

**NASA  
Technical  
Memorandum**

NASA TM -100356

**AUTOMATIC MATHEMATICAL MODELING FOR  
REAL TIME SIMULATION PROGRAM  
(AI APPLICATION)**

By Caroline Wang and Steve Purinton

Software and Data Management Division  
Information and Electronic Systems Laboratory  
Science and Engineering Directorate

February 1989

(NASA-TM-100356)  
MODELING FOR REAL  
(AI APPLICATION)

AUTOMATIC MATHEMATICAL  
TIME SIMULATION PROGRAM  
(NASA) 66 F CSCL 12A

N89-18147

Unclas  
G3/64 0192817



National Aeronautics and  
Space Administration

George C. Marshall Space Flight Center

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. NASA TM-100356		2. GOVERNMENT ACCESSION NO.		3. RECIPIENT'S CATALOG NO.	
4. TITLE AND SUBTITLE Automatic Mathematical Modeling for Real Time Simulation Program				5. REPORT DATE February 1989	
				6. PERFORMING ORGANIZATION CODE	
7. AUTHOR(S) Caroline Wang and Steve Purinton				8. PERFORMING ORGANIZATION REPORT #	
9. PERFORMING ORGANIZATION NAME AND ADDRESS George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812				10. WORK UNIT, NO.	
				11. CONTRACT OR GRANT NO.	
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D.C. 20546				13. TYPE OF REPORT & PERIOD COVERED Technical Memorandum	
				14. SPONSORING AGENCY CODE	
15. SUPPLEMENTARY NOTES Prepared by Software and Data Management Division, Information and Electronic System Laboratory, Science and Engineering Directorate.					
16. ABSTRACT  This report describes a methodology for automatic mathematical modeling and generating simulation models. The major objective was to create a user-friendly environment for engineers to design, maintain, and verify their models; to automatically convert the mathematical models into conventional code for computation; and finally, to document the model automatically.					
17. KEY WORDS  Automatic Modeling for Space Application Mathematical			18. DISTRIBUTION STATEMENT  Unclassified - Unlimited		
19. SECURITY CLASSIF. (of this report) Unclassified		20. SECURITY CLASSIF. (of this page) Unclassified		21. NO. OF PAGES ✓ 67	22. PRICE NTIS

## TABLE OF CONTENTS

	Page
I. SUMMARY .....	1
II. INTRODUCTION .....	2
III. PURPOSE .....	2
IV. INTELLIGENT INTERFACES .....	2
V. AUTOMATIC MODELING .....	3
VI. SYSTEM OVERVIEW .....	3
Flow Diagram .....	3
Basic Software and Hardware Requirement.....	
VII. MAINTAINING AND VERIFYING .....	9
VIII. GENERAL APPLICATION .....	9
IX. EXAMPLE APPLICATION .....	10
X. FUTURE PLANS .....	11
APPENDIX .....	13
Sample Problem Solving for SSME Math Model.....	14
Knowledge Base .....	14
Generic Equations .....	15
Information Base .....	21
System Generated Equations .....	37
System Generated FORTRAN Code .....	43
System Generated Documentation.....	46
Robot Manipulators.....	55
System Generated Equations .....	57

**PRECEDING PAGE BLANK NOT FILMED**

## TECHNICAL MEMORANDUM

# AUTOMATIC MATHEMATICAL MODELING FOR REAL TIME SIMULATION PROGRAM

### I. SUMMARY

This report describes a methodology for automatic mathematical modeling and generating simulation models. The models will be verified by running in a test environment using standard profiles with the results compared against known results. The major objective is to create a user-friendly environment for engineers to design, maintain, and verify their models; to automatically convert the mathematical models into conventional code for computation; and, finally, to document the model automatically.

The traditional method for simulation modeling is to define, derive, and organize the equations and then to develop the program for manual computation. If any modification or correction needs to be done in the design or mathematical equations, it will create a tremendous amount of work for the rest of the process. Generally, any model will require several modifications before it matches a real system. Historically, the modifications have been time consuming and a fertile source of errors.

Some of the available symbolic mathematics tools will help to derive the equation symbolically, and automatically produce the final equation and program code. The use of the LISP language can build the user-friendly interface and generate the knowledge base for the symbolics mathematics tool to build equations and program codes. After the equations and codes have been stored in the files, the LISP program can then combine and organize them and create a complete compiled source code for any conventional language requirement. A demonstration program was designed for modeling the Real-Time Simulation system. It is written in LISP and MACSYMA and runs on a Symbolics 3670 LISP Machine. It contains an initial set of elements for the Real-Time Simulation system and a questionnaire that allows the engineer to answer a set of questions to specify a particular model. The system is then able to automatically generate the model and FORTRAN code. The future goal which is under development is to download the FORTRAN code to the VAX/VMS system for conventional computation. The mathematical model will be verified in the test environment and the solution compared with the real data profile. Control, display and ancillary systems will be developed which will allow the execution of a predefined profile (from a file) or interactive modification of the variables. Variables can be displayed in a text format or as a plot from the control and display system. An execution control module will be available to allow time and sequence control of the model and peripheral models. A recording module will be available for execution in series with the math model and will record (optional) interface variables transparently. This module will use packet definition variables and circular queues to determine what and when to record.

The use of Artificial Intelligence (AI) techniques has shown that the process of simulation modeling can be simplified.

## **II. INTRODUCTION**

Mathematical modeling for real-time simulation programs is a very complicated process which includes analysis, design, and the generation of complex equations and programs. Generally, the model will require several modifications before it will match a real system. Historically, the modifications have been time consuming and a fertile source of errors.

The use of AI techniques has shown that this process can be simplified. Some of the AI software tools will allow us to create a user-friendly environment for engineers to design, maintain, and verify their model and also to automatically convert the mathematical model into a conventional programming language for execution.

## **III. PURPOSE**

The major objective is to develop and create a very comfortable environment for engineers to design and maintain their model. The automatic Mathematical Modeling generates knowledge base, mathematical equations, and conventional program codes. It helps to simplify the process of the modeling, and cuts down the development time and errors.

The automatic generation of a math model in simulations will add confidence in the simulation. Using interfaces which are well defined and validated will reduce programming and debug time and will allow concentration on logic and equations.

This technique can be applied to many different science and engineering projects.

## **IV. INTELLIGENT INTERFACES**

The intelligent interfaces include:

- 1) The user interface
- 2) The interface between knowledge base and automatic model generation
- 3) The interface between modules comprising the simulation.

Automatic modeling requires a very friendly work environment to collect the necessary information and to generate the knowledge base. Through the knowledge base information, the system automatically creates the mathematical equations and generic program codes for equations. Next, all the equation codes are generated and linked by another AI program to combine and organize the codes together and to build a completely compiled program.

An intermodule interface will become part of the automatic model generation process. Variables will be identified and an external interface will provide the variable and its type to external modules. These external modules will record, display, or modify the variable.

## **V. AUTOMATIC MODELING**

The traditional method for simulation modeling is to manually define, derive, and organize the equations and then to develop the program for computation. If any modifications or corrections need to be done in the design or mathematical equations, it will create a tremendous amount of work for the rest of the process. Generally the model will require several modifications before it matches a real system. Historically, the modifications have been time consuming and a fertile source of errors.

Some of the available symbolic mathematics tools will help us to derive the equation symbolically and to automatically produce the final equation and program code. The use of LISP language can build the user-friendly interface and generate the knowledge base for the symbolics mathematics tool to build equations and program codes. After the equations and codes have been stored in the files, the LISP program can then combine and organize them and create a complete compiled source code for any conventional language you require.

The use of AI techniques has shown that the process of simulation modeling can be simplified.

Control, display, and ancillary systems will be developed which will allow the execution of a predefined profile (from a file) or interactive modification of the variables. Variables can be displayed in a text format or as a plot from the control and display system. An execution control module will be available to allow time and sequence control of the model and peripheral models. A recording module will be available for execution in series with the math model and will record (or not record) interface variables transparently. This module will use packet definition variables and circular queues to determine what and when to record.

## **VI. SYSTEM OVERVIEW**

### **A. Flow Diagram**

START

Define the problem

Define generic equations

Lisp program for user interface

automatic knowledge base  
generation

Read knowledge Base and  
automatically generate the  
Macsyma code for Symbolic  
Math. tool "Macsyma" to  
recognize the information

Get in to Macsyma window to  
process the set of Macsyma  
code which has been generated  
and automatically create  
the equations and FORTRAN code  
for the model

Save the final symbolic equation  
and FORTRAN equation code into  
the disk files

Another set of Lisp function  
to read the equation code and  
build a compiled conventional  
program for computation

Verifying the result and maintain  
the equations or knowledge base  
and process the whole cycle again  
until correct

## B. Basic Software and Hardware Requirements

The Basic software tools used were "MACSYMA" and "LISP." The hardware was "Symbolics 3670" and "VAX." We used a Symbolics for the equation and conventional code generation and then downloaded to the VAX system for computation.

There are a number of tools available for Symbolic Mathematical equation and conventional program code generation. Some examples are as follows:

Symbolics Math Tool	Memory Requirements	Available Computers
MACSYMA	2 Megabytes	DEC 10 or 20 VAX Honeywell 6000 Series Symbolics LM2 or 3600
SMP	2 Megabytes	VAX 730, 750, or 780 Using UNIX operating system
REDUCE	350 kilobytes	IBM-360 or 370 DEC 10 or 20 VAX UNIVAC 1100 CDC CYBER CRAY-1 BURROUGHS 6700 APOLLO DOMAIN AND OTHERS
MuMATH	512 kilobytes	Personal Computer
MAPLE	350 kilobytes	VAX Using UNIX 4.3
ALTRAN	270 kilobytes	1966 Standard FORTRAN
FORMAC	150 kilobytes	IBM-360 or compatible machine
SAC-2	120 kilobytes	1966 Standard Fortran

Hardware requirements are for the support of tasking and for the provision of adequate address space.



## Features for the Most Popular Tools

### MAC'S SYmbolic MAnipulation (MACSYMA)

#### Features:

1. Solve equations, factor polynomials, expand functions in series
2. Equation simplification
3. Manipulate matrices, trigonometric equation
4. Integration and differentiation
5. Automatic code generation for FORTRAN program
6. File handling system
7. Interactive help facility
8. Control function and subroutine procedures
9. 3D graphics, automatic scaling, and different viewpoints
10. Interactive sytem
11. Excellent working environment
12. Ease of use

#### Available computers:

DEC 10 or 20  
VAX  
Honeywell 6000 Series  
Symbolics LM2 or 3600

#### Memory requirements:

Two Megabytes

## Symbolic Manipulation Program (SMP)

### Features:

1. Solve equations, factor polynomials, expand functions in series
2. Equation simplification
3. Manipulate matrices, trigonometric equation
4. Integration and differentiation
5. Automatic code generation for FORTRAN or C program
6. File handling system
7. Interactive help facility
8. Control function and subroutine procedures
9. 3D graphics
10. Interactive system
11. Excellent working environment
12. Ease of use

### Available computers:

VAX

### Memory requirements:

Two megabytes

## REDUCE II

### Features:

1. Solve equations, factor polynomials
2. Equation simplification
3. Manipulate matrices

4. Integration and differentiation
5. Automatic code generation for FORTRAN program
6. File handling system
7. Control function and subroutine procedure
8. Interactive system

Available Computers:

IBM-360 or 370  
DEC 10 or 20  
VAX  
UNIVAC 1100  
CDC CYBER  
CRAY-1  
BURROUGHS 6700  
APOLLO DOMAIN AND OTHERS

Memory requirements:

350 Kilobytes

MuMath

Features:

1. Solve equations, factor polynomials
2. Equation simplification
3. Manipulate matrices
4. Integration and differentiation
5. Interactive system

Available computers:

IBM PC or compatible system

## VII. MAINTAINING AND VERIFYING

One of the most difficult problems in software today is the verification and maintenance of existing programs, especially programs built up over time with many programmers involved. This is particularly true for simulation programs. The traditional method of modifying simulation programs by rebuilding and recoding the model is a potential source of many errors.

The automatic system can eliminate most of these problems. If in error, the only place which need be considered to change is the knowledge base at the front, then the rest of the equation and program can automatically be changed. The procedure becomes much simpler, which makes it easy for the user to maintain and modify the model and program directly.

Improving maintenance and verification is a major goal for this type of model generation. Allowing maintenance to be concerned with equations and logic rather than the implementation is a goal for maintenance.

## VIII. GENERAL APPLICATION

The AI technique for automatic mathematical modeling can apply to many problems which are required to be solved:

- 1) Complicated mathematical equations derivations that are hard or impossible for humans to do manually.
- 2) Equations which share the generic theory.
- 3) Models which need to be changed considerably.

Robotics, contact dynamics, and other problems which require a lot of complicated and long equation derivation in Matrix Multiplication can very easily be solved symbolically.

The Space Shuttle Main Engine Simulation contains several equations where there are sets of equations sharing the generic theory. The engine model is required to change when the engine design in hardware is modified. This is the case used for the Automatic Modeling Example.

There are many simulation models which need to be changed under different conditions. If they are done manually, it will cost a tremendous amount time to modify the model, change the program code, and verify the result before it can be used. The AI technique totally improves the environment, and the only place which needs to be modified is the knowledge base; the rest of the product can be automatically generated.

Spacelab training applications where several physical models will be developed for a single flight. Automatic generation of individual models from equations and logic will eliminate much of the repetitive system integration.

## IX. EXAMPLE APPLICATION

A demonstration program, called Propulsion System Automatic Modeling (PSAM), was designed for modeling the Space Shuttle Main Engine Simulation Mathematic Model. PSAM is written in LISP and MACSYMA and runs on a Symbolics 3670 LISP machine.

The design goals for PSAM were to develop automatic modeling skills for propulsion systems and other scientific and engineering applications. The old Engine Model was used as an example to study.

PSAM includes the following features:

- 1) User-friendly interface
- 2) Automatic Knowledge Base generation
- 3) Automatic Equation and Coding generation
- 4) Automatic documentation generation.

The Space Shuttle Main Engine Simulation model was built up from the component process elements and their combination into the subprograms.

The component process elements are pump, hot gas turbine, hydraulic turbine, turbopump, combustor, valve, incompressible propellant flow, injector volume with priming for start, hot gas heat transfer, and regenerative cooling flow. The subprograms are fuel, oxidizer, and hot gas.

There are two types of information for a PSAM knowledge base. One is the component process elements generic equations, and the other is the information base for the combination of the Space Shuttle Main Engine model subprograms and component process elements.

The system collects the detailed requirement and generates a set of specific equations for the component process elements and subprograms.

PSAM has the ability to:

- 1) Create or maintain the knowledge base
- 2) Load different knowledge
- 3) Automatically generate equations
- 4) Output generated equation or FORTRAN code to disk file or option for printout of the Laser Printer
- 5) Automatic documentation.

The FORTRAN code is in generic conventional program format only for the equations. Another part of the LISP program will combine all the subprograms and component process elements equations and append the header program codes to become a whole compiled program. The coding format for the equations are the same for many available conventional software languages such as FORTRAN, C, or ADA, etc. So the final program can be built on whichever conventional software the user requires by changing only the header requirements and I/O functions.

Several versions and configurations of SSME will be generated and run with a version of the engine controller software.

Another example program was designed for automated kinematic equation generation for robot manipulators. Any manipulator can be considered to consist of a series of links connected together by joints. Historically, the homogeneous transformation describing the relation between one link to the next has been called an A matrix. The position and orientation of the nth link in base coordinates are given by the matrix product

$$T_n = A_1 * A_2 * A_3 * \dots A_n$$

These products of A matrices have been called T matrices. An n-link manipulator can have n degrees of freedom, one for each link, and can be positioned and oriented arbitrarily within its range of freedom. Most robot controllers equations were developed manually. The complicated matrix multiplication is inverse with trigonometric functions. It is tedious and difficult. A small, careless mistake can cost major modification of the design. Using the AI technique to derive and simplify the equations symbolically and automatically can reduce a tremendous amount of effort.

These are the proof of concept examples which lead to future projects.

## **X. FUTURE PLANS**

Currently the existing project is being used to test the method. In the future this technique can be applied to numerous new projects. The goal is to build the simulation model and maintain it automatically.

The knowledge base will also contain detailed information for automatic documentation.

The future plans for PSAM are to generate models based upon logic and flow rather than equations, and to build control and display modules which will complement the models.

## APPENDIX

### Sample Problem Solving for

- Space Shuttle Main Engine Math Model
- Robot Manipulator

PRECEDING PAGE BLANK NOT FILMED

**SSME Math Model**  
**Knowledge Base**  
**Generic Equations**  
**Information Base**



```

(D105)                                     A1:>PSAM>Macysma-kb>psam-generic-equation.out.1
(C106) BATCH("al:>psam>macysma-kb>psam-generic-equation.kb");

(C107) "*****PUMP*****"$
(C108) "Pump Flow Variable"$
(C109) PFV(F,B,DW,S):=
Block ( F = B * (DW / S);
(D109)                                     B DW
PFV(F, B, DW, S) := BLOCK(F = ----)
                                      S

(C110) "Pump Total Pressure"$
(C111) PTP(PP,P,B,S,T):=
Block ( PP = P + B * (S)^2 * T);
(D111)                                     2
PTP(PP, P, B, S, T) := BLOCK(PP = P + B S T)

(C112) "Pump Torque"$
(C113) PT(R,B,S,T):=
Block ( R = B * (S)^2 * T);
(D113)                                     2
PT(R, B, S, T) := BLOCK(R = B S T)

(C114) "*****TURBINE*****"$
(C115) "Turbine Torque"$
(C116) TT(R,B,P,T):=
Block ( R = B * P * T);
(D116)                                     B P T
TT(R, B, P, T) := BLOCK(R = B P T)

(C117) "Turbine Speed Parameter"$
(C118) TSP(M,B,S,T):=
Block ( M = B * S / (T^(1/2)));
(D118)                                     B S
TSP(M, B, S, T) := BLOCK(M = ----)
                                      1/2
                                      T

(C119) "Turbine Weight Flowrate"$
(C120) TWF(DW,B,dP,r):=
Block ( DW = (B * (dP) * r)^(1/2));
(D120)                                     1/2
TWF(DW, B, DP, R) := BLOCK(DW = (B DP R) )

(C121) "*****TURBOPUMP SPEED*****"$
(C122) "TurboPump Speed"$
(C123) TPS(S,B,Tq,S0):=
Block { S = B * Integrate ( Tq,t) + S0);
(D123)                                     /
                                     [
TPS(S, B, TQ, S0) := BLOCK(S = B I TQ dT + S0)
                                     ]
                                     /

(C124) "*****VALVE*****"$
(C125) "Valve Area"$
(C126) VA(A,T,X):=
Block ( A = T * X);
(D126)                                     VA(A, T, X) := BLOCK(A = T X)

(C127) "*****FLOW*****"$
(C128) "Fuel Flow"$
(C129) FF(DWf,Bi,P9,P,Bj,R,DW0):=
Block ( DWf:DWf(t), DW0:DW0(t),DWf = B * 'integrate (((P9 - P - (Bi / R)) * DWf^2),t) + DW0);

```

BI

**ORIGINAL PAGE IS  
OF POOR QUALITY**

```

2
(D129)      FF(DWF, BI, P9, P, BJ, R, DW0) := BLOCK(DWF : DWF(T), DW0 : DW0(T), DWF = B I (P9 - P - --
) DWF dT + DW0)
                                                    ]
                                                    /
                                                    R

(C130) "Oxidizer Flow"$

(C131) OF(DWo,Bk,Ppos,P,Bl,A,Ab,Bm,DWi,DW0):=
Block( DWo:DWo(t), DWi:DWi(t),DW0:DW0(t),Dwo = Bk * 'integrate ((Ppos - p - Bl * (DWo / (A/Ab))^2 - Bm*(DW
i)^2),t) + DW0);
(D131) OF(DWO, BK, PPOS, P, BL, A, AB, BM, DWI, DW0) := BLOCK(DWO : DW0(T), DWI : DWI(T), DW0 : DW0(T),
                                                    /
                                                    [
DWO 2          2
- BL (---) - BM DWI ) dT + DW0)
                                                    ]
                                                    /
                                                    R
A
--
AB

(C132) "Variable in injector Priming function, Dimensionless"$

(C133) VVPF(E,Bh,DW0,DWi,E0):=
Block(DW0:DW0(t), DWi:DWi(t), E = Bh * 'integrate ((DWo - DWi),t) + E0);
                                                    /
                                                    [
(D133)      VVPF(E, BH, DW0, DWI, E0) := BLOCK(DW0 : DW0(T), DWI : DWI(T), E = BH I (DW0 - DWI)
dT + E0)
                                                    ]
                                                    /

(C134) "Weight Flowrate Injector"$

(C135) WFI(DWi,DWo,Eo):=
Block( DWi = DWo * Eo);
(D135)      WFI(DWI, DWO, EO) := BLOCK(DWI = DWO EO)

(C136) "*****COMBUSTER*****"$

(C137) "Combuster Total Pressure"$

(C138) CTP(P,Bi,DW1,DW2,Bj,DW3,P0):=Block( DW1:DW1(t), DW2:DW2(t), DW3:DW3(t),P = Bi * 'integrate (( DW1
+ DW2 - Bj * DW3),t) + p0);
                                                    /
                                                    [
(D138) CTP(P, BI, DW1, DW2, BJ, DW3, P0) := BLOCK(DW1 : DW1(T), DW2 : DW2(T), DW3 : DW3(T), P = BI I (DW1
+ DW2 - BJ DW3) dT + P0)
                                                    ]
                                                    /

(C139) "Combuster Fuel Weight Flowrate"$

(C140) WFF(F,DW2,DW):=Block( F = DW2 / (DW1 + DW2));
(D140)      WFF(F, DW2, DW) := BLOCK(F = -----)
                                                    DW2
                                                    DW1 + DW2

(C141) "Combuster Temperature"$

(C142) CT(T,TT,Bk,T9):=Block ( T = TT + Bk * T9);
(D142)      CT(T, TT, BK, T9) := BLOCK(T = TT + BK T9)

(C143) "*****COOLING ELEMENT*****"$

(C144) "Cooling Element Total Pressure"$

(C145) CETP(P,B,DQw1,DQw2,H3,DWi,H,DW,P0):=
Block( DQw1:DQw1(t),DQw2:DQw2(t),DWi:DWi(t),DW:DW(t),P = B * 'integrate((DQw1 + DQw2 + H3 * DWi - H * DW
),t) + P0);
(D145) CETP(P, B, DQW1, DQW2, H3, DWI, H, DW, P0) := BLOCK(DQW1 : DQW1(T), DQW2 : DQW2(T), DWI : DWI(T),
DW : DW(T),
                                                    /
                                                    [
P = B I (DQW1

```

ORIGINAL PAGE IS  
OF POOR QUALITY

+ DQW2 + H3 DWI - H DW) dT + P0)

]  
/

(C146) "Cooling Element Specific Enthalpy"\$

(C147) CESE(H,B,T):=Block( H = B \* T);  
(D147)

CESE(H, B, T) := BLOCK(H = B T)

(C148) "Cooling Element Weight Flowrate Main Chamber Heat Exchanger"\$

(C149) CEWFM(DW,Bi,P,P5,Bj,R5,DW,DW0):=  
Block( DW:DW(t),DW = Bi \* 'integrate ((P - P5 - (Bj / R5) \* DW^2),t) +DW0);

(D149) ) dT + DW0)

CEWFM(DW, BI, P, P5, BJ, R5, DW, DW0) := BLOCK(DW : DW(T), DW = BI I (P - P5 -  $\frac{BJ DW^2}{R5}$ )

(C150) "Cooling Element Density"\$

(C151) CED(R,B,DWi,DW,R0):=  
Block( DWi:DWi(t), DW:DW(t),R = B \* 'integrate ((DWi - DW),t) + R0);

(D151) + R0)

CED(R, B, DWI, DW, R0) := BLOCK(DWI : DWI(T), DW : DW(T), R = B I (DWI - DW) dT

(C152) "Cooling Element Temperature"\$

(C153) CET(T,B,P,R):=  
Block( T = B \* ( P / R));

(D153)

CET(T, B, P, R) := BLOCK(T =  $\frac{B P}{R}$ )

(C154) "Cooling Element Heat Transfer Rate Hot Gas Wall"\$

(C155) HT(DQw,B,T,Tw,DWi):=  
Block( DQw = B \* (1.0 + 0.002 \* T) \* (Tw-T) \* (DWi)^0.8);

(D155)

HT(DQW, B, T, TW, DWI) := BLOCK(DQW = B (1.0 + 0.002 T) (TW - T) DWI<sup>0.8</sup>)

(C156) "Cooling Element Heat Transfer Rate TC"\$

(C157) HTTC(DQtc,B,Tc,Tw1,DWcn):=  
Block( DQtc = B \* (Tc - Tw1) \* DWcn^0.8);

(D157)

HTTC(DQTC, B, TC, TW1, DWCN) := BLOCK(DQTC = B (TC - TW1) DWCN<sup>0.8</sup>)

(C158) "Cooling Element Hot Gas Wall Temperature"\$

(C159) HGT(TT,B,DQtc,DQw1,T0):=  
Block( DQtc:DQtc(t), DQw1:DQw1(t), TT = B \* 'integrate ((DQtc - DQw1) ,t) + T0);

(D159) QW1) dT + T0)

HGT(TT, B, DQTC, DQW1, T0) := BLOCK(DQTC : DQTC(T), DQW1 : DQW1(T), TT = B I (DQTC - DQW1) dT + T0)

(C160) "Cooling Element Ambient thrust Chamber Temperature"\$

(C161) ATT(TT,B,DQw2,T0):=Block( DQw2:DQw2(t),TT = B \* 'integrate ((-DQw2),t) +T0);

(D161)

ATT(TT, B, DQW2, T0) := BLOCK(DQW2 : DQW2(T), TT = B I (- DQW2) dT + T0)

(C162) "\*\*\*\*\*Flowmeter\*\*\*\*\*"\$

(C163) "Volumetric Flowrate"\$

(C164) VF(Q,C1,T,C2,C3,C4,DW,C5,S):=  
Block( Q = [1.0 - C1 \* ( T - C2 )]\*[ C3 + C4 \* DW + C5 \* S]);

(D164)

VF(Q, C1, T, C2, C3, C4, DW, C5, S) := BLOCK(Q = [1.0 - C1 (T - C2)] [C3 + C4 DW

ORIGINAL PAGE IS  
OF POOR QUALITY

+ C5 S))

(C165) "Total Pressure"\$

(C166) TP(P,Pf,RF,DW,R):=  
Block( P = Pf - (RF \* DW^2)/R);

(D166) TP(P, PF, RF, DW, R) := BLOCK(P = PF -  $\frac{RF \cdot DW^2}{R}$ )

(C167) "Flowmeter Temperature"\$

(C168) FT(T,Ti,S,Coe):=  
Block ( T = Ti + Coe \* S);

(D168) FT(T, TI, S, COE) := BLOCK(T = TI + COE S)

(C169) "\*\*\*\*\*Preburner Fuel Supply Duct\*\*\*\*\*"\$

(C170) "Supply Duct Total Pressure"\$

(C171) SDTP(P,B,DW1,DW2,DW3,DW4,P0):=

Block( dw1:dw1(t), dw2:dw2(t), dw3:dw3(t), dw4:dw4(t), P = B \* 'integrate((DW1 - DW2 - DW3 + DW4), t) + P0);

(D171) SDTP(P, B, DW1, DW2, DW3, DW4, P0) := BLOCK(DW1 : DW1(T), DW2 : DW2(T), DW3 : DW3(T), DW4 : DW4(T)

(DW1 - DW2 - DW3 + DW4) dT + P0)

/  
[  
P = B I  
]  
/

(C172) "Supply Duct Temperature"\$

(C173) SDT(T, B, P, R):=  
Block( T = B \*(P / R));

(D173) SDT(T, B, P, R) := BLOCK(T =  $\frac{B \cdot P}{R}$ )

(C174) "Supply Duct Density"\$

(C175) SDD(R, DW1, DW2, Rr, B):=  
Block( R = [DW1 + DW2]/[(DW2/Rr) + B \* DW1]);

(D175) SDD(R, DW1, DW2, RR, B) := BLOCK(R =  $\frac{[DW1 + DW2]}{[--- + B \cdot DW1] \cdot RR}$ )

(C176) "\*\*\*\*\*High Pressure Oxidizer Pump Preburner Boost Stage\*\*\*\*\*"\$

(C177) "High Pressure Oxidizer boostpump Flow variable"\$

(C178) HPOPFV(F,B,DW,S):=  
Block(F = B \* (DW / S));

(D178) HPOPFV(F, B, DW, S) := BLOCK(F =  $\frac{B \cdot DW}{S}$ )

(C179) "High Pressure Oxidizer Boostpump Torque"\$

(C180) HPOPT(T,B,S,TT,F):=  
Block( T = B44 \* (S^2)\* TT \* (F));

(D180) HPOPT(T, B, S, TT, F) := BLOCK(T = B44 S<sup>2</sup> TT F)

(C181) "High Pressure Oxidizer Boostpump Pressure"\$

(C182) HPOPP(P3,B,S,TT,F,P2):=  
Block( P3 = B \* (S^2) \* TT \* F + P2);

(D182) HPOPP(P3, B, S, TT, F, P2) := BLOCK(P3 = B S<sup>2</sup> TT F + P2)

(C183) "High Pressure Oxidizer Boostpump Weight Flowrate"\$

(C184) HPOPW(DW3,B1,P3,P,B2,DW0):=  
Block( DW3:DW3(t), DW3 = B1 \* 'integrate ( (P3 - P - B2\* DW3\*DW3), t) + DW0 );

ORIGINAL PAGE IS  
OF POOR QUALITY

```

(D184)          HPOPW(DW3, B1, P3, P, B2, DW0) := BLOCK(DW3 : DW3(T), DW3 = B1 I (P3 - P - B2 DW3 DW3
) dT + DW0)
                                                    ]
                                                    /

(C185) "*****Main Chamber Injector, Fuel and Hot Gas *****"
(C186) "Main Chamber Injector Total Pressure"$
(C187) MCITP(P, B1, DW1, DW2, DW3, B2, DW4, P0) :=
Block(DW1:DW1(t), DW2:DW2(t), DW3:DW3(t), DW4:DW4(t), P = B1 * 'integrate((DW1 + DW2 + DW3 - B2* DW4), t)
+ P0);
(D187) MCITP(P, B1, DW1, DW2, DW3, B2, DW4, P0) := BLOCK(DW1 : DW1(T), DW2 : DW2(T), DW3 : DW3(T), DW4 :
DW4(T),
                                                    /
                                                    [
                                                    P = B1 I (DW
                                                    ]
                                                    /

1 + DW2 + DW3 - B2 DW4) dT + P0)

(C188) "Main Chamber Injector Weight Flowrate"$
(C189) MCIW(DW, B, P1, T, PC, P2, TT) :=
Block(DW = B * P1 * ( T * PC/P2)/TT^0.5);
                                                    B P1 (T PC)
                                                    -----
                                                    P2
(D189)          MCIW(DW, B, P1, T, PC, P2, TT) := BLOCK(DW = -----)
                                                    0.5
                                                    TT

(C190) "Main Chamber Injector Temperature"$
(C191) MCIT(TFI, B1, TFP, B2, TOP, B3, T5) :=
Block(TFI = B1 * TFP + B2 * TOP + B3 * T5);
(D191)          MCIT(TFI, B1, TFP, B2, TOP, B3, T5) := BLOCK(TFI = B1 TFP + B2 TOP + B3 T5
)

(C192) "*****Main Chamber Injector, Oxidizer*****"
(C193) "Main Chamber Injector, Oxidizer Weight Flowrate"$
(C194) MCWF(DWoi, DW, E) :=
Block(DWoi = DW * E);
(D194)          MCWF(DWOI, DW, E) := BLOCK(DWOI = DW E)

(C195) "Main Chamber Irjector, Oxidizer Weight"$
(C196) MCW(W, DW1, DW2, W0) :=
Block(DW1:DW1(t), DW2:DW2(t), W = 'integrate((DW1 - DW2), t) + W0);
                                                    /
                                                    [
(D196)          MCW(W, DW1, DW2, W0) := BLOCK(DW1 : DW1(T), DW2 : DW2(T), W = I (DW1 - DW2) dT
+ W0)
                                                    ]
                                                    /

(C197) "Main Chamber injector, Oxidizer Variable in injector Priming function"$
(C198) MCV(E, T, W) :=
Block(E = T * W);
(D198)          MCV(E, T, W) := BLOCK(E = T W)

(C199) "*****Min Thrust Chamber*****"
(C200) "Combuster Injector end static Total Pressure"$
(C201) CITP(P, B, P) :=
Block(P = B * P);
(D201)          CITP(P, B, P) := BLOCK(P = B P)

(C202) "Main thrust Chamber Total Pressure"$
(C203) MTCTP(P, B1, B2, DW1, DW2, DW3, P0) :=
Block(DW1:DW1(t), DW2:DW2(t), DW3:DW3(t), P = B1 * 'integrate((B2*DW1 + DW2 - DW3), t) + P0);

```

```

(D203) MTCTP(P, B1, B2, DW1, DW2, DW3, P0) := BLOCK(DW1 : DW1(T), DW2 : DW2(T), DW3 : DW3(T), P = B1 I (B
2 DW1 + DW2 - DW3) dT + P0)
]
/

(C204) "*****Nozzle Bypass Element*****"
(C205) "Nozzle Heat exchanger bypass Weight Flowrate"$
(C206) NHWF(DW, B1, P, P9, B2, DW1, A, DW0) :=
Block(DW1:DW1(t), DW = B1 * 'integrate((P - P9 - B2 *(DW1/(A)^2)),t) + DW0);
/
[
B2 DW
1
(D206) NHWF(DW, B1, P, P9, B2, DW1, A, DW0) := BLOCK(DW1 : DW1(T), DW = B1 I (P - P9 - -----
-) dT + DW0)
]
/
2
A

(C207) "Coolant Control Valve"$
(C208) CCV(DW , DW1) :=
Block(DW = DW1);
(D208) CCV(DW, DW1) := BLOCK(DW = DW1)
(D209) DONE
(C210) CLOSEFILE();

```

ORIGINAL PAGE IS  
OF POOR QUALITY

## Information Base

ORIGINAL PAGE IS  
OF POOR QUALITY

```
(Deframe Fop3
  (Class BOOST-STAGE)
  (Unit "Flow-Variable")
  (Nomenclature "High Pressure Oxidizer boostpump Flow variable")
  (Input-parameter B43 DWop3 So2))
(Deframe Top3
  (Class BOOST-STAGE)
  (Unit "Torque")
  (Nomenclature "High Pressure Oxidizer Boostpump Torque")
  (Input-parameter B44 So2 Trop3 Fop3))
(Deframe Pod3
  (Class BOOST-STAGE)
  (Unit "Pressure")
  (Nomenclature "High Pressure Oxidizer Boostpump Pressure")
  (Input-parameter B45 So2 Tpop3 Fop3 Pod2))
(Deframe DWop3
  (Class BOOST-STAGE)
  (Unit "Weight-Flowrate")
  (Nomenclature "High Pressure Oxidizer Boostpump Weight Flowrate")
  (Input-parameter B46 Pod3 Ppos B104 DWop30))
(Deframe Trop3
  (Class BOOST-STAGE)
  (Unit "Table-Function")
  (Nomenclature "High Pressure Oxidizer Boostpump Torque")
  (Input-parameter ))
(Deframe Tpop3
  (Class BOOST-STAGE)
  (Unit "Table-Function")
  (Nomenclature "high Pressure Oxidizer boostpump Pressure Rise Character")
  (Input-parameter ))
(Deframe DWop30
  (Class BOOST-STAGE)
  (Unit "Initial-Condition")
  (Nomenclature "DWop3 when t=0")
  (Input-parameter 105.43))
```

```

(Deframe Pfp
  (Class COMBUSTER)
  (Unit "Combuster-Total-Pressure")
  (Nomenclature "Fuel Preburner Total Pressure")
  (Input-parameter B56 DWfpf DWfpoi B111 DWft2 Pfp0))
(Deframe Pop
  (Class COMBUSTER)
  (Unit "Combuster-Total-Pressure")
  (Nomenclature "Oxidizer Preburner Total Pressure")
  (Input-parameter b63 DWopf DWopoi B107 DWot2 Pop0))
(Deframe Ffp
  (Class COMBUSTER)
  (Unit "Weight-Flowrate-Fraction")
  (Nomenclature "Fuel Preburner Flowrate Fraction")
  (Input-parameter DWfpoi DWfpoi+DWfpf))
(Deframe Fop
  (Class COMBUSTER)
  (Unit "Weight-Flowrate-Fraction")
  (Nomenclature "Oxidizer Preburner Flowrate Fraction")
  (Input-parameter DWopoi DWopoi+DWopf))
(Deframe Tfp
  (Class COMBUSTER)
  (Unit "Combuster-Temperature")
  (Nomenclature "Fuel Preburner Combuster Temperature")
  (Input-parameter TTfpf B112 T9))
(Deframe Top
  (Class COMBUSTER)
  (Unit "Combuster-Temperature")
  (Nomenclature "Oxidizer Preburner Combuster Temperature")
  (Input-parameter TTfop B113 T9))
(Deframe Pfp0
  (Class COMBUSTER)
  (Unit "Initial-Condition")
  (Nomenclature "Pfp when t=0")
  (Input-parameter 5940.0))
(Deframe Pop0
  (Class COMBUSTER)
  (Unit "Initial-Condition")
  (Nomenclature "Pop when t=0")
  (Input-parameter 5929.9))
(Deframe TTfpf
  (Class COMBUSTER)
  (Unit "Table-Function")
  (Nomenclature "Fuel Preburner Hot Gass Temperature Function")
  (Input-parameter ))
(Deframe TTfop
  (Class COMBUSTER)
  (Unit "Table-Function")
  (Nomenclature "Oxidizer Preburner Hot Gas Temperature Function")
  (Input-parameter ))

```

**ORIGINAL PAGE IS  
OF POOR QUALITY**



```

(Deframe P5
  (Class COOLING)
  (Unit "Total-Pressure")
  (Nomenclature "Main Chamber Heat Exchanger Total Pressure")
  (Input-parameter 1 DQw15 DQw25 H3 DWmc H5 DW5 P50))
(Deframe P4
  (Class COOLING)
  (Unit "Total-Pressure")
  (Nomenclature "Nozzle Regenerative Exchanger Element Total Pressure")
  (Input-parameter b84 DQw14 DQw24 H4 DW4 H3 DWfn P40))
(Deframe H5
  (Class COOLING)
  (Unit "Specific-Enthalpy")
  (Nomenclature "Main Chamber Heat Exchanger Specific Enthalpy")
  (Input-parameter b101 T5))
(Deframe H4
  (Class COOLING)
  (Unit "Specific-Enthalpy")
  (Nomenclature "Nozzle Regenerative Cooling Element Specific Enthalpy")
  (Input-parameter B102 T4))
(Deframe DW5
  (Class COOLING)
  (Unit "Weight-Flowrate")
  (Nomenclature "Main Chamber Heat Weight Flowrate")
  (Input-parameter B75 Pmfvd P5 B76 R5 DWmc DWmc0))
(Deframe DW4
  (Class COOLING)
  (Unit "Weight-Flowrate")
  (Nomenclature "Nozzle Regenerative Cooling Weight Flowrate")
  (Input-parameter B94 P4 P9 B95 R4 DW4 DW40))
(Deframe R5
  (Class COOLING)
  (Unit "Density")
  (Nomenclature "Main Chamber Element Density")
  (Input-parameter B77 DWmc DW5 R5))
(Deframe R4
  (Class COOLING)
  (Unit "Density")
  (Nomenclature "Nozzle Regenerative Element Density")
  (Input-parameter B87 DWfn DW4 R4))
(Deframe T5
  (Class COOLING)
  (Unit "Temperature")
  (Nomenclature "Main Chamber Heat Exchanger Temperature")
  (Input-parameter B78 P5 R5))
(Deframe T4
  (Class COOLING)
  (Unit "Temperature")
  (Nomenclature "Nozzle Regenerative Cooling Temperature")
  (Input-parameter B88 P4 R4))
(Deframe DQw15
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Main Chamber Hot Gas Heat Transfer Rate")
  (Input-parameter B79 T5 TW15 DWmc))
(Deframe DQw25
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Main Chamber Ambient Thrust Chamber Heat Transfer Rate")
  (Input-parameter B80 T5 TW25 DWmc))
(Deframe DQw14
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Nozzle Regenerative Hot Gas Heat Transfer Rate")
  (Input-parameter B89 T4 TW14 DWfn))
(Deframe DQw24
  (Class COOLING)
  (Unit "Heat-Transfer")
  (Nomenclature "Nozzle Regenrative Ambient Thrust Heat Transfer Rate")
  (Input-parameter B90 T4 TW24 DWfn))
(Deframe DQtc5
  (Class COOLING)
  (Unit "Heat-Transfer-TC")
  (Nomenclature "Main Chamber Cooling Element Heat Transfer Rate TC")
  (Input-parameter B81 Tc Tw15 DWcn))
(Deframe DQtc4
  (Class COOLING)
  (Unit "Heat-Transfer-TC")
  (Nomenclature "Nozzle Regenerative Heat Transfer Rate TC")

```

```

      (Input-parameter B91 Tc Tw14 DWcn))
(Deframe Tw15
  (Class COOLING)
  (Unit "Hot-Gas-Temperature")
  (Nomenclature "Hot Gas Temperature")
  (Input-parameter B82 DQtc5 DQw15 Tw150))
(Deframe Tw25
  (Class COOLING)
  (Unit "Ambient-Thrust-Temperature")
  (Nomenclature "Ambient thrust chamber Temperature")
  (Input-parameter B83 DQw25 Tw250))
(Deframe Tw14
  (Class COOLING)
  (Unit "Hot-Gas-Temperature")
  (Nomenclature "Hot Gas temperature")
  (Input-parameter B92 DQtc4 DQw14 Tw140))
(Deframe Tw24
  (Class COOLING)
  (Unit "Ambient-Thrust-Temperature")
  (Nomenclature "Ambient Thrust Temperature ")
  (Input-parameter B93 DQw24 Tw240))
(Deframe P50
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "P5 when t=0")
  (Input-parameter 5143.0))
(Deframe P40
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "P4 when t=0")
  (Input-parameter 6794.0))
(Deframe DWmc0
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "DWmc when t=0")
  (Input-parameter 35.24))
(Deframe DW40
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "DW4 when t=0")
  (Input-parameter 41.98))
(Deframe R50
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "R5 when t=0")
  (Input-parameter 0.001402))
(Deframe R40
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "R4 when t=0")
  (Input-parameter 6650.0))
(Deframe Tw150
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "TW15 when t=0")
  (Input-parameter 1330.0))
(Deframe Tw250
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "Tw25 when t=0")
  (Input-parameter 544.5))
(Deframe Tw140
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "Tw14 when t=0")
  (Input-parameter 1260.0))
(Deframe Tw240
  (Class COOLING)
  (Unit "Initial-Condition")
  (Nomenclature "Tw24 when t=0")
  (Input-parameter 679.3))

```

**ORIGINAL PAGE IS  
OF POOR QUALITY**

```

(Deframe DWfpf
  (Class FLOW)
  (Unit "Fuel-Flow")
  (Nomenclature "Fuel Preburner Fuel Flow")
  (Input-parameter B50 P9 Pfp B51 R9 DWfpf0))
(Deframe DWopf
  (Class FLOW)
  (Unit "Fuel-Flow")
  (Nomenclature "Oxidizer Preburner Fuel Flow")
  (Input-parameter B57 P9 Pop B58 R9 DWopf0))
(Deframe DWfpo
  (Class FLOW)
  (Unit "Oxidizer-Flow")
  (Nomenclature "Fuel Preburner Oxidizer Weight Flowrate")
  (Input-parameter B52 Ppos Pfp B53 A Ab B54 DWfpoi DWfpo0))
(Deframe DWopo
  (Class FLOW)
  (Unit "Oxidizer-Flow")
  (Nomenclature "Oxidizer Preburner Oxidizer Weight Flowrate")
  (Input-parameter B59 Ppos Pop B61 A Ab B60 DWopoi DWopo0))
(Deframe Efpo
  (Class FLOW)
  (Unit "Variable-in-injector-Priming")
  (Nomenclature "Fuel Preburner Oxidizer Variable in injector Priming")
  (Input-parameter B55 DWfpo DWfpoi Efpo0))
(Deframe Eopo
  (Class FLOW)
  (Unit "Variable-in-injector-Priming")
  (Nomenclature "Oxidizer Preburner Oxidizer Variable in injector Priming")
  (Input-parameter B62 DWopo DWopoi Eopo0))
(Deframe DWfpoi
  (Class FLOW)
  (Unit "Weight-Flowrate-Injector")
  (Nomenclature "Fuel Preburner Oxidizer Injector Weight Flowrate")
  (Input-parameter DWfpo Efpo
))
(Deframe DWopoi
  (Class FLOW)
  (Unit "Weight-Flowrate-Injector")
  (Nomenclature "Oxidizer Preburner Oxidizer Injector Weight Flowrate")
  (Input-parameter DWopo Eopo))
(Deframe DWfpf0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWfpf when t=0")
  (Input-parameter 83.15))
(Deframe DWopf0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWopf when t=0")
  (Input-parameter 38.93))
(Deframe DWfpo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWfpo when t=0")
  (Input-parameter 79.30))
(Deframe DWopo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "DWopo when t=0")
  (Input-parameter 26.13))
(Deframe Efpo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "EFpo when t=0")
  (Input-parameter 1.0))
(Deframe Eopo0
  (Class PUMP)
  (Unit "Initial-Condition")
  (Nomenclature "Eopo when t=0")
  (Input-parameter 1.0))

```

```
(Deframe QF
  (Class FLOWMETER)
  (Unit "Volumetric-Flowrate")
  (Nomenclature "Fuel Volumetric Flowrate")
  (Input-parameter 0.008783 Tfs 37.0 -204.852 101.323 DWfd2 0.20206 Sf1))

(Deframe Pffm
  (Class FLOWMETER)
  (Unit "Total-Pressure")
  (Nomenclature "Fuel Flowmeter Total Pressure")
  (Input-parameter Pfd1 RFfcd DWfd2 Rfd1))

(Deframe Tfpl
  (Class FLOWMETER)
  (Unit "Flowmeter-Temperature")
  (Nomenclature "Low Pressure Fuel Pump Flowmeter Temperature")
  (Input-parameter Tfs 0.0015789 Sf1))

(Deframe Tod2
  (Class FLOWMETER)
  (Unit "Flowmeter-Temperature")
  (Nomenclature "Low pressure Oxidizer pump Flowmeter Temperature")
  (Input-parameter Tos 0.00547 So2))
```

**ORIGINAL PAGE IS  
OF POOR QUALITY**

```
(Deframe Pfi
  (Class MAIN-CHAMBER)
  (Unit "Pressure")
  (Nomenclature "Main Chamber Injector, fuel and Hot Gas Total Pressure")
  (Input-parameter B64 DWft1 DWot2 DWft2 B108 DWfi Pfi0))
(Deframe DWfi
  (Class MAIN-CHAMBER)
  (Unit "Weight-Flowrate")
  (Nomenclature "Main Chamber injector weight flowrate")
  (Input-parameter B64 Pfi Tpr Pc Ppi Tfi))
(Deframe Tfi
  (Class MAIN-CHAMBER)
  (Unit "Temperature")
  (Nomenclature "Main Chamber Temperature")
  (Input-parameter B66 Tfp B67 Top B68 T5))
(Deframe DWoi
  (Class MAIN-CHAMBER)
  (Unit "Oxidizer-Weight-Flowrate")
  (Nomenclature "Main Chamber injector, Oxidizer Weight Flowrate")
  (Input-parameter DWmov Eoi))
(Deframe Woi
  (Class MAIN-CHAMBER)
  (Unit "Oxidizer-Weight")
  (Nomenclature "Main Chamber Injector, Oxidizer Weight")
  (Input-parameter DWmov DWoi Woi))
(Deframe Eoi
  (Class MAIN-CHAMBER)
  (Unit "Oxidizer-Variable")
  (Nomenclature "Main Chamber injector, Oxidizer variable in injector priming function")
  (Input-parameter TEoi Woi))
(Deframe Pfi0
  (Class MAIN-CHAMBER)
  (Unit "Initial-Condition")
  (Nomenclature "Pfi when t=0")
  (Input-parameter 3662.0))
```

```
(Deframe Pcies
  (Class MAIN-THRUST-CHAMBER)
  (Unit "Combuster-Total-Pressure")
  (Nomenclature "Combuster Injector end static Total Pressure")
  (Input-parameter B114 Pc))
(Deframe Pc
  (Class MAIN-THRUST-CHAMBER)
  (Unit "Main-thrust-Total-Pressure")
  (Nomenclature "Main thrust chamber Total Pressure")
  (Input-parameter B72 B109 DWfi DWoi DWcn Pc0))
(Deframe Pc0
  (Class MAIN-THRUST-CHAMBER)
  (Unit "Initial-Condition")
  (Nomenclature "Pc when t=0")
  (Input-parameter 3330.0))
```

**ORIGINAL PAGE IS  
OF POOR QUALITY**

```
(Deframe DWfnbp
  (Class NOZZLE-BYPASS)
  (Unit "Nozzle-Heat-Weight-Flowrate")
  (Nomenclature "Nozzle Heat Exchanger bypass weight flowrate")
  (Input-parameter B98 Pmfvd P9 B99 DWfnbp Accv DWfnbp0))
(Deframe DWccv
  (Class NOZZLE-BYPASS)
  (Unit "Coolant-Control-Weight-Flowrate")
  (Nomenclature "Coolant Control Valve")
  (Input-parameter DWfnbp))
(Deframe DWfnbp0
  (Class NOZZLE-BYPASS)
  (Unit "Initial-Condition")
  (Nomenclature "DWfnbp when t=0")
  (Input-parameter 80.19))
```

```

(Deframe P9
  (Class PREBURNER-SUPPLY-DUCT)
  (Unit "Supply-Duct-Total-Pressure")
  (Nomenclature "Preburner Fuel Supply Element Total Pressure")
  (Input-parameter B96 DW4 DWopf DWfpf DWfnbp P90))
(Deframe Ppos
  (Class PREBURNER-SUPPLY-DUCT)
  (Unit "Supply-Duct-Total-Pressure")
  (Nomenclature "Preburner Oxidizer Supply Total Pressure")
  (Input-parameter B49 DWop3 DWfpo Dwopo DW Ppos0))
(Deframe T9
  (Class PREBURNER-SUPPLY-DUCT)
  (Unit "Supply-Duct-Temperature")
  (Nomenclature "Preburner Fuel Supply Element Temperature")
  (Input-parameter B100 P9 R9))
(Deframe R9
  (Class PREBURNER-SUPPLY-DUCT)
  (Unit "Supply-Duct-Density")
  (Nomenclature "Preburner Fuel Supply Element Density")
  (Input-parameter DWfnbp DW4 R4 B105))
(Deframe P90
  (Class PREBURNER-SUPPLY-DUCT)
  (Unit "Initial-Condition")
  (Nomenclature "P9 when t=0")
  (Input-parameter 6650.0))
(Deframe Ppos0
  (Class PREBURNER-SUPPLY-DUCT)
  (Unit "Initial-Condition")
  (Nomenclature "Ppos when t=0")
  (Input-parameter 8457.0))

```

**ORIGINAL PAGE IS  
OF POOR QUALITY**



ORIGINAL PAGE IS  
OF POOR QUALITY

```
(Deframe Ffp1
  (Class PUMP)
  (Unit "Flow-Variable")
  (Nomenclature "Low Pressure Fuel Pump Flow variable")
  (Input-parameter B11 DWfd1 Sf1))

(Deframe Ffp2
  (Class PUMP)
  (Unit "Flow-Variable")
  (Nomenclature "High Pressure Fuel Pump Flow Variable")
  (Input-parameter B18 DWfd2 Sf2))

(Deframe Fdp1
  (Class PUMP)
  (Unit "Flow-Variable")
  (Nomenclature "Low Pressure Oxidizer Pump Flow variable")
  (Input-parameter B27 DWmov+DWop3 Sol))

(Deframe Fop2
  (Class PUMP)
  (Unit "Flow-Variable")
  (Nomenclature "High Pressure Oxidizer Pump Flow variable")
  (Input-parameter B39 DWmov+DWot1+DWop3 So2))

(Deframe Pfd1
  (Class PUMP)
  (Unit "Total-Pressure")
  (Nomenclature "Low Pressure Fuel Pump Discharge Total pressure")
  (Input-parameter Pfs B12 Sf1 Tpfpl))

(Deframe Pfd2
  (Class PUMP)
  (Unit "Total-Pressure")
  (Nomenclature "High Pressure Fuel Pump Total Pressure")
  (Input-parameter Pfd1 B19 Sf2 Tpfp2))

(Deframe Pod1
  (Class PUMP)
  (Unit "Total-Pressure")
  (Nomenclature "Low Pressure Oxidizer Pump Total Pressure")
  (Input-parameter Pos B28 S01 Tpop1))

(Deframe Pod2
  (Class PUMP)
  (Unit "Total-Pressure")
  (Nomenclature "High Pressure Oxidizer Pump Total Pressure")
  (Input-parameter Pod1 B40 So2 Tpop2))

(Deframe Rfp1
  (Class PUMP)
  (Unit "Torque")
  (Nomenclature "Low Pressure Fuel Pump "Torque"")
  (Input-parameter B13 Sf1 Trfp1))

(Deframe Rfp2
  (Class PUMP)
  (Unit "Torque")
  (Nomenclature "High Pressure Fuel Pump "Torque"")
  (Input-parameter B20 Sf2 Trfp2))

(Deframe Rop1
  (Class PUMP)
  (Unit "Torque")
  (Nomenclature "Low Pressure Oxidizer Pump "Torque"")
  (Input-parameter B34 Sol Tropl))

(Deframe Tpfpl
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "Low Pressure Fuel Pump Pump Pressure Rise Characteristics")
  (Input-parameter ))

(Deframe Tpfp2
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "High Pressure Fuel Pump Pump Pressure Rise Characteristics")
  (Input-parameter ))

(Deframe Tpop1
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "Low Pressure Oxidizer Pump")
  (Input-parameter ))

(Deframe Tpop2
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "High Pressure Oxidizer Pump Pump Pressure Rise Characteristics"))
```

```

      (Input-parameter ))
(Deframe Trfp1
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "Low Pressure Fuel Pump Pump Torque Characteristics")
  (Input-parameter ))
(Deframe Trfp2
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "High Pressure Fuel Pump Pump Pump Torque Characteristics")
  (Input-parameter ))
(Deframe Trop1
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "Low Pressure Oxidizer Pump Pump Torque Characteristics")
  (Input-parameter ))
(Deframe Torque
  (Class PUMP)
  (Unit "Torque")
  (Nomenclature "High Pressure Oxidizer Pump \"Torque\"")
  (Input-parameter B41 So2 Trop2))
(Deframe Trop2
  (Class PUMP)
  (Unit "Table-Function")
  (Nomenclature "High Pressure Oxidizer Pump Pump Torque Characteristics")
  (Input-parameter ))

```

**ORIGINAL PAGE IS  
OF POOR QUALITY**

```

(Deframe TQft1
  (Class TURBINE)
  (Unit "Torque")
  (Nomenclature "Low Pressure Fuel Turbine Torque")
  (Input-parameter B16 P5 Trft1))
(Deframe TQft2
  (Class TURBINE)
  (Unit "Torque")
  (Nomenclature "High Pressure Fuel Turbine Torque")
  (Input-parameter B26 Pfp Trft2))
(Deframe TQot1
  (Class TURBINE)
  (Unit "Torque")
  (Nomenclature "Low Pressure Oxidizer Turbine Torque")
  (Input-parameter B36 DWot1^2 Trot1))
(Deframe TQot2
  (Class TURBINE)
  (Unit "Torque")
  (Nomenclature "High Pressure Oxidizer Turbine")
  (Input-parameter B29 Pop Trot2))
(Deframe Uft1
  (Class TURBINE)
  (Unit "Speed-Parameter")
  (Nomenclature "Low Pressure Fuel Turbine Speed Parameter")
  (Input-parameter B15 Sf1 T5))
(Deframe Uft2
  (Class TURBINE)
  (Unit "Speed-Parameter")
  (Nomenclature "High Pressure Fuel Turbine Speed Parameter")
  (Input-parameter B25 Sf2 Tfp))
(Deframe Fot1
  (Class TURBINE)
  (Unit "Speed-Parameter")
  (Nomenclature "Low Pressure Oxidizer Turbine Flow Variable")
  (Input-parameter B38 Sol DWot1))
(Deframe Uot2
  (Class TURBINE)
  (Unit "Speed-Parameter")
  (Nomenclature "High Pressure Oxidizer Turbine Speed Parameter")
  (Input-parameter B47 So2 Rop*Top))
(Deframe DWft1
  (Class TURBINE)
  (Unit "Weight-Flowrate")
  (Nomenclature "Low Pressure Fuel Turbine Weight Flowrate")
  (Input-parameter B17 P5-Pfi R5))
(Deframe DWft2
  (Class TURBINE)
  (Unit "Weight-Flowrate")
  (Nomenclature "High Pressure Fuel Turbine Weight Flowrate")
  (Input-parameter B24 Pfp Tpr))
(Deframe DWot1
  (Class TURBINE)
  (Unit "Weight-Flowrate")
  (Nomenclature "Low Pressure Oxidizer Turbine Weight Flowrate")
  (Input-parameter (1/(B37 + Rot1)) Pod2-Pod1 1))
(Deframe DWot2
  (Class TURBINE)
  (Unit "Weight-Flowrate")
  (Nomenclature "High Pressure Oxidizer Turbine Weight Flowrate")
  (Input-parameter B48 Pop Tpr/Top))
(Deframe Trft1
  (Class TURBINE)
  (Unit "Table-Function")
  (Nomenclature "Low Pressure Fuel Turbine Turbine Torque Characteristics")
  (Input-parameter ))
(Deframe Trft2
  (Class TURBINE)
  (Unit "Table-Function")
  (Nomenclature "High Pressure Fuel Turbine Turbine Torque Characteristics")
  (Input-parameter ))
(Deframe Trot1
  (Class TURBINE)
  (Unit "Table-Function")
  (Nomenclature "High Pressure Oxidizer Turbine Turbine Torque Characteristics")
  (Input-parameter ))
(Deframe Trot2
  (Class TURBINE)

```

```
(Unit "Table-Function")
(Nomenclature "Low Pressure Oxidizer Turbine Turbine Torque Characteristics")
(Input-parameter )
(Deframe Tpr
(Class TURBINE)
(Unit "Table-Function")
(Nomenclature "Pressure Ratio")
(Input-parameter ))
```

**ORIGINAL PAGE IS  
OF POOR QUALITY**

```
(Deframe Sf1
  (Class TURBOPUMP)
  (Unit "Rotational-Speed")
  (Nomenclature "Low Pressure Fuel Turbopump Rotational Speed")
  (Input-parameter B14 TQft1-TQfp1 Sf10))
(Deframe Sf2
  (Class TURBOPUMP)
  (Unit "Rotational-Speed")
  (Nomenclature "High Pressure Fuel Turbopump Rotational Speed")
  (Input-parameter B23 TQft2-TQfp2 Sf20))
(Deframe Sol
  (Class TURBOPUMP)
  (Unit "Rotational-Speed")
  (Nomenclature "Low Pressure Oxidizer Turbopump Rotational Speed")
  (Input-parameter B35 TQot1-TQop1 Sol0))
(Deframe So2
  (Class TURBOPUMP)
  (Unit "Rotational-Speed")
  (Nomenclature "High Pressure Oxidizer Turbopump Rotational Speed")
  (Input-parameter B42 TQot2-TQop2-TQop3 So20))
(Deframe Sf10
  (Class TURBOPUMP)
  (Unit "Initial-Condition")
  (Nomenclature "Sf1 when t=0")
  (Input-parameter 1638.0))
(Deframe Sf20
  (Class TURBOPUMP)
  (Unit "Initial-Condition")
  (Nomenclature "Sf2 when t=0")
  (Input-parameter 3922.0))
(Deframe Sol0
  (Class TURBOPUMP)
  (Unit "Initial-Condition")
  (Nomenclature "Sol when t=0")
  (Input-parameter 570.9))
(Deframe So20
  (Class TURBOPUMP)
  (Unit "Initial-Condition")
  (Nomenclature "So2 when t=0")
  (Input-parameter 3262.0))
```

```
(Deframe Amfv
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Main Fuel Valve Area")
  (Input-parameter TQmfv Xmfv))
(Deframe Amov
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Main Oxidizer Valve Area")
  (Input-parameter TQmov Xmov))
(Deframe Afpv
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Fuel Preburner Oxidizer Valve Area")
  (Input-parameter TQfpv Xfpv))
(Deframe Aopv
  (Class VALVE)
  (Unit "Area")
  (Nomenclature "Oxidizer Preburner Oxidizer Valve Area")
  (Input-parameter TQopv Xopv))
```

**ORIGINAL PAGE IS  
OF POOR QUALITY**

**System Generated Equations**

Component Processor Element Pump

$$SO2 = B42 T (TQOT2 - TQOP3 - TQOP2) + SO20$$

$$SO1 = B35 T (TQOT1 - TQOP1) + SO10$$

$$SF2 = B23 T (TQFT2 - TQFP2) + SF20$$

$$SF1 = B14 T (TQFT1 - TQFP1) + SF10$$

Component Processor Element Turbine

$$DWOT2 = \text{SQRT} \left( \frac{B48 \text{ POP TPR}}{\text{TOP}} \right)$$

$$DWOT1 = \text{SQRT} \left( \frac{\text{POD2} - \text{POD1}}{\text{ROT1} + B37} \right)$$

$$DWFT2 = \text{SQRT}(B24 \text{ PFP TPR})$$

$$DWFT1 = \text{SQRT}(B17 (P5 - \text{PFI}) R5)$$

$$UOT2 = \frac{B47 \text{ SO2}}{\text{SQRT}(\text{ROP TOP})}$$

$$FOT1 = \frac{B38 \text{ SO1}}{\text{SQRT}(DWOT1)}$$

$$UFT2 = \frac{B25 \text{ SF2}}{\text{SQRT}(\text{TFP})}$$

$$UFT1 = \frac{B15 \text{ SF1}}{\text{SQRT}(T5)}$$

$$TQOT2 = B29 \text{ POP TROT2}$$

$$TQOT1 = B36 \text{ DWOT1}^2 \text{ TROT1}$$

$$TQFT2 = B26 \text{ PFP TRFT2}$$

$$TQFT1 = B16 \text{ P5 TRFT1}$$

Component Processor Element Turbopump

$$SO2(T) = B42 \frac{\int [ 1 - dT ] (TROT2 - TROP3 - TROP2) + SOO2(T)}{G(T)}$$

$$SO1(T) = B35 \frac{\int [ 1 - dT ] (TROT1 - TROP1) + SOO1(T)}{G(T)}$$

$$SF2(T) = B23 \frac{\int [ 1 - dT ] (TRFT2 - TRFP2) + SOF2(T)}{G(T)}$$

$$SF1(T) = B14 \frac{\int [ 1 - dT ] (TRFT1 - TRFP1) + SOF1(T)}{G(T)}$$

ORIGINAL PAGE IS  
OF POOR QUALITY



ORIGINAL PAGE IS  
OF POOR QUALITY

Component Processor Element Valve

AOPV = TQOPV XOPV

AFPV = TQFPV XFPV

AMOV = TQMOV XMOV

AMFV = TQMFV XMFV

Component Processor Element Flow

DWOPOI = DWOPO EOPO

DWFPOI = DWFPO EFPO

$$\begin{aligned}
 & / \\
 & [ \\
 \text{OO (T)} \quad & \text{DWOPO (T) = B59 I (- B60 DWOPOI (T) - } \frac{\text{AB B61 DWOPO (T)}^2}{\text{A}} + \text{PPOS - POP) dT + DWOP} \\
 & ] \\
 & /
 \end{aligned}$$

$$\begin{aligned}
 & / \\
 & [ \\
 \text{OO (T)} \quad & \text{DWFPO (T) = B52 I (- B54 DWFPOI (T) - } \frac{\text{AB B53 DWFPO (T)}^2}{\text{A}} + \text{PPOS - PFP) dT + DWFP} \\
 & ] \\
 & /
 \end{aligned}$$

$$\begin{aligned}
 & / \\
 & [ \\
 \text{DWOPI (T) = B} & \left( \frac{\text{B57}}{\text{R9}} - \text{POP} + \text{P9} \right) \text{I DWOPF (T) dT + DWOO (T)} \\
 & ] \\
 & /
 \end{aligned}$$

$$\begin{aligned}
 & / \\
 & [ \\
 \text{DWFPI (T) = B} & \left( \frac{\text{B50}}{\text{R9}} - \text{PFP} + \text{P9} \right) \text{I DWFPF (T) dT + DWF0 (T)} \\
 & ] \\
 & /
 \end{aligned}$$

Component Processor Element Combuster

TOP = TTFOP + B113 T9

TFP = TTFPP + B112 T9

$$\text{FOP} = \frac{\text{DWOPOI}}{\text{DWOPOI} + \text{DW1}}$$

$$\text{FFP} = \frac{\text{DWFPOI}}{\text{DWFPOI} + \text{DW1}}$$

$$\begin{aligned}
 & / \\
 & [ \\
 \text{POP} = \text{B63 I} & \left( - \text{B107 DWOT2 (T) + DWOPOI (T) + DWOPF (T) \right) \text{dT + POPO} \\
 & ] \\
 & /
 \end{aligned}$$

$$\begin{aligned}
 & / \\
 & [ \\
 \text{PFP} = \text{B56 I} & \left( - \text{B111 DWFT2 (T) + DWFPOI (T) + DWFPF (T) \right) \text{dT + PFPO} \\
 & ] \\
 & /
 \end{aligned}$$

Component Processor Element Cooling

$$\begin{aligned}
 & / \\
 & [ \\
 \text{TW24} = \text{TW240} & - \text{B93 I DQW24 (T) dT} \\
 & ] \\
 & /
 \end{aligned}$$

ORIGINAL PAGE IS  
OF POOR QUALITY

$$TW14 = TW140 + B92 \int \left[ (DQTC4(T) - DQW14(T)) dT \right]$$

$$TW25 = TW250 - B83 \int DQW25(T) dT$$

$$TW15 = TW150 + B82 \int \left[ (DQTC5(T) - DQW15(T)) dT \right]$$

$$DQTC4 = B91 \text{ DWCN}^{0.8} (TC - TW14)$$

$$DQTC5 = B81 \text{ DWCN}^{0.8} (TC - TW15)$$

$$DQW24 = B90 \text{ DWFN}^{0.8} (0.002 T4 + 1.0) (TW24 - T4)$$

$$DQW14 = B89 \text{ DWFN}^{0.8} (0.002 T4 + 1.0) (TW14 - T4)$$

$$DQW25 = B80 \text{ DWMC}^{0.8} (0.002 T5 + 1.0) (TW25 - T5)$$

$$DQW15 = B79 \text{ DWMC}^{0.8} (0.002 T5 + 1.0) (TW15 - T5)$$

$$T4 = \frac{B88 P4}{R4}$$

$$T5 = \frac{B78 P5}{R5}$$

$$R4 = B87 \int \left[ (DWFN(T) - DW4(T)) dT + R40 \right]$$

$$R5 = B77 \int \left[ (DWMC(T) - DW5(T)) dT + R50 \right]$$

$$DW4(T) = B94 \int \left[ \left( - \frac{B95 DW4(T)^2}{R4} - P9 + P4 \right) dT + DW40 \right]$$

$$DWMC(T) = B75 \int \left[ \left( - \frac{B76 DWMC(T)^2}{R5} + PMFVD - P5 \right) dT + DWMC0 \right]$$

$$H4 = B102 T4$$

$$H5 = B101 T5$$

$$P4 = B84 \int \left[ (- H3 DWFN(T) + H4 DW4(T) + DQW24(T) + DQW14(T)) dT + P40 \right]$$

ORIGINAL PAGE IS  
OF POOR QUALITY

$$P5 = \int (H3 \text{ DWMC}(T) - H5 \text{ DW5}(T) + \text{DQW25}(T) + \text{DQW15}(T)) \text{ dT} + P50$$

Component Processor Element flowmeter

$$\text{TOD2} = \text{TOS} + 0.00547 \text{ SO2}$$

$$\text{TFP1} = \text{TFS} + 0.0015789 \text{ SF1}$$

$$\text{POFM} = \text{POD2} - \frac{(\text{DWOTPR} + \text{DWOP3} + \text{DWMOV})^2 \text{ RFOD2}}{\text{ROD2}}$$

$$\text{PFFM} = \text{PFD1} - \frac{\text{DWFD2}^2 \text{ RFFCD}}{\text{RFD1}}$$

$$\text{QOFM} = [(0.0209912 \text{ SO2} + 6.19891 (\text{DWOTPR} + \text{DWOP3} + \text{DWMOV}) + 17.5339) (1.0 - 0.002789 (\text{TOS} - 164.0))]$$

$$\text{QF} = [(0.20206 \text{ SF1} + 101.323 \text{ DWFD2} - 204.852) (1.0 - 0.008783 (\text{TFS} - 37.0))]$$

Component Processor Element Preburner Supply Duct

$$\text{R9} = \left[ \frac{\text{DWFNBP} + \text{DW4}}{\text{DW4} + \text{B105 DWFNBP}} \right] \text{R4}$$

$$\text{T9} = \frac{\text{B100 P9}}{\text{R9}}$$

$$\text{PPOS} = \int (- \text{DWOP3}(T) + \text{DWOP3}(T) - \text{DWFPO}(T) + \text{DW}(T)) \text{ dT} + \text{PPOS0}$$

$$\text{P9} = \int (- \text{DWOPF}(T) - \text{DWFPP}(T) + 2 \text{ DWFNBP}(T)) \text{ dT} + \text{P90}$$

Component Processor Element High Pressure Oxidizer Pump Preburner Boost-stage

$$\text{DWOP3}(T) = \int (- \text{B2 DWOP3}(T) + \text{P3} - \text{P}) \text{ dT} + \text{DW0}$$

$$\text{POD3} = \text{B F S}^2 \text{ TT} + \text{P2}$$

$$\text{TOP3} = \text{B44 F S}^2 \text{ TT}$$

$$\text{FOP3} = \frac{\text{B DW}}{\text{S}}$$

Component Processor Element Main Chamber Injector, Fuel, Hot Gas and Oxidizer

$$\text{EOI} = \text{TEOI WOI}$$

ORIGINAL PAGE IS  
OF POOR QUALITY

$$\begin{aligned} & / \\ & [ \\ \text{WOI} = & \text{WOI} + \int (\text{DWMOV}(T) - \text{DWOI}(T)) \, dT \\ & ] \\ & / \end{aligned}$$

$$\text{DWOI} = \text{DWMOV} \, \text{EOI}$$

$$\text{TFI} = \text{B67} \, \text{TOP} + \text{B66} \, \text{TFP} + \text{B68} \, \text{T5}$$

$$\text{DWFI} = \frac{\text{B64} \, \text{PC} \, \text{PFI} \, \text{TPR}}{0.5 \, \text{PPI} \, \text{TFI}}$$

$$\begin{aligned} & / \\ & [ \\ \text{PFI} = & \text{B64} \int (\text{DWOT2}(T) + \text{DWFT2}(T) + \text{DWFT1}(T) - \text{B108} \, \text{DWFI}(T)) \, dT + \text{PFI0} \\ & ] \\ & / \end{aligned}$$

Component Processor Element Main Thrust Chamber

$$\begin{aligned} & / \\ & [ \\ \text{PC} = & \text{B72} \int (\text{DWOI}(T) + \text{B109} \, \text{DWFI}(T) - \text{DWCN}(T)) \, dT + \text{PC0} \\ & ] \\ & / \end{aligned}$$

$$\text{PC} = \text{B114} \, \text{PC}$$

Component Processor Element Nozzle Bypass Element

$$\text{DWCCV} = \text{DWFNBP}$$

$$\begin{aligned} & / \\ & [ \\ \text{DWFNBP} = & \text{B98} \int \left( - \frac{\text{B99} \, \text{DWFNBP}(T)}{\text{ACCV}^2} + \text{PMFVD} - \text{P9} \right) \, dT + \text{DWFNBP0} \\ & ] \\ & / \end{aligned}$$

## **System Generated FORTRAN Code**

## Component Processor Element Pump

TORQUE = B41\*SO2\*\*2\*TROP2  
 ROP1 = B34\*SO1\*\*2\*TROP1  
 RFP2 = B20\*SF2\*\*2\*TRFP2  
 RFP1 = B13\*SF1\*\*2\*TRFP1  
 POD2 = B40\*SO2\*\*2\*TPOP2+POD1  
 POD1 = B28\*SO1\*\*2\*TPOP1+POS  
 PFD2 = B19\*SF2\*\*2\*TPFP2+PFD1  
 PFD1 = B12\*SF1\*\*2\*TPFP1+PFS  
 FOP2 = B39\*(DWOT1+DWOP3+DWMOV)/SO2  
 FDP1 = B27\*(DWOP3+DWMOV)/SO1  
 FFP2 = B18\*DWFD2/SF2  
 FFP1 = B11\*DWFD1/SF1

ORIGINAL PAGE IS  
 OF POOR QUALITY

## Component Processor Element Turbine

DWOT2 = SQRT(B48\*POP\*TPR/TOP)  
 DWOT1 = SQRT((POD2-POD1)/(ROT1+B37))  
 DWFT2 = SQRT(B24\*PPF\*TPR)  
 DWFT1 = SQRT(B17\*(P5-PFI)\*R5)  
 UOT2 = B47\*SO2/SQRT(ROP\*TOP)  
 FOT1 = B38\*SO1/SQRT(DWOT1)  
 UFT2 = B25\*SF2/SQRT(TFP)  
 UFT1 = B15\*SF1/SQRT(T5)  
 TQOT2 = B29\*POP\*TROT2  
 TQOT1 = B36\*DWOT1\*\*2\*TROT1  
 TQFT2 = B26\*PPF\*TRFT2  
 TQFT1 = B16\*P5\*TRFT1

## Component Processor Element Turbopump

SO2 = B42\*T\*(TQOT2-TQOP3-TQOP2)+SO20  
 SO1 = B35\*T\*(TQOT1-TQOP1)+SO10  
 SF2 = B23\*T\*(TQFT2-TQFP2)+SF20  
 SF1 = B14\*T\*(TQFT1-TQFP1)+SF10

## Component Processor Element Valve

AOPV = TQOPV\*XOPV  
 AFPV = TQFPV\*XFPV  
 AMOV = TQMOV\*XMOV  
 AMFV = TQMFV\*XMFV

## Component Processor Element Flow

DWOPOI = DWOPO\*EOPO  
 DWFPOI = DWFPO\*EFPO  
 DWOPO(T) = B59\*' INTEGRATE (-B60\*DWOPOI(T)\*\*2-AB\*\*2\*B61\*DWOPO(T)\*\*2/  
 1 A\*\*2+PPOS-POP,T)+DWOPO0(T)  
 DWFPO(T) = B52\*' INTEGRATE (-B54\*DWFPOI(T)\*\*2-AB\*\*2\*B53\*DWFPO(T)\*\*2/  
 1 A\*\*2+PPOS-PPF,T)+DWFPO0(T)  
 DWOPF(T) = B\*(-B57/R9-POP+P9)'\* INTEGRATE (DWOPF(T)\*\*2,T)+DWO0(T)  
 DWFPF(T) = B\*(-B50/R9-PPF+P9)'\* INTEGRATE (DWFPF(T)\*\*2,T)+DWF0(T)

## Component Processor Element Combuster

TOP = TTFOP+B113\*T9  
 TFP = TTPFP+B112\*T9  
 FOP = DWOPOI/(DWOPOI+DW1)  
 FFP = DWFPOI/(DWFPOI+DW1)  
 POP = B63\*' INTEGRATE (-B107\*DWOT2(T)+DWOPOI(T)+DWOPF(T),T)+POPO  
 PFP = B56\*' INTEGRATE (-B111\*DWFT2(T)+DWFPOI(T)+DWFPF(T),T)+PFPO

## Component Processor Element Cooling

TW24 = TW240-B93\*' INTEGRATE (DQW24(T),T)

TW14 = TW140+B92\*' INTEGRATE (DQTC4 (T)-DQW14 (T) , T)  
 TW25 = TW250-B83\*' INTEGRATE (DQW25 (T) , T)  
 TW15 = TW150+B82\*' INTEGRATE (DQTC5 (T)-DQW15 (T) , T)  
 DQTC4 = B91\*DWCN\*\*0.8\*(TC-TW14)  
 DQTC5 = B81\*DWCN\*\*0.8\*(TC-TW15)  
 DQW24 = B90\*DWFN\*\*0.8\*(0.002\*T4+1.0)\*(TW24-T4)  
 DQW14 = B89\*DWFN\*\*0.8\*(0.002\*T4+1.0)\*(TW14-T4)  
 DQW25 = B80\*DWMC\*\*0.8\*(0.002\*T5+1.0)\*(TW25-T5)  
 DQW15 = B79\*DWMC\*\*0.8\*(0.002\*T5+1.0)\*(TW15-T5)  
 T4 = B88\*P4/R4  
 T5 = B78\*P5/R5  
 R4 = B87\*' INTEGRATE (DWFN (T)-DW4 (T) , T)+R40  
 R5 = B77\*' INTEGRATE (DWMC (T)-DW5 (T) , T)+R50  
 DW4 (T) = B94\*' INTEGRATE (-B95\*DW4 (T)\*\*2/R4-P9+P4 , T)+DW40  
 DWMC (T) = B75\*' INTEGRATE (-B76\*DWMC (T)\*\*2/R5+PMFVD-P5 , T)+DWMC0  
 H4 = B102\*T4  
 H5 = B101\*T5  
 P4 = B84\*' INTEGRATE (-H3\*DWFN (T)+H4\*DW4 (T)+DQW24 (T)+DQW14 (T) , T)+P40  
 P5 = ' INTEGRATE (H3\*DWMC (T)-H5\*DW5 (T)+DQW25 (T)+DQW15 (T) , T)+P50

ORIGINAL PAGE IS  
OF POOR QUALITY

Component Processor Element flowmeter

TOD2 = TOS+0.00547\*SO2  
 TFP1 = TFS+0.0015789\*SF1  
 POFM = POD2-(DWOTPR+DWOP3+DWMOV)\*\*2\*RFOD2/ROD2  
 PFFM = PFD1-DWFD2\*\*2\*RFFCD/RFD1  
 QOFM = [(0.0209912\*SO2+6.19891\*(DWOTPR+DWOP3+DWMOV)+17.5339)\*(1.0-  
 1 0.002789\*(TOSS-164.0))]  
 QF = [(0.20206\*SF1+101.323\*DWFD2-204.852)\*(1.0-0.008783\*(TFS-37.0)  
 1 )]

Component Processor Element Preburner Supply Duct

R9 = [(DWFNBP+DW4)/(DW4/R4+B105\*DWFNBP)]  
 T9 = B100\*P9/R9  
 PPOS = B49\*' INTEGRATE (-DWOP3 (T)+DWOP3 (T)-DWFPO (T)+DW (T) , T)+PPOS0  
 P9 = B96\*' INTEGRATE (-DWOPF (T)-DWFPP (T)+2\*DWFNBP (T) , T)+P90

Component Processor Element High Pressure Oxidizer Pump Preburner Boost-stage

DWOP3 (T) = B1\*' INTEGRATE (-B2\*DWOP3 (T)\*\*2+P3-P , T)+DWO  
 POD3 = B\*F\*S\*\*2\*TT+P2  
 TOP3 = B44\*F\*S\*\*2\*TT  
 FOP3 = B\*DW/S

Component Processor Element Main Chamber Injector, Fuel, Hot Gas and Oxidizer

EOI = TEOI\*WOI  
 WOI = WOI+' INTEGRATE (DWMOV (T)-DWOI (T) , T)  
 DWOI = DWMOV\*EOI  
 TFI = B67\*TOP+B66\*TFP+B68\*T5  
 DWFI = B64\*PC\*PFI\*TPR/(PPI\*TFI\*\*0.5)  
 PFI = B64\*' INTEGRATE (DWOT2 (T)+DWFT2 (T)+DWFT1 (T)-B108\*DWFI (T) , T)+PF  
 1 IO

Component Processor Element Main Thrust Chamber

PC = B72\*' INTEGRATE (DWOI (T)+B109\*DWFI (T)-DWCN (T) , T)+PC0  
 PC = B114\*PC

Component Processor Element Nozzle Bypass Element

DWCCV = DWFNBP  
 DWFNBP = B98\*' INTEGRATE (-B99\*DWFNBP (T)/ACCV\*\*2+PMFVD-P9 , T)+DWFNBP0

## **System Generated Documentation**



\*\*\*\*\*

Documentation for Component Processure PUMP

1. Low Pressure Fuel Pump
2. High Pressure Fuel Pump
3. Low Pressure Oxidizer Pump
4. High Pressure Oxidizer Pump

Generic Equations

$F = B_i * [ DW / S ]$  ; Flow variable , in<sup>3</sup>  
 $P = P_s + B_j * (S)^2 * T_p(F)$  ; Total pressure , Psia  
 $R = B_k * (S)^2 * T_r(F)$  ; Torque , in-lb

EOPO0 : Eopo when t=0

EFPO0 : EFpo when t=0

DWOPO0 : DWopo when t=0

DWFPO0 : DWfpo when t=0

DWOPF0 : DWopf when t=0

DWFPF0 : DWfpf when t=0

TROP2 : High Pressure Oxidizer Pump Pump Torque Characteristics

TORQUE : High Pressure Oxidizer Pump "Torque"

TROP1 : Low Pressure Oxidizer Pump Pump Torque Characteristics

TRFP2 : High Pressure Fuel Pump Pump Pump Torque Characteristics

TRFP1 : Low Pressure Fuel Pump Pump Torque Characteristics

TPOP2 : High Pressure Oxidizer Pump Pump Pressure Rise Characteristics

TPOP1 : Low Pressure Oxidizer Pump

TPFP2 : High Pressure Fuel Pump Pump Pressure Rise Characteristics

TPFP1 : Low Pressure Fuel Pump Pump Pressure Rise Characteristics

ROP1 : Low Pressure Oxidizer Pump

RFP2 : High Pressure Fuel Pump

RFP1 : Low Pressure Fuel Pump

POD2 : High Pressure Oxidizer Pump Total Pressure

POD1 : Low Pressure Oxidizer Pump Total Pressure

PFD2 : High Pressure Fuel Pump Total Pressure

PFD1 : Low Pressure Fuel Pump Discharge Total pressure

FOP2 : High Pressure Oxidizer Pump Flow variable

FDP1 : Low Pressure Oxidizer Pump Flow variable

FFP2 : High Pressure Fuel Pump Flow Variable

FFP1 : Low Pressure Fuel Pump Flow variable

\*\*\*\*\*  
\*\*\*\*\*

Documentation for TURBINE section

- 1. Low Pressure Fuel Turbine
- 2. High Pressure Fuel Turbine
- 3. Low Pressure Oxidizer Turbine
- 4. High Pressure Oxidizer Turbine

Generic Equations

$R = B_i * P * Tr$  ; Torque , in-lb f  
 $T_s = B_j * S / (T^{(1/2)})$  ; Speed Parameter  
 $R = B_k * (S)^2 * Tr(F)$  ; Weight Flowrate , lb m/sec

TPR : Pressure Ratio

TROT2 : Low Pressure Oxidizer Turbine Turbine Torque Characteristics

TROT1 : High Pressure Oxidizer Turbine Turbine Torque Characteristics

TRFT2 : High Pressure Fuel Turbine Turbine Torque Characteristics

TRFT1 : Low Pressure Fuel Turbine Turbine Torque Characteristics

DWOT2 : High Pressure Oxidizer Turbine Weight Flowrate

DWOT1 : Low Pressure Oxidizer Turbine Weight Flowrate

DWFT2 : High Pressure Fuel Turbine Weight Flowrate

DWFT1 : Low Pressure Fuel Turbine Weight Flowrate

UOT2 : High Pressure Oxidizer Turbine Speed Parameter

FOT1 : Low Pressure Oxidizer Turbine Flow Variable

UFT2 : High Pressure Fuel Turbine Speed Parameter

UFT1 : Low Pressure Fuel Turbine Speed Parameter

TQOT2 : High Pressure Oxidizer Turbine

TQOT1 : Low Pressure Oxidizer Turbine Torque

TQFT2 : High Pressure Fuel Turbine Torque

TQFT1 : Low Pressure Fuel Turbine Torque

\*\*\*\*\*  
\*\*\*\*\*

Documentationfor TURBOPUMP section

- 1. Low Pressure Fuel Turbopump
- 2. High Pressure Fuel Turbopump
- 3. Low Pressure Oxidizer Turbopump
- 4. High Pressure Oxidizer Turbopump

Generic Equations

$S = (1/G) * \text{'Integrate (Tq*temp(t),t) +S0}$  ; Speed

ORIGINAL PAGE IS  
OF POOR QUALITY

SO20 : So2 when t=0

SO10 : So1 when t=0

SF20 : Sf2 when t=0

SF10 : Sf1 when t=0

SO2 : High Pressure Oxidizer Turbopump Rotational Speed

SO1 : Low Pressure Oxidizer Turbopump Rotational Speed

SF2 : High Pressure Fuel Turbopump Rotational Speed

SF1 : Low Pressure Fuel Turbopump Rotational Speed

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for VALVE section

1. Main Fuel Valve
2. Main Oxidizer Valve
3. Fuel Preburner Oxidizer Valve
4. Oxidizer Preburner Oxidizer Valve

Generic Equations

$$A = Ab * (T * Th | Thb)$$

Input-parameter : Ab T Th Thb

Ab: Average of area, T \* Th | Thb =Valve Effective Area

$$Pit = Pi - RF * DW^2 | RHO$$

Input-parameter : Pi RF DW RHO

Pit: Inlet Total Pressure, Pi: Inlet Pressure

RF: Resistance to Flow, DW: Weight Flowrate, Rho: Density

$$Pis = Pit - DW^2 | [(772.8) * (Ai^2) * RHO]$$

Input-parameter : Pit DW Ai RHO

$$dP = B * (RHOb | RHO) (DW | (A|Ab)^2 - RF * DW | RHO)$$

Input-parameter : B Rhob Rho DW A Ab RF

AOPV : Oxidizer Preburner Oxidizer Valve Area

AFPV : Fuel Preburner Oxidizer Valve Area

AMOV : Main Oxidizer Valve Area

AMFV : Main Fuel Valve Area

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for FLOW section

1. Fuel Preburner Fuel Flow
2. Fuel Preburner Oxidizer Flow
3. Oxidizer Preburner Fuel Flow
4. Oxidizer Preburner Oxidizer Flow

Generic Equations

$$DWf = Bi \int \{P(9) - P - [Bj/R(9)] (DWf)^2\} dt + [DWf]_{t=0} ; \text{Fuel Flow}$$

$$DWO = Bk \int \{Ppos - P - Be[DWO/(A/A')]^2 - Em(DWi)^2\} dt + [DWO]_{t=0} ; \text{Oxidizer Flow}$$

$$Eo = Bn \int \{DWO - DWi\} dt + [Eo]_{t=0} ; \text{Variable in injector Priming function}$$

$$DWi = DWO [Eo]$$

DWOPOI : Oxidizer Preburner Oxidizer Injector Weight Flowrate

DWFPOI : Fuel Preburner Oxidizer Injector Weight Flowrate  
 EOPO : Oxidizer Preburner Oxidizer Variable in injector Priming  
 EFPO : Fuel Preburner Oxidizer Variable in injector Priming  
 DWOPO : Oxidizer Preburner Oxidizer Weight Flowrate  
 DWFPO : Fuel Preburner Oxidizer Weight Flowrate  
 DWOPF : Oxidizer Preburner Fuel Flow  
 DWFPF : Fuel Preburner Fuel Flow

ORIGINAL PAGE IS  
 OF POOR QUALITY

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for COMBUSTER section

1. Fuel Preburner COMBUSTER
2. Oxidizer Preburner COMBUSTER

Generic Equations

$P = B_i * \text{Integrate} \{ [DW1 + DW2 - B_j * DW3], t \} + P_0$  ; Total Pressure, Psia

$F = DW2 / (DW1 + DW2)$  ; Fraction

$T = T_{qt}(F) + B_k * T_9$  ; Temperature

TTFOP : Oxidizer Preburner Hot Gas Temperature Function

TTFPP : Fuel Preburner Hot Gas Temperature Function

POPO : Pop when t=0

PFPO : Pfp when t=0

TOP : Oxidizer Preburner Combuster Temperature

TFP : Fuel Preburner Combuster Temperature

FOP : Oxidizer Preburner Flowrate Fraction

FFP : Fuel Preburner Flowrate Fraction

POP : Oxidizer Preburner Total Pressure

PFP : Fuel Preburner Total Pressure

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for COOLING section

1. Main Chamber Regenerative Cooling Element
2. Nozzle Regenerative Cooling Element

Generic Equations

$P = B * \text{integrate} \{ [DQw1 + DQw2 + H3 * DWi - H * DW], t \} + P_0$  ; Total Pressure

$H = b * T$  ; Specific Enthalpy

$DW = B_i * \text{integrate} \{ [P - P_5 - (B_j/R_5) * DW^2], t \} + DW_0$  ; Weight Flowrate

$R = B * \text{integrate} \{ [DWi - DW], t \} + R_0$  ; Density

$T_5 = B * (P / R)$  ; Temperature

$DQw1 = b * [1.0 + 0.002 * T_5] * [Tw15 - T_5] * DW_{mc}^{0.8}$  ; Heat transfer rate

$DQw2 = B * [1.0 + 0.002 * T_5] * [Tw25 - T_5] * DW_{mc}^8$  ; Heat Transfer Rate

DQtc = B \* [Tc - Tw15] \* [DWcn]^0.8  
Tw1 = B \* integrate {[DQtc - DQw1],t} + Tw10 ; Hot Gas Wall Temperature  
Tw2 = B \* integrate {[-DQw2],t} + Tw20 ; Ambient thrust chamber Temperature  
TW240 : Tw24 when t=0  
  
TW140 : Tw14 when t=0  
  
TW250 : Tw25 when t=0  
  
TW150 : TW15 when t=0  
  
R40 : R4 when t=0  
  
R50 : R5 when t=0  
  
DW40 : DW4 when t=0  
  
DWMC0 : DWmc when t=0  
  
P40 : P4 when t=0  
  
P50 : P5 when t=0  
  
TW24 : Ambient Thrust Temperature  
  
TW14 : Hot Gas temperature  
  
TW25 : Ambient thrust chamber Temperature  
  
TW15 : Hot Gas Temperature  
  
DQTC4 : Nozzle Regenerative Heat Transfer Rate TC  
  
DQTC5 : Main Chamber Cooling Element Heat Transfer Rate TC  
  
DQW24 : Nozzle Regenrative Ambient Thrust Heat Transfer Rate  
  
DQW14 : Nozzle Regenerative Hot Gas Heat Transfer Rate  
  
DQW25 : Main Chamber Ambient Thrust Chamber Heat Transfer Rate  
  
DQW15 : Main Chamber Hot Gas Heat Transfer Rate  
  
T4 : Nozzle Regenerative Cooling Temperature  
  
T5 : Main Chamber Heat Exchanger Temperature  
  
R4 : Nozzle Regenerative Element Density  
  
R5 : Main Chamber Element Density  
  
DW4 : Nozzle Regenerative Cooling Weight Flowrate  
  
DW5 : Main Chamber Heat Weight Flowrate  
  
H4 : Nozzle Regenerative Cooling Element Specific Enthalpy  
  
H5 : Main Chamber Heat Exchanger Specific Enthalpy  
  
P4 : Nozzle Regenerative Exchanger Element Total Pressure  
  
P5 : Main Chamber Heat Exchanger Total Pressure

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for FLOWMETER section

1. Fuel Flowmeter
2. Oxidizer Flowmeter

Generic Equations

$Q = P - (RF * DW^2) / Rb$  ; Volumetric Flowrate ,GPM

$P = Pf - (RF * DW^2) / R$  ; Total Pressure ,Psia

$T = Ti + B * S$  ; Temperature

TOD2 : Low pressure Oxidizer pump Flowmeter Temperature

TFP1 : Low Pressure Fuel Pump Flowmeter Temperature

PFFM : Fuel Flowmeter Total Pressure

QF : Fuel Volumetric Flowrate

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for PREBURNER-SUPPLY-DUCT section

1. Preburner Fuel Supply Duct
2. Preburner Oxidizer Supply Duct

Generic Equations

$P = B * \text{integrate}([DW], t) + P0$  ; Total Pressure , Psia

$T = B * [P/R]$  ; Temperature

$R = [DW1 + DW2] / [(DW2/R) + B * DW3]$  ; Density

PPOS0 : Ppos when t=0

P90 : P9 when t=0

R9 : Preburner Fuel Supply Element Density

T9 : Preburner Fuel Supply Element Temperature

PPOS : Preburner Oxidizer Supply Total Pressure

P9 : Preburner Fuel Supply Element Total Pressure

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for BOOST-STAGE section

1. High Pressure Oxidizer boostpump Flow variable
2. High Pressure Oxidizer boostpump rotational Torque
3. High Pressure Oxidizer Boostpump pressure
4. High Pressure Oxidizer Boostpump weight flowrate

Generic Equations

$F = B * (DW / S)$  ; Flow variable , in<sup>3</sup>

$T = B * (S^2) * TT * (F)$  ; Torque , in-lb

$P3 = B * (S^2) * TT * F + P2$  ; Pressure ,

$DW3 = B1 * \text{integrate} [(P3 - P - B2^2), t] + DW0$ ; Weight Flowrate, lbm/sec

DWOP30 : DWop3 when t=0

TPOP3 : high Pressure Oxidizer boostpump Pressure Rise Character

TROP3 : High Pressure Oxidizer Boostpump Torque  
 DWOP3 : High Pressure Oxidizer Boostpump Weight Flowrate  
 POD3 : High Pressure Oxidizer Boostpump Pressure  
 TOP3 : High Pressure Oxidizer Boostpump Torque  
 FOP3 : High Pressure Oxidizer boostpump Flow variable

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for BOOST-STAGE section

1. High Pressure Oxidizer boostpump Flow variable
2. High Pressure Oxidizer boostpump rotational Torque
3. High Pressure Oxidizer Boostpump pressure
4. High Pressure Oxidizer Boostpump weight flowrate

Generic Equations

$F = B * (DW / S)$  ; Flow variable , in<sup>3</sup>  
 $T = B * (S^2) * TT * (F)$  ; Torque , in-lb  
 $P3 = B * (S^2) * TT * F + P2$  ; Pressure ,  
 $DW3 = B1 * \text{integrate} [(P3 - P - B2^2), t] + DW0$ ; Weight Flowrate, lbm/sec  
 PFIO : Pfi when t=0

EOI : Main Chamber injector, Oxidizer variable in injector priming function  
 WOI : Main Chamber Injector, Oxidizer Weight  
 DWOI : Main Chamber injector, Oxidizer Weight Flowrate  
 TFI : Main Chamber Temperature  
 DWFI : Main Chamber injector weight flowrate  
 PFI : Main Chamber Injector, fuel and Hot Gas Total Pressure

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for MAIN-THRUST-CHAMBER section

1. Combuster Injector end static Total Pressure
2. Main thrust Chamber Total Pressure

Generic Equations

$P = b * P$  ; Combuster Injection Total pressure , Psia  
 $P = B1 * \text{'integrate} [(B2 * DW1 + DW2 - DW3), t] + P0$ ; Main Thrust Chamber Total Pressure, Psia  
 PCO : Pc when t=0

PC : Main thrust chamber Total Pressure  
 PCIES : Combuster Injector end static Total Pressure

\*\*\*\*\*

\*\*\*\*\*

Create Knowledge Base for NOZZLE-BYPASS section

- 1. Nozzle Heat exchanger bypass Weight Flowrate
- 2. Coolant Control valve

Generic Equations

DW = B1 \* 'integrate([P - P9 - B2 \* ( DW1/(A/Ab)^2],t) + DW0 ; Nozzle Heate weight flowrate , lbm/sec

DW = DW1 ; Coolant Control Valve , lbm/sec

DWFNBPO : DWfnbp when t=0

DWCCV : Coolant Control Valve

DWFNBP : Nozzle Heat Exchanger bypass weight flowrate

\*\*\*\*\*



## Example for the Robot Manipulators

Example steps to solve the end position.

1. Define rules.  $\sin^2 \theta_i = 1 - \cos^2 \theta_i$

$SS_i = \sin \theta_i$        $i$  is from 1 to 6

$CC_i = \cos \theta_i$        $i$  is from 1 to 6

2. Define matrices  $A_1, A_2, A_3, A_4, A_5, A_6, T_6$  .

3.  $T_6 = A_1 A_2 A_3 A_4 A_5 A_6$

$T_6$  is the end position and  $A_i$  is the matrix for each joint.

$T_6$  is the product of  $A_1 \dots A_6$

4.  $A_1^{-1} T_6 = A_2 A_3 A_4 A_5 A_6$

$A_1^{-1} A_2^{-1} T_6 = A_3 A_4 A_5 A_6$

$A_1^{-1} A_2^{-1} \dots A_5^{-1} T_6 = A_6$

5. Compare the above equations and solve

$$T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} CC_6 & -SS_6 & 0 & 0 \\ SS_6 & CC_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{End position } T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6. The end position is a product of the A matrices

$$T_6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6$$

7. Matrix inverse, matrix multiplication

$$T_{16} = A_1^{-1} T_6 = A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6$$

$$T_{26} = A_1^{-1} A_2^{-1} T_6 = A_3 \cdot A_4 \cdot A_5 \cdot A_6$$

$$T_{36} = A_1^{-1} A_2^{-1} A_3^{-1} T_6 = A_4 \cdot A_5 \cdot A_6$$

$$T_{46} = A_1^{-1} A_2^{-1} A_3^{-1} A_4^{-1} T_6 = A_5 \cdot A_6$$

$$T_{56} = A_1^{-1} A_2^{-1} A_3^{-1} A_4^{-1} A_5^{-1} T_6 = A_6$$

8. Evaluating equations to obtain the end position

$$n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, a_z, p_x, p_y, p_z$$

ORIGINAL PAGE IS  
OF POOR QUALITY

A1:>demo>demo-stanford.out.1

(D4)  
(C5) BATCH("A1:>DEMO>DEMO-STANFORD.DEMO");

(C6) tellsimp(ss1^2,1-cc1^2) \$

(C7) tellsimp(ss2^2,1-cc2^2) \$

(C8) tellsimp(ss3^2,1-cc3^2) \$

(C9) tellsimp(ss4^2,1-cc4^2) \$

(C10) tellsimp(ss5^2,1-cc5^2) \$

(C11) tellsimp(ss6^2,1-cc6^2) \$

(C12) A1:MATRIX([CC1,0,-SS1,0],[SS1,0,CC1,0],[0,-1,0,0],[0,0,0,1]);  

$$\begin{bmatrix} [CC1 & 0 & -SS1 & 0] \\ [SS1 & 0 & CC1 & 0] \\ [0 & -1 & 0 & 0] \\ [0 & 0 & 0 & 1] \end{bmatrix}$$

(D12)

(C13) A2:MATRIX([CC2,0,SS2,0],[SS2,0,-CC2,0],[0,1,0,DD2],[0,0,0,1]);

$$\begin{bmatrix} [CC2 & 0 & SS2 & 0] \\ [SS2 & 0 & -CC2 & 0] \\ [0 & 1 & 0 & DD2] \\ [0 & 0 & 0 & 1] \end{bmatrix}$$

(D13)

(C14) A3:MATRIX([1,0,0,0],[0,1,0,0],[0,0,1,DD3],[0,0,0,1]);

$$\begin{bmatrix} [1 & 0 & 0 & 0] \\ [0 & 1 & 0 & 0] \\ [0 & 0 & 1 & DD3] \\ [0 & 0 & 0 & 1] \end{bmatrix}$$

(D14)

(C15) A4:MATRIX([CC4,0,-SS4,0],[SS4,0,CC4,0],[0,-1,0,0],[0,0,0,1]);

$$\begin{bmatrix} [CC4 & 0 & -SS4 & 0] \\ [SS4 & 0 & CC4 & 0] \\ [0 & -1 & 0 & 0] \\ [0 & 0 & 0 & 1] \end{bmatrix}$$

(D15)

(C16) A5:MATRIX([CC5,0,SS5,0],[SS5,0,-CC5,0],[0,1,0,0],[0,0,0,1]);

$$\begin{bmatrix} [CC5 & 0 & SS5 & 0] \\ [SS5 & 0 & -CC5 & 0] \\ [0 & 1 & 0 & 0] \\ [0 & 0 & 0 & 1] \end{bmatrix}$$



```

[      CC4 SS6 + CC5 CC6 SS4      ]
[      ]
[      ]
(C23) a2i:invert(a2);
[ CC2 SS2 0 0 0 ]
[      ]
[      ]
[ 0 0 1 - CC2 DD2 - (1 - CC2 ) DD2 ]
[      ]
[ SS2 - CC2 0 0 ]
[      ]
[ 0 0 0 0 1 ]
(C24) sa2i:ratsimp(a2i);
[ CC2 SS2 0 0 ]
[      ]
[ 0 0 1 - DD2 ]
[      ]
[ SS2 - CC2 0 0 ]
[      ]
[ 0 0 0 0 1 ]
(C25) a2i6:sa2i.a1i.t6;
[ CC2 (NY SS1 + CC1 NX) - NZ SS2 CC2 (OY SS1 + CC1 OX) - OZ SS2 CC2 (AY SS1 + AX CC1) - AZ SS2 CC2 (PY SS1 + CC1 PX) - PZ SS2 ]
[      ]
[      CC1 NY - NX SS1      ]
(C25) [      ]
[ (NY SS1 + CC1 NX) SS2 + CC2 NZ (OY SS1 + CC1 OX) SS2 + CC2 OZ (AY SS1 + AX CC1) SS2 + AZ CC2 (PY SS1 + CC1 PX) SS2 + CC2 PZ ]
[      ]
[      ]
[      ]
(C26) t26:a3.a4.a5.a6;
[ CC4 CC5 CC6 - SS4 SS6 - CC4 CC5 SS6 - CC6 SS4 CC4 SS5 0 ]
[      ]
[ CC4 SS6 + CC5 CC6 SS4 CC4 CC6 - CC5 SS4 SS6 SS4 SS5 0 ]
[      ]
[ - CC6 SS5 SS5 SS6 CC5 DD3 ]
[      ]
[ 0 0 0 0 1 ]
(C27) a3i:invert(a3);
[ 1 0 0 0 ]
[      ]
[ 0 1 0 0 ]
[      ]
[ 0 0 1 - DD3 ]
[      ]
[ 0 0 0 1 ]
(C28) sa3i:ratsimp(a3i);
[ 1 0 0 0 ]
[      ]
[ 0 1 0 0 ]
[      ]
[ 0 0 1 - DD3 ]
[      ]
[ 0 0 1 - DD3 ]

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

[ 0 0 0 1 ]
[ 0 0 0 1 ]

(C29) a3i6:sa3i.sa2i.sali.t6;
(D29)
[ CC2 (NY SS1 + CC1 NX) - NZ SS2 CC2 (OY SS1 + CC1 OX) - OZ SS2 CC2 (AY SS1 + AX CC1) - AZ SS2 CC2 (PY SS1 + CC1 PX) - PZ SS2 ]
[ [ CC1 NY - NX SS1 ] ]
[ [ CC1 OY - OX SS1 ] ]
[ [ (NY SS1 + CC1 NX) SS2 + CC2 NZ (OY SS1 + CC1 OX) SS2 + CC2 OZ (AY SS1 + AX CC1) SS2 + AZ CC2 (PY SS1 + CC1 PX) SS2 + CC2 PZ - DD3 ] ]
[ [ 0 0 ] ]
[ [ 0 ] ]
(C30) t36:a4.a5.a6;
[ CC4 CC5 CC6 - SS4 SS6 - CC4 CC5 SS6 - CC6 SS4 CC4 SS5 0 ]
[ [ CC4 SS6 + CC5 CC6 SS4 CC4 CC6 - CC5 SS4 SS6 SS4 SS5 0 ] ]
[ [ - CC6 SS5 SS5 SS6 CC5 0 ] ]
[ [ 0 0 ] ]
[ [ 0 0 0 1 ] ]

(C31) a4i:invert(a4);
[ CC4 SS4 0 0 ]
[ [ 0 0 - 1 0 ] ]
[ [ - SS4 CC4 0 0 ] ]
[ [ 0 0 0 1 ] ]

(C32) sa4i:ratsimp(a4i);
[ CC4 SS4 0 0 ]
[ [ 0 0 - 1 0 ] ]
[ [ - SS4 CC4 0 0 ] ]
[ [ 0 0 0 1 ] ]

(C33) a4i6:sa4i.sa3i.sa2i.sali.t6;
[ (CC1 NY - NX SS1) SS4 + CC4 (CC2 (NY SS1 + CC1 NX) - NZ SS2) ]
[ [ - (NY SS1 + CC1 NX) SS2 - CC2 NZ ] ]
(D33) Col 1 = [ [ CC4 (CC1 NY - NX SS1) - (CC2 (NY SS1 + CC1 NX) - NZ SS2) SS4 ] ]
[ [ 0 ] ]
[ (CC1 OY - OX SS1) SS4 + CC4 (CC2 (OY SS1 + CC1 OX) - OZ SS2) ]
[ [ - (OY SS1 + CC1 OX) SS2 - CC2 OZ ] ]
Col 2 = [ [ CC4 (CC1 OY - OX SS1) - (CC2 (OY SS1 + CC1 OX) - OZ SS2) SS4 ] ]
[ [ 0 ] ]

```

ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

```

[
  0
]
[ (AY CC1 - AX SS1) SS4 + CC4 (CC2 (AY SS1 + AX CC1) - AZ SS2) ]
[
[
  - (AY SS1 + AX CC1) SS2 - AZ CC2
]
]
Col 3 = [
[ CC4 (AY CC1 - AX SS1) - (CC2 (AY SS1 + AX CC1) - AZ SS2) SS4 ]
[
  0
]
]
[ (- PX SS1 + CC1 PY - DD2) SS4 + CC4 (CC2 (PY SS1 + CC1 PX) - PZ SS2) ]
[
[
  - (PY SS1 + CC1 PX) SS2 - CC2 PZ + DD3
]
]
Col 4 = [
[ CC4 (- PX SS1 + CC1 PY - DD2) - (CC2 (PY SS1 + CC1 PX) - PZ SS2) SS4 ]
[
  1
]
]
(C34) t46:a5.a6;
[ CC5 CC6 - CC5 SS6 SS5 0 ]
[
[ CC6 SS5 - SS5 SS6 - CC5 0 ]
[
  SS6 CC6 0 0 ]
[
  0 0 0 1 ]
]
(D34)
(C35) a5i:invert(a5);
[ CC5 SS5 0 0 ]
[
  0 0 1 0 ]
[
  SS5 - CC5 0 0 ]
[
  0 0 0 1 ]
]
(D35)
(C36) sa5i:ratsimp(a5i);
[ CC5 SS5 0 0 ]
[
  0 0 1 0 ]
[
  SS5 - CC5 0 0 ]
[
  0 0 0 1 ]
]
(D36)
(C37) a5i6:sa5i.sa4i.sa3i.sa2i.sa1i.t6$
(C38) col(a5i6,1);
[ (- (NY SS1 + CC1 NX) SS2 - CC2 NZ) SS5 + CC5 ((CC1 NY - NX SS1) SS4 + CC4 (CC2 (NY SS1 + CC1 NX) - NZ SS2)) ]
[
  CC4 (CC1 NY - NX SS1) - (CC2 (NY SS1 + CC1 NX) - NZ SS2) SS4
]
[ ((CC1 NY - NX SS1) SS4 + CC4 (CC2 (NY SS1 + CC1 NX) - NZ SS2)) SS5 - CC5 (- (NY SS1 + CC1 NX) SS2 - CC2 NZ) ]
[
  0
]
]
(D38)

```





## REFERENCES

1. Rockwell International Corporation, Rocketdyne Division, "Engine Balance and Dynamic Model," Model No. RL00001, Documentation Ident. No. 02602.
2. Paul, Richard P.: Robot Manipulators: Mathematics, Programming, and Control. Pages 42-62.

APPROVAL

AUTOMATIC MATHEMATICAL MODELING FOR  
REAL TIME SIMULATION PROGRAM

By Caroline Wang and Steve Purinton

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.



---

WILLIAM B. CHUBB

Director, Information and Electronic  
Systems Laboratory