4=13500000.
CDX5=1372.56
Z1=Z-BETA
A41=LC*(IAKR(2)/2**23
U=U+DELTA*LCOT
IF(U.GT.20.3)GO TC 4
FSD=CDX1*(L/30.3)**3)
GO TC 3
CONTINUE
IF(U.GT.27.03)GO TC 1
FSD=635200.59-CDX5*U
GO TC 2
CONTINUE
IF(U.GT.63.97)GO TC 2
FSD=CDX2*(L-37721.
GO TC 3
FSD=CDX3*U**2+CDX4/(1+U**1.5)
CONTINUE
V=V+DELTA*VDOIT
R=R+DELTA*RCCT
P=P+DELTA*EDGT
C=C+DELTA*CCGT

```

```

PHI=PHI+ULL*P
THETA=THETA+DELT*Q
LUCT=((1/AM)*COS(Z1))-1S/(AM*V*R
VUCT=((1/AM)*SIN(Z1))-((CDY/AM)*V*ABS(V))-U*R
WUCT=((1-1.0)*(1/RZ)*CL*SIN(Z1))*V*H*ABS(V)
C*((A22*L*V*H)/RZ
KDP=6.-5C*((SIN(PHI))/(COS(PHI)))
KDS=6.-5C*((SIN(PHI))/(COS(PHI)))
FSP=128CCG.*RDS
FSS=128QCG.*RDS
PDCI=((FSP-FSS)*50.)-(1*(SIN(Z1)))+(CLY*V*(ABS(V))*30.)-(
C(CD/P*V*(ABS(V))*5C.)-(A3*U*P)/RX
F1=192C00.-32C0000.*(SIN(THETA))/(COS(THETA))
F2=192C00.-32C0000.*(SIN(THETA))/(COS(THETA))
F3=COP*PE*E*(LD-L3*(SIN(THETA))/(COS(THETA)))
LDCT=((1)+(COS(Z1))*LP)+F3*L3+F2*L3-FSD*L2-F1*L3-A34*U*Q)/RY
HYSESA=>ST
HYSESY=YST
RETURN
END

```

```

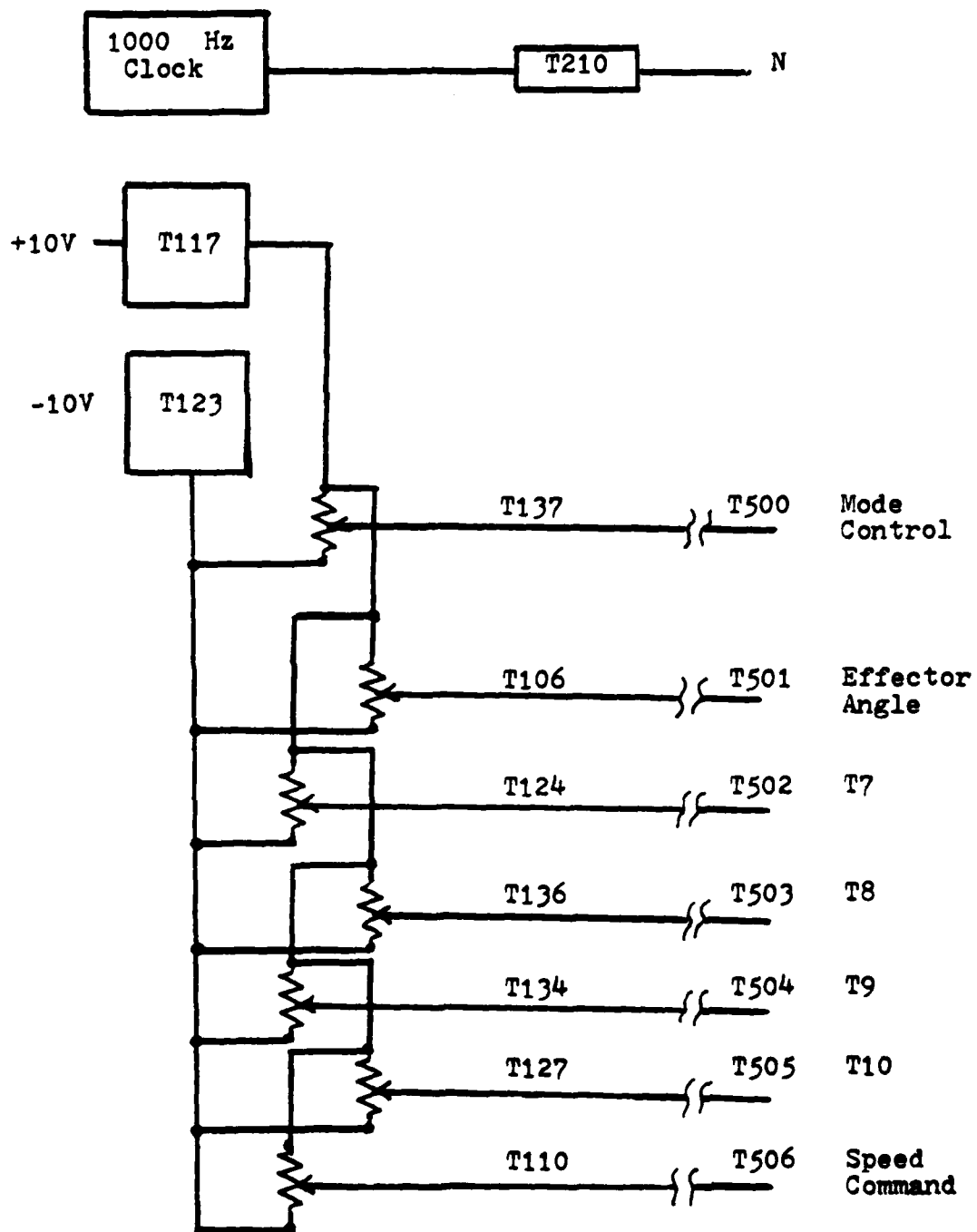
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172

```

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026

9SETUPJ
1
BUF
=C77777
CLMM
=7
15
=CCNM
CCNTR
ENDBRM
040
C34001
CCNTR
AL
ENDAD
034001
CCNTR
ENDAD
7
PZE
BRM
PZE
PZE
LEA
EIK
STA
LLA
LLSA
ALD
STA
LUA
STA
ELM
PLT
BRM
BRM
PZE
DIR
ECH
PLT
EIK
BRL
RES
PZE
END
PZE
BRM
ENDAD
LUAH
CUNTR
END

Appendix D RTS5D Modified Wiring Diagram



LIST OF REFERENCES

1. Nelson, T.P. III, Real Time Simulation and Control 3000 Ton Surface Effect Ship M. S. Thesis, Naval Postgraduate School, Monterey, California, 1979.
2. Kaplan, P., Bentson, J., Sargent, T.P., A Study of Surface Effect Ship (SES) Craft, Loads and Motions, Part I, Equations of Motion of SES Craft with Six Degrees of Freedom, Contract No. C-1-35055, Report No. 71-84A, Oceanics Inc., Technical Industrial Park, Plainview, N.Y., August 1971.
3. Lockheed Missiles and Space Company, Inc., Surface Effect Ships Aero/Hydrodynamics Technology Design Manual, Vol. I, Contract No. N00024-73-A-0919, Task Order No. 0003, May 1974.
4. Gerba, A.Jr., Thaler, G.J., Development of a Real Time Hybrid Computer Simulation for the 3K-SES, 500F, Data Based Program, Part I- Familiarization and Planning, Progress Report to SESPO, U.S. Naval Postgraduate School, Monterey, California, July 1979.
5. Department of the Navy, Code 1630: PRV, Surface Effect Ship Data Based Maneuvering Simulation, SES Project PMS-304, September, 1979.
6. David W. Taylor Naval Ship Research and Development Center, Viars, P.R. Unclassified Letter 1630: PRV:wjj to Gerba, A. U.S. Naval Postgraduate School, Subject: Conditions for Constant Speed 3KSES Maneuvers, 5 December 1979.
7. Greensite, A., Pascol, L., Wiemer, M., 3K SES Ship Control System Design Information Report, Volume 1, CDRL No. E05P, Document No. TER023A, Rohr Marine, Inc., Chula Vista California, 24 July 1978.
8. Sturgeon, M.E., The Array Processor: A Real Time Simulator, Paper Presented at the Summer Computer Simulation Conference, Seattle, Washington, August 1980.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Professor Alex Gerba, Jr., Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5. Professor George J. Thaler, Code 62Tr Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Mr. H. Robertson PMS 304, Code 141 Surface Effects Ships Project Office P. O. Box 34401 Bethesda, Maryland 20034	6
7. Commanding Officer Naval Electronic Systems Engineering Center, Vallejo Attn: LT Lee L. Oliphant, USN Vallejo, California 94590	1

DATE
FILMED
8