Pratt & Whitney
FR-20282-3
25 July 1989

LARGE LIQUID ROCKET ENGINE TRANSIENT PERFORMANCE SIMULATION SYSTEM

CONTRACT NO. NAS8-36994
SIX MONTH REPORT

PERIOD OF PERFORMANCE
1 OCTOBER 1988 THROUGH 31 MAY 1989

Prepared for
NASA-George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

Prepared by:

J. R. Mason
Program Manager

R. D. Southwick
Propulsion Systems
Technology Manager

Pratt & Whitney
P. O. Box 109600
West Palm Beach, Fl. 33410-9600

UNITED TECHNOLOGIES PRATT&WHITNEY

CONTENTS

CONTENTS (Continued)

INTRODUCTION

Large Liquid Rocket Engine Transient Performance Simulation, Contract NAS8-36994, is a program to design and develop a simulation system for rocket engine transient performance models. The system was designed and conceptually demonstrated during Phase I of the contract. The current Phase II of the contract will enhance the system, develop a detailed simulation of the Technology Test Bed Engine (TTBE), and deliver the model and system software to the National Aeronautics and Space Administration (NASA) George C. Marshall Space Flight Center (MSFC).The program is sponsored by NASA/MSFC with Mr. W.A. Adams acting as project manager. The Pratt & Whitney program manager is Mr. J.R. Mason.

The objective of the program is to apply state-of-the-art modeling and simulation technology to the rocket engine models. This application will be used to design, develop, and produce all liquid rocket transient simulation system compatible with MSFC control analysis software, simulation facilities, and computers. The simulation system will be referred to as the ROCETS (Rocket Engine Transient Simulation) System.

The report covers the third six months of ROCETS development.

SUMMARY

Phase 1 of the ROCETS program consists of seven technical tasks plus provision for reports and review; these tasks are:

1. Architecture
2. System Requirements
3. Component and Submodel Requirements
4. Submodel Implementation
5. Component Implementation
6. Submodel Testing and Verification
7. Subsystem Testing and Verification

During Phase I, the above tasks were completed and a Critical Design Review and Phase I completion review were conducted at MSFC. Phase II of Rocets consists of two technical tasks plus reports and reviews; these tasks are:

1. TTBE Model Data Generation
2. System Testing and Verification

During this period specific coding of the system processors was begun and the engineering representations of Phase I were expanded to produce a simple model of the Technology Test Bed Engine (TTBE). As code was completed, some minor modifications to the system architecture centering on the global variable common, GLOBVAR, were necessary to increase processor efficiency.

The engineering modules completed during Phase II are listed below:

1. INJT00 - Main Injector
2. MCHB00 - Main Chamber
3. NOZL00 - Nozzle Thrust Calculations
4. PBRN00 - Preburner
5. PIPE02 - Compressible flow without inertia
6. PUMP00 - Polytropic Pump
7. ROTR00 - Rotor Torque Balance/Speed Derivative
8. TURB00 - Turbine

Detailed documentation of these modules is in the appendix. In addition to the engineering modules, several submodules were also completed during the reporting period. These submodules include combustion properties, component performance characteristics (maps) and specific utilities. Documentation of the submodules will be included in updates to the System Design Specification (SDS).

Specific coding was begun on the system configuration processor. All functions necessary for multiple module operation have been completed but the SOLVER implementation is still under development. This portion of the processor has been incorporated into an in-house system for automated verification of the engineering modules. This system, the Verification Checkout Facility (VCF) allows interactive comparison of module results to store data as well as provides an intermediate checkout of the processor code.

After validation using the VCF, the engineering modules and submodules were used to build a simple Technology Test Bed Engine (TTBE). The simple TTBE model has 55 states and 2 algebraic loops. This simulation of main-stage TTBE operation demonstrates steady-state and transient operation of a full rocket simulation, used the ROCETS general property package, and provided additional information on system operation. The simple TTBE model provides a basis for completion of the detailed TTBE model which will be used to verify the ROCETS system.

SYSTEM REQUIREMENT CHANGES

The system requirements remain unchanged.

SYSTEM ARCHITECTURE CHANGES

Minor modifications to the global common structure are being implemented during coding. These changes center around the GLOBVAR common. Instead of breaking out all variables by type and storing them in separate arrays, all variables are being stored in the order encountered into the global common. The only variable types currently being used are real*4, character*4 and integer*4. The global variable names are still be stored in the GLOBNAM common as character*8. The changes should increase processor speed in the linked list search which verifies variable existence.

COMPONENT/SUBMODULE OVERVIEW

Modules

Thirteen component modules have been developed for the ROCETS system. Five modules were developed during Phase I of the contract and were used to model a simple engine sub-system. Eight more modules were developed under Phase II of the contract in order to model a complete rocket engine. The Phase I modules have been slightly revised under Phase II and have been incorporated into the rocket engine simulation (Simple Technology Test Bed Engine).

The component modules fall into two general categories -- state-derivative modules and non state-derivative modules. The state-derivative modules perform a function that can be described by differential equations. Associated with the module are "states" whose time derivatives are calculated by the module. Since the solver controls the value of the states, the modules require them as inputs and use them along with other inputs to calculate the state-derivatives which are module outputs. Non state-derivative modules have no states associated with them and are not directly controlled by the solver. The format for each module consists of an argument list, software identification number, history (revision dates), interface data, a description of inputs and outputs, a brief engineering description, a list of required subroutines/commons and finally the FORTRAN code.

Following is a list of the component modules that have been developed or updated, along with a brief description.

STATE-DERIVATIVE MODULES

* PBRN00 - PREBURNER MODULE. Combined combustion and volume dynamics module using overall density, temperature and oxidizer fraction as states.

* MCHB00 - MAIN COMBUSTION CHAMBER. Combustion and volume dynamics module using overall density, temperature and oxidizer fraction as states.

* INJT00 - MAIN FUEL INJECTOR. Perfect gas volume dynamics using overall density, temperature and oxidizer fraction as states

* ROTR00 - ROTOR MODULE. Simple torque balance that handles two supply torques, two required torques, drag torque and extraction torque. Rotational speed is the state.

* PIPE00 - PIPE WITH INERTIA. Updated pipe flow module using flow as a state for incompressible flow.

* MIXR00 - FLOW MIXER. Updated flow mixer with reverse flow capability using density and internal energy as states.

* SPLT00 - FLOW SPLITTER. Updated flow splitter with reverse flow capability using density and internal energy as states.

* VOLM00 - LUMPED VOLUME. Updated volume module with reverse flow capability and heat transfer. Density and internal energy are the states.

NON STATE-DERIVATIVE MODULES

* PUMP00 - PUMP MODULE. Polytropic pump with an internal iteration on density ratio.

* TURB00 - TURBINE MODULE. Isentropic turbine with heat transfer and compressibility effects.

* NOZL00 - NOZZLE MODULE. Isentropic nozzle with normal shock.

* PIPE01 - PIPE WITH LOSS. Updated pipe flow module without inertia. Pressure loss for incompressible flow.

* PIPE02 - FLOW THROUGH ORIFICE. Compressible flow through an orifice using a standard flow parameter relation.

Submodules

Twenty-one engineering submodules which are used to support the rocket engine component modules have been created or updated during this phase of the program. Basically, the sub-modules are divided into maps, utility and property submodules.

The map submodules, described in Table I, consist of component maps used to provide specific engine characteristics.

The utility submodules, described in Table II, consist of routines which perform specific functions or tasks.

The property submodules, described in Table III, consist of routines and maps used to calculate both combustion properties and real fluid properties. The combustion property submodules which were created during this phase consist of a system driver, model type interface routine and a series of combustion maps which are limited to reactions involving hydrogen and oxygen as the reactants. The real fluid property submodule was updated to meet the growing requirements of the ROCETS systems.

TABLE I
MAP SUBMODULES

SUBROUTINE	DESCRIPTION
PMAP01	- SSME ATD high pressure fuel pump map of both head and torque coefficients vs flow coefficient.
PMAP02	- SSME ATD high pressure oxidizer (main stage) pump map of both head and torque coefficients vs flow coefficient.
PMAP03	- SSME ATD high pressure oxidizer (preburner stage) pump map of both head and torque coefficients vs flow coefficient.
TBMP01	- SSME ATD high pressure oxidizer turbine map of flow parameter vs PR with lines of speed parameter. Also, efficiency over velocity ratio vs velocity ratio.
TBMP02	- SSME ATD high pressure fuel turbine map of flow parameter efficiency over velocity ratio vs velocity ratio vs PR with lines of speed parameter. Also, efficiency over velocity ratio vs velocity ratio.
RDMP20	- Preburner/main chamber temperature rise curves.

TABLE II
UTILITY SUBMODULES

SUBROUTINE	DESCRIPTION
CDNZ00	- Calculates the performance of an isentropic nozzle with normal shock.
MACH03	- Calculates Mach number
UNIT00	- Determines all the SI/English units conversion constants

TABLE III
PROPERTY SUBMODULES

SUBROUTINE	DESCRIPTION
COMBUSTION PROPERTIES -	CREATED
COMB00	- The combustion property driver routine
PGAS00	- The perfect gas model interface routine
CPGM00	- Calculate CP and gamma for the products of combustion. The routine also allows for pressure ratio effects.
CPMP01	- CP as a function pressure & temperature for hydrogen
RDM12	- Molecular weight as a function of mixture ratio
RDMP13	- Gamma as a function of oxidizer fraction
RDMP21	- CP as a function of oxidizer fraction
ZGAS00	- Calculates compressibility (Z) effects as a function pressure, temperature and mixture ratio
ZZMP01	- Compressibility as a function of pressure and temperature for hydrogen
	REAL FLUID PROPERTIES - UPDATED
PROP00	- Provides thermodynamic properties for the five real fluid properties.

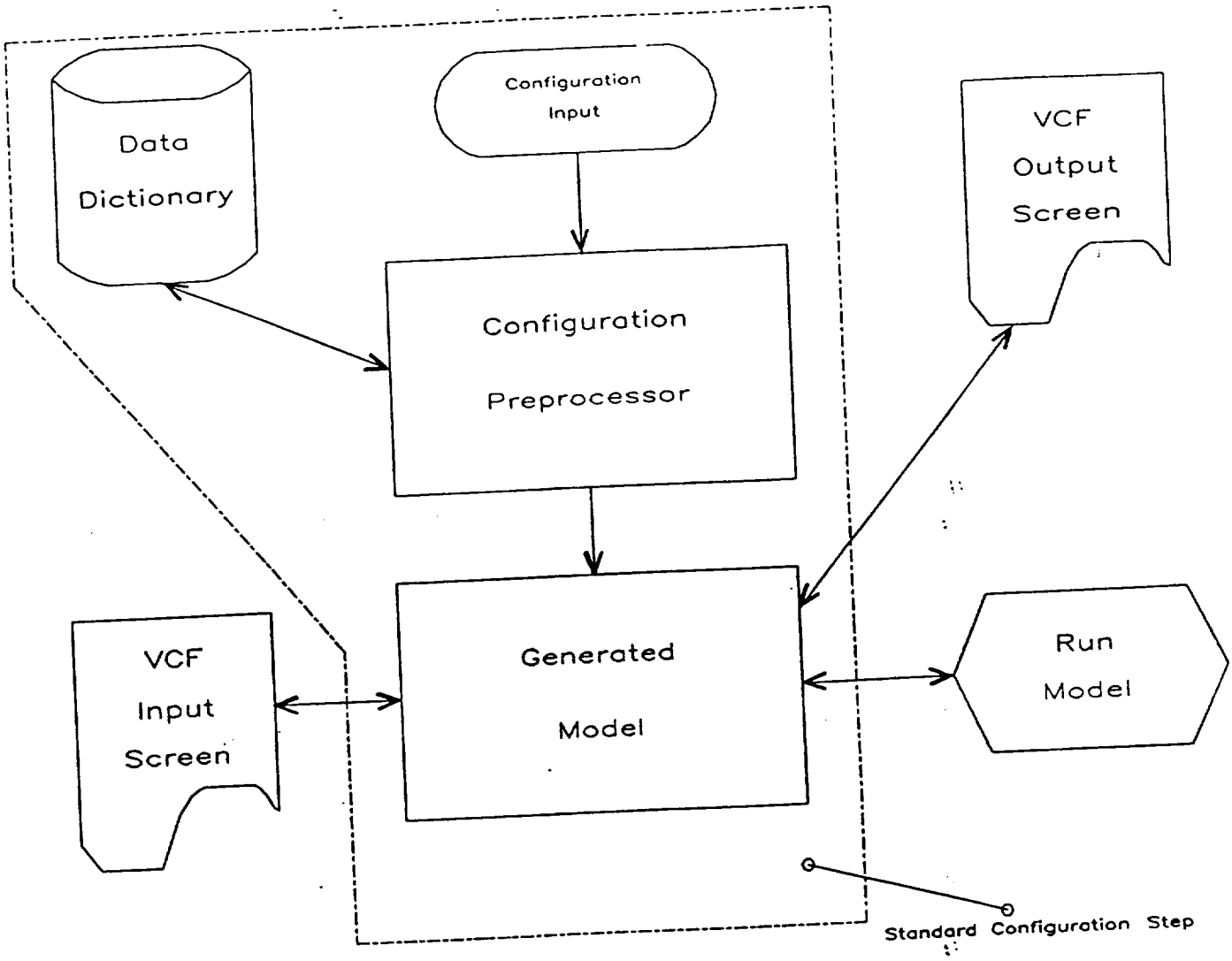
System Processors

The configuration processor is currently under full scale development. All functions necessary for a single module by module checkout have been completed. The system can currently interpret a single module/multiple node input and configure the necessary FORTRAN calls. Property capability has been added and currently state, state derivative and additional balance capabilities are being coded to allow multiple module checkouts to begin. Unit definition is also definable through configuration input. A common structure has been defined and is currently in place and is described in greater detail in the architecture section. A linked list system is used for determining variable existence and adding new variables to the list during configuration.

A system was created called VCF (Verification Checkout Facility) which allows interactive comparisons to be made between a ROCETS module and a section of

code representing the same function in the DTM. Binary CAPA files are created using the DTM. This allows the engineering equations to be verified during the development phase.

Configuration Preprocessor

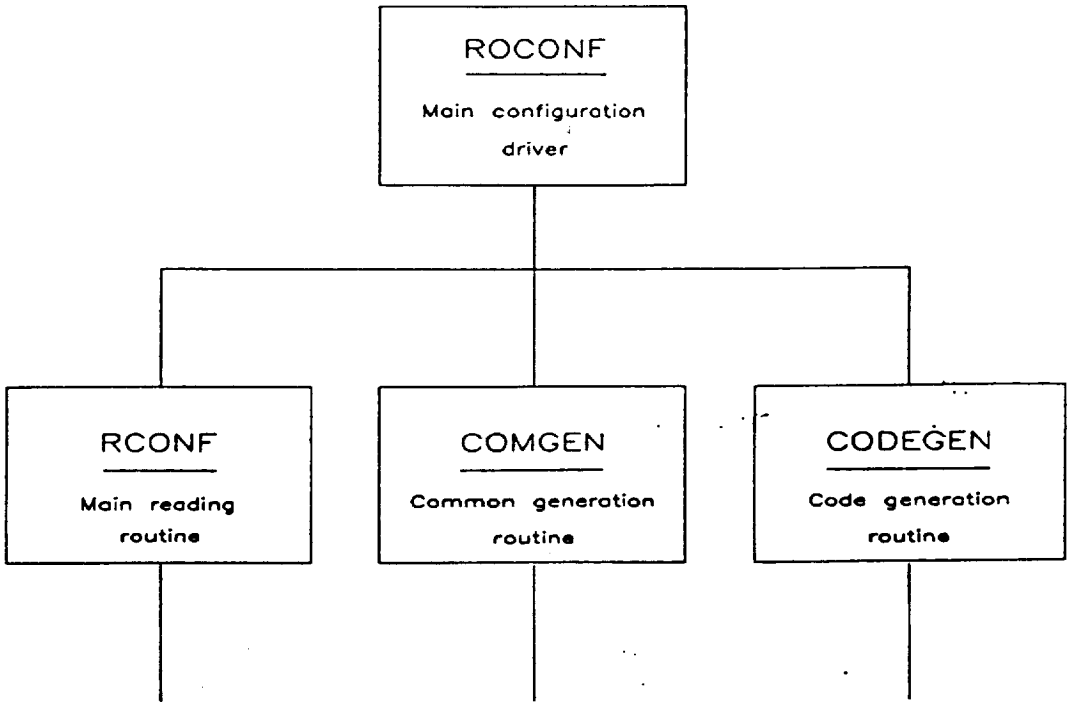


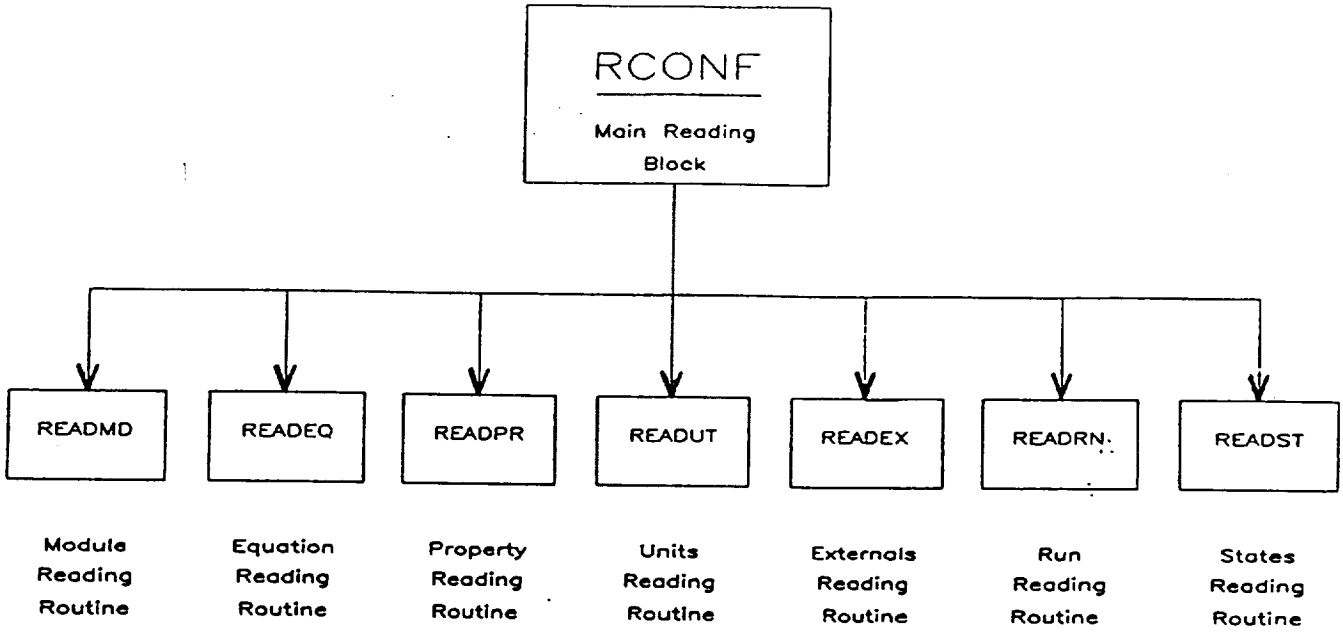
The Configuration Preprocessor and the Verification Checkout Facility combine as illustrated above. A simple flag in the configuration input file either turns on or turns off the VCF system.

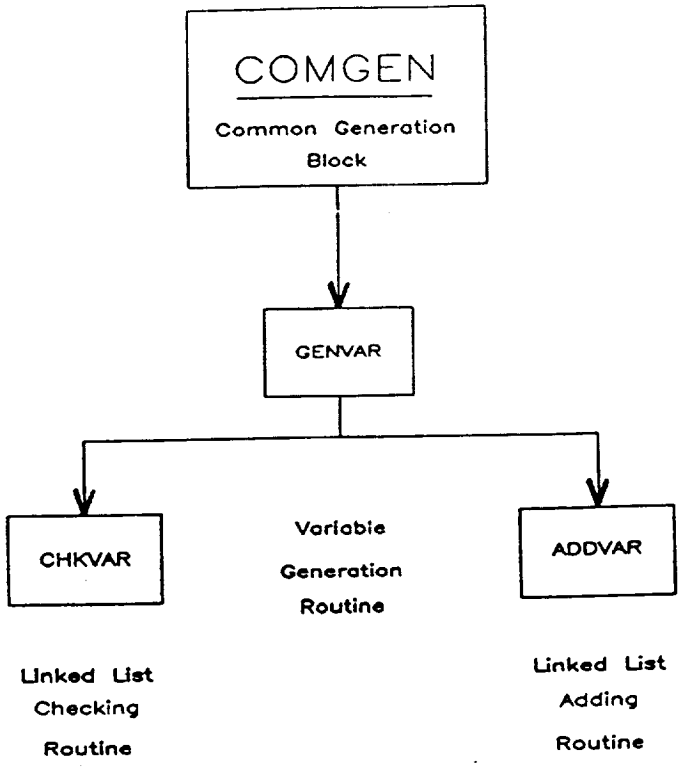
ROCETS Main System

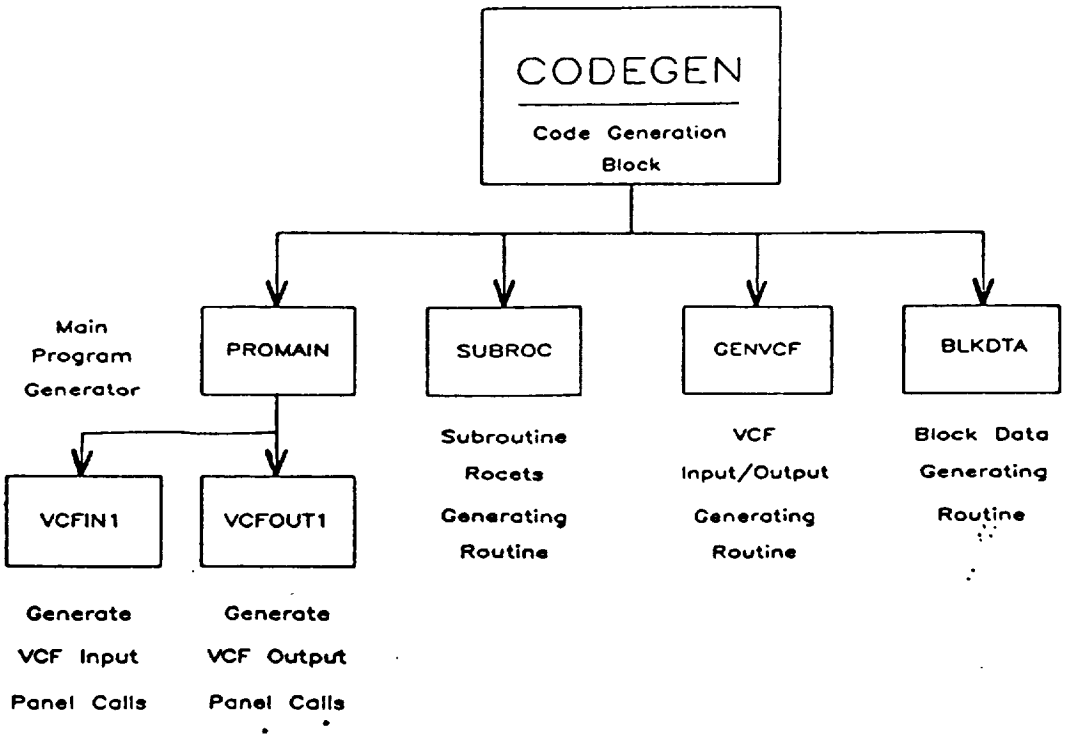
Configuration PreProcessor

First level calls from main driver routine.




RCONF
Main Reading Block
READMD
READEQ
READPR
READUT
READEX
READRN
READST
Module Reading Routine
Equation Reading Routine
Property Reading Routine
Units Reading Routine
Externals Reading Routine
Run Reading Routine
States Reading Routine


COMGEN
Common Generation Block
GENVAR
CHKVAR
Variable Generation Routine
ADDVAR
Linked List Checking Routine
Linked List Adding Routine


CODEGEN
Code Generation Block
Main Program Generator
PROMAIN
SUBROC
GENVCF
BLKDTA
Subroutine Rocets Generating Routine
VCF Input/Output Generating Routine
Block Data Generating Routine
VCFIN1
VCFOUT1
Generate VCF Input Panel Calls
Generate VCF Output Panel Calls

A example of the module input and the generated code follows. For a two module input, such as:

```
    DEFINE SYSTEM

      MODULE: 1 USING PIPE00     ,
              NODE 1 IS   IN     ,
              NODE 2 IS   VOL1   ;
      MODULE: 2 USING PIPE01     ,
              NODE 3 IS   IN2    ,
              NODE 4 IS   VOL2   ;

    END SYSTEM
```

The following common would have been generated by the configuration processor:

```
      COMMON / GLOBVAR /
     *      IPRPL    ,   IUPDAT   ,   MODN1    ,   NOD1IN   ,   NOD2VOL1,
     *      AREA1    ,   CF1      ,   PTIN     ,   PTVOL1   ,   RHOIN    ,
     *      RHOVOL1  ,   XLEN1    ,   W1       ,   DWDT1    ,   MODN2    ,
     *      NOD1IN2  ,   NOD2VOL2,   CF2      ,   PTIN2    ,   PTVOL2   ,
     *      RHOIN2   ,   W2
```

All of these variables were created using the standard naming convention described in the ROCETS standards.

The above input would also have resulted in the following calls to the engineering modules as listed below:

```
      CALL PIPE00   ( IPRPL    ,IUPDAT   ,MODN1    ,NOD1IN   ,
     *      NOD2VOL1 ,AREA1    ,CF1      ,PTIN     ,PTVOL1   ,
     *      RHOIN    ,RHOVOL1  ,XLEN1    ,W1       ,DWDT1    )
      CALL PIPE01   ( IPRPL    ,IUPDAT   ,MODN2    ,NOD1IN2  ,
     *      NOD2VOL2 ,CF2      ,PTIN2    ,PTVOL2   ,RHOIN2   ,
     *      W2       )
```

WORKING EXAMPLE (cont.)

The requirements for the property calls have recently been revised. Previously all properties were called as the following example shows:

```
PROPERTY: PTVOL1 = F(RHOVOL1, UTVOL1 ) USING   HYDROGEN;
PROPERTY: PTVOL2 = F(RHOVOL2, UTVOL2 ) USING   OXYGEN  ;
```

This would have produced the following code:

```
C
C  BEGIN PROPERTY CALLS
C
      CALL PROP00 ('HYDROGEN',14,RHOVOL1 ,UTVOL1  ,PTVOL1  ,
     *             DUMYVOL1,DUMYVOL1,TEMP)
      CALL PROP00 ('OXYGEN  ',14,RHOVOL2 ,UTVOL2  ,PTVOL2  ,
     *             DUMYVOL2,DUMYVOL2,TEMP)
```

Recently, the need for real combustion property calculations have altered the form of the property definition to the following:

```
PROPERTY

   PROP00
     PTVOL1 = F(RHOVOL1, UTVOL1 ) USING   HYDROGEN
   COMB00
     <combustion definition>;
```

The combustion definition has not been resolved as of this printing. The type of calls generated would be exact like those listed above for PROP00 and would be similiar for COMB00.

The VCF calls to the input panels from the program main for the above example look like:

```
C
C  CHECKOUT VERIFICATION FACILITY
C  ENTRY CALL

      DO 100 I = 1,  2
      IRC = 0
      DO 50  J = 1, ISIZE(I)
   50 ARRAY(J) = ARGLIST(I,J)
      IF(I.EQ.  2) IRC = -1
  100 CALL VCFIN ( MODNM(I), MMNEM(I), ARRAY, ISIZE(I), IRC )
```

The arrays MODNM and MMENM contain the character names of the module's actual subroutine name and the modules mnemonic, respectively. The array ARGLIST contains the argument lists as read from the data dictionary and ISIZE contains the actual size of each argument list. The variable IRC is used to verify that data was input to the simulation and as a single to the VCF routine that the last call has occurred.

The VCF output panel is also invoked from the main program with the following calls:

```
C
C CALL ROUTINE TO WRITE OUTPUT DATA TO VCF FILE
C
      DO 1000 I = 1,  2
      DO 500  J = 1, ISIZE(I)
      IARRY(J) = IPTLIST(I,J)
  500 ARRAY(J) = ARGLIST(I,J)
 1000 CALL VCFWRT ( MODNM(I), MMNEM(I), ARRAY, IARRY, ISIZE(I) )
C
C INVOKE VCF OUT PANEL SYSTEM
C
      CALL VCFOUT
```

The calls to VCFWRT simply place the output from the simulation into a location where the VCF output routine can have access to it. The VCFOUT routine is the actual output panel driver. The double assignment of ARRAY will be removed at next update since it is redundant.

An example of what the VCF input screen would look like for the first module configured in this example:

```
                          Verification Checkout Facility
19 Jul 1989 ====================(DTM Input Map)====================  3:28:37pm
Module: PIPE00   Mnemonic: 1        DTM: 048010DTM   Time: 0          Page     1
       ROCETS  I/O                          DTM
   1  IPRPL     _  ____________________________________________________________
   2  IUPDAT    _  ____________________________________________________________
   3  MODN1     _  ____________________________________________________________
   4  NOD1IN    _  ____________________________________________________________
   5  NOD2VOL1  _  ____________________________________________________________
   6  AREA1     _  ____________________________________________________________
   7  CF1       _  ____________________________________________________________
   8  PTIN      _  ____________________________________________________________
   9  PTVOL1    _  ____________________________________________________________
  10  RHOIN     _  ____________________________________________________________
  11  RHOVOL1   _  ____________________________________________________________
  12  XLEN1     _  ____________________________________________________________
  13  W1        _  ____________________________________________________________
  14  DWDT1     _  ____________________________________________________________
```

An example of what the VCF output screen format would look like for the first module configured in this example (without values filled in):

```
                          Verification Checkout Facility
19 Jul 1989 ====================(DTM Output Map)====================  3:28:37pm
Module: PIPE00   Mnemonic: 1        DTM: 048010DTM   Time: 0          Page     1
     ROCETS_VAR DTM_VAR  ROCETS_VAL     DTM_VAL      DELTA         %DELTA
   1 IPRPL      ________ _____________ _____________ _____________ _____________
   2 IUPDAT     ________ _____________ _____________ _____________ _____________
   3 MODN1      ________ _____________ _____________ _____________ _____________
   4 NOD1IN     ________ _____________ _____________ _____________ _____________
   5 NOD2VOL1   ________ _____________ _____________ _____________ _____________
   6 AREA1      ________ _____________ _____________ _____________ _____________
   7 CF1        ________ _____________ _____________ _____________ _____________
   8 PTIN       ________ _____________ _____________ _____________ _____________
   9 PTVOL1     ________ _____________ _____________ _____________ _____________
  10 RHOIN      ________ _____________ _____________ _____________ _____________
  11 RHOVOL1    ________ _____________ _____________ _____________ _____________
  12 XLEN1      ________ _____________ _____________ _____________ _____________
  13 W1         ________ _____________ _____________ _____________ _____________
  14 DWDT1      ________ _____________ _____________ _____________ _____________
```

This is an full and current example of a configuration input file:

```
DEFINE RUN

  VCF = ON

END RUN

DEFINE EXTERNALS

PTIN , HTIN , PTEX,
CFLIN1, HTEX , VOLVOL,
CFLIN2;

END EXTERNALS

DEFINE SYSTEM

  MODULE: 1 USING PIPE00     ,
          NODE 1 IS   IN     ,
          NODE 2 IS   VOL1   ;
  MODULE: 2 USING PIPE01     ,
          NODE 3 IS   IN2    ,
          NODE 4 IS   VOL2   ;

  PROPERTY: PTVOL1 = F(RHOVOL1, UTVOL1 ) USING   HYDROGEN;
  PROPERTY: PTVOL2 = F(RHOVOL2, UTVOL2 ) USING   OXYGEN  ;

  EQUATION: PT = PS * 1000/21.2 * TTIN *
            T;
END SYSTEM
```

THIS IS THE FORTRAN THAT WOULD BE CURRENTLY PRODUCED BY THE CONFIN FILE LISTED ON THE PREVIOUS PAGE:

```
      PROGRAM MAIN
      COMMON / ARGLST / ARGLIST(  2, 14)
      COMMON / ARGVAR / IPTLIST(  2, 14)
      COMMON / MODULE / MMNEM(  2),MODNM(  2)
      COMMON / COMSIZ / ISIZE(  2)
      CHARACTER*8 A , ARGLIST , ARRAY(50) , MODNM , MMNEM
      INTEGER IPTLIST , IARRY(50)
C
C  CHECKOUT VERIFICATION FACILITY
C  ENTRY CALL

      DO 100 I = 1,  2
      IRC = 0
      DO 50  J = 1, ISIZE(I)
   50 ARRAY(J) = ARGLIST(I,J)
      IF(I.EQ.  2) IRC = -1
  100 CALL VCFIN ( MODNM(I), MMNEM(I), ARRAY, ISIZE(I), IRC )
C
C  CALL READER ROUTINE TO BRING IN VALUES
C
      CALL VCFREAD
C
C  CALL MAIN SIMULATION SUBROUTINE
C
      CALL ROCETS
C
C CALL ROUTINE TO WRITE OUTPUT DATA TO VCF FILE
C
      DO 1000 I = 1,  2
      DO 500  J = 1, ISIZE(I)
      IARRY(J) = IPTLIST(I,J)
  500 ARRAY(J) = ARGLIST(I,J)
 1000 CALL VCFWRT ( MODNM(I), MMNEM(I), ARRAY, IARRY, ISIZE(I) )
C
C INVOKE VCF OUT PANEL SYSTEM
C
      CALL VCFOUT
      END
      SUBROUTINE ROCETS
C
      COMMON / GLOBVAR /
     *      IPRPL   ,   IUPDAT  ,   MODN1   ,   NOD1IN  ,   NOD2VOL1,
     *      AREA1   ,   CF1     ,   PTIN    ,   PTVOL1  ,   RHOIN   ,
     *      RHOVOL1 ,   XLEN1   ,   W1      ,   DWDT1   ,   MODN2   ,
     *      NOD1IN2 ,   NOD2VOL2,   CF2     ,   PTIN2   ,   PTVOL2  ,
     *      RHOIN2  ,   W2
      CHARACTER*4
     *      MODN1       ,NOD1IN     ,NOD2VOL1   ,NOD2VOL2   ,CF2
     *      ,PTIN2
C
C  CALL UNIT DEFINITION ROUTINE
C
      CALL UNITOO('ENGLISH ')
C
C  BEGIN PROPERTY CALLS
C
      CALL PROPOO ('HYDROGEN',14,RHOVOL1 ,UTVOL1  ,PTVOL1  ,
     *             DUMYVOL1,DUMYVOL1,TEMP)
```

```
      CALL PROP00 ('OXYGEN  ',14,RHOVOL2 ,UTVOL2  ,PTVOL2  ,
     *             DUMYVOL2,DUMYVOL2,TEMP)
C
C  BEGIN MODULE CALLS
C
      PT       = PS * 1000/21.2 * TTIN *
      CALL PIPE00   ( IPRPL    ,IUPDAT   ,MODN1    ,NOD1IN   ,
     *      NOD2VOL1 ,AREA1    ,CF1      ,PTIN     ,PTVOL1   ,
     *      RHOIN    ,RHOVOL1  ,XLEN1    ,W1       ,DWDT1    )
C
      RETURN
      END
      SUBROUTINE VCFREAD
C
      COMMON / GLOBVAR /
     *      IPRPL    ,  IUPDAT  ,  MODN1   ,  NOD1IN  ,  NOD2VOL1,
     *      AREA1    ,  CF1     ,  PTIN    ,  PTVOL1  ,  RHOIN   ,
     *      RHOVOL1  ,  XLEN1   ,  W1      ,  DWDT1   ,  MODN2   ,
     *      NOD1IN2  ,  NOD2VOL2,  CF2     ,  PTIN2   ,  PTVOL2  ,
     *      RHOIN2   ,  W2
      CHARACTER*4
     *      MODN1       ,NOD1IN     ,NOD2VOL1   ,NOD2VOL2   ,CF2
     *      ,PTIN2
      NAMELIST / ROCK /
     *      IPRPL    ,  IUPDAT  ,  MODN1   ,  NOD1IN  ,  NOD2VOL1,
     *      AREA1    ,  CF1     ,  PTIN    ,  PTVOL1  ,  RHOIN   ,
     *      RHOVOL1  ,  XLEN1   ,  W1      ,  DWDT1   ,  MODN2   ,
     *      NOD1IN2  ,  NOD2VOL2,  CF2     ,  PTIN2   ,  PTVOL2  ,
     *      RHOIN2   ,  W2
C
      DO 100 I = 1,  2
  100 READ (15,ROCK)
C
      RETURN
      END
      SUBROUTINE VCFWRT (MODNM,MMNEM,ARRAY,IARRY,ISIZE)
C
      COMMON / GLOBVAR /
     *      IPRPL    ,  IUPDAT  ,  MODN1   ,  NOD1IN  ,  NOD2VOL1,
     *      AREA1    ,  CF1     ,  PTIN    ,  PTVOL1  ,  RHOIN   ,
     *      RHOVOL1  ,  XLEN1   ,  W1      ,  DWDT1   ,  MODN2   ,
     *      NOD1IN2  ,  NOD2VOL2,  CF2     ,  PTIN2   ,  PTVOL2  ,
     *      RHOIN2   ,  W2
      CHARACTER*4
     *      MODN1       ,NOD1IN     ,NOD2VOL1   ,NOD2VOL2   ,CF2
     *      ,PTIN2
C
      COMMON / VARTYP / TYPE(  22)
C
      CHARACTER*8 MODNM,MMNEM,ARRAY(*)
C
      REAL*4 VAL(  22)
      INTEGER TYPE
C
      INTEGER IARRY(*)
C
      EQUIVALENCE (IPRPL    ,VAL(1))
      WRITE(16,10) MODNM,MMNEM
   10 FORMAT('MODULE:',A8,' MNEMONIC:',A8)
      DO 50 I = 1,ISIZE
      IF (TYPE(IARRY(I)).EQ.3) GOTO 20
      IF (TYPE(IARRY(I)).EQ.2) GOTO 30
      WRITE(16,100) ARRAY(I),VAL(IARRY(I))
      GOTO 50
```

```
      20   WRITE(16,120) ARRAY(I),VAL(IARRY(I))
           GOTO 50
      30   WRITE(16,140) ARRAY(I),VAL(IARRY(I))
      50   CONTINUE
      100  FORMAT(3X,A8,'=',1X,F15.5)
      120  FORMAT(3X,A8,'=',1X,A8)
      140  FORMAT(3X,A8,'=',1X,I4)
C
           RETURN
           END
           BLOCK DATA ONE
           COMMON / GLOBNAM / A(   22)
           COMMON / VARTYP / TYPE(   22)
           COMMON / ARGLST / ARGLIST(  2, 14)
           COMMON / ARGVAR / IPTLIST(  2, 14)
           COMMON / MODULE / MMNEM(  2),MODNM(  2)
           COMMON / COMSIZ / ISIZE(  2)
           CHARACTER*8 A , ARGLIST , MMNEM , MODNM
           INTEGER IPTLIST
C
           DATA A /
          *      'IPRPL   ', 'IUPDAT  ', 'MODN1   ', 'NOD1IN  ', 'NOD2VOL1',
          *      'AREA1   ', 'CF1     ', 'PTIN    ', 'PTVOL1  ', 'RHOIN   ',
          *      'RHOVOL1 ', 'XLEN1   ', 'W1      ', 'DWDT1   ', 'MODN2   ',
          *      'NOD1IN2 ', 'NOD2VOL2', 'CF2     ', 'PTIN2   ', 'PTVOL2  ',
          *      'RHOIN2  ', 'W2      '/
C
           DATA ARGLIST /
          *      'IPRPL   ', 'IPRPL   ', 'IUPDAT  ', 'IUPDAT  ', 'MODN1   ',
          *      'MODN2   ', 'NOD1IN  ', 'NOD1IN2 ', 'NOD2VOL1', 'NOD2VOL2',
          *      'AREA1   ', 'CF2     ', 'CF1     ', 'PTIN2   ', 'PTIN    ',
          *      'PTVOL2  ', 'PTVOL1  ', 'RHOIN2  ', 'RHOIN   ', 'W2      ',
          *      'RHOVOL1 ', '        ', 'XLEN1   ', '        ', 'W1      ',
          *      '        ', 'DWDT1   ', '        '/
C
           DATA IPTLIST /
          *        1,         1,         2,         2,         3,
          *       15,         4,        16,         5,        17,
          *        6,        18,         7,        19,         8,
          *       20,         9,        21,        10,        22,
          *       11,         0,        12,         0,        13,
          *        0,        14,         0/
C
           DATA MMNEM /
          *      '1       ', '2       '/
C
           DATA MODNM /
          *      'PIPE00  ', 'PIPE01  '/
C
           DATA ISIZE /
          *       14,        10/
C
           DATA TYPE /
          *        2,         2,         3,         3,         3,
          *        1,         1,         1,         1,         1,
          *        1,         1,         1,         1,         2,
          *        2,         3,         3,         3,         1,
          *        1,         1/
           END
```

Simple TTBE Simulation

In order to complete system testing and module verification, a simple model of the Technology Test Bed Engine (TTBE) was completed. The simple TTBE demonstrates the ability to model a complete engine using the ROCETS methodology of combining generic engineering modules with an implicit solver. Figure 1 shows a schematic of the simple TTBE along with the 42 specific stations in the simulation. By using generic code, only the following 13 componet modules are required by the simulation:

1. INJT00 - Main Injector
2. MCHB00 - Main Chamber
3. MIXR00 - Flow Mixer
4. NOZL00 - Nozzle Thrust Calculations
5. PBRN00 - Preburner
6. PIPE00 - Incompressible flow with inertia
7. PIPE01 - Incompresisble flow without inertia
8. PIPE02 - Compressible flow without inertia
9. PUMP00 - Polytropic Pump
10. ROTR00 - Rotor Torque Balance/Speed Derivative
11. SPLT00 - Flow Splitter
12. TURB00 - Turbine
13. VOLM00 - Volume

The modules described above were hand processed into the simple TTBE simulation along with required property relationships and numerical utilities. There are 55 state variables and 2 algebraic loops required in the simulation as shown in Table IV. State derivatives and outputs are calculated from model inputs and states. The simulation employs the multivariable Newton-Raphson solver, SMITE, to simultaneously drive all state derivatives and algebraic loop errors to within a certain tolerance. SMITE operates on the matrix equation:

$$\Delta Y = J\Delta X$$

Where ΔY is the amount that the errors need to change to be zero and ΔX is the required change in the state or algebraic loop parameter. J, the solver Jocabian, is the matrix of partial derivatives for the states and algebraic loops. The J matrix is evaluated by perturbing each state and balance parameter. Then the J matrix is inverted and the new X values are calculated until the Y variables are within tolerance.

Using initial guesses from data of the Digital Transient Model (DTM) run at 100% RPL, SMITE successfully drives all TTBE model state derivatives and algebraic loop parameters to within a specific tolerance. This demonstrates the capability

of the ROCETS system to converge a rocket simulation to a steady-state point without running a transient.

Transient capability was demonstrated by running the simple TTBE simulation with small perturbations of valve areas about the 100% RPL point. Figures 2-3 show the results of stepping the oxidizer preburner valve open then closed. Figures 4-5 show the results for the same type of movement of the fuel preburner valve.

Additional algebraic loops (balances) were placed in the model to set preburner valve coefficients at points other than 100% RPL. The fuel preburner valve coefficient was iterated until chamber pressure (PTMCHB) was equal to the request:

$$PTMCHB_{Request} = PTMCHB_{100\ RPL}\ X\ RPL$$

The LOX preburner valve coefficient was iterated until chamber oxidizer fraction (OFRMCHB) was equal to a constant value of 0.865. (This is equivalent to a mixture ratio of 6.407). Figure 6 shows main chamber pressure and oxidizer fraction as a function of RPL. A series of steady-state points between 60% and 119% RPL were then run with the solver iterating on valve coefficients until chamber pressure and LOX fraction were satisified. This demonstrates the ability of the model to use the solver as a means to set a model parameter based on an input constraint. Output of the run gives a reference steady-state operating characteristic for the model and provides data for SMITE guess curves. Figure 7 shows fuel and LOX preburner valve coefficients and Figure 8 shows turbine speeds as a function of RPL.

The data shown in Figure 7 for valve coefficients as a function of RPL was tabularized and used to construct and open-loop control with RPL request input and valve data area requests calculated from the table. The valve requests were put through a first order lag to simulate actuator dynamics. Gross throttle transients were run by inputting an RPL request as a function of time and using the open-loop control to provide valve areas. Figures 9-13 show results of a run from 100% to 65% RPL decel and figures 14-18 show a run from 65% to 109% RPL.

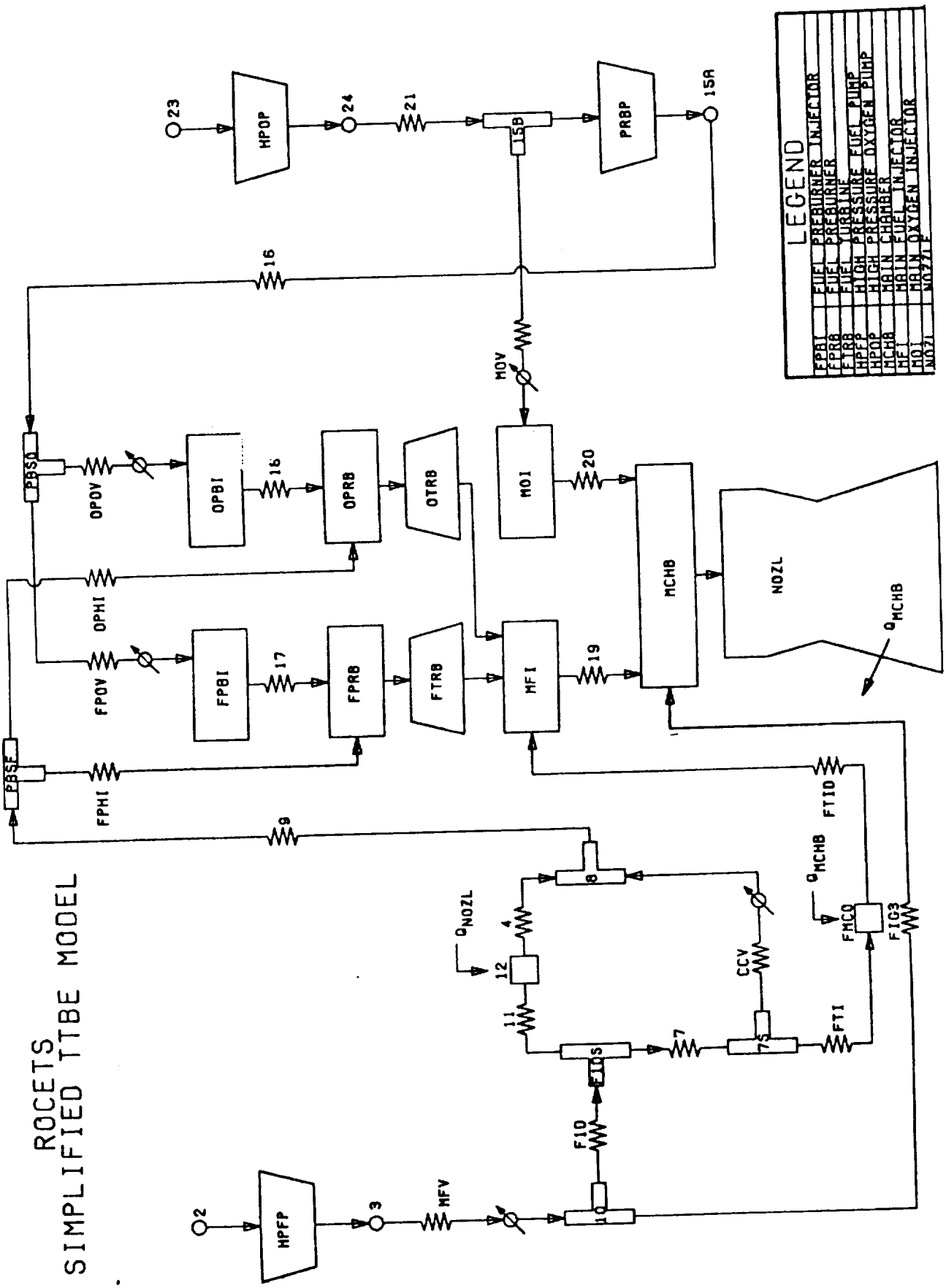

ROCETS
SIMPLIFIED TTBE MODEL
LEGEND
FPBI FUEL PREBURNER INJECTOR
FPRB FUEL PREBURNER
FTRB FUEL TURBINE
HPFP HIGH PRESSURE FUEL PUMP
HPOP HIGH PRESSURE OXYGEN PUMP
MCHB MAIN CHAMBER
MFI MAIN FUEL INJECTOR
MOI MAIN OXYGEN INJECTOR
NOZL NOZZLE
HPFP
HPOP
PRBP
OPBI
OPRB
OTRB
FPBI
FPRB
FTRB
MFI
MOI
MCHB
NOZL
MFV
MOV
OPOV
FPOV
OPHI
FPHI
FTID
FTI
FIG3
FMCO
CCV
FIOS
FIO
PBSO
PBSF
Q_{NOZL}
Q_{MCHB}

Figure 1

TABLE IV – Simple TTBE Simulation States and Algebraic Loops

STATE	DESCRIPTION
OFRFPRB	Fuel Preburner Oxidizer Fraction
OFRMCHB	Main Chamber Oxidizer Fraction
OFRMFI	Main Fuel Injector Oxidizer Fraction
OFROPRB	Oxidizer Preburner Oxidizer Fraction
RHOFMCO	Volume FMCO Density
RHOFPBI	Fuel Preburner Injector Density
RHOFPRB	Oxidizer Preburner Injector Density
RHOF10S	Volume F10S Density
RHOMCHB	Main Chamber Density
RHOMFI	Main Fuel Injector Density
RHOMOI	Main Oxidizer Injector Density
RHOOPBI	Oxidizer Preburner Injector Density
RHOOPRB	Oxidizer Preburner Density
RHOPBSF	Preburner Fuel Splitter Density
RHOPBSO	Preburner Oxidizer Splitter Density
RHO10	Volume 10 Density
RHO12	Volume 12 Density
RHO15B	Volume 15B Density
RHO8	Volume 8 Density
SNF2	Fuel Turbomachinery Speed
SN02	Oxidizer Turbomachinery Speed
TTFPRB	Fuel Preburner Temperature
TTMCHB	Main Chamber Temperature
TTMFI	Main Fuel Injector Temperature
TTOPRB	Oxidizer Preburner Temperature
UTFMCO	Volume FMCO Internal Energy
UTFPBI	Fuel Preburner Injector Internal Energy
UTF10S	Voume F10S Internal Energy
UTMOI	Main Oxidizer Injector Internal Energy
UTOPBI	Oxidizer Preburner Injector Internal Energy
UTPBSF	Preburner Fuel Splitter Internal Energy
UTPBSO	Preburner Oxidizer Splitter Internal Energy
UT10	Volume 10 Internal Energy
UT12	Volume 12 Internal Energy
UT15B	Volume 15B Internal Energy
UT7S	Volume 7S Internal Energy
UT8	Volume 8 Internal Energy

STATE	DESCRIPTION
WCCV	Coolant Control Valve Flow
WFPHI	Fuel Preburner Fuel Flow
WFPOV	Fuel Preburner Oxidizer Flow
WFTI	Line FTI Flow
WFTID	Line FTID Flow
WF10	Line F10 Flow
WMFV	Main Fuel Valve Flow
WMOV	Main Oxidizer Valve Flow
WOPHI	Oxidizer Preburner Fuel Flow
WOPOV	Oxidizer Preburner Oxidizer Flow
W11	Line 11 Flow
W16	Line 16 Flow
W20	Line 20 Flow
W21	Line 21 Flow
W4	Line 4 Flow
W7	Line 7 Flow
W9	Line 9 Flow

ITERATION VARIABLE	DESCRIPTION
WFTRB	Fuel Turbine Flow - Iterated until equal to calculated value
WOTRB	LOX Turbine flow - Iterated until equal to calculated value

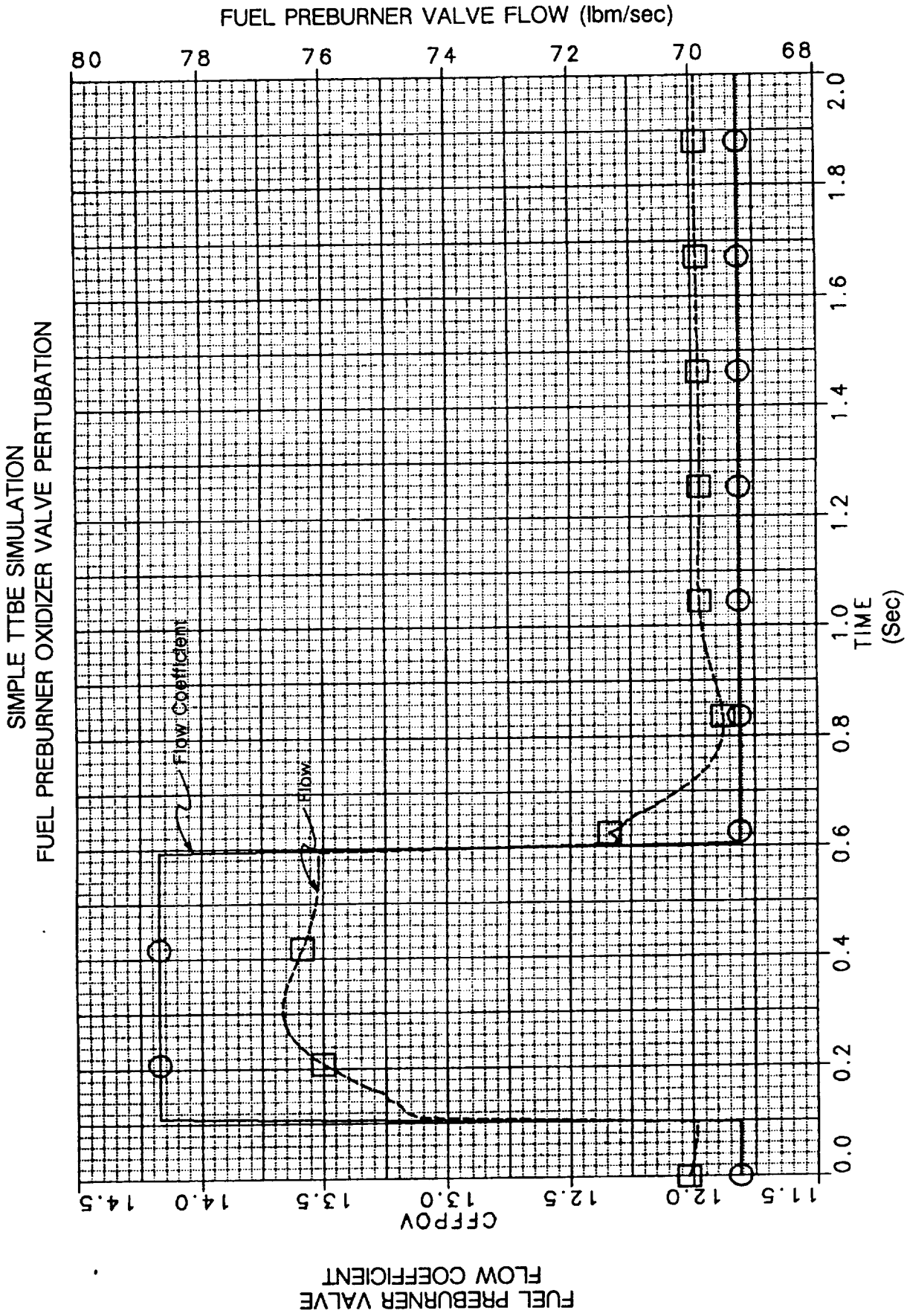


Figure 2 – Fuel Preburner Flow Coefficient and Flow vs Time

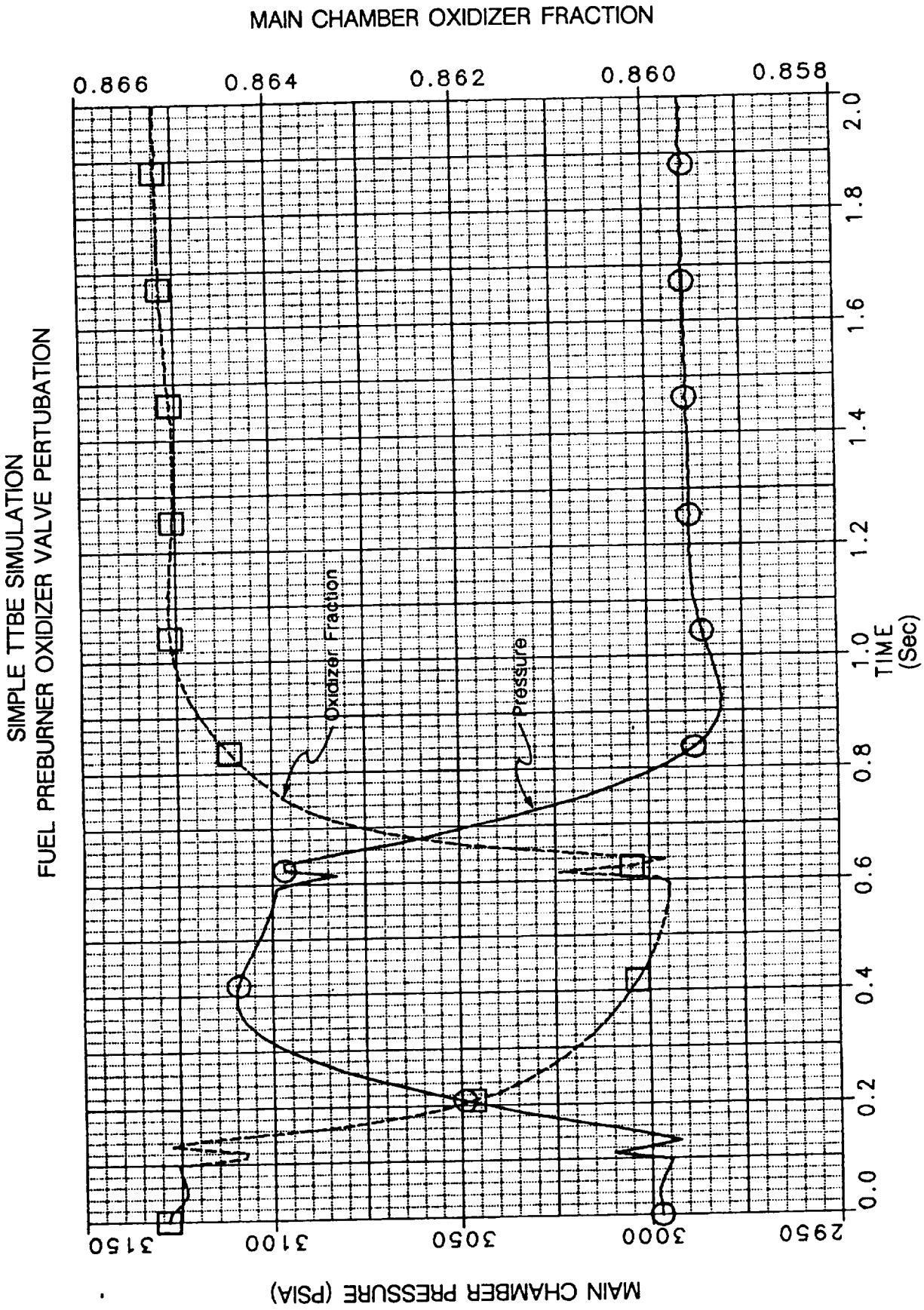


Figure 3 – Main Chamber Conditions as a Function of Time

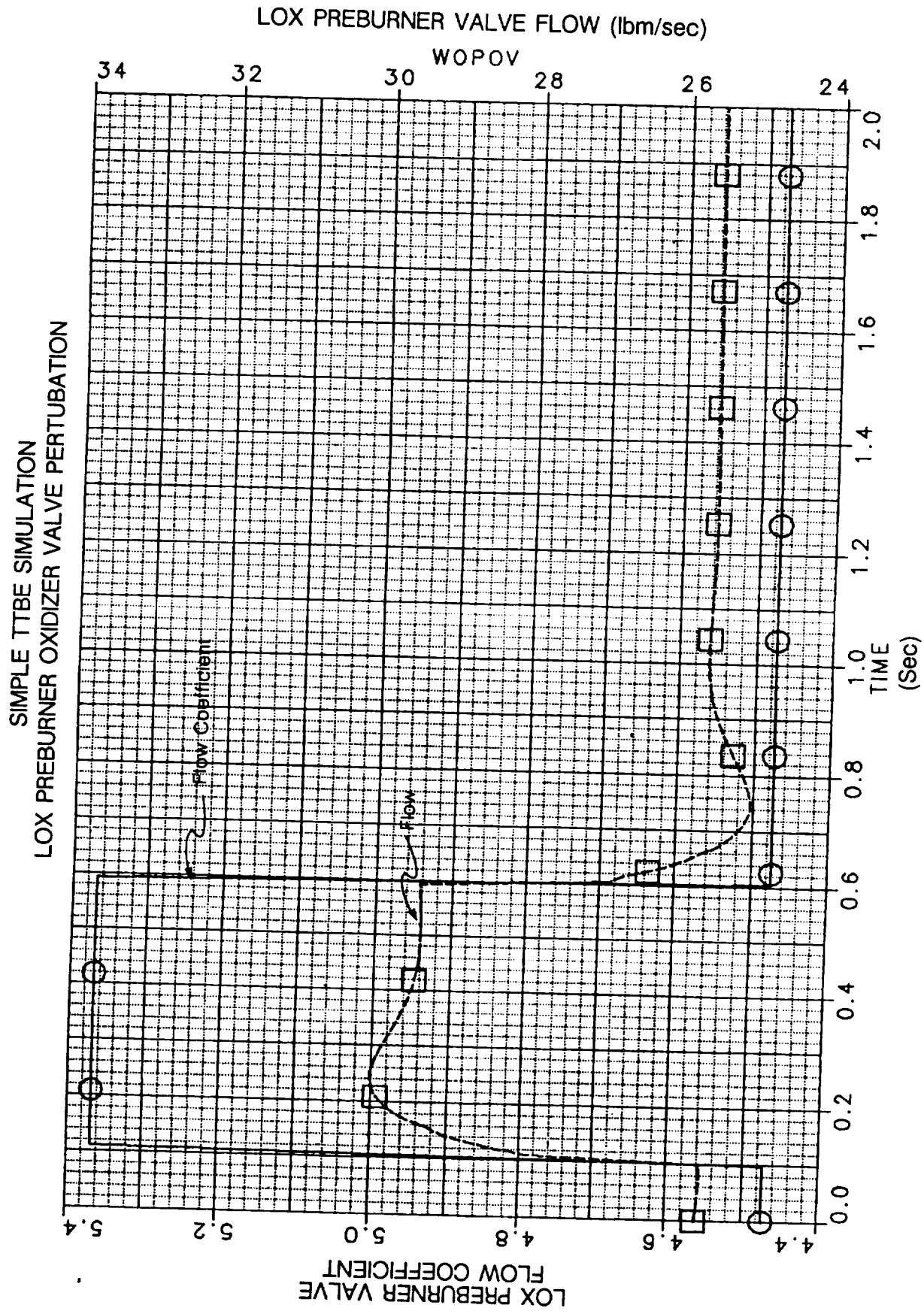


FIGURE 4 – Lox Preburner Flow Coefficient and Flow vs Time

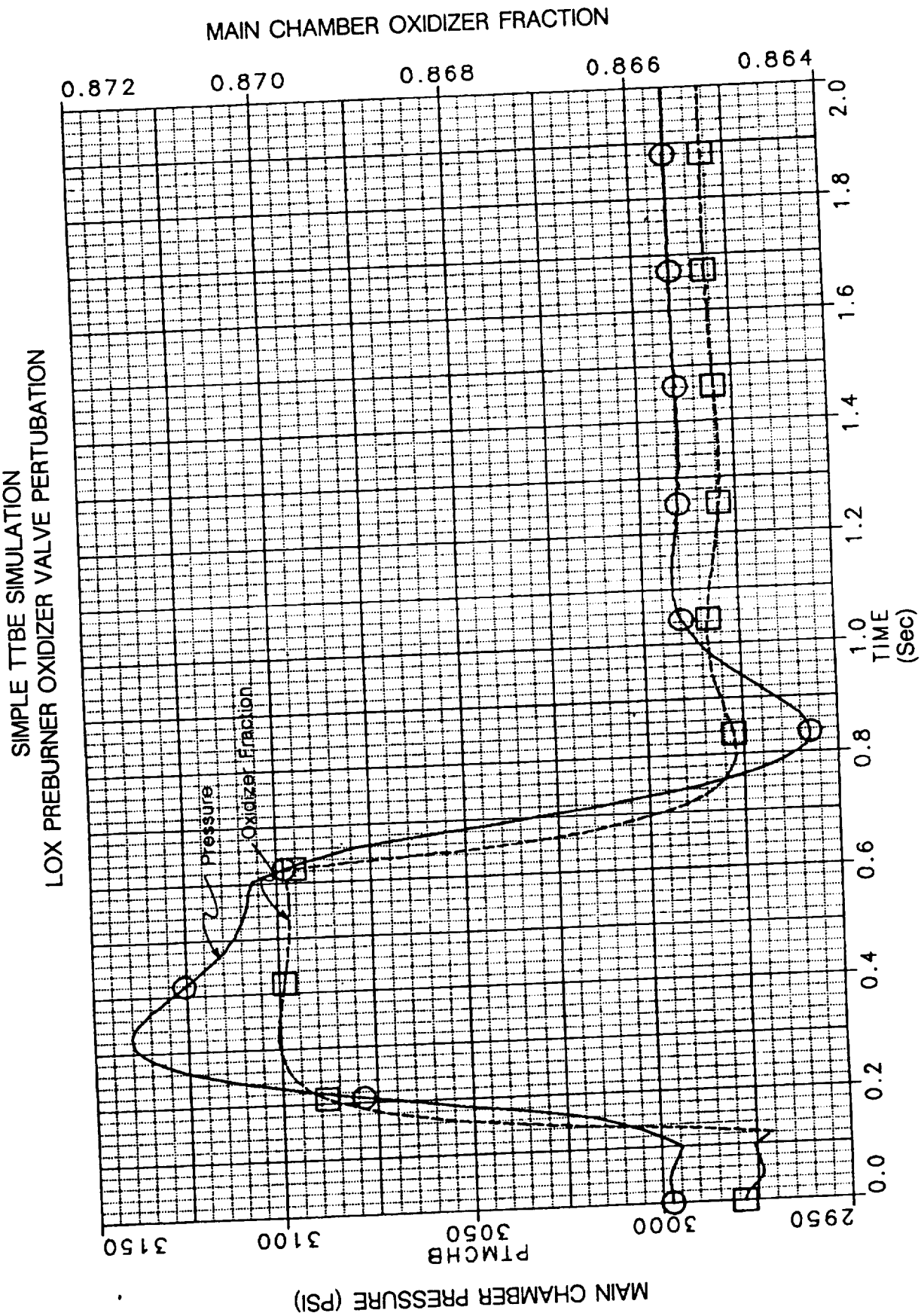


Figure 5 – Main Chamber Conditions as a Function of Time

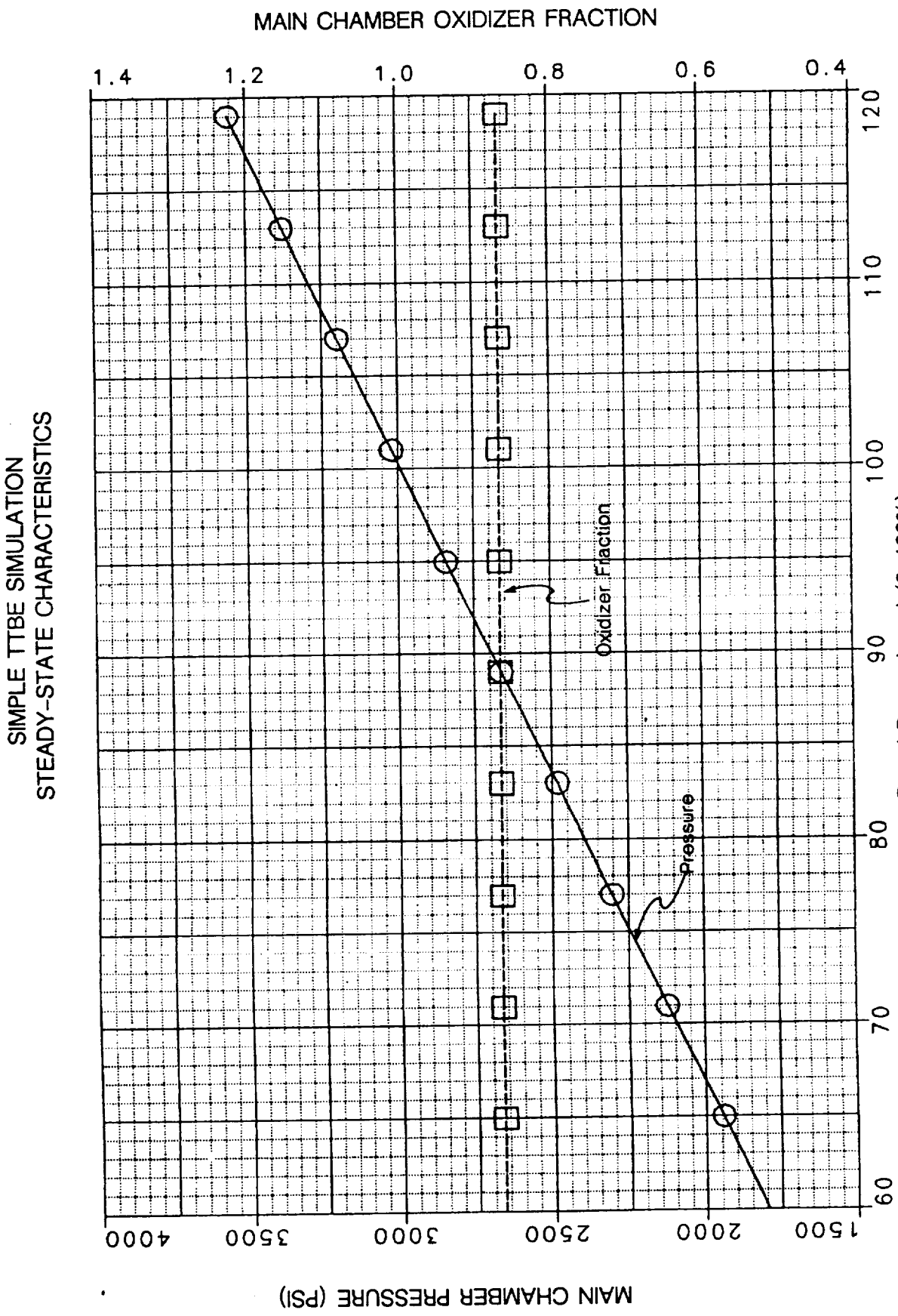


Figure 6 - Main Chamber Conditions as a Function of RPL

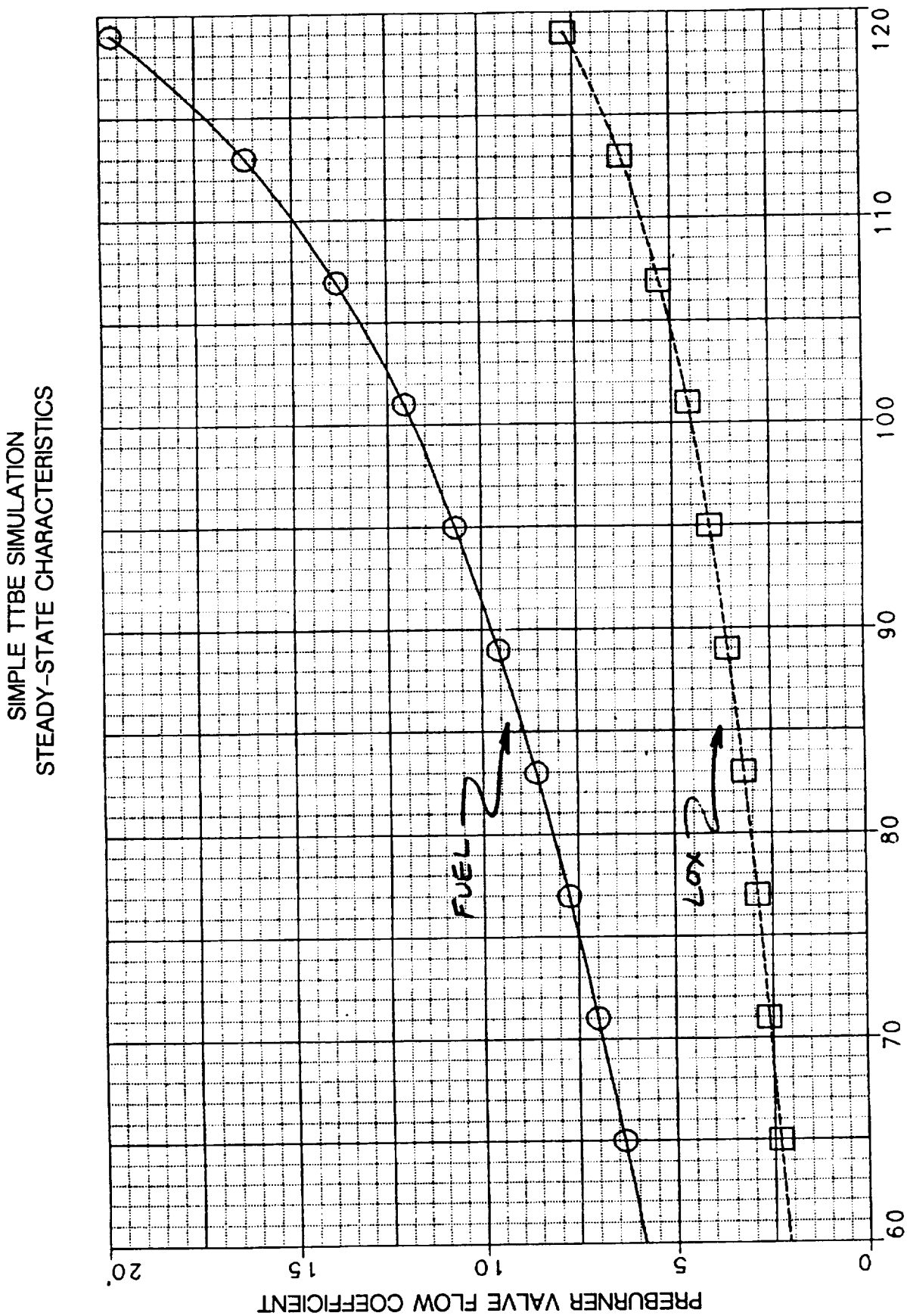


Figure 7 - Preburner Valve Coefficients as a Function of RPL

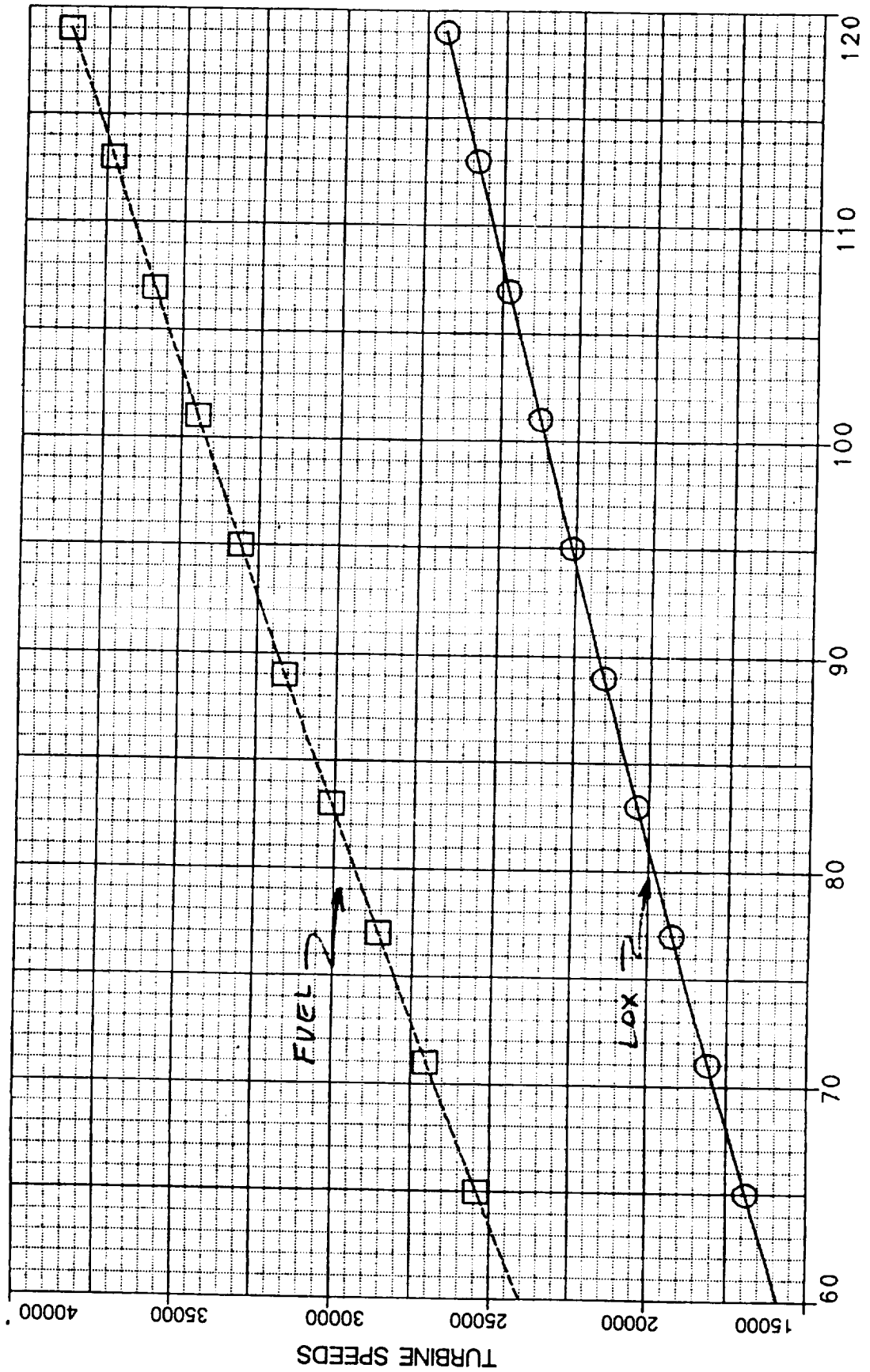


Figure 8 - Turbine Speeds as a Function of RPL

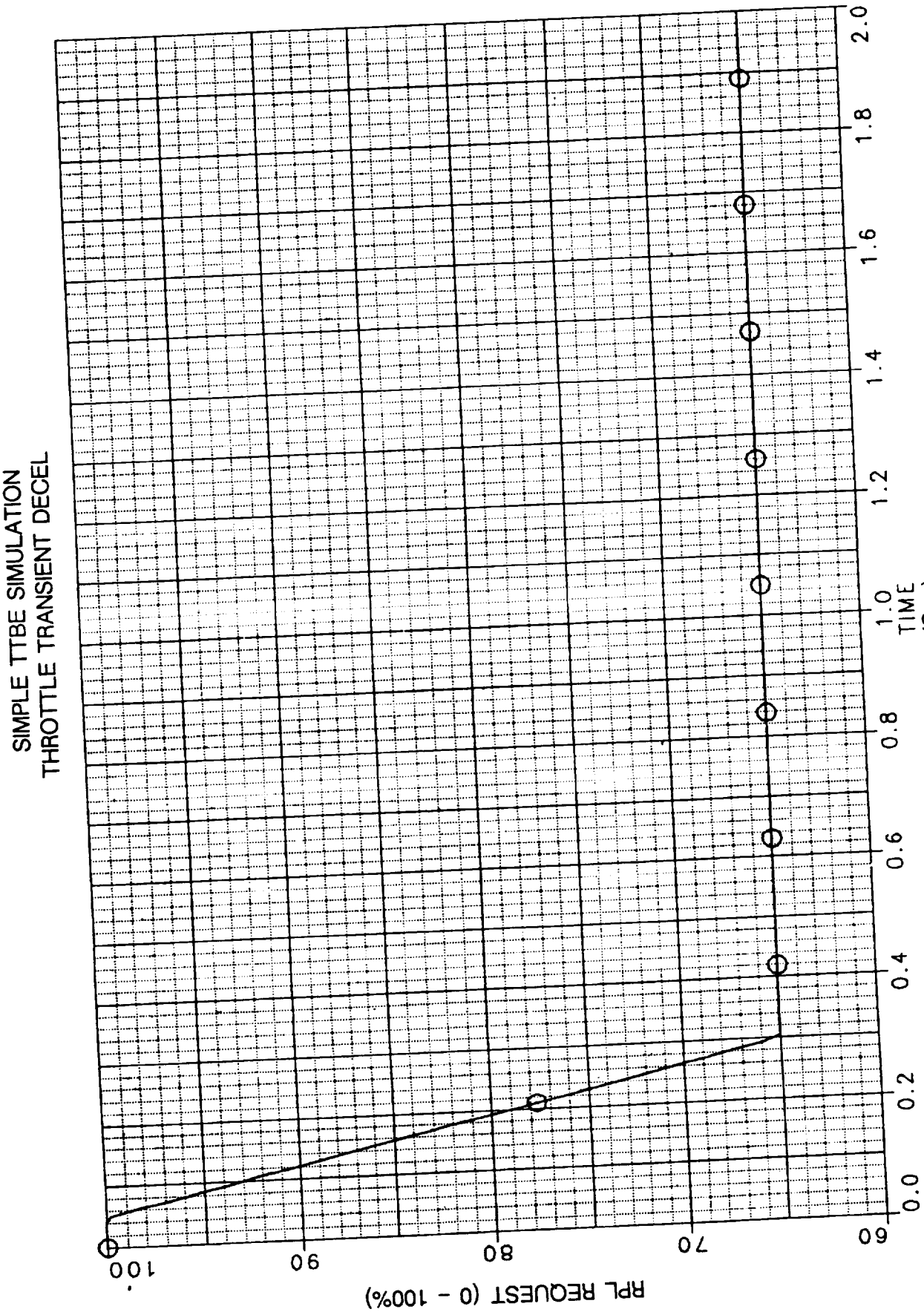


Figure 9 – RPL Request as a Function of Time

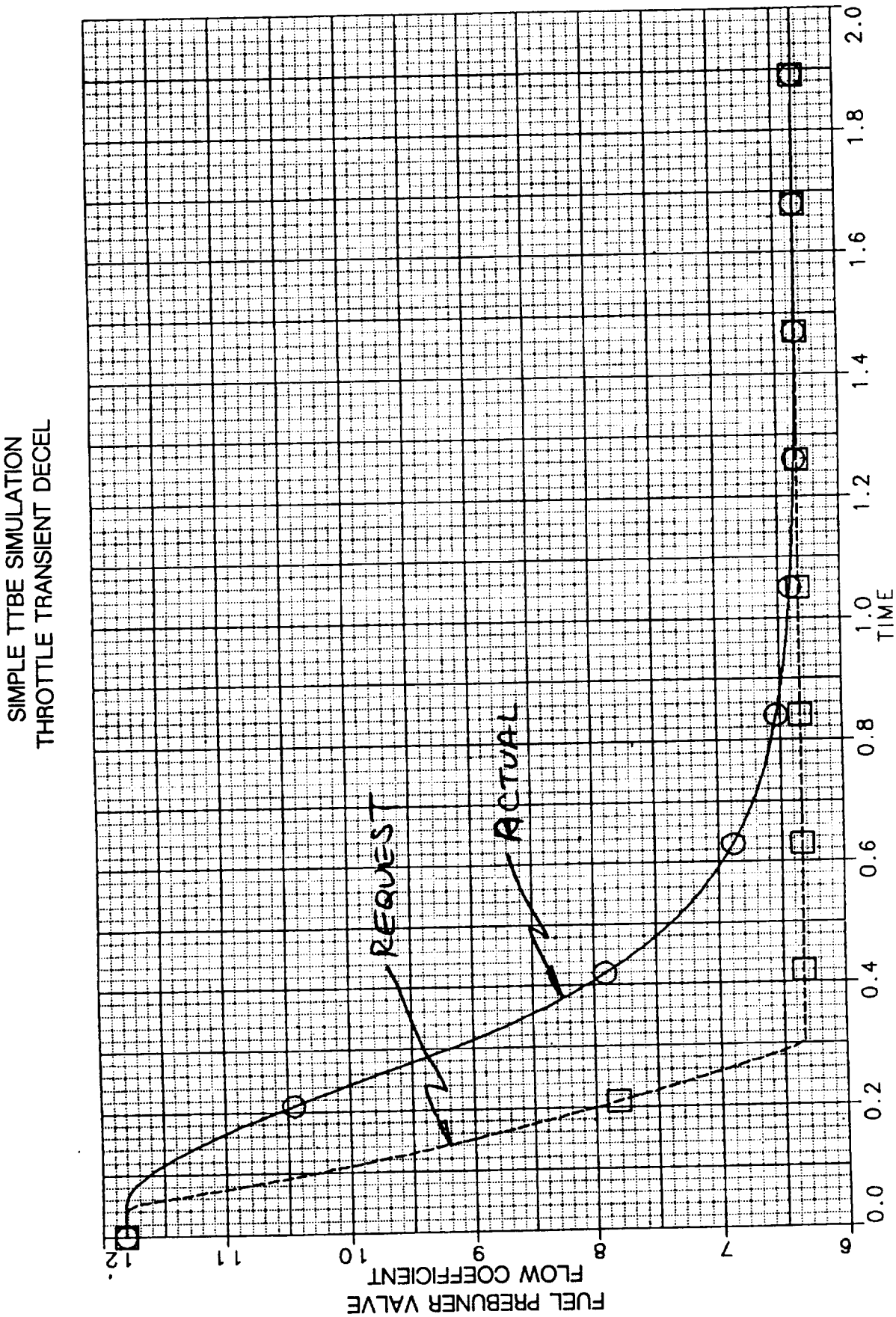


Figure 10 – Fuel Preburner Valve as a Function of Time

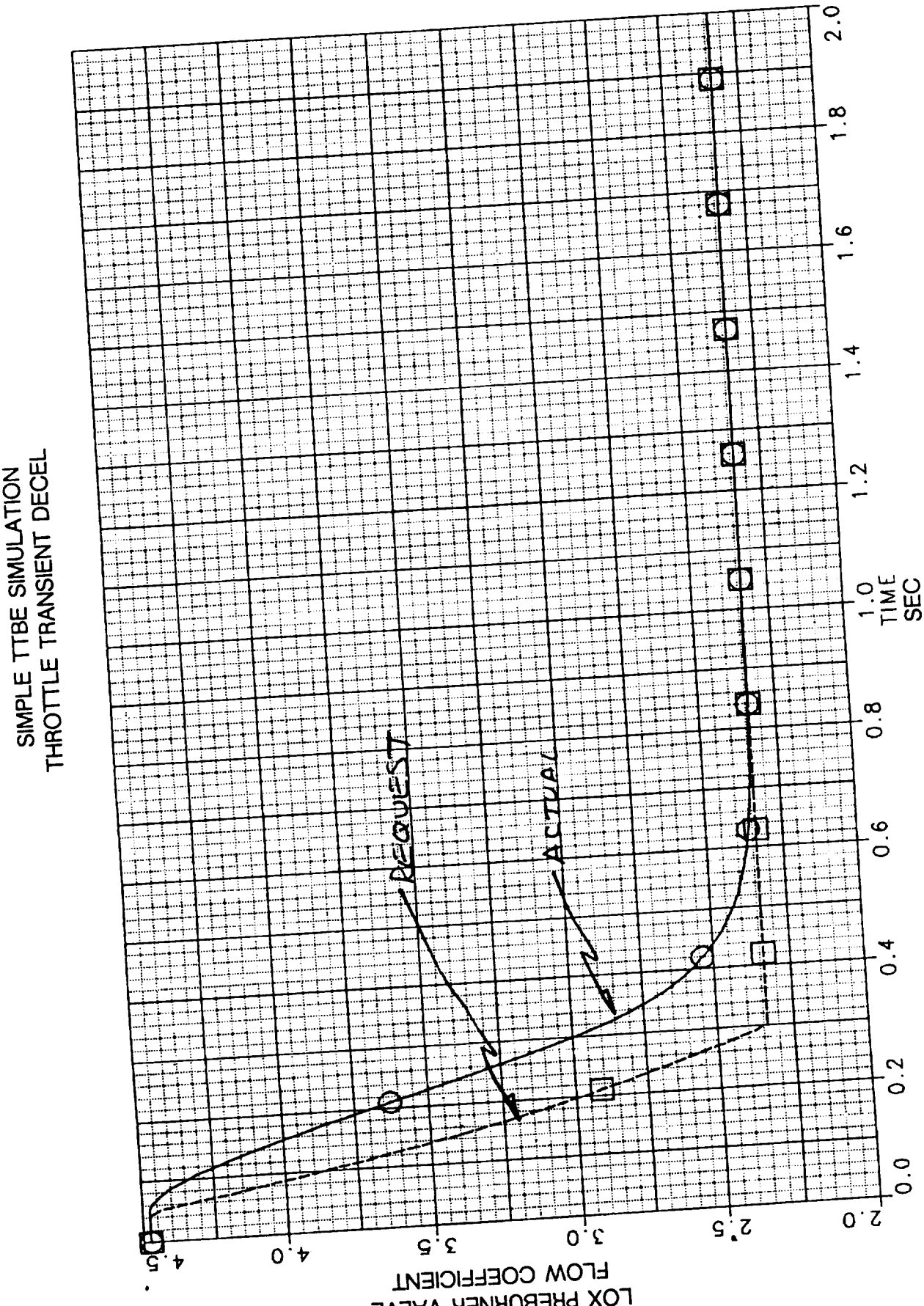


Figure 11 – Lox Preburner Valve as a Function of Time

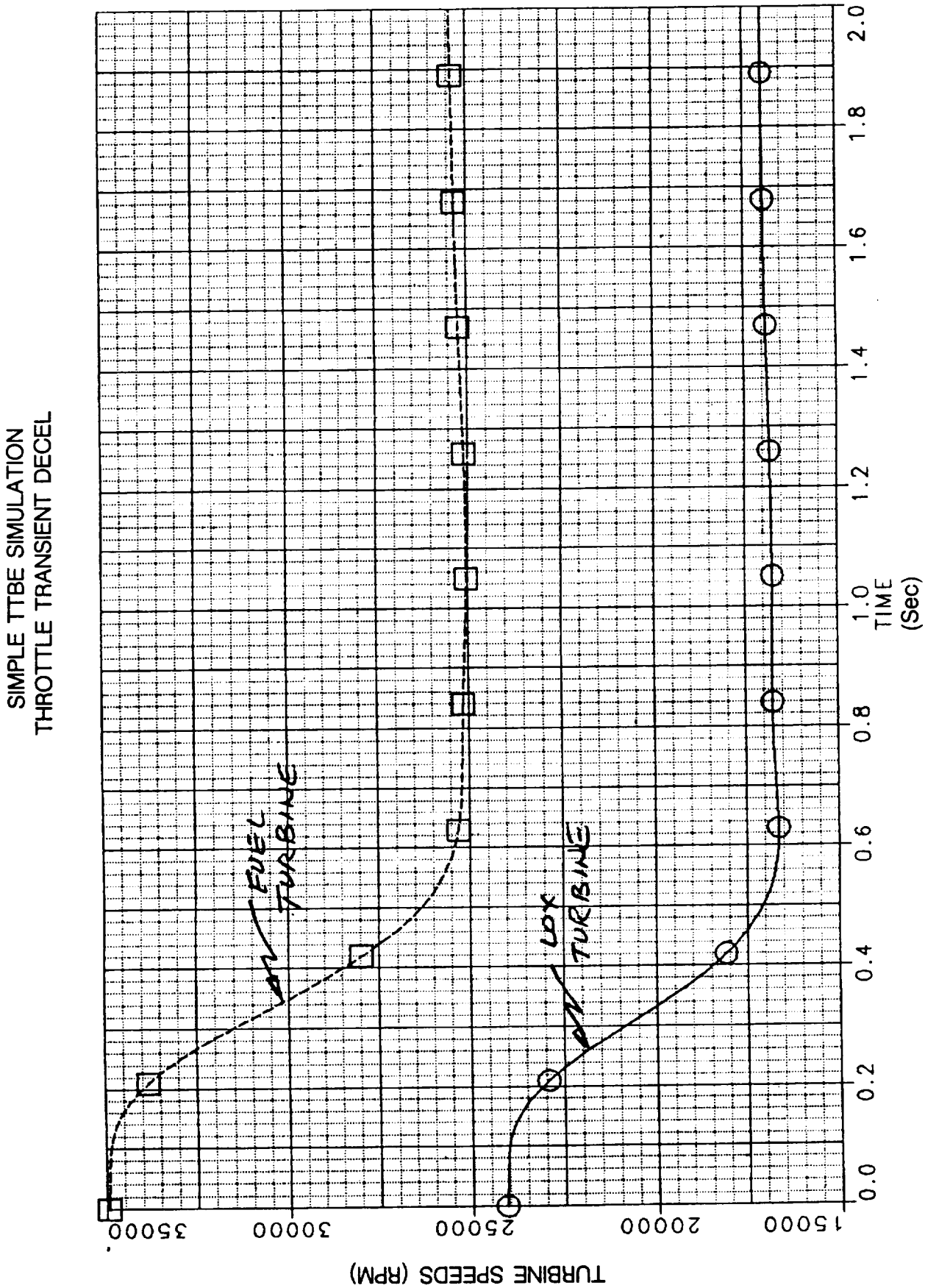


Figure 12 – Turbine Speeds as a Funciton of Time

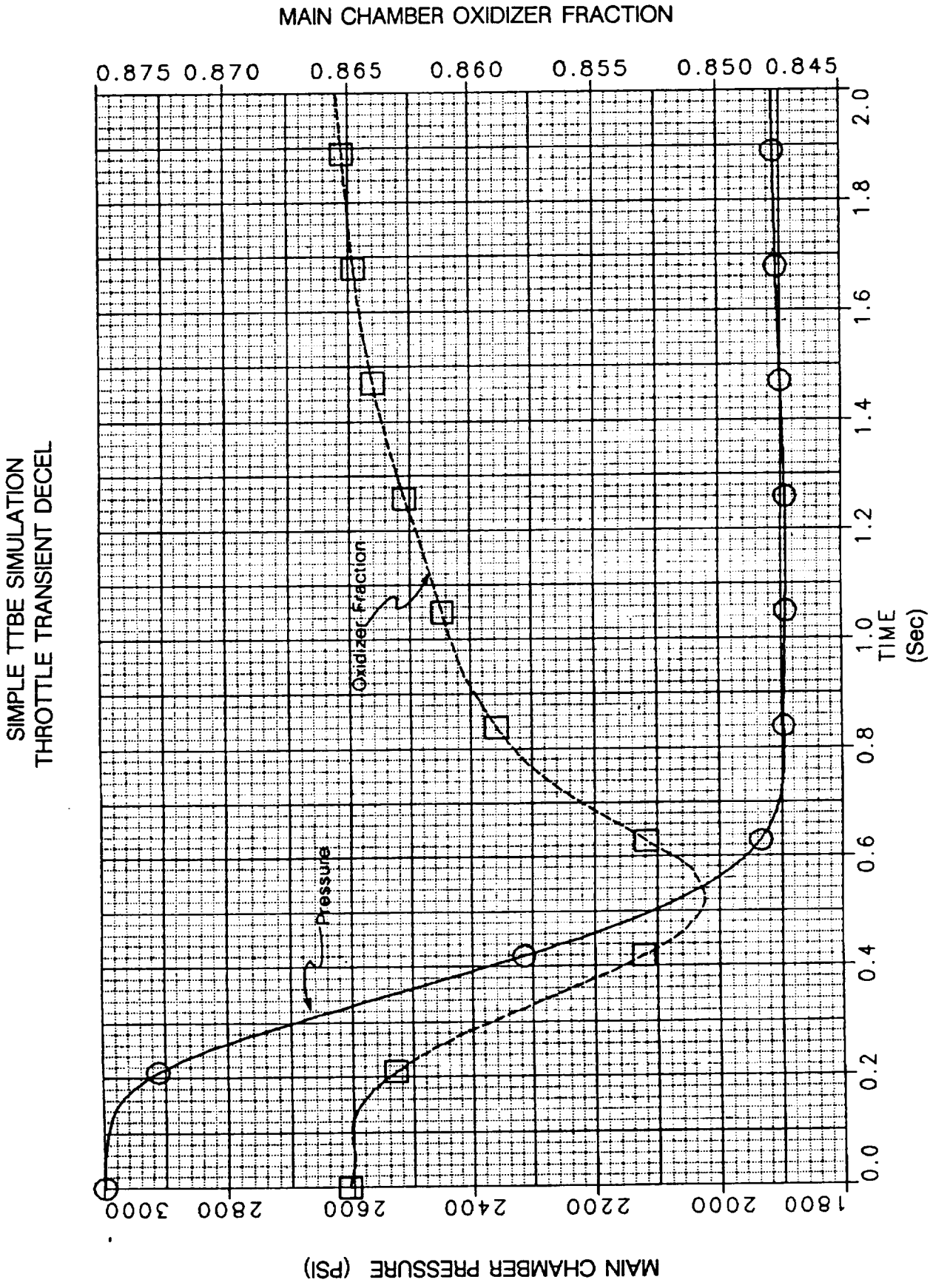


Figure 13 - Main Chamber Conditions as a Function of Time

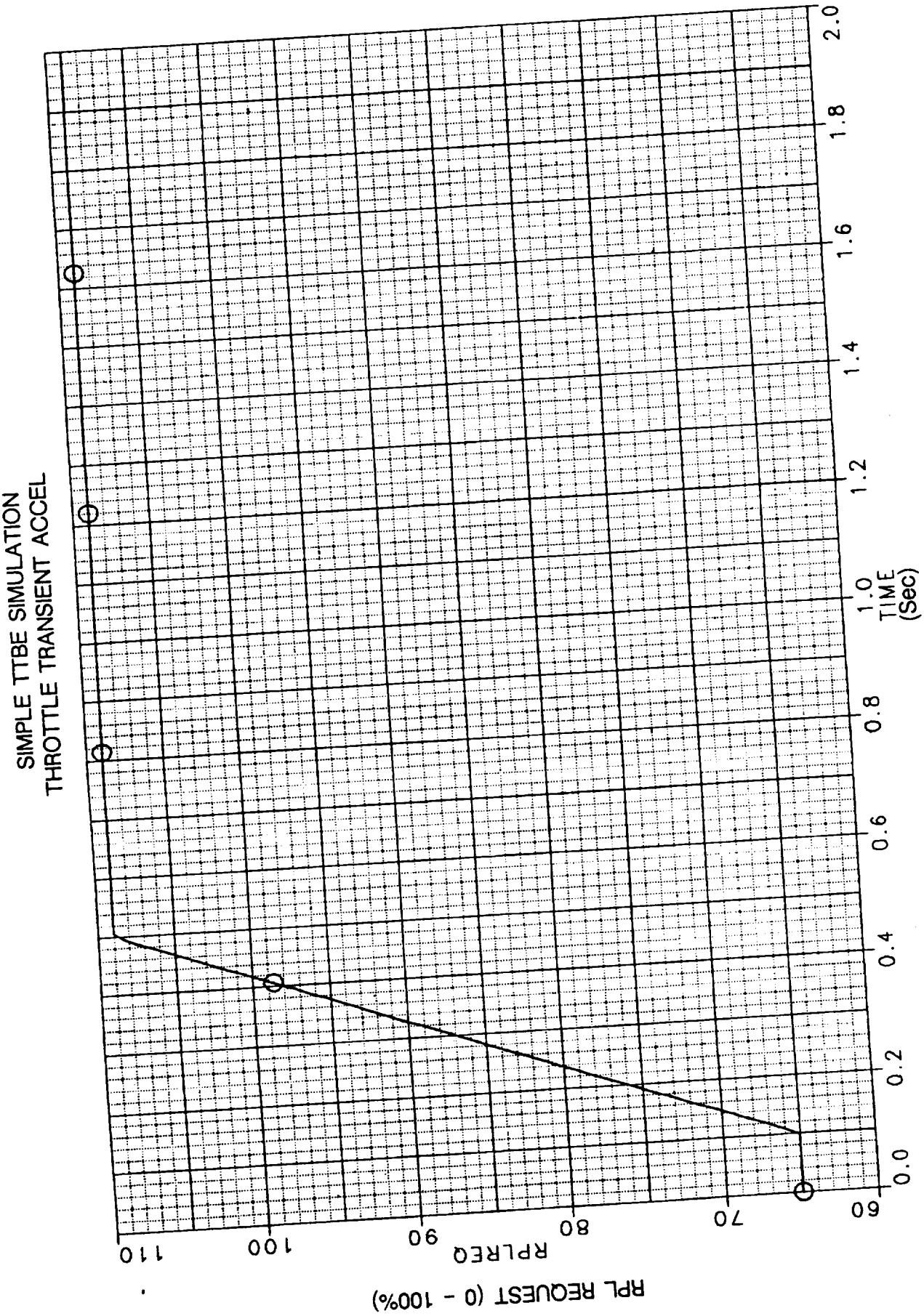


Figure 14 – RPL Request as a Function of Time

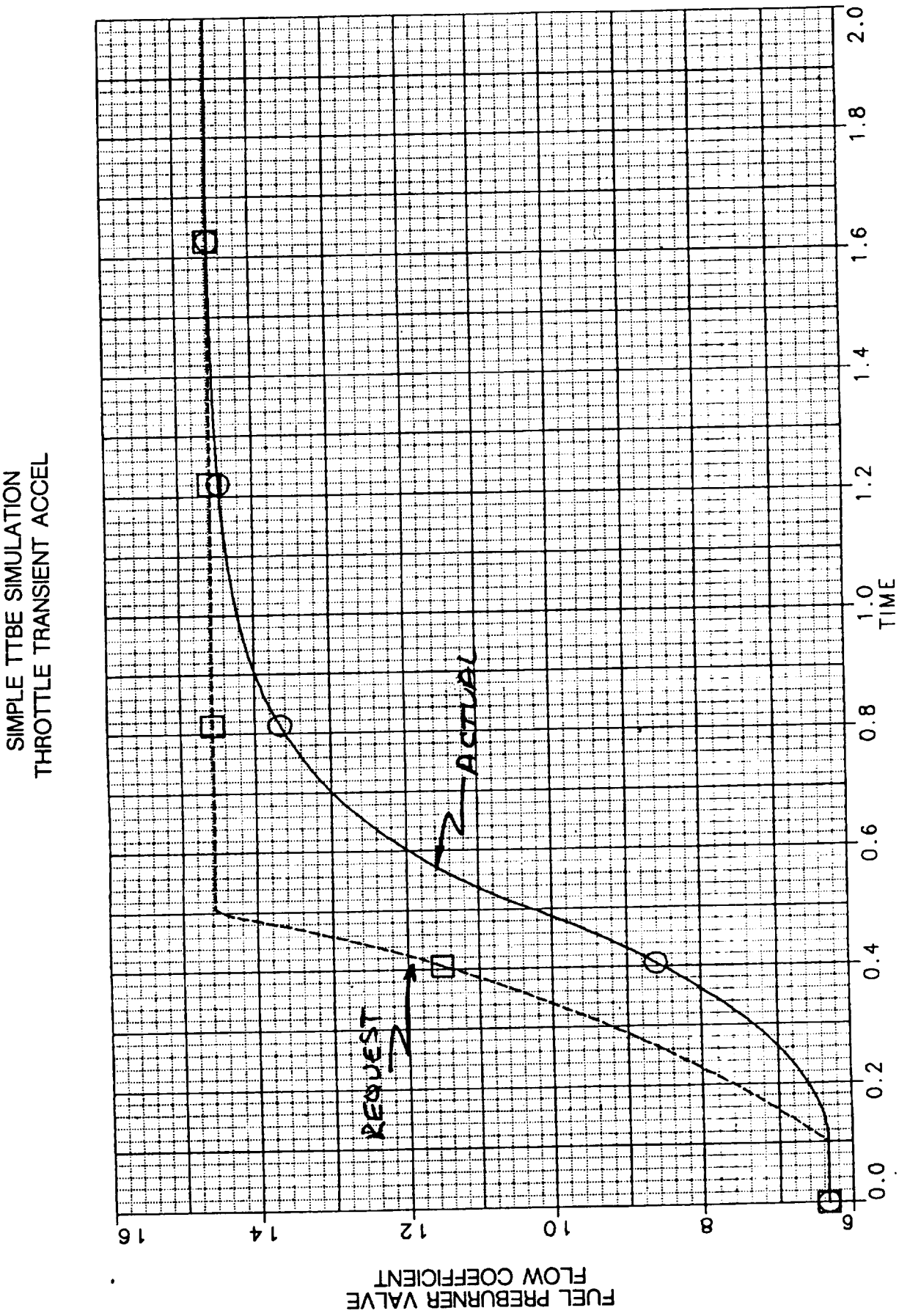


Figure 15 – Fuel Preburner Valve as a Function of Time

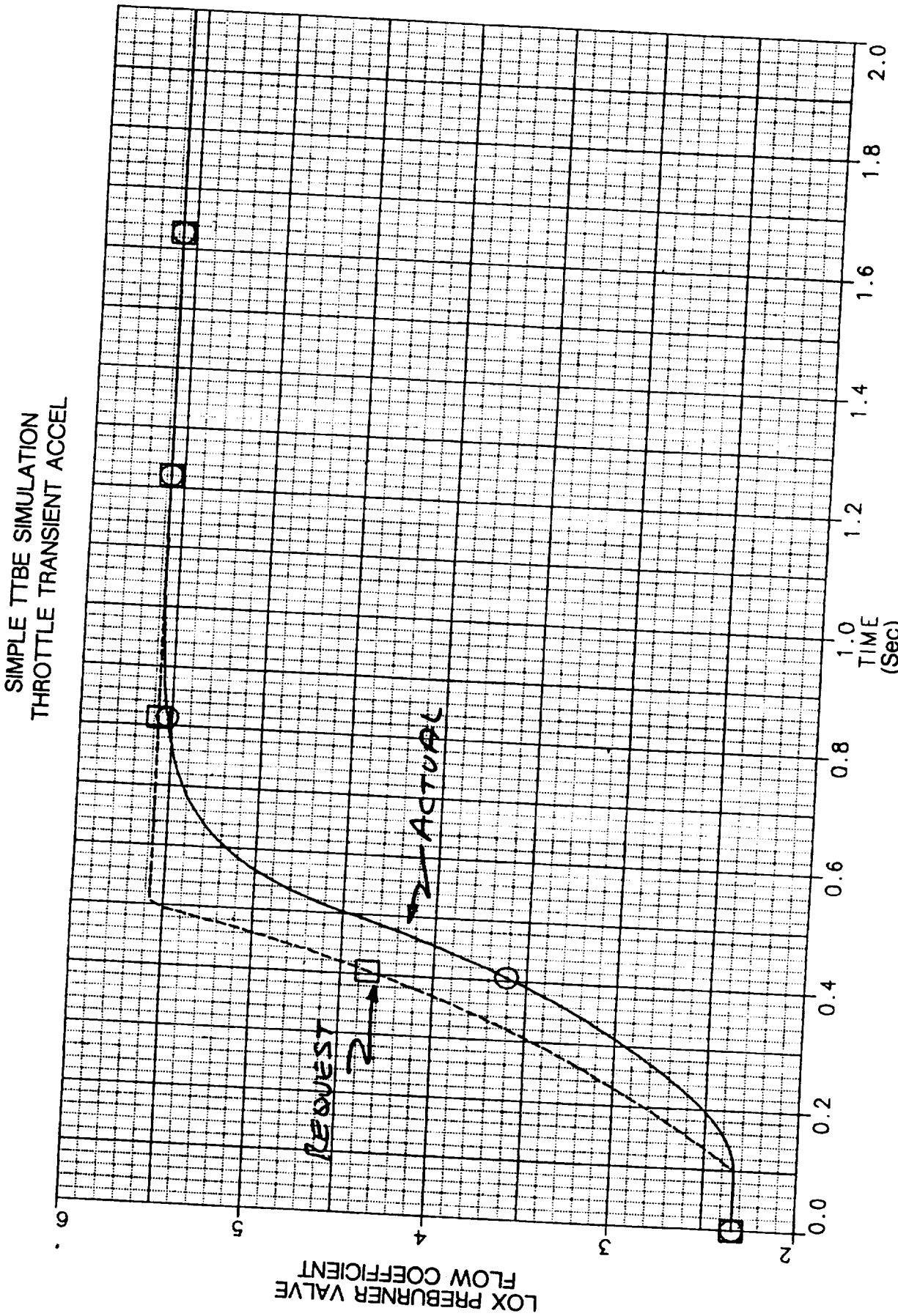


Figure 16 – Lox Preburner Valve as a Funciton of Time

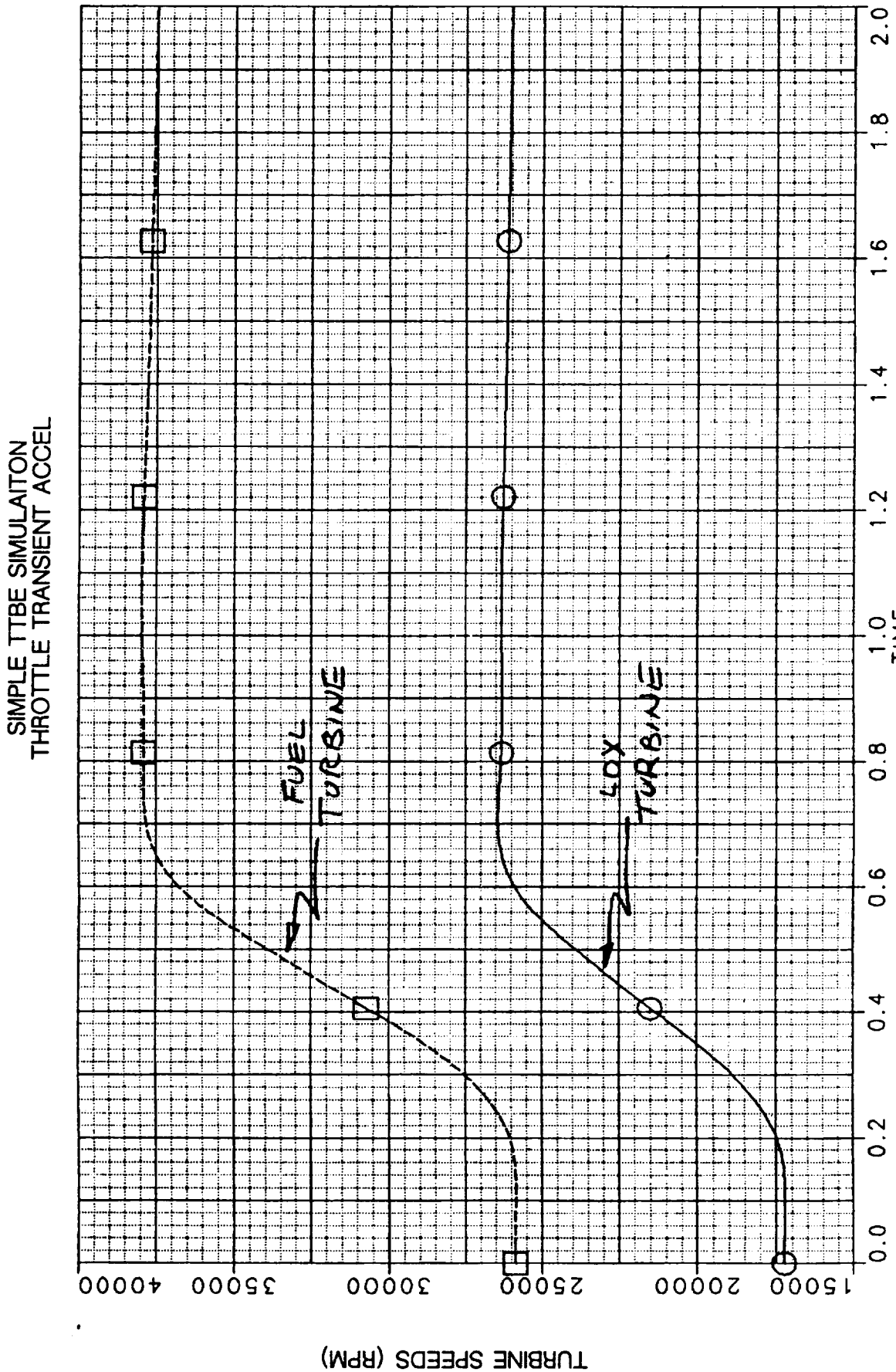


Figure 17 – Turbine Speeds as a Function of Time

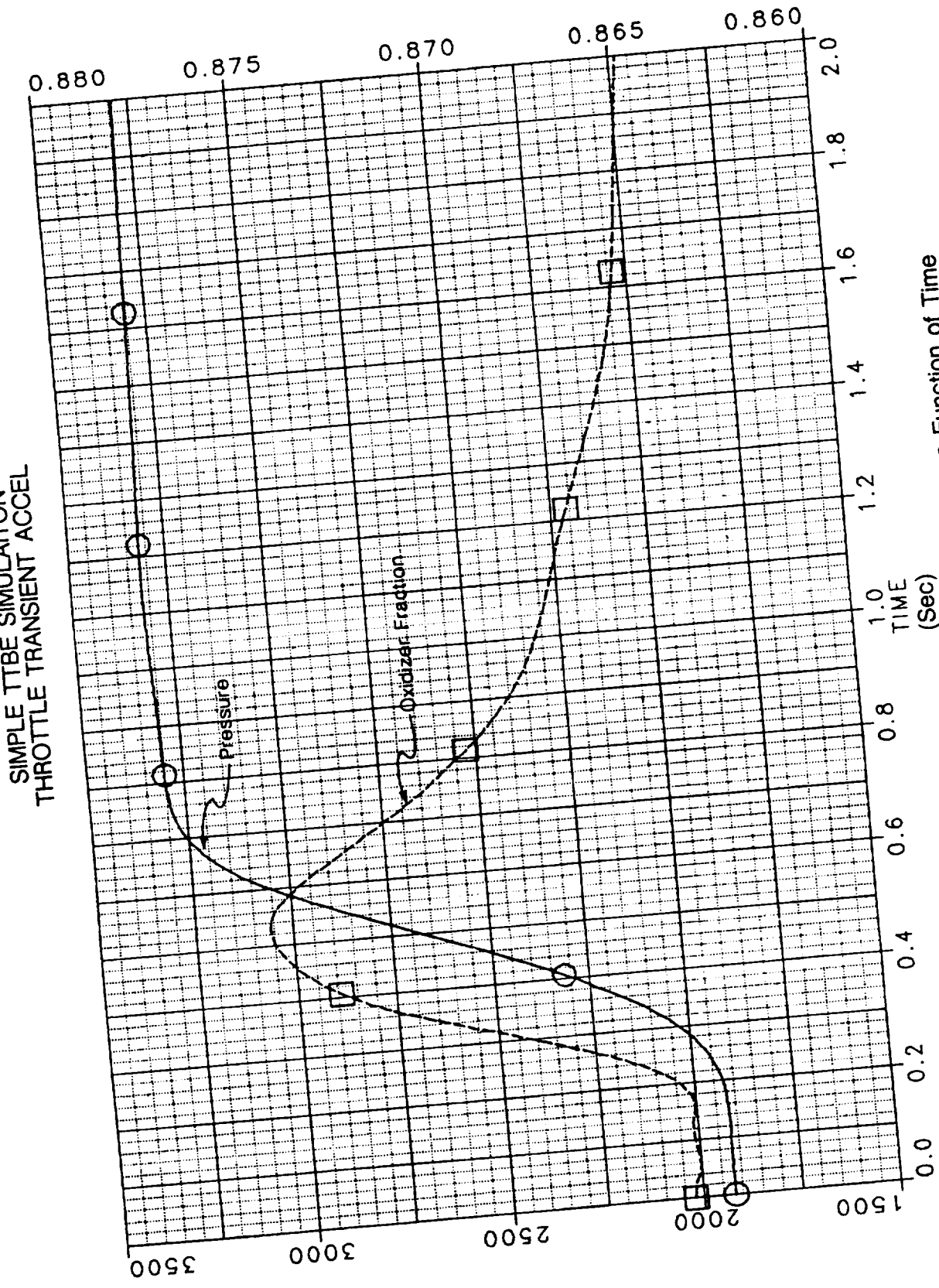


Figure 18 – Main Chamber Conditions as a Function of Time

APPENDIX

3.4.2.1 Module Cross Reference

E-1	INJT00	-	Main fuel injector
F-1	PIPE00	-	Incompressible fluid flow in pipe with inertia
F-2	PIPE01	-	Incompressible fluid flow in pipe with loss
F-3	PIPE02	-	Compressible flow through an orifice
G-1	MCHB00	-	Main combustion chamber
H-1	MIXR00	-	Simple flow mixer
I-1	NOZL00	-	Isentropic nozzle
J-1	PRBN00	-	Preburner without purge
K-1	PUMP00	-	Pump module for polytropic process
L-1	ROTR00	-	Rotor
O-1	SPLT00	-	Simple flow splitter
P-1	TURB00	-	Turbine
R-1	VOLM00	-	Volume/Accumulator/Tank

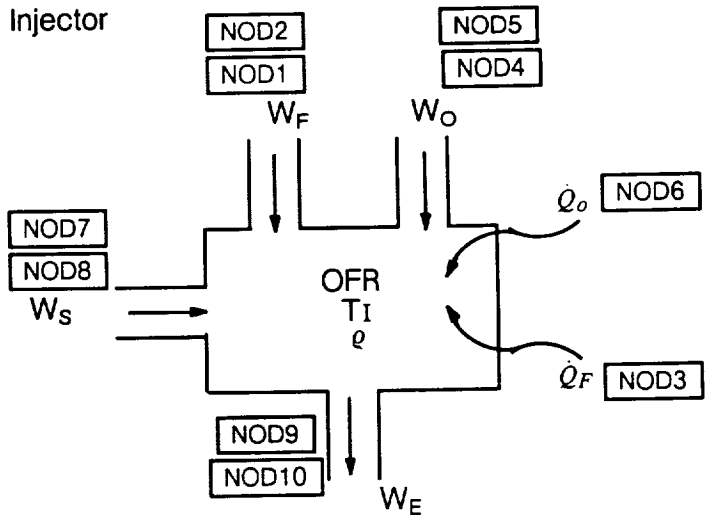
3.4.2.2 Map Cross Reference

1. PMAP01 - Pratt and Whitney SSME High Pressure Fuel Pump
2. PMAP02 - Pratt and Whitney SSME High Pressure Oxidizer Pump (main stage)
3. PMAP03 - Pratt and Whitney SSME High Pressure Oxidizer Pump (Preburner stage)
4. TBMP01 - Pratt and Whitney SSME High Pressure Oxidizer Turbine
5. TBMP02 - Pratt and Whitney SSME High Pressure Fuel Turbine

3.4.2.3 Module + Map Compatibility

1. PUMP00 - PMAP01, PMAP02, PMAP03
2. TURB00 - TBMP01, TBMP02

INJTØØ – Main Fuel Injector



INPUTS:

NOD1 – Fuel Inlet Node
NOD2 – Fuel Inlet Thermal Node
NOD3 – Heat Flux Node – Fuel
NOD4 – Oxidizer Inlet Node
NOD5 – Oxidizer Inlet Thermal Node
NOD6 – Heat Flux Node – Oxidizer
NOD7 – Sleeve Inlet Node
NOD8 – Sleeve Inlet Thermal Node
NOD9 – Exit node
NOD10 – Exit Thermal Node
C_{PF} – Specific Heat–Fuel Turbine Discharge
C_{PO} – Specific Heat–Oxidizer Turbine Discharge
γ_F – Specific Heat Ratio – Fuel Turbine Discharge
γ_O – Specific Heat Ratio – Oxidizer Turbine Discharge
γ_S – Specific Heat Ratio – Sleeve
γ_I – Specific Heat Ratio – Main Fuel Injector
γ_E – Specific Heat Ratio – Exit
$\dot{Q}_F$ – Heat Flux – Main Chamber Cooling – Oxidizer
$\dot{Q}_O$ – Heat Flux – Main Chamber Cooling – Fuel
W_F – Fuel Flowrate
W_O – Oxidizer Flowrate
W_S – Sleeve Flowrate
W_E – Exit Flowrate
T_F – Fuel Temperature
T_O – Oxidizer Temperature
OFR_F – Oxidizer Fraction – Fuel Inlet
OFR_O – Oxidizer Fraction – Oxidizer Inlet

STATES:

OFR – Oxidizer Fraction
T_I – Injector Temperature
ϱ – Overall Density

T_S – Sleeve Temperature
T_E – Exit Temperature
Vol – Main Fuel Injector Volume

OUTPUTS:

$\frac{d(OFR)}{dt}$ – Oxidizer Fraction Derivative

$\frac{d\varrho}{dt}$ – Density Derivative

$\frac{dT_I}{dt}$ – Temperature Derivative

CALCULATIONS:

$$W_{IN} = W_s + W_F + W_O$$

Total oxygen flow in: $$W_{O_{IN}} = OFR_F \times W_F + OFR_O \times W_O$$

DERIVATIVE CALCULATIONS:

$$m = \varrho \times Vol$$

$$\varrho = \frac{m}{Vol}$$

$$\frac{d\varrho}{dt} = \frac{\frac{dm}{dt}(Vol) - (m)\frac{dVol}{dt}}{(Vol)^2}$$

For a constant volume: $$\frac{d\varrho}{dt} = \frac{dm}{dt}\left(\frac{1}{Vol}\right)$$

$$\boxed{\frac{d\varrho}{dt} = \frac{W_{IN} - W_{out}}{Vol}}$$

ENERGY ANALYSIS:

$$\frac{dU}{dt} = \dot{Q}_o + \dot{Q}_F + W_F h_F + W_o h_o + W_s h_s - W_E h_E \qquad (1)$$

$$\frac{dU}{dt} = \frac{d(mu)}{dt} = \frac{du}{dt}(m) + (u)\frac{dm}{dt}$$

$$u = C_v T_I$$

$$\frac{dU}{dt} = mC_v\frac{dT_I}{dt} + C_v T_I(W_F + W_O + W_S - W_E) \qquad (2)$$

Substitute (2) into (1) and H = CpT and $T_{EXIT} = T_I$

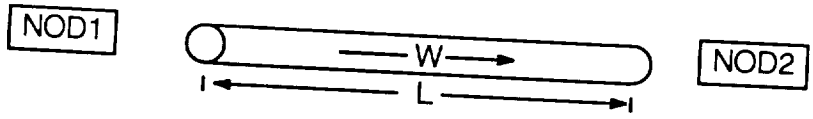
$$mC_v\frac{dT_I}{dt} + C_v T_I(W_F + W_O + W_S - W_E) = \dot{Q}_F + \dot{Q}_o + W_F C_P T_F + W_S C_P T_S - W_E C_P T_I$$

Solving for $\frac{dT_I}{dt}$:

$$\frac{dT_I}{dt} = [W_F\gamma_F T_F + W_o\gamma_o T_o + W_E\gamma_E T_I - (T_I(W_F + W_o + W_S - W_E))]\frac{1}{m} + \frac{\dot{Q}_F + \dot{Q}_o}{C_v m}$$

$$\boxed{\frac{dT_I}{dt} = \left[W_F(\gamma_F T_F - T_I) + W_o(\gamma_o T_o - T_I) + W_S(\gamma_S T_S - T_I) - W_E(\gamma_E - 1)T_I + \frac{(\dot{Q}_F + \dot{Q}_o)}{C_v}\right]\frac{1}{m}}$$

PIPEØØ – Incompressible fluid flow in pipe with inertia and loss



INPUTS:

NOD1 – Inlet Thermal Node
NOD2 – Exit Thermal Node
A – Inlet Flow Area
CF – Flow Coefficient
Pin – Inlet Pressure
Pout – Exit Pressure
ϱ in – Inlet Fluid Density
L – Length of Pipe

STATES

W – Flow Rate

OUTPUTS:

$\frac{dW}{dt}$ – Flow derivative

CALCULATIONS:

Definition of flow coefficient, CF: $CF = Q \times \sqrt{GF/\Delta P}$ where $GF = \frac{\varrho}{\varrho_{H_2O}}$ (specific gravity)
and Q = Volumetric flowrate (USGPM)

$$\varrho_{av} = \frac{(\varrho_{in} + \varrho_{out})}{2}$$

$$Q = \frac{W}{\varrho_{AV}}$$

$$CF = \frac{W}{\varrho_{av}} \sqrt{\frac{\varrho_{av}}{\varrho_{H_2O}} \left(\frac{1}{\Delta P}\right)}$$

solving for ΔP : $$CF^2 = \frac{W^2}{\varrho_{av}{}^2} \frac{\varrho_{av}}{\varrho_{H_2O}} \left(\frac{1}{\Delta P}\right)$$

$$\Delta P = \frac{W^2}{\varrho_{av}{}^2 \, \varrho_{H_2O} \, CF^2} \qquad (1)$$

Define α : $\alpha = \rho_{AV} \times CF^2$

Substitute into (1): $\Delta P = \frac{W^2}{a}$ x const. (2)

Equation (2) represents the steady state relationship for W and ΔP. Lets call this friction loss ΔP_{loss}.

DERIVATIVE CALCULATIONS:

Momentum analysis:

$$F = \frac{Ma}{g_c} \tag{1}$$

$$a = \frac{dV}{dt} \tag{2}$$

$$M = \varrho AL \tag{3}$$

$$W = \varrho AV \Rightarrow V = \frac{W}{\varrho A} \tag{4}$$

Substitute (2), (3) and (4) into (1):

$$F = \frac{\varrho AL}{g_c}\left(\frac{dW}{dt}\right)\frac{1}{\varrho A}$$

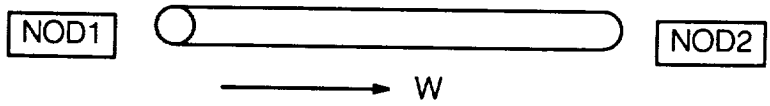
$$F = \frac{L}{g_c}\left(\frac{dW}{dt}\right) \tag{5}$$

Also: $F = (\Delta P - \Delta P_{loss})\, A$ (6)

Substitute (6) into (5) and solve for: $\frac{dW}{dt}$

$$\boxed{\frac{dW}{dt} = (\Delta P - \Delta P_{loss})\left(\frac{Ag_c}{L}\right)}$$ (where ΔP = Pin - Pout)

PIPEØ1 – Incompressible fluid flow in pipe with loss.



INPUTS:

NOD1	–	Inlet Thermal Node
NOD2	–	Exit Thermal Node
CF	–	Flow Coefficient
Pin	–	Inlet Pressure
Pout	–	Exit Pressure
ϱ in	–	Inlet Fluid Density

OUTPUT:

W – Flow Rate

CALCULATIONS:

Definition of flow coefficient, CF:

$$CF = Q \times \sqrt{GF/\Delta P} \qquad where\ GF = \frac{\varrho}{\varrho_{H_2O}}\ \text{(specific gravity)}$$

and Q = Volumetric Flowrate (USGPM)

$$Q = \frac{W}{\varrho_{in}}$$

$$CF = \frac{W}{\varrho_{in}} \sqrt{\frac{\varrho_{in}}{\varrho_{H_2O}} \left(\frac{1}{\Delta P}\right)}$$

Solving for W:

$$CF^2 = \frac{W^2}{\varrho_{in}^2} \frac{\varrho_{in}}{\varrho_{H_2O}} \frac{1}{\Delta P}$$

$$W = \sqrt{\Delta P\ \varrho_{in}\ CF^2\ \varrho_{H_2O}} \qquad (1)$$

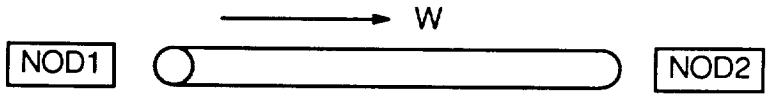
Define α:

$$\alpha = \varrho_{in} \times CF^2$$

Substitute into (1)

$$\boxed{W = \sqrt{\alpha\ \Delta P}\ x\ const.}$$

PIPEØ2 - Compressible flow through an orifice



INPUTS:

NOD1	-	Inlet Thermal Node
NOD2	-	Exit Thermal Node
K	-	Number of Head Losses
A	-	Throat Area
Pin	-	Inlet Pressure
Pout	-	Exit Presure
R	-	Gas Constant
Tin	-	Inlet Temperature
γin	-	Inlet Specific Heat Ratio

OUTPUTS:

W	-	Flowrate

CALCULATIONS:

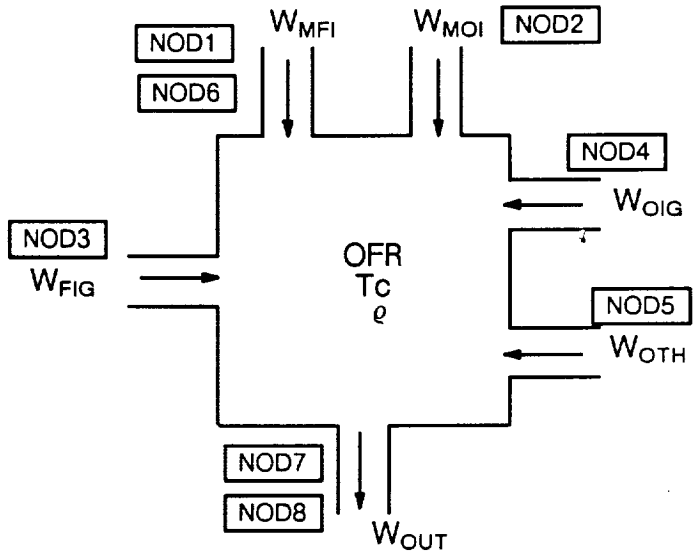
PR = Pin/Pout

Submodule FLPMØ2 gives a value for FP from PR, K, γ_{IN}.

Definition of flow parameter: $FP = \frac{W\sqrt{R \times T_{IN}}}{P_{IN} \times A}$

Solve for W : $\boxed{W = \frac{FP \times P_{IN} \times A}{\sqrt{R \times T_{IN}}}}$

MCHBØØ - Main Combustion Chamber



INPUTS:

NOD1 - Inlet Thermal Node
NOD2 - Inlet Node
NOD3 - Exit Node
NOD4 - Exit Thermal Node
NOD5 - Exit Node
NOD6 - Exit Thermal Node
NOD7 - Main Chamber Exit Node
NOD8 - Exit Thermal Node
OFR_{MFI} - Oxidizer Fraction, Main Fuel Inj.
W_{out} - Exit Flowrate
W_{FIG} - Fuel Igniter Flowrate
W_{MFI} - Main Fuel Injector Flowrate
W_{MOI} - Main Oxid. Injector Flowrate
W_{OIG} - Oxid. Igniter Flowrate
W_{OTH} - Other Fuel Flowrate
VOL - Main Chamber Volume
T_F - Fuel Temperature
γ - Specific Heat Ratio (Chamber)

STATES:

OFR - Oxidizer Fraction
Tc - Main Comb. Ch. Temperature
ϱ - Overall Density

OUTPUTS:

$\frac{d(OFR)}{dt}$ - Oxidizer fraction derivative

$\frac{d\varrho}{dt}$ - Density derivative

$\frac{dT_c}{dt}$ - Temperature derivative

CALCULATIONS:

$$m = \varrho \times Vol$$

Total oxidizer & Fuel Flows:

$$W_{O_{IN}} = W_{MOI} + W_{OIG} + W_{MFI} \times OFR_{MFI}$$
$$W_{F_{IN}} = W_{FIG} + W_{OTH} + W_{MFI} \times (1 - OFR_{MFI})$$

Flow-Based Mixture Ratio:

$$OF_W = W_{O_{IN}}/W_{F_{IN}}$$

Mass-based mixture ratio:

$$OF_m = OFR/(1 - OFR)$$

Temperature Rise ΔT
Where $\Delta T = f(\text{OFR})$

$$T_c^{\bullet} = T_F + \Delta T$$

DERIVATIVE CALCULATIONS:

$$\varrho = m/Vol$$

$$\frac{d\varrho}{dt} = \frac{\frac{dm}{dt}(Vol) - \frac{dVol}{dt}(m)}{(Vol)^2}$$

For a constant volume:

$$\frac{d\varrho}{dt} = \frac{\frac{dm}{dt}(Vol)}{(Vol)^2} = \frac{dm}{dt}\left(\frac{1}{Vol}\right)$$

$$\boxed{\frac{d\varrho}{dt} = \frac{W_{F_{IN}} + W_{o_{IN}} - W_{out}}{Vol}}$$

Energy analysis:

$$\frac{dU}{dt} = (W_{F_{IN}} + W_{o_{IN}})\, h^{\bullet} - (W_{out})\, h_c \qquad (1)$$

$$\frac{dU}{dt} = m\frac{du}{dt} + u\frac{dm}{dt} \qquad (2)$$

Substituting (2) into (1) and solving for $\frac{du}{dt}$ yields.

$$\frac{du}{dt} = \left[(W_{F_{IN}} + W_{O_{IN}}) h^* - (W_{out})\ h_c - u\frac{dm}{dt} \right] / m \qquad (3)$$

For a perfect gas:

$$h = C_P T \Rightarrow dh = C_P dT$$
$$u = C_v T \Rightarrow du = C_v dT$$

From continuity:

$$m = m_F + m_O$$
$$\frac{dm}{dt} = (W_{F_{IN}} + W_{O_{IN}}) - W_{out}$$

Substituting perfect gas relationships and $\frac{dm}{dt}$ into (3) yields:

$$\frac{du}{dt} = \left[(W_{F_{IN}} + W_{O_{IN}})\ C_P T^* - W_{out}\ C_P T_c - C_V T_c (W_{F_{IN}} + W_{O_{IN}} - W_{out}) \right] \left(\frac{1}{m} \right)$$

$$\frac{du}{dt} = \frac{d(C_v T_c)}{dt} = C_v \left(\frac{dT_c}{dt} \right)$$

Simplifying yields:

$$\boxed{\frac{dT_c}{dt} = \left[(W_{F_{IN}} + W_{O_{IN}})(\gamma_c T_c^* - T_c) - W_{out}(\gamma_c - 1) T_c \right] \left(\frac{1}{m} \right)}$$

OFR derivative:

$$OFR = \frac{m_O}{m_T} = \frac{m_O}{m_O + m_F}$$

Assuming homogeneous volume: $(OFR)\ W_{T_{EX}} = W_{O_{EX}}$

$$(1 - OFR)\ W_{T_{EX}} = W_{F_{EX}}$$

$$\frac{d(OFR)}{dt} = \left(\frac{dm_O}{dt}\ m_T - \frac{dm_T}{dt}\ m_O \right) / m_T^2$$

$$= \frac{1}{m_T} \left[\frac{dm_O}{dt} - \frac{dm_T}{dt}\ \frac{m_O}{m_T} \right]$$

$$= \frac{1}{m_T}\left[\frac{dm_o}{dt} - (OFR)\frac{dm_T}{dt}\right] \quad (1)$$

$$\frac{dm_o}{dt} = W_{o_{IN}} - W_{o_{EX}} = W_{o_{IN}} - (OFR)W_{T_{EX}} \quad (2)$$

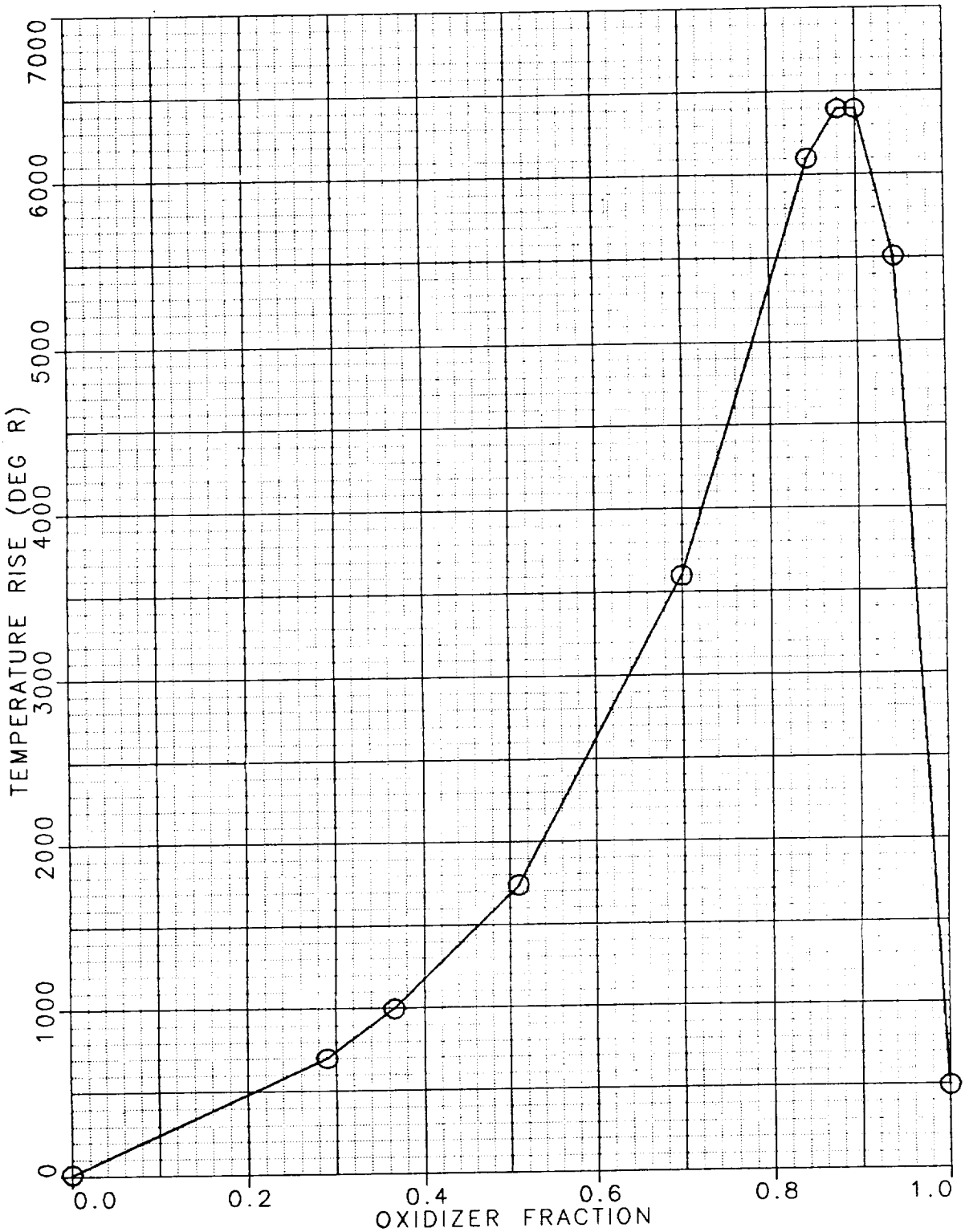
$$\frac{dm_T}{dt} = W_{T_{IN}} - W_{T_{EX}} \quad (3)$$

Substituting (2) and (3) into (1) yields:

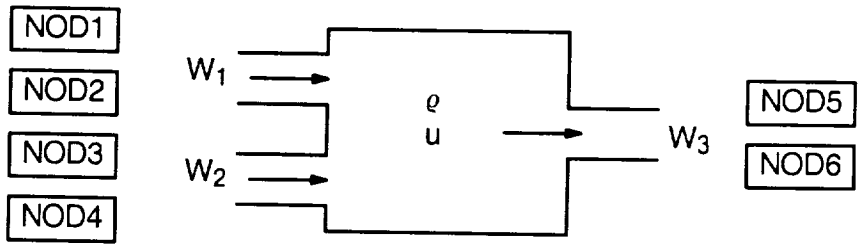
$$\frac{d(OFR)}{dt} = \frac{1}{m_T}\left[W_{o_{IN}} - (OFR)W_{T_{EX}} - (OFR)(W_{T_{IN}} - W_{T_{EX}})\right]$$

Simplifying: $$\boxed{\frac{d(OFR)}{dt} = \frac{1}{m_T}\left[W_{o_{IN}} - (OFR)W_{T_{IN}}\right]}$$

TEMPERATURE RISE VS. OXIDIZER FRACTION



MIXRØØ - Simple Flow Mixer



INPUTS:

NOD1 - Inlet 1 Thermal Node
NOD2 - Inlet 1 Node
NOD3 - Inlet 2 Thermal Node
NOD4 - Inlet 2 Node
NOD5 - Exit Node
NOD6 - Exit Thermal node
h_1 - Inlet Stream 1 Enthalpy
h_2 - Inlet Stream 2 Enthalpy
h_3 - Exit Enthalpy
h_{mix} - Mixer Enthalpy
W_1 - Inlet 1 Flowrate
W_2 - Inlet 2 Flowrate
W_3 - Exit Flowrate
VOL - Mixer Volume

STATES:

ϱ - Mixer Density
u - Mixer Internal Energy

OUTPUTS:

$\frac{d\varrho}{dt}$ - Density Derivative

$\frac{du}{dt}$ - Internal Energy Derivative

DERIVATIVE CALCULATIONS:

$m = \varrho \times vol$

$$\frac{dm}{dt} = W_1 + W_2 - W_3$$

$$\varrho = \frac{m}{Vol}$$

$$\frac{d\varrho}{dt} = \frac{\left(\frac{dm}{dt}\right)(Vol) - \left(\frac{dVol}{dt}\right)(m)}{Vol^2}$$

For a constant Volume:

$$\frac{d\varrho}{dt} = \frac{dm}{dt}\left(\frac{1}{Vol}\right)$$

$$\boxed{\frac{d\varrho}{dt} = \frac{(W_1 + W_2 - W_3)}{Vol}}$$

Energy:

$$\frac{dU}{dt} = W_1 h_1 + W_2 h_2 - W_3 h_{mix}$$ (assuming no heat transfer) (1)

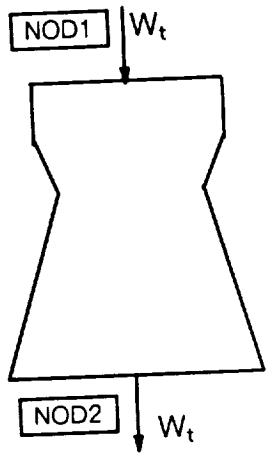
$$\frac{dU}{dt} = \frac{d(mu)}{dt} = m\left(\frac{du}{dt}\right) + u\left(\frac{dm}{dt}\right)$$ (2)

Substituting (2) into (1): $$m\left(\frac{du}{dt}\right) + u\left(\frac{dm}{dt}\right) = W_1 h_1 + W_2 h_2 - W_3 h_{mix}$$

Solving for $\frac{du}{dt}$:

$$\boxed{\frac{du}{dt} = \left[W_1 h_1 + W_2 h_2 - W_3 h_{mix} - u\left(\frac{dm}{dt}\right)\right]\left(\frac{1}{m}\right)}$$

NOZLØØ - Isentropic Nozzle



INPUTS:

NOD1 - Inlet Thermal Node
NOD2 - Exit Thermal Node
P_{IN} - Inlet Pressure Main Chamber
P_{OUT} - Exit Pressure (Ambient)
T_{IN} - Inlet Temperature
AR - Area Ratio
AREA - Throat Area
γ - Specific Heat Ratio
R - Gas Constant
CS - Loss Coefficient

OUTPUTS:

Fg - Gross Thrust
SI - Specific Impulse
M_e - Mach Number At Exit
W_T - Flowrate At Throat

CALCULATIONS:

PR = P_{IN} / P_{OUT}

Submodule CDNZØØ gives values for FP and CF.

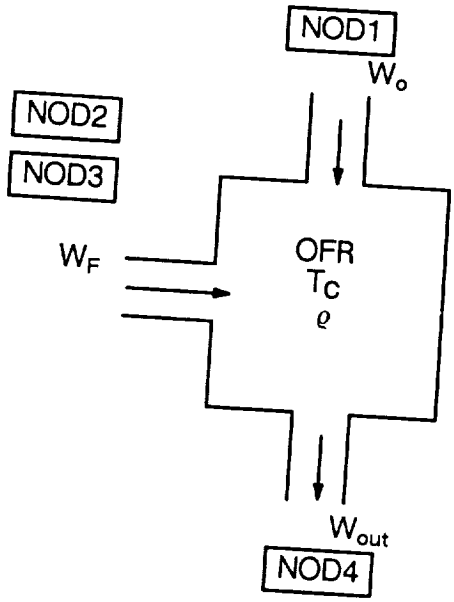
Definition of flow parameter: $FP = \frac{Wt\sqrt{R \times Tin}}{AREA \times P_{IN}}$

Solve for W_t: $\boxed{W_t = \frac{(FP)\ (P_{IN})(AREA)}{\sqrt{(R)(T_{IN})}}}$

Thrust: $\boxed{Fg = P_{IN} \times AREA \times CF}$ where CF is a thrust coefficient

Specific impulse: $\boxed{SI = Fg / W_t}$

PBRNØØ - Preburner without purge



INPUTS:

NOD1 - Oxidizer Inlet Flow Node
NOD2 - Fuel Inlet Flow Node
NOD3 - Fuel Inlet Thermal node
NOD4 - Exit Flow Node
OFLT - Mixture Ratio (Light Level)
T_F - Fuel Temperature
VOL - Preburner Volume
W_F - Fuel Flowrate
W_O - Oxidizer Flowrate
W_{OUT} - Exit Flowrate
γ - Specific Heat Ratio

STATES:

OFR - Oxidizer Fraction
T_C - Preburner Temperature
ϱ - Overall Density

OUTPUTS:

$\frac{d(OFR)}{dt}$ - Oxidizer fraction derivative

$\frac{dT_c}{dt}$ - Temperature derivative

$\frac{d\varrho}{dt}$ - Density derivative

CALCULATIONS:

Flow - Based
Mixture Ratio : $OF_w = W_o / W_F$

Mass - Based
Mixture Ratio : $OF_m = OFR \;/\; (1 - OFR)$

Temperature Rise (ΔT): $T_c^* = T_F + \Delta T$

Where $\Delta T = f(\text{OFR})$ if preburner is lit
or $\Delta T = \text{const.}$ if preburner is not lit

DERIVATIVE CALCULATIONS:

$$\varrho = m/Vol$$

$$\frac{d\varrho}{dt} = \frac{\frac{dm}{dt}(VOL) - \frac{dVol}{dt}(m)}{(Vol)^2}$$

For a constant volume: $\frac{d\varrho}{dt} = \frac{\frac{dm}{dt}(Vol)}{(Vol)^2} = \frac{dm}{dt}\left(\frac{1}{Vol}\right)$

$$\boxed{\frac{d\varrho}{dt} = \frac{W_F + W_o - W_{out}}{Vol}}$$

Energy analysis: $\frac{dU}{dt} = (W_F + W_o)\, h^* - (W_{out})\, h_c$ (1)

$$\frac{dU}{dt} = m\frac{du}{dt} + u\frac{dm}{dt} \quad (2)$$

Substituting (2) into (1) and solving for $\frac{du}{dt}$ yields.

$$\frac{du}{dt} = \left[(W_F + W_o)h^* - (W_{out})\, h_c - u\frac{dm}{dt}\right]/m \quad (3)$$

For a perfect gas: $h = C_P T \Rightarrow dh = C_P dT$

$$u = C_v T \Rightarrow du = C_v dT$$

From continuity: $m = m_F + m_o$

$$\frac{dm}{dt} = (W_F + W_o) - W_{out}$$

Substituting perfect gas relationships and $\frac{dm}{dt}$ into equation (3) yields:

$$\frac{du}{dt} = \left[(W_F + W_o)\; C_P T_c^* - W_{out} C_P T_c - C_v T_c (W_F + W_o - W_{out})\right]\left(\frac{1}{m}\right)$$

$$\frac{du}{dt} = \frac{d(C_v\, T_c)}{dt} = C_v\left(\frac{dT_c}{dt}\right)$$

Simplifying yields: $$\boxed{\frac{dT_C}{dt} = \left[(W_F + W_O)(\gamma_C T_C - T_C) - W_{out}(\gamma_C - 1)T_C\right]\left(\frac{1}{m}\right)}$$

OFR derivative:

$$OFR = \frac{m_o}{m_T} = \frac{m_o}{m_o + m_F}$$

Assuming homogeneous volume: $(OFR)\ W_{T_{EX}} = W_{o_{EX}}$

$$(1 - OFR)\ W_{T_{EX}} = W_{F_{EX}}$$

$$\frac{d(OFR)}{dt} = \left(\frac{dm_o}{dt}\ m_T - \frac{dm_T}{dt}\ m_o\right) / m_T^2$$

$$= \frac{1}{m_T}\left[\frac{dm_o}{dt} - \frac{dm_T}{dt}\ \frac{m_o}{m_T}\right]$$

$$= \frac{1}{m_T}\left[\frac{dm_o}{dt} - (OFR)\ \frac{dm_T}{dt}\right] \quad (1)$$

$$\frac{dm_o}{dt} = W_o - W_{o_{EX}} = W_o - (OFR)W_{T_{EX}} \quad (2)$$

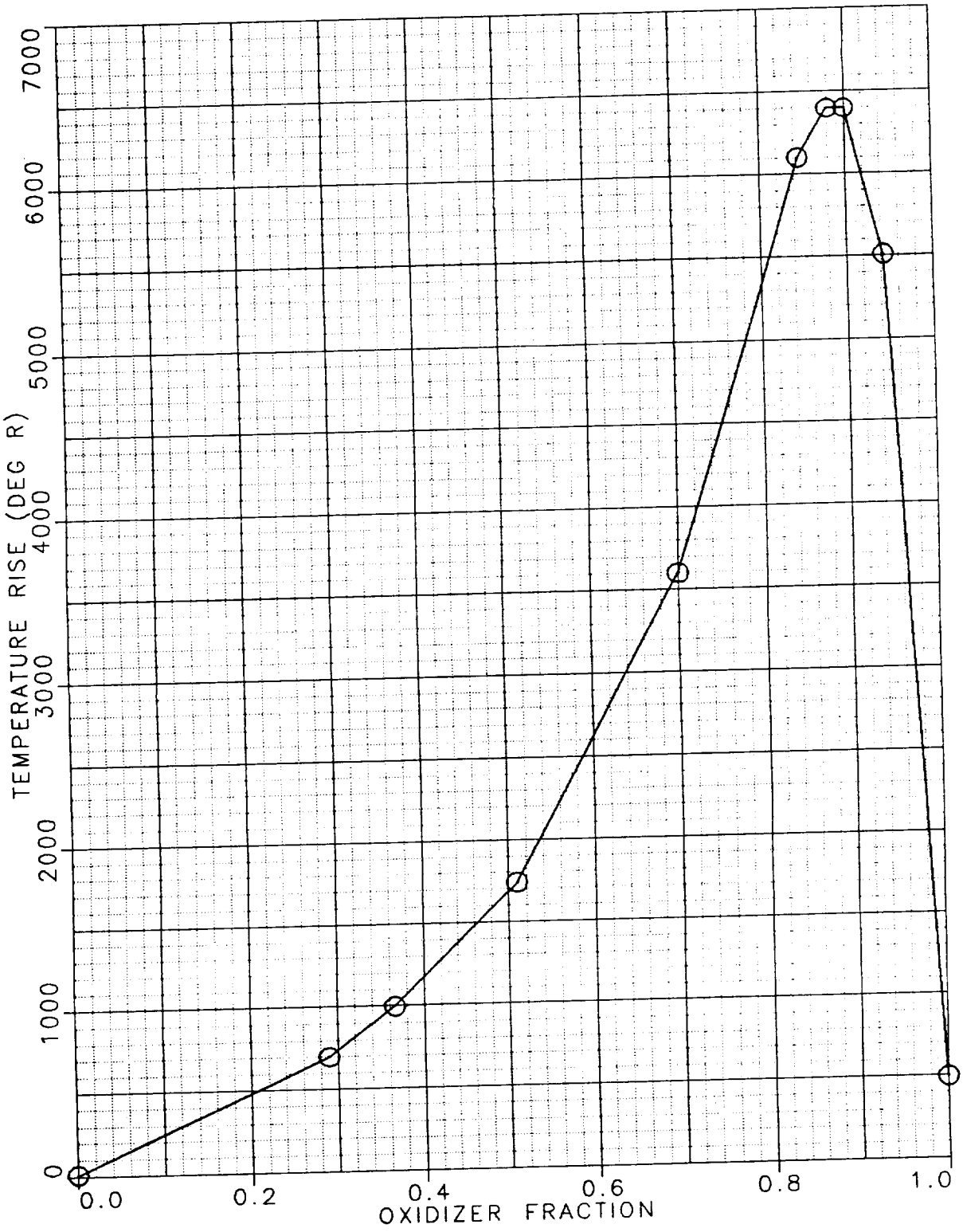
$$\frac{dm_T}{dt} = W_{T_{IN}} - W_{T_{EX}} \quad (3)$$

Substituting (2) and (3) into (1) yields:

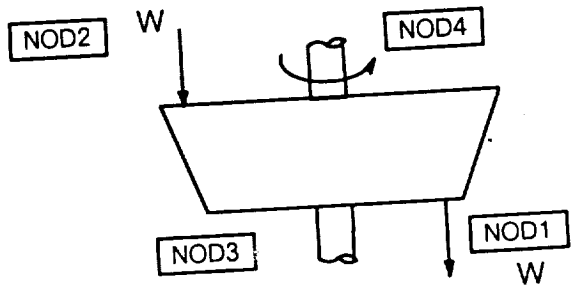
$$\frac{d(OFR)}{dt} = \frac{1}{m_T}\left[W_o - (OFR)W_{T_{EX}} - (OFR)(W_{T_{IN}} - W_{T_{EX}})\right]$$

Simplifying: $$\boxed{\frac{d(OFR)}{dt} = \frac{1}{m_T}\left[W_o - (OFR)W_{T_{IN}}\right]}$$

TEMPERATURE RISE VS. OXIDIZER FRACTION



PUMPØØ - Pump Module For Polytropic Process



INPUTS:

NOD1 - Flow Node
NOD2 - Inlet Thermal Node
NOD3 - Exit Thermal Node
NOD4 - Shaft Work Node
W - Flowrate
P_{IN} - Inlet Pressure
ϱ_{IN} - Inlet Density
h_{IN} - Inlet Enthalpy
N - Rotational Speed

OUTPUTS:

P_{OUT} - Discharge Pressure
h_{OUT} - Discharge Enthalpy
ϱ_{OUT} - Discharge Density
T_{ORQ} - Torque Required

This module iterates on $\left(\frac{\varrho_{OUT}}{\varrho_{IN}}\right)$ until the exit pressure that results from the property call using h_{out} and ϱ_{OUT} equals the exit pressure calculated from the polytropic equation (iterate until $P_{OUT} = P_{OUTc}$).

CALCULATIONS:

$$N_{rad} = N \times \left(\frac{2\pi}{60}\right)$$

Submodule MAP gives values for H (head) and Torque from W, ϱ_{OUT} and N.

$$\eta \equiv \frac{work\ out}{work\ in} = \frac{(W)(H)(gr)}{(N)(Torq)(gc)}$$

where gr = acceleration due to gravity

$$\Delta h = \frac{(H)(gr)}{(\eta)(gc)}$$

$h_{OUT} = h_{IN} + \Delta h$

A property call using ϱ_{OUT} and h_{OUT} gives a value for P_{OUT}. To get P_{OUTc} lets look at a polytropic process:

Polytropic Headrise

For a Polytropic Process $P \cdot v^{n} = C$ (1)

$P \cdot v^{1/n} = C^{1/n}$ (2)

$v = C^{1/n} P^{-1/n}$ (3)

H = headrise $= \int_1^2 v dP$ (4)

$H = \int_1^2 v dP$

Substitute (3) into (4)

$H = \int_1^2 C^{1/n} P^{-1/n} dP$

$H = C^{1/n} \int_1^2 P^{-1/n} dP$

$H = \left(C^{1/n} P_2^{1-1/n} - C^{1/n} P_1^{1-1/n}\right) \left(\frac{n}{n-1}\right)$

$H = \left(\frac{n}{n-1}\right)\left(C^{1/n} P_2^{1-1/n} - C^{1/n} P_1^{1-1/n}\right)$ (5)

Substitute (2) into (5)

$H = \left(\frac{n}{n-1}\right)\left[v_2 P_2^{1/n} P_2^{1-1/n} - v_1 P_1^{1/n} P_1^{1-1/n}\right]$

$$H = \left(\frac{n}{n-1}\right)\left\{v_2 P_2 - v_1 P_1\right\}$$

For a polytropic process (1)

$$P \cdot v^{n} = k$$

or

$$P_1 \; v_1^{n} = P_2 \; v_2^{n}$$

(2)

$$\frac{P_2}{P_1} = \left(\frac{v_1}{v_2}\right)$$

(3)

$$\ln \left(\frac{P_2}{P_1}\right) = n \ln \left(\frac{v_1}{v_2}\right)$$

(4)

$$n = \frac{\ln \left(\frac{P_2}{P_1}\right)}{\ln \left(\frac{v_1}{v_2}\right)}$$

also from 3:

$$\frac{P_2}{P_1} \cdot \frac{v_2}{v_1} = \left(\frac{v_1}{v_2}\right)^{n} \cdot \frac{v_2}{v_1}$$

$$\frac{P_2 \; v_2}{P_1 \; v_1} = \left(\frac{v_2}{v_1}\right)^{-n} \cdot \left(\frac{v_2}{v_1}\right)^{1}$$

$$\frac{P_2 \; v_2}{P_1 \; v_1} = \left(\frac{v_2}{v_1}\right)^{1-n}$$

$$\ln \left(\frac{P_2 \; v_2}{P_1 \; v_1}\right) = (1-n) \; \ln \left(\frac{v_2}{v_1}\right)$$

(5)

$$n = 1 - \frac{\ln \left(\frac{P_2 \; v_2}{P_1 \; v_1}\right)}{\ln \left(\frac{v_2}{v_1}\right)}$$

let $\beta = \dfrac{n}{n-1}$

$$\frac{n}{n-1} = \frac{1 - \dfrac{\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right]}{\ln\left(\frac{v_2}{v_1}\right)}}{-\dfrac{\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right]}{\ln\left(\frac{v_2}{v_1}\right)}}$$

$$\frac{n}{n-1} = \frac{\ln\left(\frac{v_2}{v_1}\right) - \ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right]}{-\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right]}$$

$$\frac{n}{n-1} = \frac{\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right] - \ln\left(\frac{v_2}{v_1}\right)}{\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right]}$$

$$\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right] - \ln\left(\frac{v_2}{V_1}\right) = \ln\frac{\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)}{\left(\frac{P_2}{P_1}\right)} = \ln\left(\frac{P_2}{P_1}\right)$$

$$\boxed{\frac{n}{n-1} = \frac{\ln\left(\frac{P_2}{P_1}\right)}{\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right]}}$$

$$H = \left(\frac{n}{n-1}\right)[P_2 v_2 - P_1 v_1]$$

$$H = \left(\frac{n}{n-1}\right)\left[\frac{P_2}{\varrho_2} - \frac{P_1}{\varrho_1}\right]$$

$$H\left(\frac{n-1}{n}\right) = \frac{P_2}{\varrho_2} - \frac{P_1}{\varrho_1}$$

$$H\left(\frac{n-1}{n}\right) + \frac{P_1}{\varrho_1} = \frac{P_2}{\varrho_2}$$

$$\therefore\ P_2 = \varrho_2\left[H\left(\frac{n-1}{n}\right) + \frac{P_1}{\varrho_1}\right]$$

If $\beta = \dfrac{n}{n-1} \Rightarrow \dfrac{1}{\beta} = \dfrac{n-1}{n}$

$$P_1 = \varrho_2\left[\frac{H}{\beta} + \frac{P_1}{\varrho_1}\right]$$

Module implementation with ϱ and P instead of v and P

$$\boxed{P_{OUT_C} = \varrho_{OUT}\left[\frac{H}{\beta} + \frac{P_{IN}}{\varrho_{IN}}\right]}$$

FORTRAN implementation of β calculation

$$\beta = \frac{\ln\left(\frac{P_2}{P_1}\right)}{\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{v_2}{v_1}\right)\right]}$$

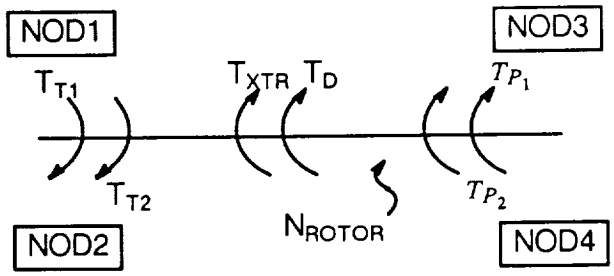
$$\frac{v_2}{v_1} = \frac{\varrho_1}{\varrho_2}$$

then $$\beta = \frac{\ln\left(\frac{P_2}{P_1}\right)}{\ln\left[\left(\frac{P_2}{P_1}\right)\left(\frac{\varrho_1}{\varrho_2}\right)\right]} = \frac{\ln\left(\frac{P_2}{P_1}\right)}{\ln\left[\frac{\left(\frac{P_2}{P_1}\right)}{\left(\frac{\varrho_2}{\varrho_1}\right)}\right]}$$

$$\beta = \frac{\ln\left(\frac{P_2}{P_1}\right)}{\ln\left(\frac{P_2}{P_1}\right) - \ln\left(\frac{\varrho_2}{\varrho_1}\right)}$$

$$\boxed{\beta = \frac{1}{1 - \left[\frac{\ln\left(\frac{\varrho_2}{\varrho_1}\right)}{\ln\left(\frac{P_2}{P_1}\right)}\right]}}$$

ROTRØØ - Rotor Torque Balance



INPUTS:

NOD1 - High Pressure Turbine Node
NOD2 - Low pressure Turbine node
NOD3 - High Pressure Pump Node
NOD4 - Low Pressure Pump Node
T_{T1} - High Pressure Turbine Torque
T_{T2} - Low Pressure Turbine Torque
T_{P1} - High Pressure Pump Torque
T_{P2} - Low Pressure Pump Torque
T_D - Drag Torque
T_{XTR} - Extraction Torque
n_{T1} - High Pressure Turbine Gear Ratio
n_{T2} - Low Pressure Turbine Gear Ratio
n_{P1} - High Pressure Pump Gear Ratio
n_{P2} - Low Pressure Pump Gear Ratio
n_{XTR} - Extraction Gear Ratio
I_P - Polar Moment of Inertia

STATES:

N_{ROTOR} - Rotor Speed

OUTPUTS:

$\frac{dN_{ROTOR}}{dt}$ - Rotor speed derivative

CALCULATIONS:

$$\Sigma Tn = I_P \alpha \qquad (1)$$

$$\alpha = \frac{dN_{rotor}}{dt} \qquad (2)$$

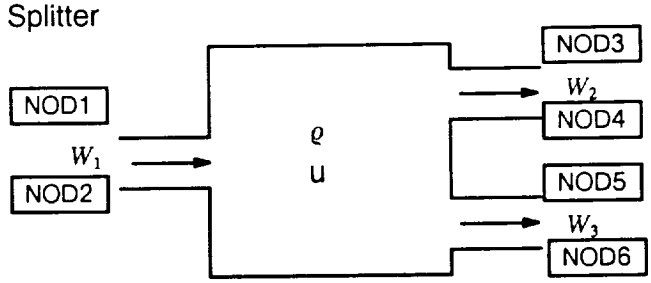
Substituting (2) into (1)

$$\Sigma Tn = I_P \frac{dN_{rotor}}{dt}$$

$$\frac{dN_{rotor}}{dt} = \frac{\Sigma Tn}{I_P}$$

$$\boxed{\frac{dN_{rotor}}{dt} = \frac{(T_{T_1}\, n_{T_1} + T_{T_2}\, n_{T_2} - T_{P_1}\, n_{P_1} - T_{P_2}\, n_{P_2} - T_D - T_{XTR}\, n_{XTR})}{I_P}}$$

SPLTØØ - Simple Flow Splitter



INPUTS:

NOD1 - Inlet Thermal Node
NOD2 - Inlet Node
NOD3 - Exit Node
NOD4 - Exit Thermal Node
NOD5 - Exit Node
NOD6 - Exit Thermal Node
h_1 - Inlet Enthalpy
h_2 - Exit Enthalpy
h_3 - Exit Enthalpy
h_{SPLT} - Splitter Enthalpy
VOL - Splitter Volume
W_1 - Inlet Flowrate
W_2 - Exit Flowrate
W_3 - Exit Flowrate

STATES:

ϱ - Splitter Density
u - Splitter Internal Energy

OUTPUTS:

$\frac{d\varrho}{dt}$ - Density derivative

$\frac{du}{dt}$ - Internal energy derivative

CALCULATIONS:

$m = \varrho \times vol$

$\frac{dm}{dt} = W_1 - W_2 - W_3$

$\varrho = \frac{m}{Vol}$

$\frac{d\varrho}{dt} = \frac{\left(\frac{dm}{dt}\right)(Vol) - \left(\frac{dVol}{dt}\right)(m)}{Vol^2}$

For a constant Volume:

$$\frac{d\varrho}{dt} = \frac{dm}{dt}\left(\frac{1}{Vol}\right)$$

$$\boxed{\frac{d\varrho}{dt} = \frac{(W_1 + W_2 - W_3)}{Vol}}$$

$$\frac{dU}{dt} - W_1 h_1 - W_2 h_{SPLT} - W_3 h_{SPLT} \qquad \text{(assuming no heat transfer)} \quad (1)$$

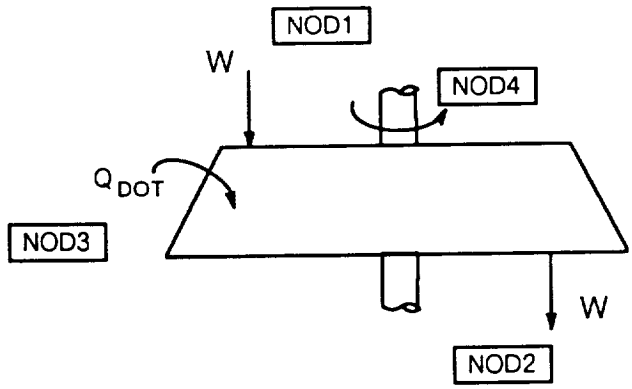
$$\frac{dU}{dt} = \frac{d(mu)}{dt} = m\left(\frac{du}{dt}\right) + u\left(\frac{dm}{dt}\right) \quad (2)$$

Substituting (2) into (1): $m\left(\frac{du}{dt}\right) + u\left(\frac{dm}{dt}\right) = W_1 h_1 - W_2 h_{SPLT} - W_3 h_{SPLT}$

Solving for $\frac{du}{dt}$:

$$\boxed{\frac{du}{dt} = \left[W_1 h_1 - W_2 h_{SPLT} - W_3 h_{SPLT} - u\left(\frac{dm}{dt}\right)\right]\left(\frac{1}{m}\right)}$$

TURBØØ - Turbine



INPUTS:

NOD1 - Inlet Thermal Node
NOD2 - Discharge Thermal Node
NOD3 - Heat Transfer Node
NOD4 - Shaft Work Node
R - Inlet Gas Constant
γ - Inlet Specific Heat Ratio
Z - Inlet Compressibility Factor
P_{IN} - Inlet Pressure
T_{IN} - Inlet Temperature
P_{OUT} - Exit Pressure
N - Rotational Speed
Q_{DOT} - Heat Transfer Rate Into Turbine
Q_{FRC} - Fraction of Heat That Goes To Turbine Work
MAP - Turbine Characteristic Map

OUTPUTS:

W - Turbine Flowrate
T_{ORQ} - Turbine Torque
T_{OUT} - Discharge Temperature

CALCULATIONS:

$PR = P_{IN} / P_{OUT}$

$N_{rad} = N \left(\frac{2\pi}{60}\right)$

Submodule MAP gives values for FP, $\Delta h'$ and η

Definition of Flow Parameter: $FP = \frac{W\sqrt{R \times T_{IN}}}{P}$

Solve for W: $W = \frac{FP \times P}{\sqrt{R \times T_{IN}}}$

Real Δh: $\Delta h = \Delta h' \times \eta + (Q_{DOT} \times Q_{FRC})$

$$TORQ = \frac{W\Delta h}{N_{rad}}$$

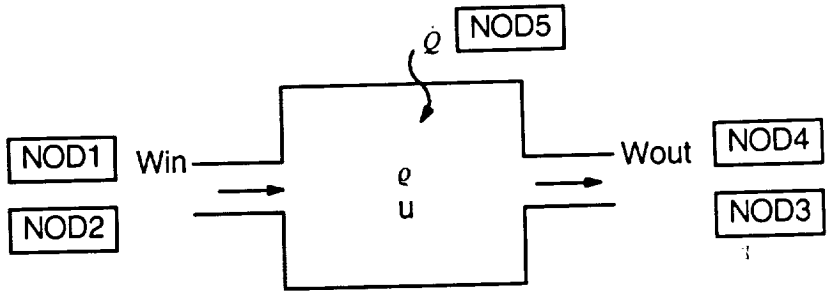
Exit Temperature: $\Delta h = C_p \Delta T$

$$C_P = R\left(\frac{\gamma}{\gamma - 1}\right)$$

$$T_{IN} - T_{OUT} = \frac{(\Delta h - Q_{DOT})}{C_P}$$

$$T_{OUT} = T_{IN} - \frac{(\Delta h - Q_{DOT})(\gamma - 1)}{\gamma}$$

VOLMØØ - Lumped Volume With Energy and Continuity



INPUTS

NOD1 - Inlet Thermal Node
NOD2 - Inlet Node
NOD3 - Exit Node
NOD4 - Exit Thermal Node
NOD5 - Heat Transfer Node
Vol - Volume
hin - Inlet Enthalpy
hout - Exit Enthalpy
hvol - Volume Enthalpy
$\dot{Q}$ - Heat Flux into the Volume
Win - Inlet Flowrate
Wout - Exit Flowrate

STATES:

ϱ - Volume Density
u - Volume Internal Energy

OUTPUTS:

$\frac{d\varrho}{dt}$ - Density Derivative

$\frac{du}{dt}$ - Internal Energy Derivative

DERIVATIVE CALCULATIONS:

$m = \varrho \times vol$

$$\frac{dm}{dt} = W_{in} - W_{out}$$

$$\varrho = \frac{m}{Vol}$$

$$\frac{d\varrho}{dt} = \frac{\left(\frac{dm}{dt}\right)(Vol) - \left(\frac{dVol}{dt}\right)(m)}{Vol^2}$$

For a constant volume: $\frac{dQ}{dt} = \frac{dm}{dt}\left(\frac{1}{Vol}\right)$

$$\boxed{\frac{dQ}{dt} = \frac{W_{in} - W_{out}}{Vol}}$$

Energy:

$$\frac{dU}{dt} = (Win)\ (hin) - (Wout)\ (hvol) + \dot{Q} \tag{1}$$

$$\frac{dU}{dt} = \frac{d(mu)}{dt} = (m)\left(\frac{du}{dt}\right) + (u)\left(\frac{dm}{dt}\right) \tag{2}$$

substituting (2) into (1)

$$m\left(\frac{du}{dt}\right) + u\left(\frac{dm}{dt}\right) = (Win)(hin) - (Wout)(hvol) + \dot{Q}$$

solving for $\frac{du}{dt}$: $$\boxed{\frac{du}{dt} = \left((Win)\ (hin) - (Wout)\ (hvol) - u\frac{dm}{dt} + \dot{Q}\right)\left(\frac{1}{m}\right)}$$

SSBL03 - Steady State Balance Routine

SSBL03 is an interface to SMIT01 which is designed to perform a steady-state balance on a given model. The call list for SSBL03 is defined as follows:

ISSPNT = Print option flag
- 0 = No steady-state print
- 1 = Steady-state print to unit 6

NSTATE = Number of States
STATE = Array of State Variables
DSTATE = Array of State Variable Derivatives
SNORM = Array of normalizing values for State Variables
SNAME = Array of State Variable Names
ISON = Array of on/off codes for State Variables
- 0 = State not iterated
- 1 = State iterated

NVAR = Number of algebraic loops (balances)
XVAR = Array of balance independent variables
YVAR = Array of balance dependent variables
XNORM = Array of normalizing values for balances
XNAME = Array of names for balances
IXON = Array of on/off codes for balances
- 0 = Balance not iterated
- 1 = Balance iterated

MXPASS = Maximum number of convergence passes
ISAMAT = Same Jacobian Flag
- 0 = Do no use previous matrix
- 1 = Use previous Jacobian

ISIG = Control flag
- 0 = On first pass to set initialization
- 1 = Evaluate derivatives and errors
- 2 = Converged

SSBL03 provides input to SMIT01 so that both states and algebraic loop (balance) parameters are iterated until their respective errors are within tolerance. The error terms for states are calculated as the derivative divided by the state and the error terms for the balance parameters are the input error terms (YVAR).

The states and balance parameters are normalized before going into SMIT01 by the input normalization values, SNORM, and XNORM. If the input normalizing value is zero, the initial value of the state or balance parameter will be used.

Values for the SMIT01 input DELTAX and DXALOW are input based on the normalization values.

Specific states and balances may be turned on or off based on the input switching arrays ISON and IXON. SSBL03 builds the dependent and independent variable arrays for SMIT01 based on these inputs and the normalization discussed above. If the switch value is 1, the variable will be included in the SMIT01 array and therefore iterated, if the switch value is zero, the variable will not be operated on by SMIT01.

If requested via the flag ISSPNT, SSBL03 will provide a detailed print of the steady-state balance. The steady-state print includes the following:

1. The base point dependent and independent variables sent to SMIT01
2. The dependent and independent SMIT01 variables during the generation of the J matrix.
3. The dependent and independent SMIT01 variables during each convergence attempt along with the changes in each variable and the percent reduction in the sum of the squared errors.
4. A message providing SMIT01 statistics upon completion.

TRAP03 - Transient Integration Routine

TRAP03 is the transient counterpart of SSBL03 and has the same inputs as SSBL03 with the following additions:

1. DT = Transient time increment

2. TIME = Transient time

TRAP03 is an implicit integration routine which can simultaneously solve algebraic loops. The predicted STATE values at TIME = TIME + DT are the previous time values and MSIT01 is used to solve a trapezoidal corrector. This is performed by interfacing to MSIT01 similarly to SSBL03 except that the state error terms become:

$$\epsilon_{STATE} = \frac{STATE_{input} - STATE_{calculated}}{SNORM}$$

$STATE_{calculated}$ is determined from the previous time STATE and DSTATE, the current time DSTATE and the time increment DT as follows:

$STATE_{calculated} = STATE_{previous} + (DSTATE + DSTATE_{previous})$ X DT/2.

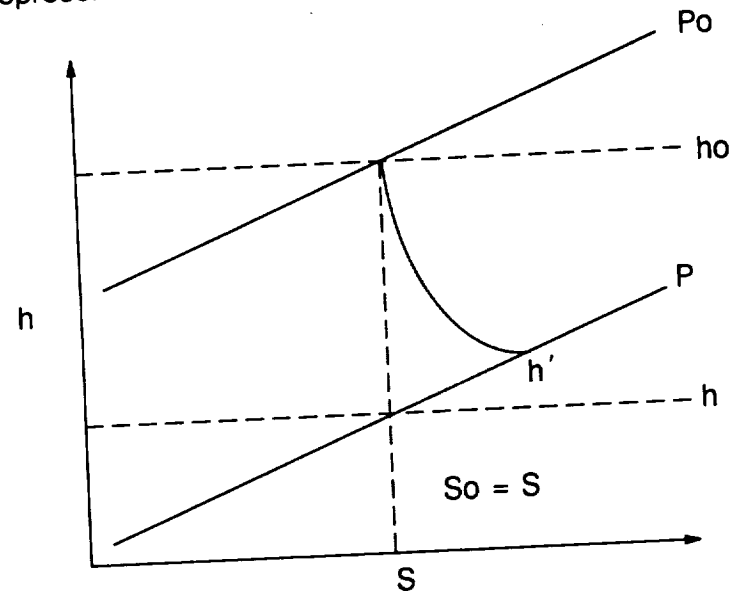
Because the current derivative is used in the calculation of $STATE_{calculated}$, SMIT01 is required.

Additionally, TRAP03 can solve multiple algebraic loops at the same time as solving for the states. This is accomplished by constructing the SMIT01 input arrays in the same manner as SSBL03, except that the error terms for the STATEs are as ϵ_{STATE} rather than DSTATE divided by STATE as in SSBL03. The error terms for the algebraic loops are calculated in the same way as in SSBL03.

Internal checking is performed so that the J matrix used by SMIT01 is consistent. If a state or algebraic loop is switched on or off during a run a new J matrix is immediately evaluated. If convergence fails the SMIT01 variables are reset to the initial values (previous converged point) and DT is reduced by a factor of DTRDCN and convergence is attempted again. DT may be reduced a total number MXDTRD times on any given integration step. The current values of DTRDCN and MXDTRD are both 3 therefore allowing a factor of 27 reduction in DT. If the point still fails, they are set equal to the initial failed values. TRAP03 will fail when the number of failures above exceeds the value of MXFAIL. MXFAIL is currently set at 5. DTRDCN, MXDTRD and MXFAIL are all internal variables to TRAP03.

SUBMODULE CDNZØØ : Isentropic Nozzle Analysis

Basically, all nozzles can be analyzed as an isentropic process which allows the use of the isentropic flow rationships. Hence, the expansion process of nozzles can be represented on an H-S diagram as drawn below:



A nozzle analysis requires the following input as known conditions:

They are: PR, AR, γ, P_o, T_o, Ath

The input parameters defined above represent pressure ratio (PR), area ratio (AR), ratio of specific heats (γ), chamber pressure (Po), chamber temperature (To) and throat area (Ath).

For a first test, we need to check and see if the PR is choke. The isentropic total to static pressure can be defined as:

$$\frac{P_o}{P_s} = \left(1 + \frac{\gamma - 1}{2} mn^2\right)^{\frac{\gamma}{\gamma-1}} \qquad (1)$$

At the throat, the choked PR occures at Mn = 1

$$\left(\frac{P_o}{P_s}\right)_{CHOKED} = \left(1 + \frac{\gamma - 1}{2} mn^2\right)^{\frac{\gamma}{\gamma-1}} = \left(\frac{\gamma + 1}{2}\right)^{\frac{\gamma}{\gamma-1}}$$

$$\therefore \left(\frac{P_o}{P_s}\right)_{CHOKED} = \left(\frac{\gamma + 1}{2}\right)^{\frac{\gamma}{\gamma-1}} \qquad (2)$$

If PR > $\left(\frac{P_o}{P_s}\right)_{CHOKED}$ then the nozzle is choked

For a second test, we need to check and see if the flow parameter at the throat (FP_{THROAT}) is choked. The flow parameter is defined as:

$$FP = \frac{W\sqrt{R\ T_o}}{P_o A} = \sqrt{\gamma}\frac{mn}{\sqrt{\left(1 + \frac{\gamma-1}{2}mn^2\right)^{\frac{\gamma+1}{\gamma-1}}}} \qquad (3)$$

The flow parameter defined by equation (2) is a function of γ and Mn. This relationship can also be defined as a function of γ and PR by defining Mn as function of PR.

Solving equation (1) for Mn as a F(PR).

$$1 + \frac{\gamma-1}{2}\,mn^2 = \left(\frac{P_o}{P_s}\right)^{\frac{\gamma-1}{\gamma}}$$

$$mn^2 = \frac{2}{\gamma-1}\left[\left(\frac{P_o}{P_s}\right)^{\frac{\gamma-1}{\gamma}} - 1\right]$$

$$\boxed{mn = \left\{\left(\frac{2}{\gamma-1}\right)\left[\left(\frac{P_o}{P_s}\right)^{\frac{\gamma-1}{\gamma}} - 1\right]\right\}^{\frac{1}{2}}} \qquad (4)$$

Now that a relationship for Mn as a F(PR) has been determined; it can be used to define the flow parameter as function of γ and PR. Let's substitute equation (4) into (3)

$$FP = \frac{\left\{\left(\frac{2\gamma}{\gamma-1}\right)\left[\left(\frac{P_o}{P_s}\right)^{\frac{\gamma-1}{\gamma}} - 1\right]\right\}^{\frac{1}{2}}}{\left\{1 + \left(\frac{\gamma-1}{2}\right)\left(\frac{2}{\gamma-1}\right)\left[\left(\frac{P_o}{P_s}\right)^{\frac{\gamma-1}{\gamma}} - 1\right]\right\}^{\frac{\gamma+1}{2(\gamma-1)}}}$$

Simplifying:

$$\boxed{FP = \frac{\left\{\frac{2\gamma}{\gamma-1}\left[\left(\frac{P_o}{P_s}\right)^{\frac{\gamma-1}{\gamma}} - 1\right]\right\}^{\frac{1}{2}}}{\left\{\left(\frac{P_o}{P_s}\right)^{\frac{\gamma+1}{\gamma}}\right\}^{\frac{1}{2}}}} \qquad (5)$$

The choked flow parameter can now be defined by substituting the choked pressure ratio defined by equation (2) into equation (5).

$$FP_{CHOKED} = \frac{\left(\frac{2\gamma}{\gamma-1}\right)^{\frac{1}{2}}\left[\left(\frac{\gamma+1}{2}\right)^{\frac{\gamma-1}{2}\cdot\frac{\gamma}{\gamma-1}} - 1\right]^{\frac{1}{2}}}{\left[\left(\frac{\gamma+1}{2}\right)^{\frac{\gamma+1}{\gamma-1}}\right]^{\frac{1}{2}}}$$

$$FP_{CHOKED} = \frac{\left(\frac{2\gamma}{\gamma-1}\right)^{\frac{1}{2}}\left(\frac{\gamma-1}{2}\right)^{\frac{1}{2}}}{\left\{\left(\frac{\gamma+1}{2}\right)^{\frac{\gamma+1}{\gamma-1}}\right\}^{\frac{1}{2}}}$$

Let the checked pressure term be defined as CHOKED

$$CHOKED = \left\{\left(\frac{\gamma-1}{2}\right)\left(\frac{\gamma+1}{2}\right)^{\frac{\gamma+1}{\gamma-1}}\right\}^{\frac{1}{2}}$$

$$\therefore FP_{CHOKED} = \left(\frac{2\gamma}{\gamma-1}\right)^{\frac{1}{2}} \cdot CHOKED$$

Let the nozzle pressure term be defined as EXIT:

$$EXIT = \left\{PR^{\frac{\gamma-1}{\gamma}} - 1\right\}^{\frac{1}{2}}$$

$$\therefore FP_{EXIT} = \left(\frac{2\gamma}{\gamma-1}\right)^{\frac{1}{2}} \cdot EXIT$$

Finally,

$$FP_{THROAT} = FP_{EXIT} \cdot AR$$

If $FP_{THROAT} > FP_{CHOKED}$ then the nozzle is choked.

UN-CHOKED nozzle

Let's calculate the exit Mach number using subroutine Mach03 which guess Mn_{EXIT} as a function of FP_{EXIT}.

$$mn_{EXIT} = F(FP_{EXIT}) \quad .$$

Vary Mn_{EXIT} until $(FP_{EXIT})_{CALC} - FP_{EXIT} \leq$ tolerance

The above balance uses subroutine ITER01 and calculates $(FP_{EXIT})_{CALC}$.

$$(FR_{EXIT})_{CALC} = \frac{\sqrt{\gamma}\ mn_{EXIT}}{\left(1 + \frac{\gamma-1}{2} mn^{\,2}{}_{EXIT}\right)^{\frac{\gamma+1}{2(\gamma-1)}}}$$

$$FP_{THROAT} = FP_{EXIT} * AR$$

$$\left(\frac{P_o}{P_s}\right)_{EXIT} = PR$$

Go calculate the thrust coefficient

CHOKED nozzle

If AR < 1 then

CONVERGENT AREA

$$mn_{EXIT} = 1.0$$

$$\left(\frac{P_o}{P_s}\right)_{EXIT} = PR_{CHOKED}$$

$$FP_{THROAT} = FP_{CHOKED} * AR$$

Go calculate the thrust coefficient

Else

DIVERGENT AREA

$$FP_{THROAT} = FP_{CHOKED}$$

$$\therefore\ FP_{EXIT} = FP_{CHOKED} \,/\, AR$$

Since this is a divergent nozzle, the analysis needs to check for internal shocks. Let's calculate the exit Mach number using subroutine Mach03 which guess Mn_{EXIT} as a function of FP_{EXIT}.

$$mn_{EXIT} = F(FP_{EXIT})$$

Vary Mn_{EXIT} until $(FP_{EXIT})_{CALC} - FP_{EXIT} \leq$ tolerance

$$\left(\frac{P_o}{P_s}\right)_{EXIT} = \left(1 + \frac{\gamma - 1}{2}\, mn^2_{EXIT}\right)^{\frac{\gamma}{\gamma-1}}$$

1st test:
$P_{EXIT} > P_{AMB}$

$$RPR = \frac{PR}{\left(\frac{P_o}{P_s}\right)_{EXIT}}$$

If RPR $\geq$ 1 then

The exit pressure (P_{EXIT}) is greater than ambient pressure (P_{AMB}) which indicates that there is no shock in the nozzle. Go calculate the thrust coefficient.

$$Ps2 = \frac{2\gamma mn^2_{EXIT}}{\gamma + 1} - \frac{\gamma - 1}{\gamma + 1}$$

2nd test P_{EXIT} (after shock > P_{AMB})

$$RPRS = \left(\frac{Ps2}{Ps1}\right) * RPR$$

If RPRS > 1, then

The exit static pressure (P_{EXIT}) is greater than ambient pressure are the check. Hence the nozzle does not have a shock wave. Go calculate the thrust coefficient.

A <u>normal shock</u> exist in the nozzle. Let's guess Mn, (Mach number upstream of the shock).

$$mn_1 = 1 + \frac{(mn_{EXIT} - 1)}{2}$$

This allow the total to total pressure ratio (P_{T1} / P_{T2}) across the shock to calculate.

$$\left(\frac{T_o}{T_s}\right)_1 = 1 + \frac{\gamma - 1}{2} mn_1^2$$

$$\frac{P_{T2}}{P_{T1}} = \left[\frac{\frac{\gamma+1}{2} mn_1^2}{1 + \frac{\gamma-1}{2} mn_1^2} \right]^{\frac{\gamma}{\gamma-1}} \Bigg/ \left[\frac{2\gamma}{\gamma-1} mn_1^2 - \frac{\gamma-1}{\gamma+1} \right]^{\frac{1}{\gamma-1}}$$

Calculating exit Mach number downstream of the shock.

$$FP_{EXIT} = \frac{FP_{THROAT}}{\left(\frac{P_{T_2}}{P_{T_1}}\right) AR}$$

Mn_{EXIT} = F (FP_{EXIT}) using subroutine Mach03

$$\left(\frac{P_o}{P_s}\right)_{EXIT} = \left(1 + \frac{\gamma - 1}{2} mn_{EXIT}^2\right)^{\frac{\gamma}{\gamma-1}}$$

Vary Mn_1 until $[(Po/Ps)_{EXIT} - (PR)\ (\ P_{T2} \setminus P_{T1}\ )] \leq$ tolerance

$$\left(\frac{P_o}{P_s}\right)_{EXIT} = PR$$

Go calculate the thrust coefficient.

Calculation of The Thrust Coefficient

$$\frac{P_{s_{EXIT}}}{P_o} = \frac{1}{\left(\frac{P_o}{P_s}\right)_{EXIT}}$$

The ideal throat coefficient { with $P_{EXIT} = P_{AMB}$) is defined as:

$$CF_{IDL} = \gamma \left\{ \left(\frac{2}{\gamma-1}\right)\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}} \left[1 - (1/PR)^{\frac{\gamma-1}{\gamma}} \right] \right\}^{\frac{1}{2}}$$

The ideal thrust coefficient (no real losses, but includes expansion losses) is defined as:

$$CF' = FP_{THROAT}\ mn_{EXIT}\left\{\frac{\gamma}{g_2\left(1 + \frac{\gamma-1}{2}\ m^2_{EXIT}\right)}\right\}^{\frac{1}{2}}$$

$$+ AR\left(\frac{P_{s_{EXIT}}}{P_o} - \frac{1}{PR}\right)$$

$$Loss_{REAL} = CF'_{VAC}(1 - CS)$$

Velocity coefficient is defined as:

$$CV = (CF' - LOSS_{REAL})/CF_{IDL}$$

SUBMODULE FLPMØ2 - Flow Calculation Through A Restriction

It is often necessary to include the effects of a flow path restriction in a simulation. This involves developing a relationship between flow and total pressure loss.

FLOW CALCULATION:

The general problem of calculating the mass flow rate through a restriction can be simplified considerably if the restriction is considered to be an orifice. This reduces the scope of the problem from developing a general theory applicable to all possible geometries to deriving a calculational procedure for only one geometry. The general situation is thus reduced to calculation of flow through an orifice and proper sizing of an orifice.

As flow moves through an orifice it is first accelerated until it attains a maximum velocity at the throat. The flow is then diffused as it is decelerated to the downstream conditions. During this process, the total and static pressures change as depicted in Figure I. There is little loss in total pressure as the flow nears the throat, although static pressure changes considerably due to the increased fluid velocity. Most of the total pressure loss occurs as the fluid is diffused downstream of the throat. The static pressure is increasing, however, as a portion of the velocity head available at the throat is converted to pressure. Total temperature does not change throughout the process.

Typically, the upstream and downstream total pressures and temperatures are known, from which flow is to be determined. Since the total pressure and temperature at the throat is nearly the same as upstream, flow could be calculated using compressible flow relations if static pressure at the throat were known.

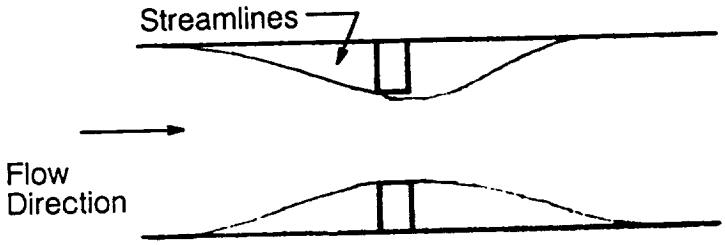
Static pressure at the throat can be calculated if it is assumed that the total pressure loss across the orifice is related to the velocity head at the throat. Numerous experiments have been conducted which justify this assumption, although most of the empirical data is presented in terms of discharge and flow coefficients. Discharge and flow coefficients are useful for control valve work but are generally restricted in their range of applicability. It is more convenient, therefore, to define a loss coefficient from which static pressure at the throat can be determined.

Referring to Figure II, the loss coefficient is defined as:

$$L = \frac{Total\ pressure\ loss}{Velocity\ head\ at\ throat} = \frac{P_{T\ up} - P_{T\ DOWN}}{P_{T\ up} - P_{S\ THROAT}}$$

Rearranging, the total to static pressure ratio at the throat is given by

$$\frac{P_{T_{UP}}}{P_{s_{throat}}} = \frac{PR(L)}{1 + PR\ (L-1)}$$



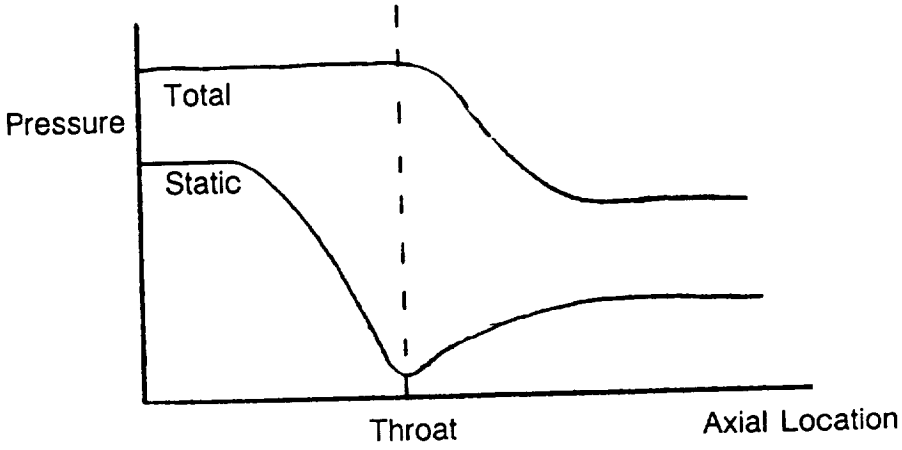


Figure I. Total and Static Pressure Variation in an Orifice

where PR = $P_{T\ up}/P_{T\ down}$ and it is assumed the orifice is not choked. If the calculated total to static pressure ratio is greater than the choke value, then the orifice is choked, in which case, the actual total to static pressure ratio at the throat is equal to the choke value.

With the total to static pressure ratio known, the flow parameter $(W\sqrt{RT_t}/P_tA)$ can be determined at the throat based on compressible flow relations;

$$W\frac{\sqrt{RT_T}}{P_TA} = FP = \frac{\sqrt{2g_c\gamma\left(PR^{\frac{\gamma-1}{\gamma}}-1\right)}}{(\gamma-1)\ PR^{\frac{\gamma+1}{\gamma}}}$$

C 2

Since total temperature remains constant, and total pressure at the throat is assumed equal to the upstream total pressure, the mass flow rate can be determined,

$$W = FP\frac{(P_{T\,up})(A_{eff})}{R(T_{t\,up})}$$

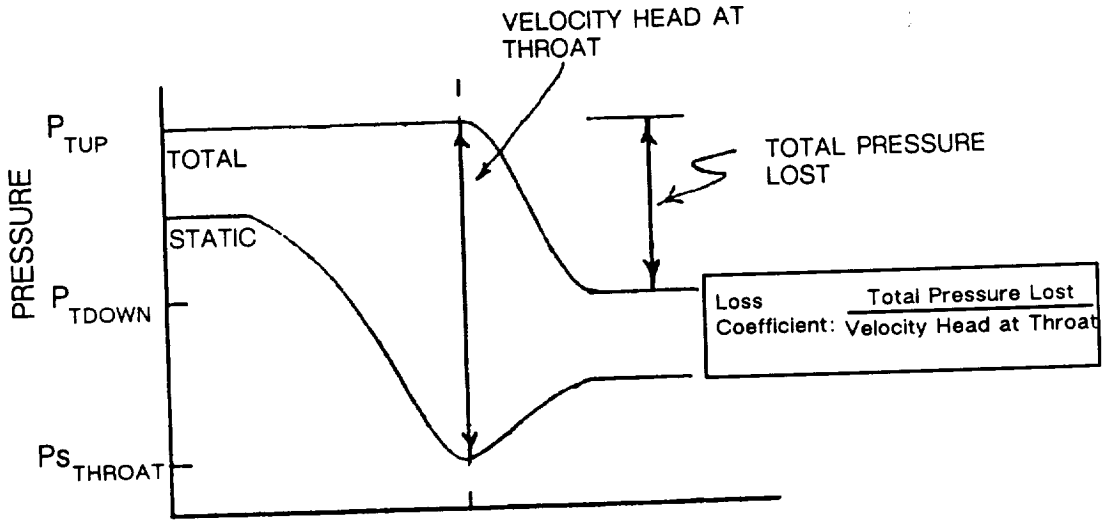
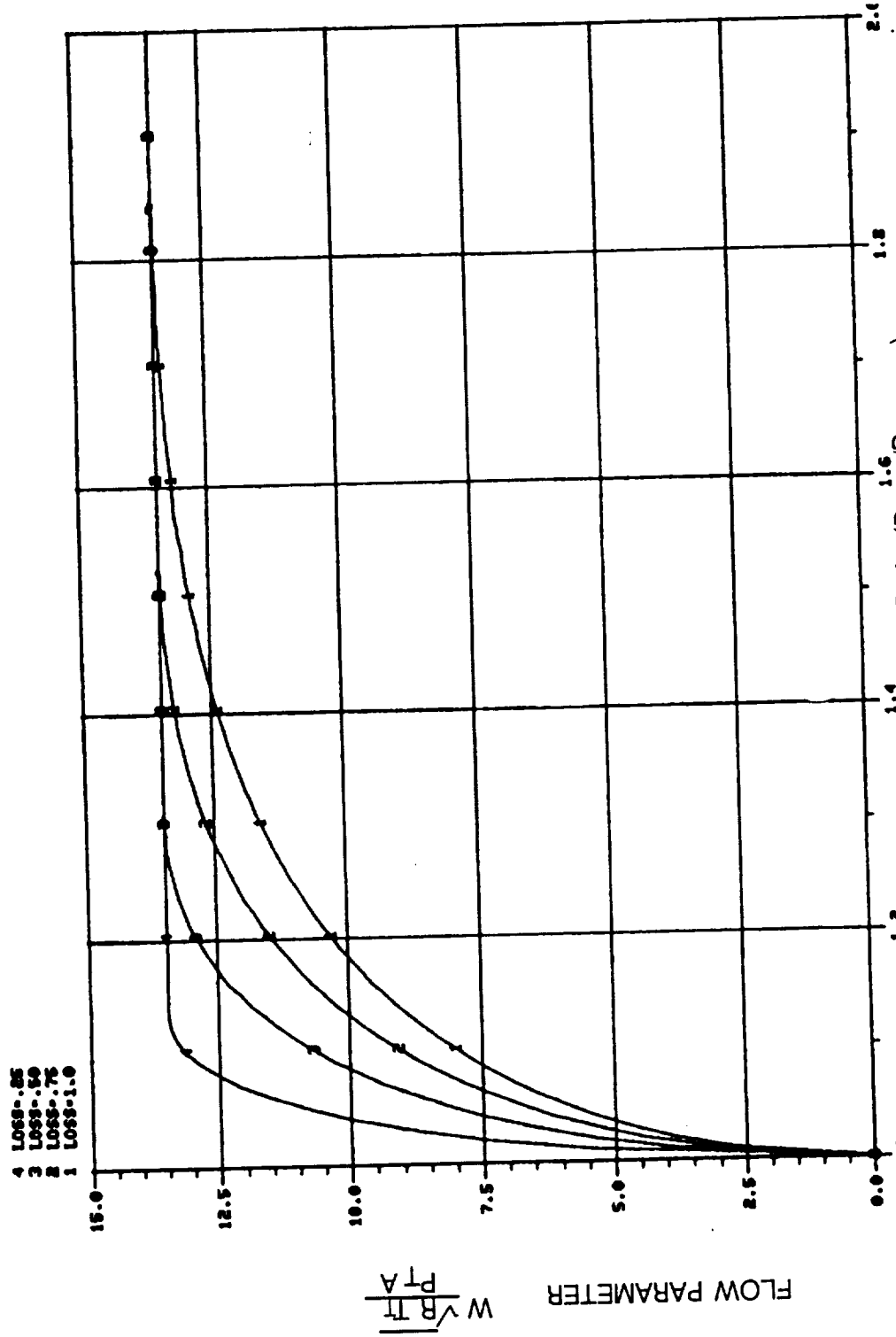
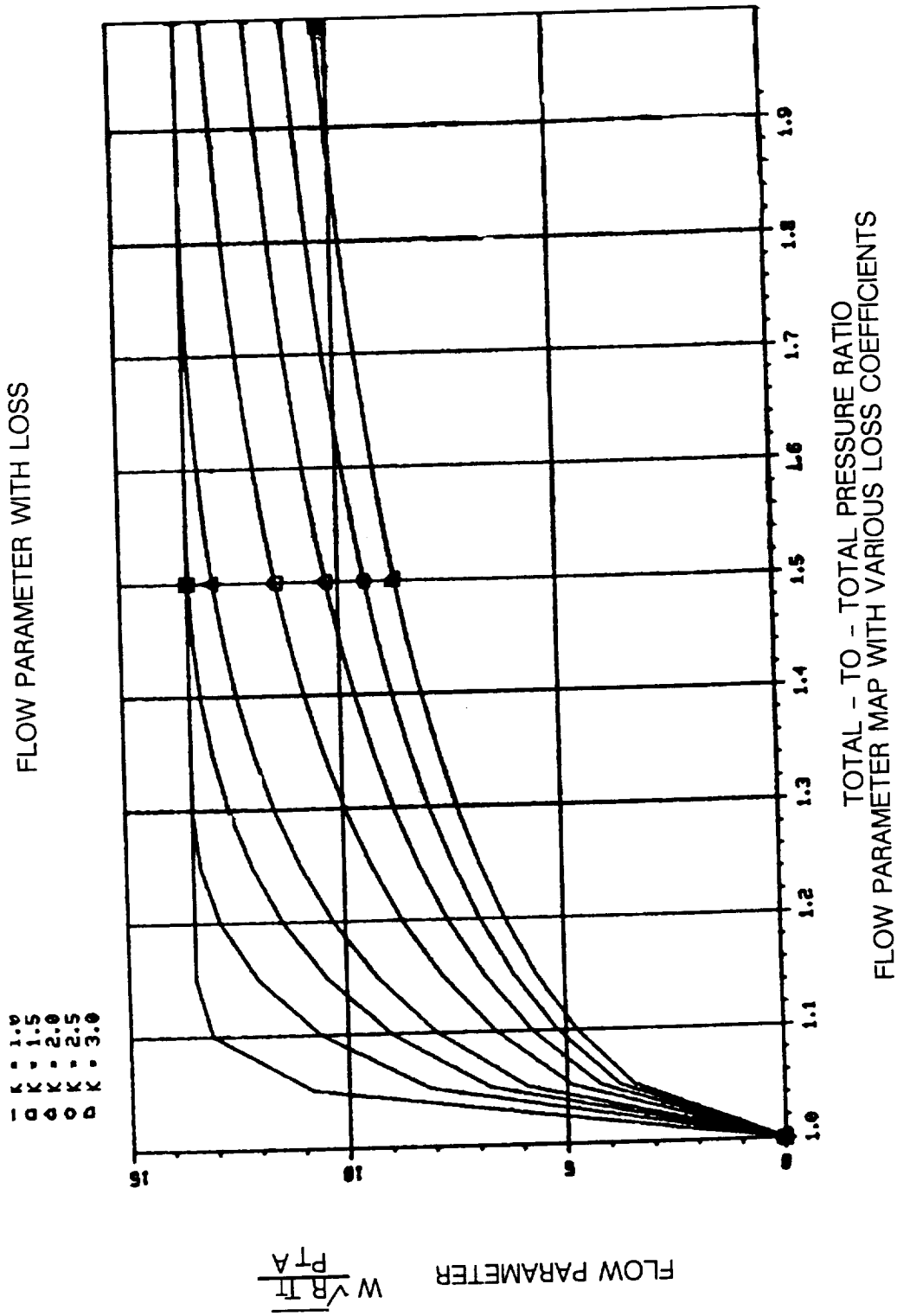
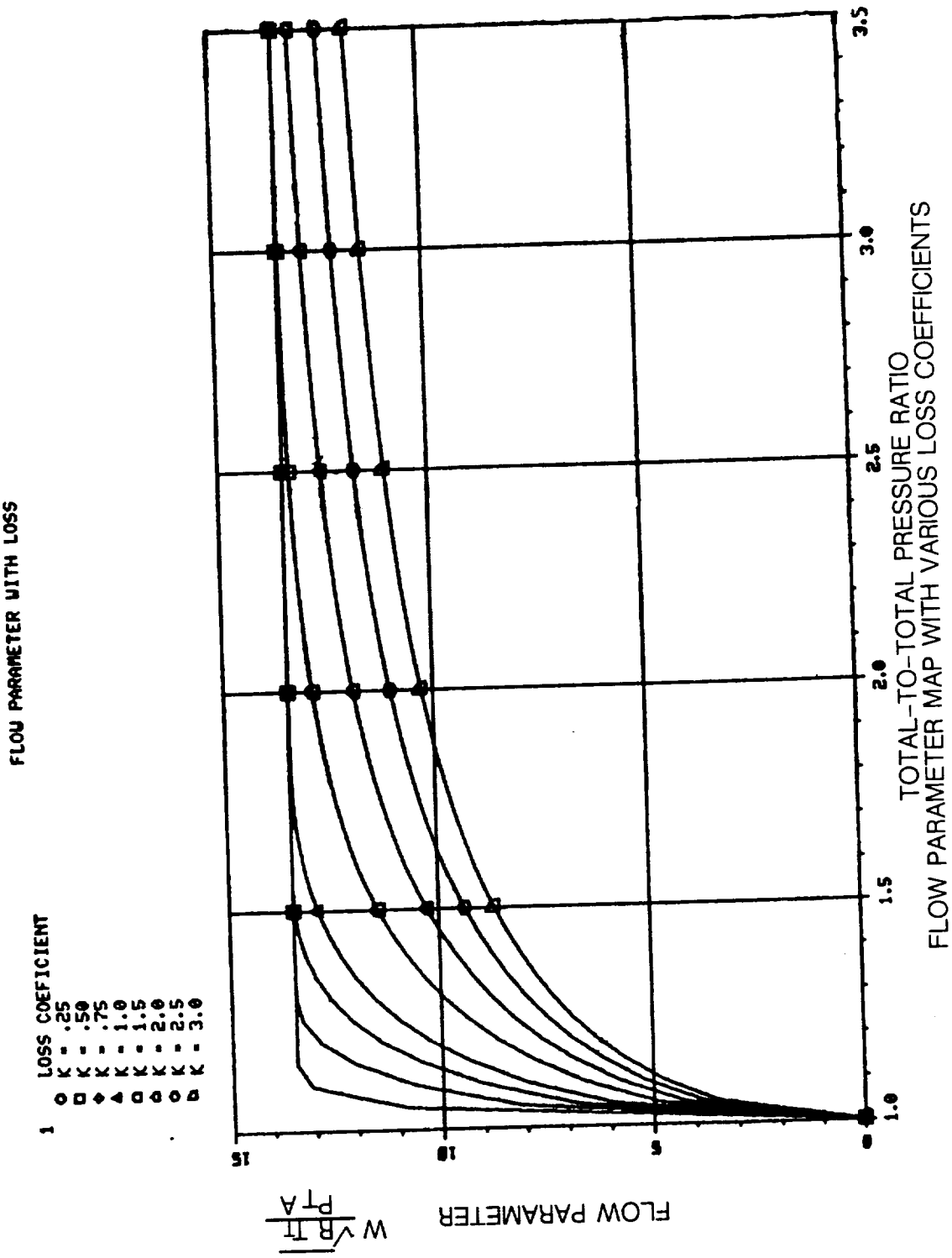


Figure II. Loss coefficient relates total pressure lost to velocity head at the orifice throat



Flow Parameter MAP with Various Loss Coefficients


FLOW PARAMETER WITH LOSS
TOTAL - TO - TOTAL PRESSURE RATIO
FLOW PARAMETER MAP WITH VARIOUS LOSS COEFFICIENTS
K = 1.0
K = 1.5
K = 2.0
K = 2.5
K = 3.0
FLOW PARAMETER
W √RT / PTA
15
10
5
0
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9


FLOW PARAMETER WITH LOSS
LOSS COEFICIENT
K = .25
K = .50
K = .75
K = 1.0
K = 1.5
K = 2.0
K = 2.5
K = 3.0
TOTAL-TO-TOTAL PRESSURE RATIO
FLOW PARAMETER MAP WITH VARIOUS LOSS COEFFICIENTS
FLOW PARAMETER
W √RT / PTA

ROCETS REAL FLUID AND COMBUSTION PROPERTY PACKAGES

The ROCETS property package consists of five real fluid properties and one "perfect gas" combustion property. Both property packages consist of a driver subroutine and a property/model interface subroutine for each of the five fluids and the combustion method. The driver routine is the decision maker and then routes all information to the correct property or model interface which the accesses the property maps for the real fluid properties or the subroutines for the combustion properties to look up the desired results. The program logic of the ROCETS real fluid properties and the combustion property package is shown Figures 3.4.2.4.1 and 3.4.2.4.2 respectively. The five basic properties are para-hydrogen, oxygen, methane, nitrogen and helium. The combustion properties provide a perfect gas combustion analysis using hydrogen and oxygen as the reactants. The driver has been set up such that it can be expanded to allow real gas combustion and Brinkly combustion at some future date. The combustion properties are from the digital transient model (DTM), but have been upgraded to meet ROCETS software standards.

In the combustion property package, the driver receives basic inputs such as model type (MTYPE), property option (IOPT), the independent variables (XVAR1, XVAR2, XVAR3 and XVAR4) and the map indicies IHOLD1 and IHOLD2. This information is then passed to the model interface which accesses the required information from the combustion property subroutines and maps. The call statement for the driver routine has the following form:

CALL COMB00 (MTYPE, IOPT, XVAR1, XVAR2, XVAR3, XVAR4, OVAR1, OVAR2, IHOLD1, IHOLD2)

It should be noted that MTYPE is a character*8 variable which defines the method to be used. For example, if the user wishes to access combustion properties using a perfect gas model, they would define MTYPE = 'PGAS '.

Once the combustion property model type has been defined, the correct model interface is accessed. The model interface routines have the following form:

CALL XGASAA (IOPT, XVARI, XVAR2, XVAR3, XVAR4, OVAR1, OVAR2, IHOLD1, IHOLD2)

The single character prefix X defines the model to be used and two character suffix defines the version number. The following models are available or are planned: P-perfect gas model (available), R-Real gas model (not available) and B-Brinkly model (not available). A complete list of all options and parameters used in the argument list are defined in Table 3.4.2.4.4 and the units are defined in Table 3.4.2.4.5.

The perfect gas model interface defined above obtains property information in both subroutines and property maps from the Digital Transient Model (DTM). However, the subroutines found in the DTM have been upgraded to ROCETS software standards and the maps which are from the DTM input file were put into subroutine format. Basically, the subroutines which are called from the model interface access results from tabular data which are in uni-variant or bi-variant form. The univariant maps re set up in x-axis and y-axis coordinate system and use a linear map reader (SUNBØØ) to interpolate. The bi-variant maps are set up in an x-axis, y-axis and z-axis coordinate system and use a corresponding point map reader (CPMRØ2) to interpolate. The map indices IHOLD1 and IHOLD2 care used to track the corresponding map locations and are responsible for a significant reduction in the cost of accessing the maps.

At this stage of development, the real gas and Brinkly gas combustion models are not available. However, as they are implemented into the ROCETS system, these models will be discussed in greater detail.

The ROCETS combustion property package has 7 property maps that are used to drive the perfect gas model interface. The real gas and Brinkly gas interface have not yet been implemented. The following tables define the subroutines and maps which are used.

COMBUSTION PROPERTY ROUTINES

PERFECT GAS	REAL GAS	BRINKLY GAS
CPGMØØ	NONE	NONE
CPMPØ1		
RDMP13		
RDMP21		
RDMP12		
ZGASØØ		
ZZMPØ1		

These maps are presented in figures 3.4.2.4.27 - 3.4.2.4.31. The perfect gas combustion property package allows the calculation of basic thermodynamic properties which are corrected for compressibility effects for the hot gas system components such as the preburner, turbine, injectors and the main chamber.

In the basic fluid property package, the driver routine receives basic property inputs such as fluid type (FLUID), property option (IOPT), the independent

variables (XVAR1 and XVAR2) and the station niumber (KSTA). The user supplied information is then passed to the particular property interface which accesses the required information from the property maps. The call statement for the driver routine has the following from:

CALL PROP00 (FLUID, IOPT, XVAR1, XVAR2, PVAR, OVAR1, OVAR2, KSTA)

It should be noted that FLUID is a character*8 variable which defines the fluid by name. For example, if the user whishes to access hydrogen property data, he would define FLUID = 'HYDROGEN'.

Once the fluid property has been defined, the correct property interface can be accessed. The property interface routines for the five basic properties have the following form:

CALL XXPROP (IOPT, XVAR1, XVAR2, DVAR, OVAR1, OVAR2)

The two character prefix XX defines the fluid to be used. The following fluids are available: H2 - para-hydrogen, O2 - oxygen, N2 - nitrogen, HE - helium and ME - methane. A complete list of all options and parameters used in the argument list are defined in Table 3.4.2.4.1 and units are defined in Table 3.4.2.4.2.

Each property interface defined above obtains property data from a series of real fluid property maps. The call statements to the maps follow a basic format which uses a one letter prefix to define the fluid, a three character map variable name and two character version number. The map variables define how a particular map is set up. Let's consider an example for a hydrogen map. The fluid prefix is H. Hence, the call for a hydrogen map whould have the form HXYZAA. A hydrogen map call would have the collowing form:

CALL HXYZAA (XX, YY, ZZ, IOPT, IHLD1, IHLD2)

All property maps are set up in an x-axis, y-axis and z-axis coordinate system and uses a corresponding point map reader (CPMR02) to interpolate. The advantage of a corresponding point reader is that a family of discontinuous curves such as a property map can be adequately handled. The map reader has two options. Option one determines zz as a function of xx and yy. Option two determines xx as a function of zz and yy. The integer variables IHLD1 and IHLD2 are map indicies which are stored internally for each call to a particular property interface routine by specifying KSTA.

These indicies significantly reduces the amount of CPU time required to access information from the maps.

The ROCETS property package has 19 property data maps that are used to drive the interface routines. The following table defines the maps which are available:

REAL FLUID PROPERTY ROUTINES

PARA-HYDROGEN	OXYGEN	METHANE	NITROGEN	HELIUM
HRHP01	ORHP01	MRPH01	NRHP01	ERHP01
HRUP01	ORUP01	MTPH01	NPHT01	EPHT01
HPUT01	OPUT01	MHPS01	NHPS01	EHPS01
HHPS01	OHPS01	MCPT01		
		MCVT01		

These maps are presented in plot form in Figures 3.4.2.4.3 - 3.4.2.4.26 and are restricted to the pressure and temperature ranges defined in Table 3.4.2.4.3. It should be noted that MCPT and MCVT define Cp and Cv as a function of temperature with lines of constant pressure.

This particular property package has also been optimized to minimize cost. This was accomplished by tracking the indices of the property maps for up to 100 calls for each fluid by storing unique integer values assigned to KSTA for each call to a particular fluid. A speed trial was performed which ran 100,000 calls to PROP00 for para-hydrogen on the IBM 3083 for both an iterative and noniterative solution. The iterative solution required 510 seconds of real time and 1.14 CRU's of execution time. The noniterative solution required 207 seconds of real time and 0.46 CRU's of execution time.

TABLE 3.4.2.4.1

PROPERTY INTERFACE PARAMETERS OPTIONS

IOPT	XVAR1	XVAR2	DVAR	OVAR1	OVAR2
1	PRES	TEMP	ENTH	DENS	TEMP
2	PRES	TEMP	ENTR	DENS	TEMP
4	PRES	ENTH	ENTR	DENS	TEMP
5	PRES	ENTR	TEMP	DENS	TEMP
6	PRES	ENTR	TEMP	DENS	TEMP
7	PRES	TEMP	DENS	DENS	TEMP
8	PRES	TEMP	CP	GAMMA	DENS
9	PRES	TEMP	TH	CONDVISC	DENS
10	TEMP	DENS	PRES	ENTR	ENTH
11	PRES	DENS	ENTH	TEMP	ENTR
12	DENS	ENTH	PRES	--	--
13	PRES	ENTH	DENS	--	--
14	DENS	INT.ENERGY	PRES	--	--
15	PRES	INT.ENERGY	DENS	--	--

NOTE: Of the eleven options above, only (9) for helium and methane are not available.

TABLE 3.4.2.4.2

PARAMETERS AND UNITS USED IN XXPROP

PARAMETER		UNITS	
		ENGLISH	SI
PRESSURE	-	Lbf/IN^2	N/m^2
TEMPERATURE	-	DEG R	DEG K
DENSITY	-	LBM/IN**3	KG/m^3
ENTHALPY	-	BTU/LBM	J/Kg
INTERNAL ENERGY	-	BTU/LBM	J/Kg
CP	-	BTU/IN-SEC-DEG R	J/Kg - DEG K
THERM.COND.	-	BTU/FT-HR-R	J/m - SEC - DEG K
VISCOSITY	-	LBM/FT-SEC	KG/m - SEC
ENTROPY	-	BTU/LBM-DEG R	J/Kg - DEG K

TABLE 3.4.2.4.3

NBS PROPERTY PRESSURE AND TEMPERATURE LIMITS

FLUID PROPERTY	PRESSURE RANGE (Lbf/IN^2)	TEMPERATURE RANGE (DEG R)
Para-hydrogen	0.0 - 17,404	24.8 - 720.0
Oxygen	0.0 - 17,404	97.8 - 720.0
Nitrogen	0.0 - 20,000	113.7 - 3420.0
Helium	0.0 - 14,500	5.4 - 2,700.0
Methane	1.7 - 10,000	163.2 - 900.0

TABLE 3.4.2.4.4

COMBUSTION PROPERTY INTERFACE PARAMETER OPTIONS

IOPT	XVAR1	XVAR2	XVAR3	XVAR4	OVAR1	OVAR2
1	PT	TT	OFR	PR	CP	GAMA
2	OFR	NA	NA	NA	RGAS	XMW
3	PT	TT	OFR	NA	Z	NA
4	RHO	TT	OFR	NA	PT	RGAS

TABLE 3.4.2.4.5

PARAMETERS AND UNITS USED IN XGASAA

PARAMETER (ARG)	UNITS ENGLISH	UNITS SI
PRESSURE (PT)	Lbf/IN^2	N/m^2
TEMPERATURE (TT)	°R	°K
DENSITY (RHO)	Lbm/IN^3	Kg/m^3
GAS CONSTANT (RGAS)	Lbf IN/Lbm °R	N/m/(Kg°K)
SPECIFIC HEAT RATIO (GAMA)	D'LESS	D'LESS
SPECIFIC HEAT (CP)	Btu/Lbm°R)	J/(Kg°K)
COMPRESSIBILITY (Z)	D'LESS	D'LESS
MOLECULAR WEIGHT (XMW)	Lbm/Lbm-MOLE	Kg/Kg-MOLE
PRESSURE RATIO (PR)	D'LESS	D'LESS
OXIDIZER FRACTION (OFR)	D'LESS	D'LESS

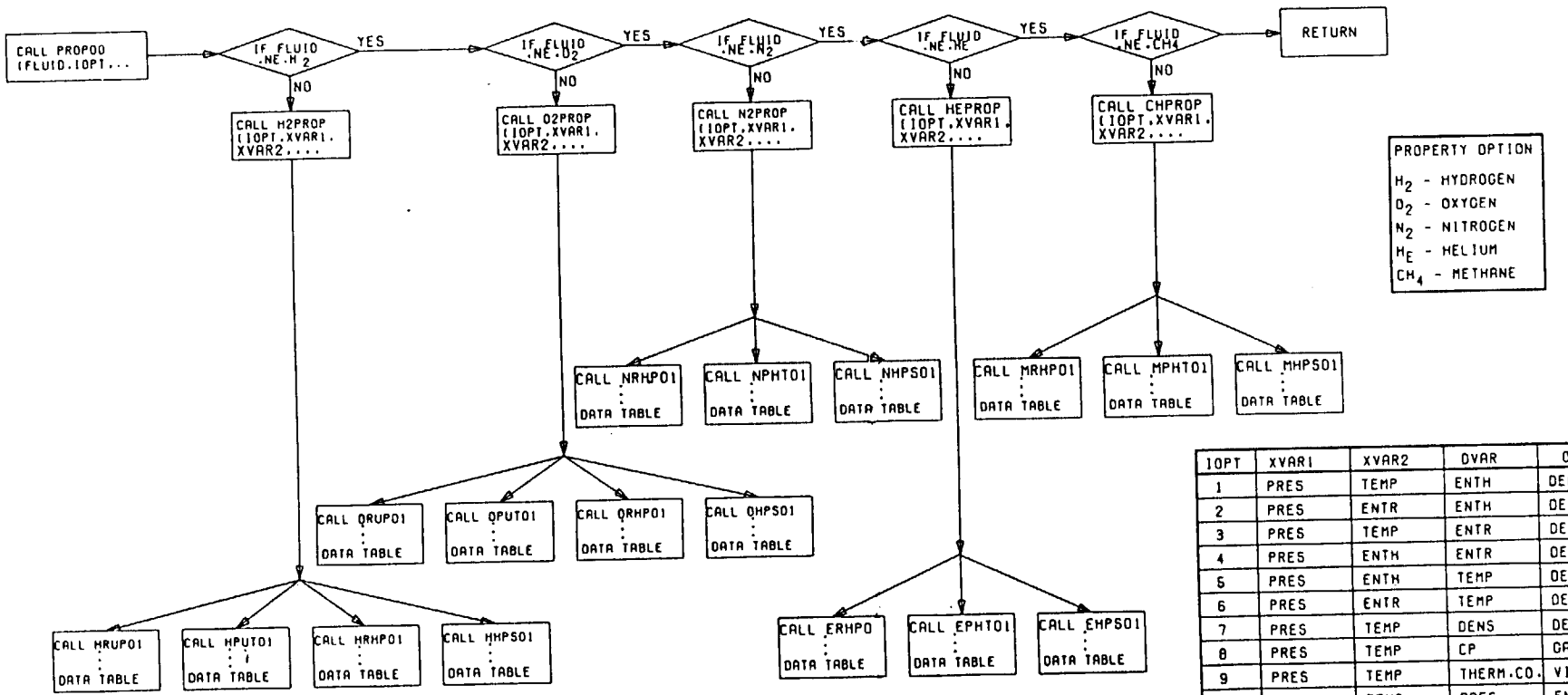
TABLE 3.4.2.4.6

SUBROUTINE PARAMETERS AND RANGES

SUBROUTINE	PARAMETER	RANGE	UNITS
ZZMPØ1	PT	14.7 - 2000	Lbf/IN2
	TT	50 - 400	°R
	ZZ	0.45 - 1.55	D'LESS
RDMP12	OFR	0 - 9999	D'LESS
	XMW	2 - 32	Lbm/Lbm-mole
CPMPØ1 (1)	PT	50 - 5000	Lbg/IN2
	TT	100 - 2000	°R
	PR	1 - 4	D'LESS
	GAMA	1.4	D'LESS
RDMP21	OFR	0 - 1.0	D'LESS
	CP	0.22 - 3.6	Btu/(Lbm°R)
RDMP13	OFR	0 - 1.0	D'LESS
	GAMA	1.2 - 1.4	D'LESS

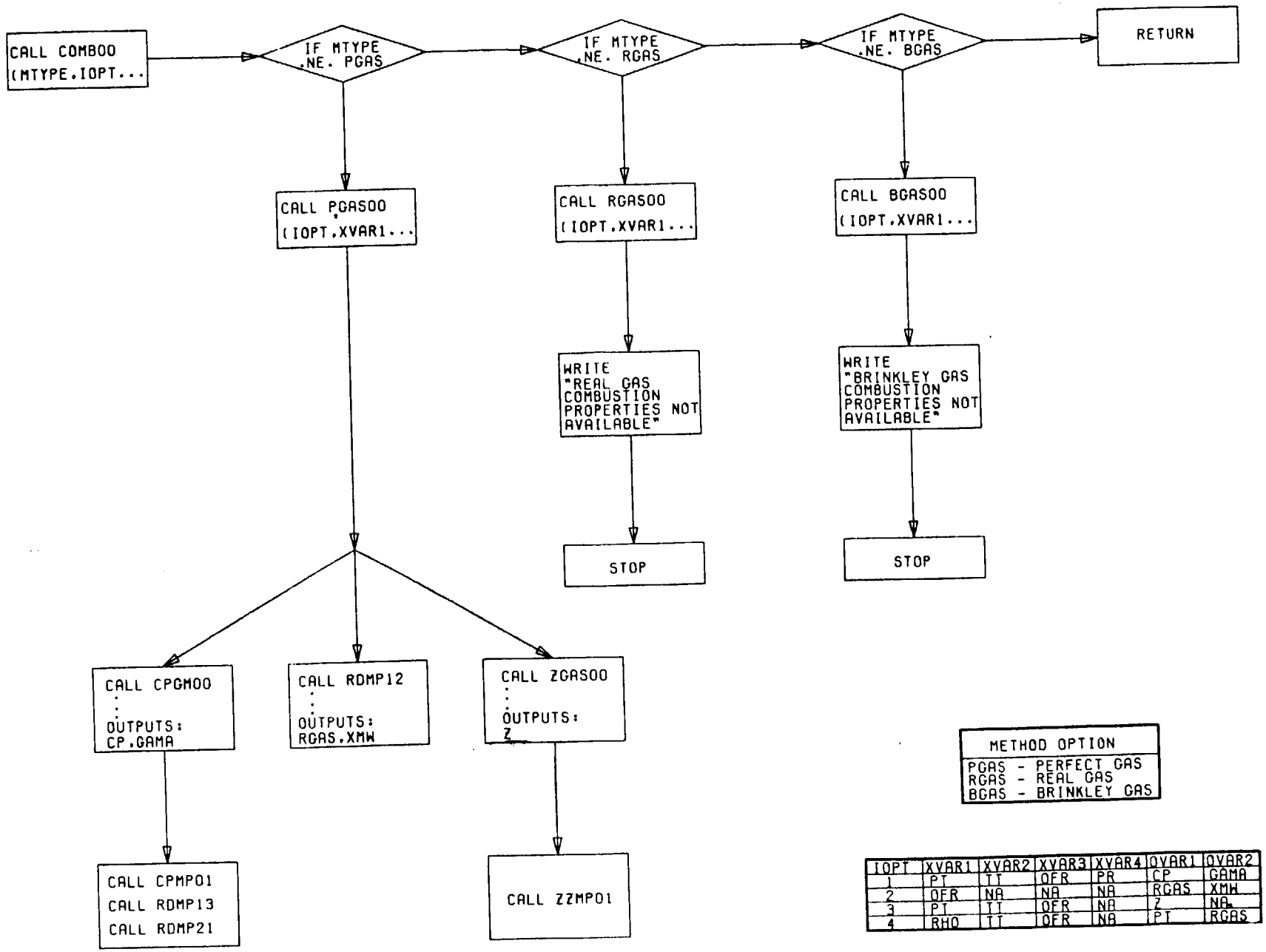
NOTE: (1) C_P agrees with H_2 (NBS) within +/- 2% when inlet pressure (for turbines) is less than 2000 PSIA and within +/- 60% when inlet pressure is less than 5000 PSIA.

ROCETS Real Fluid Property Package



IOPT	XVAR1	XVAR2	DVAR	OVAR1	OVAR2
1	PRES	TEMP	ENTH	DENS	TEMP
2	PRES	ENTR	ENTH	DENS	TEMP
3	PRES	TEMP	ENTR	DENS	TEMP
4	PRES	ENTH	ENTR	DENS	TEMP
5	PRES	ENTH	TEMP	DENS	TEMP
6	PRES	ENTR	TEMP	DENS	TEMP
7	PRES	TEMP	DENS	DENS	TEMP
8	PRES	TEMP	CP	GAMMA	DENS
9	PRES	TEMP	THERM.CO.	VISCOSITY	DENS
10	TEMP	DENS	PRES	ENTR	ENTH
11	PRES	DENS	ENTH	TEMP	ENTR
12	DENS	ENTH	PRES	XXXX	XXXX
13	PRES	ENTH	DENS	XXXX	XXXX
14	DENS	INTERNAL ENERGY	PRES	XXXX	XXXX
15	PRES	INTERNAL ENERGY	DENS	XXXX	XXXX

ROCETS Combustion Property Package



METHOD OPTION
PGAS - PERFECT GAS
RGAS - REAL GAS
BGAS - BRINKLEY GAS

IOPT	XVAR1	XVAR2	XVAR3	XVAR4	OVAR1	OVAR2
1	PT	TT	OFR	PR	CP	GAMA
2	OFR	NA	NA	NA	RGAS	XMW
3	PT	TT	OFR	NA	Z	NA
4	RHO	TT	OFR	NA	PT	RGAS

PRATT & WHITNEY

HYDROGEN PROPERTY MAP

DENSITY VS PRESSURE AT CONSTANT U

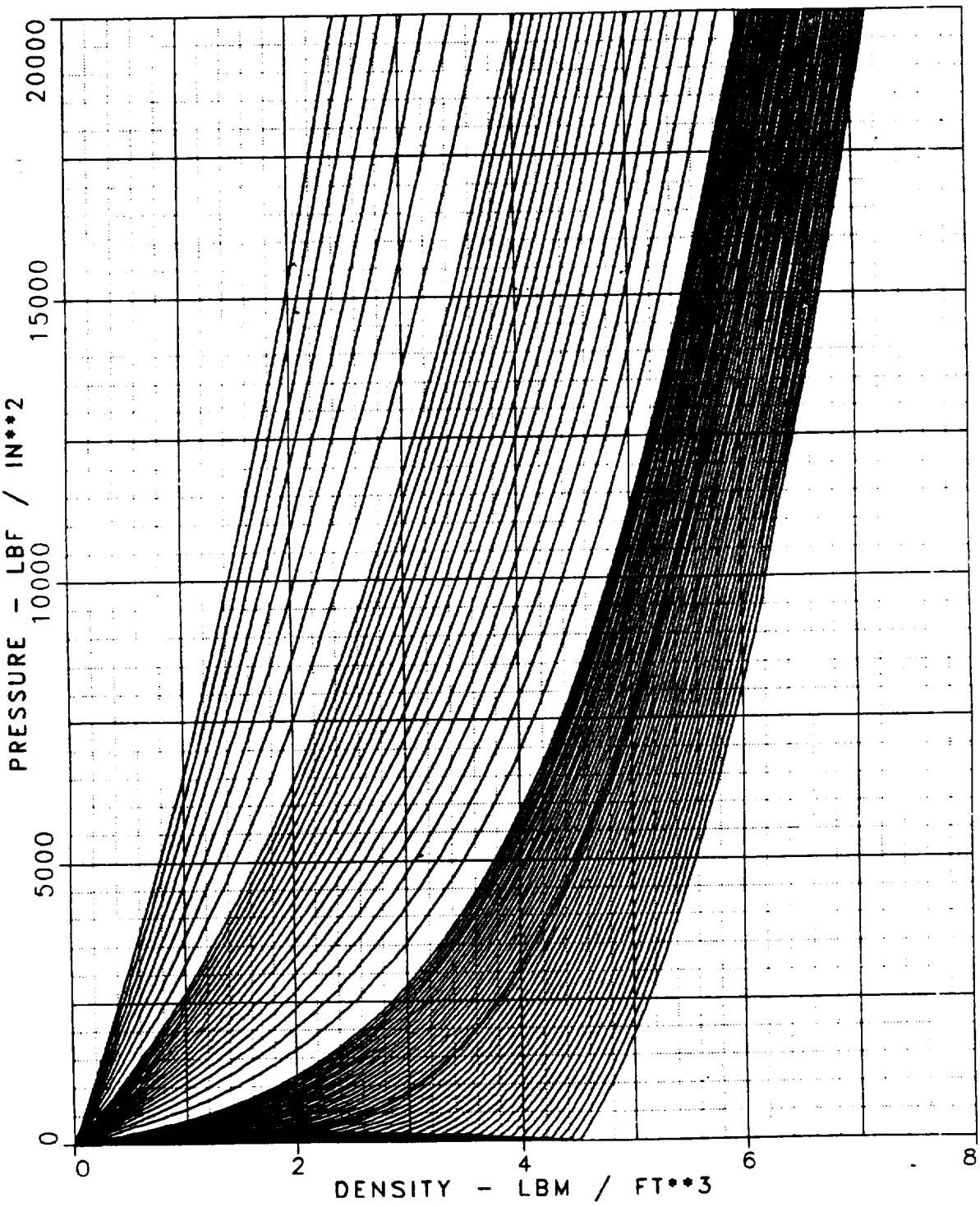


FIGURE 3.4.2.4.3

PRATT & WHITNEY

HYDROGEN PROPERTY MAP

DENSITY VS PRESSURE AT CONSTANT ENTHALPY

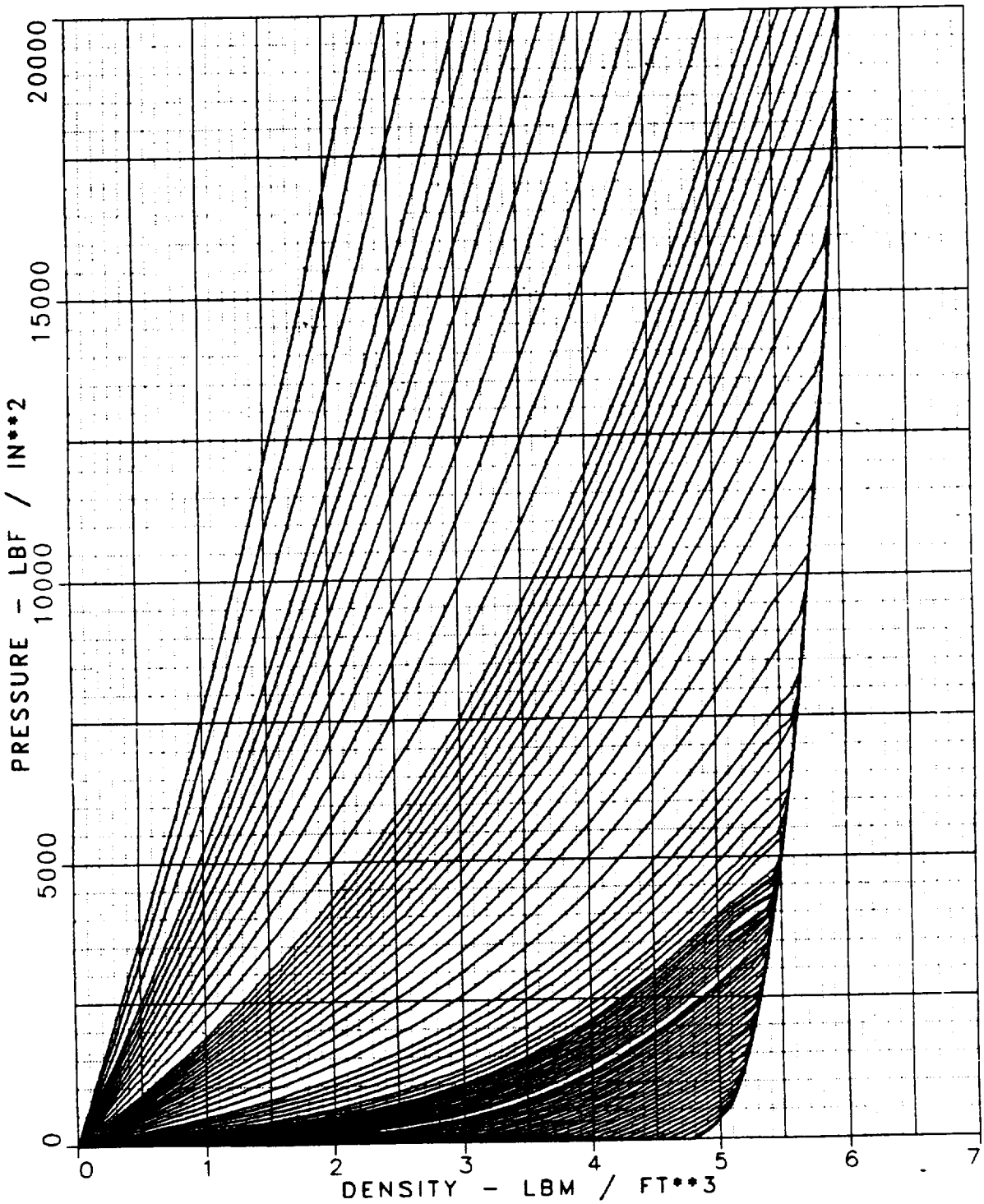


FIGURE 3.4.2.4.4

PRATT & WHITNEY

HYDROGEN PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT U

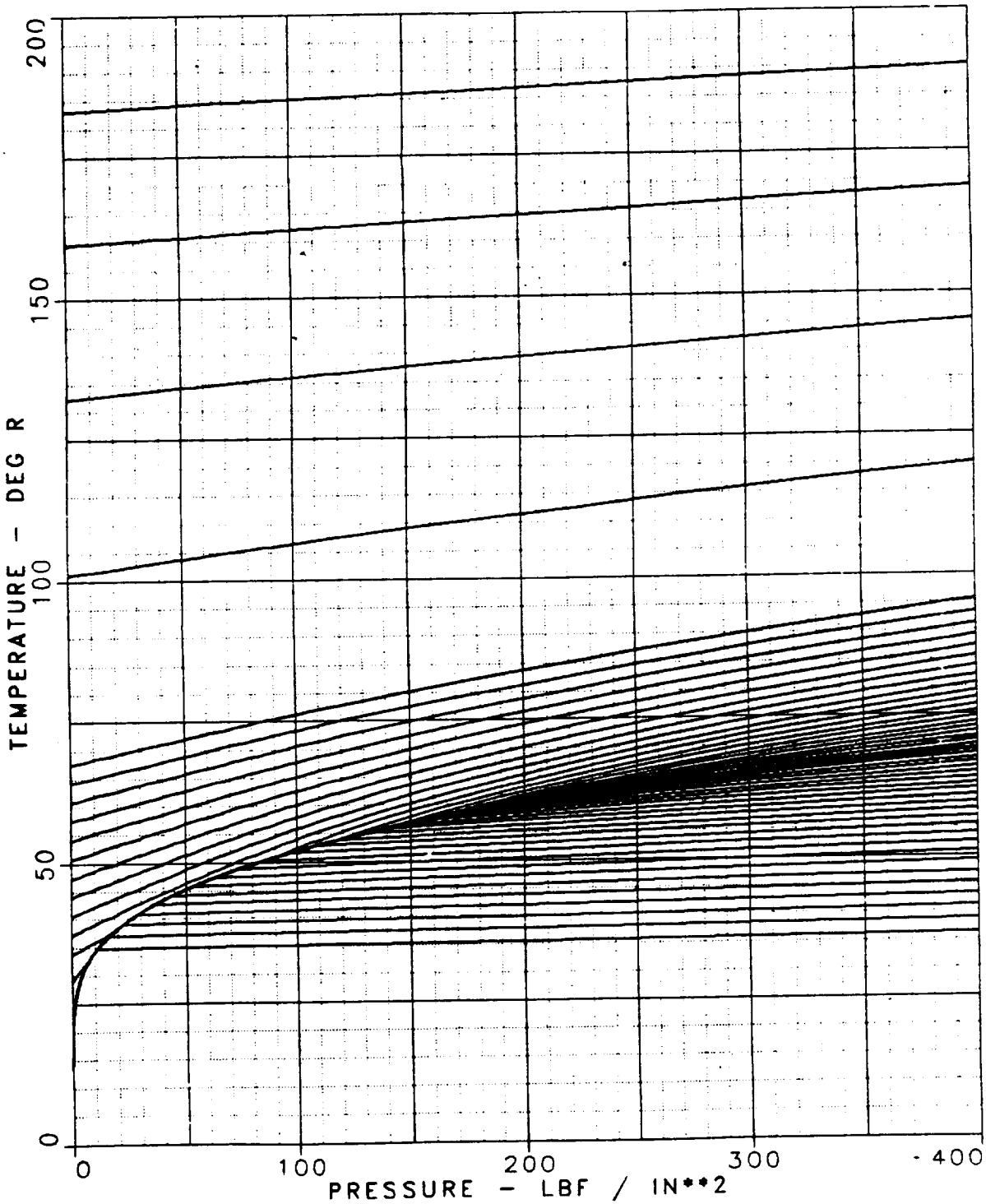


FIGURE 3.4.2.4.5

PRATT & WHITNEY

HYDROGEN PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT U

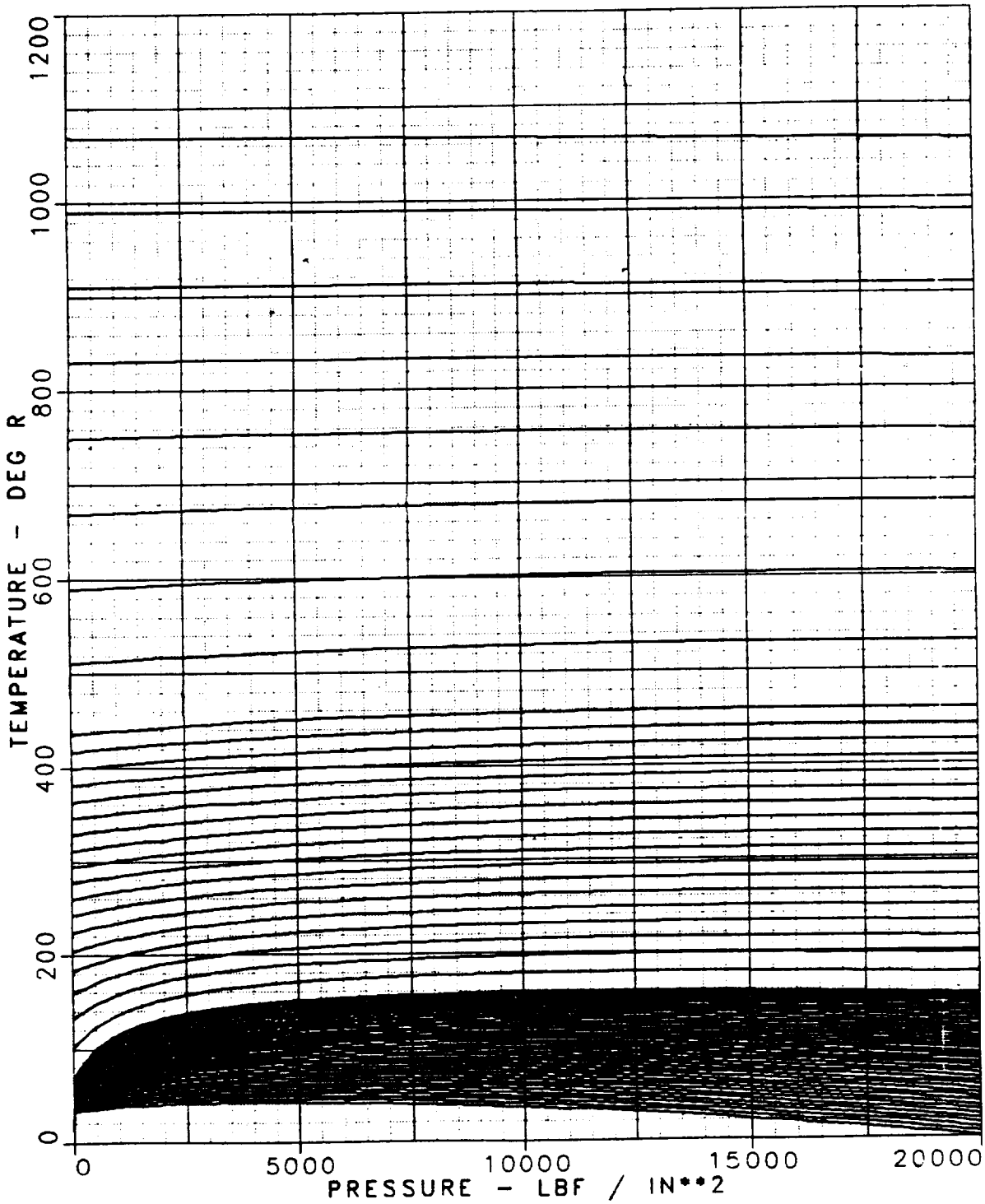


FIGURE 3.4.2.4.6

PRATT & WHITNEY

HYDROGEN PROPERTY MAP

ENTROPY VS ENTHALPY AT CONSTANT PRESSURE

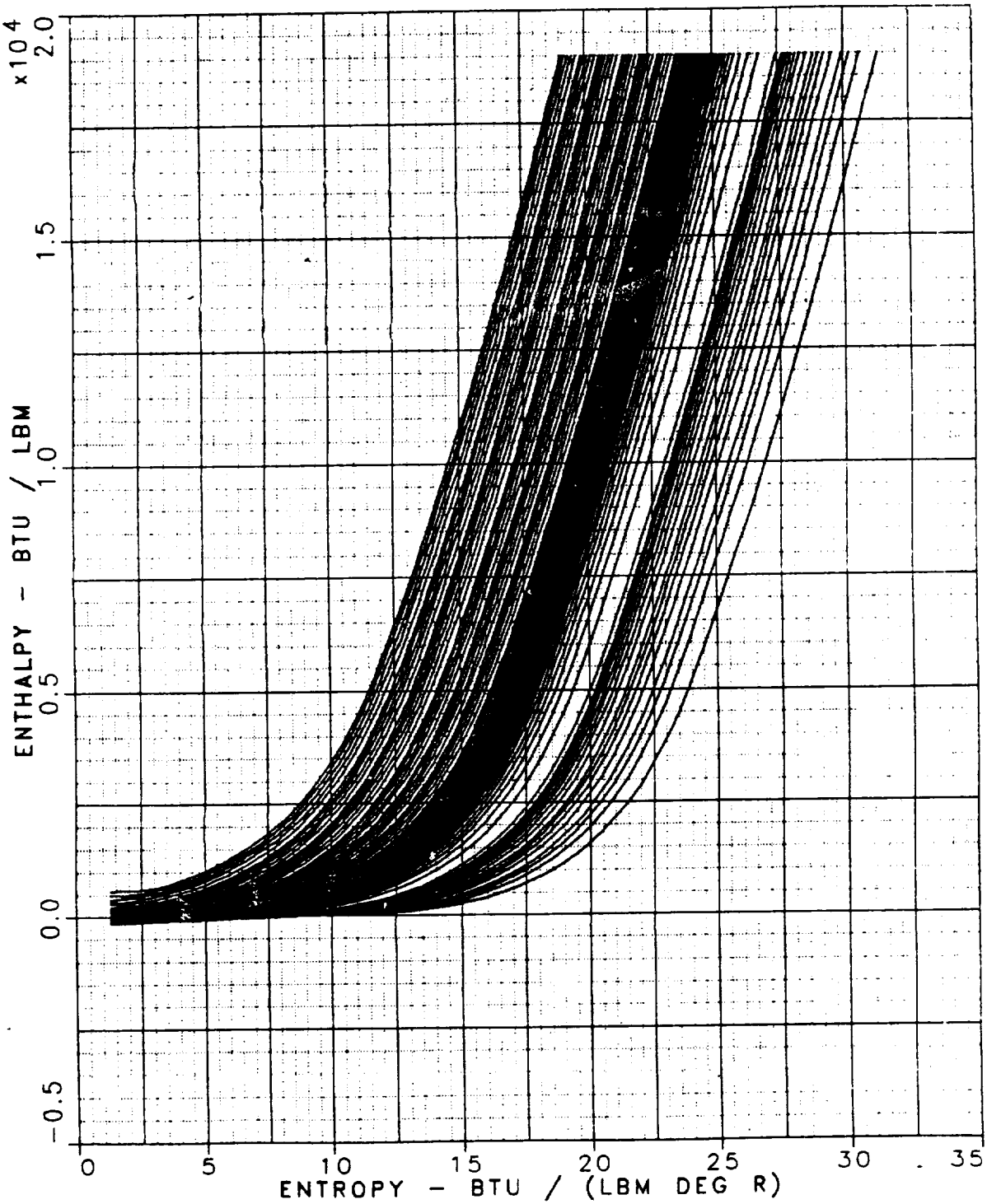


FIGURE 3.4.2.4.7

PRATT & WHITNEY

OXYGEN PROPERTY MAP

DENSITY VS PRESSURE AT CONSTANT U

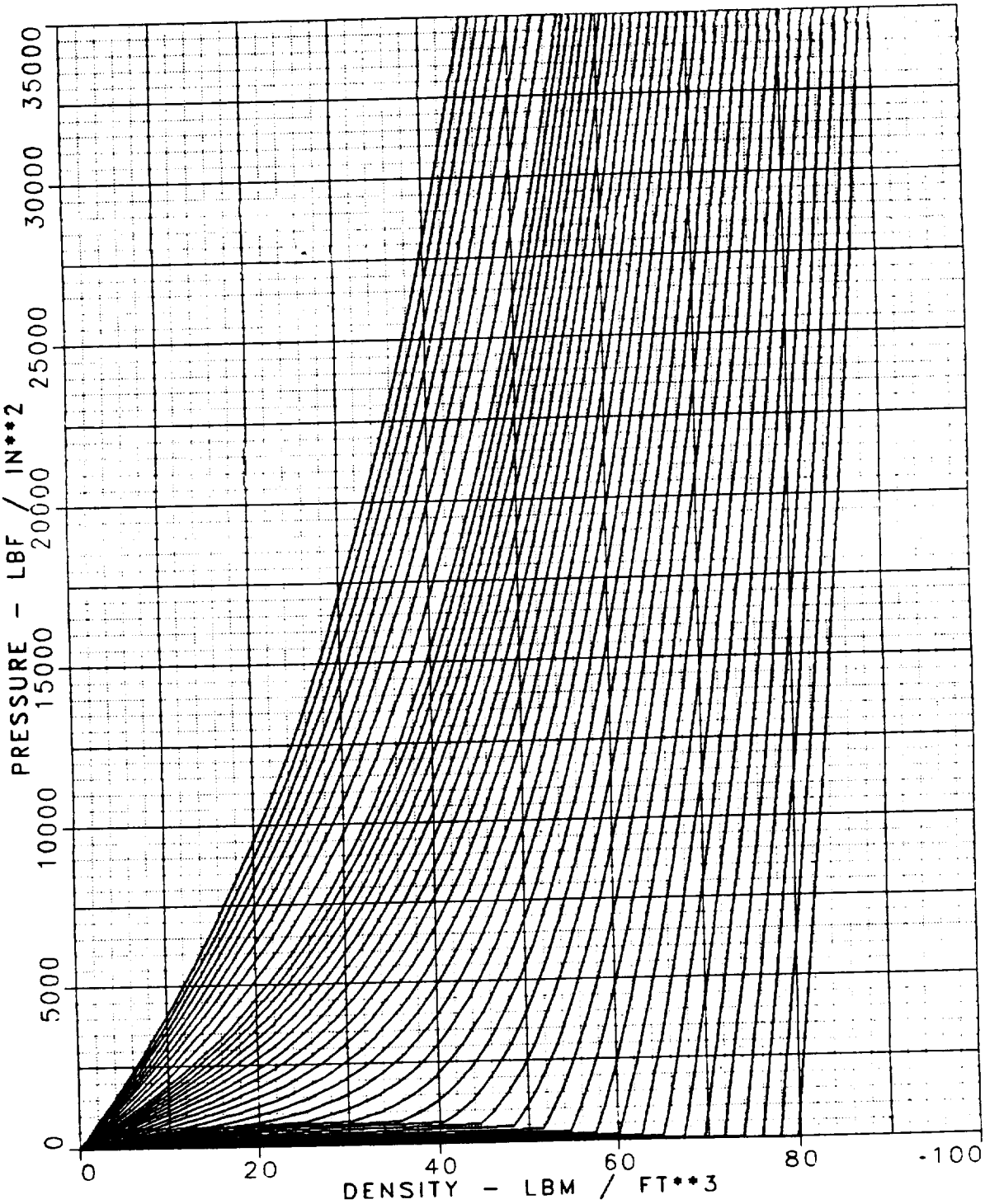


FIGURE 3.4.2.4.8

PRATT & WHITNEY

OXYGEN PROPERTY MAP

DENSITY VS PRESSURE AT CONSTANT ENTHALPY

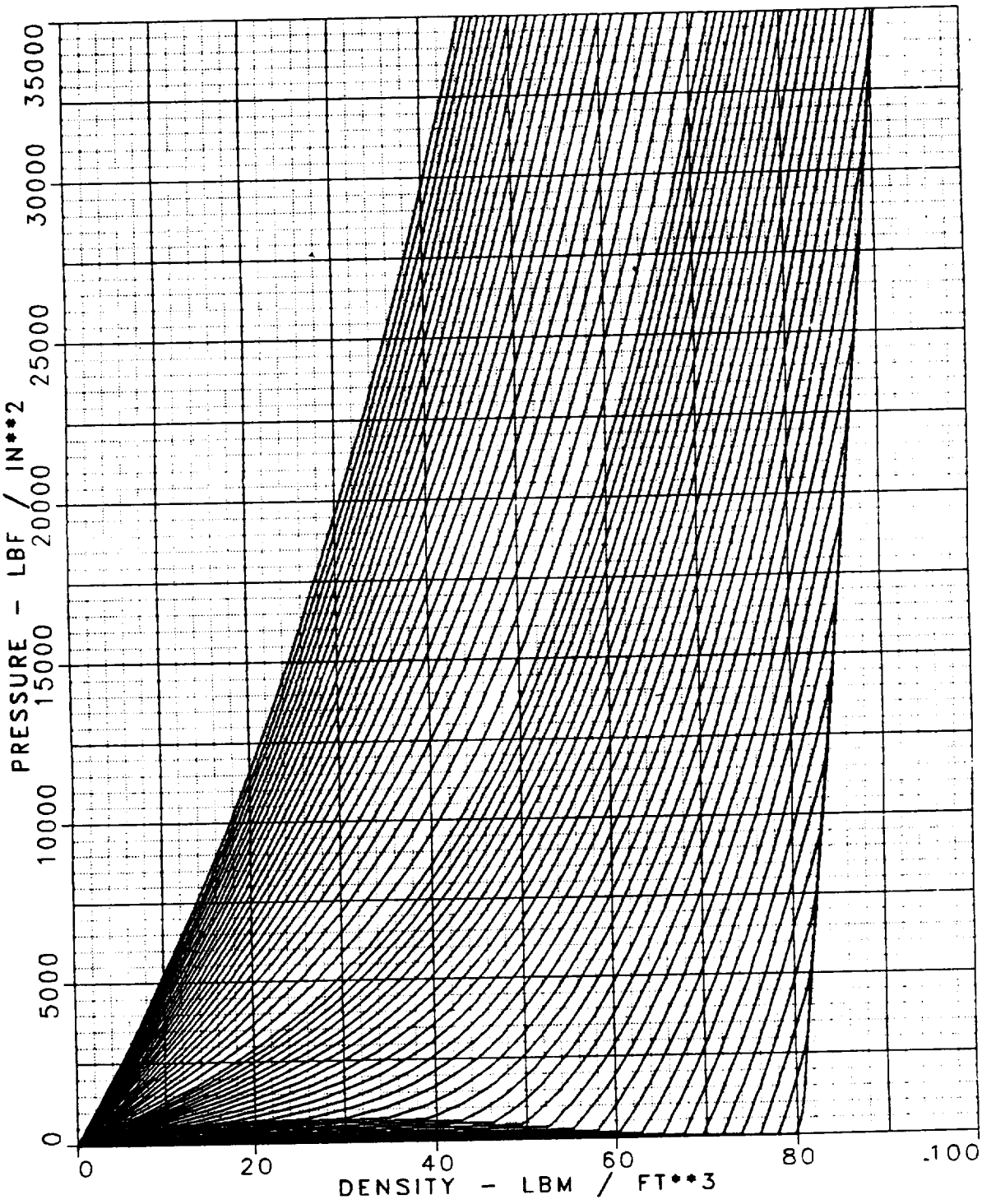


FIGURE 3.4.2.4.9

PRATT & WHITNEY

OXYGEN PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT U

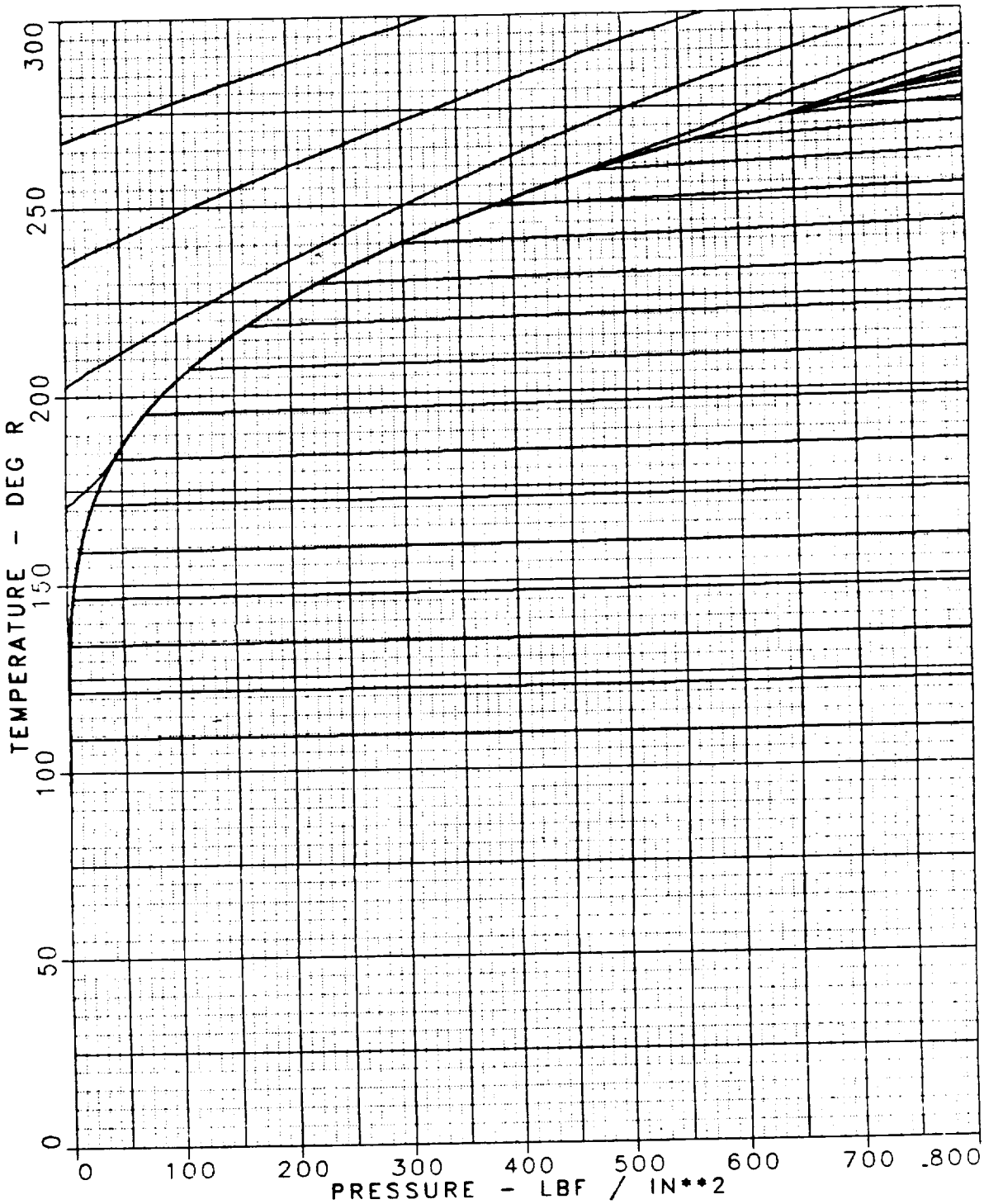


FIGURE 3.4.2.4.10

PRATT & WHITNEY

OXYGEN PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT U

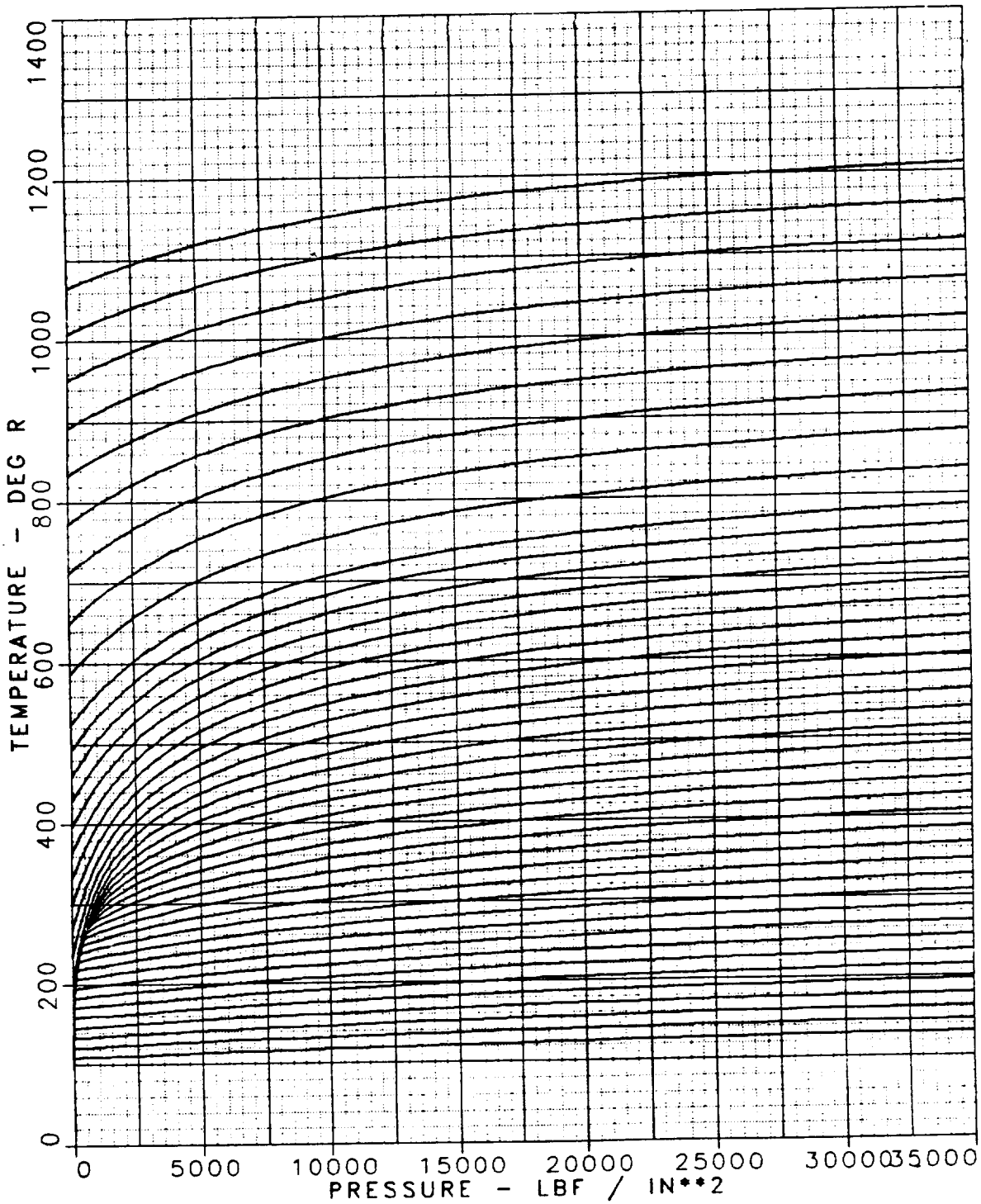


FIGURE 3.4.2.4.11

PRATT & WHITNEY

OXYGEN PROPERTY MAP

ENTROPY VS ENTHALPY AT CONSTANT PRESSURE

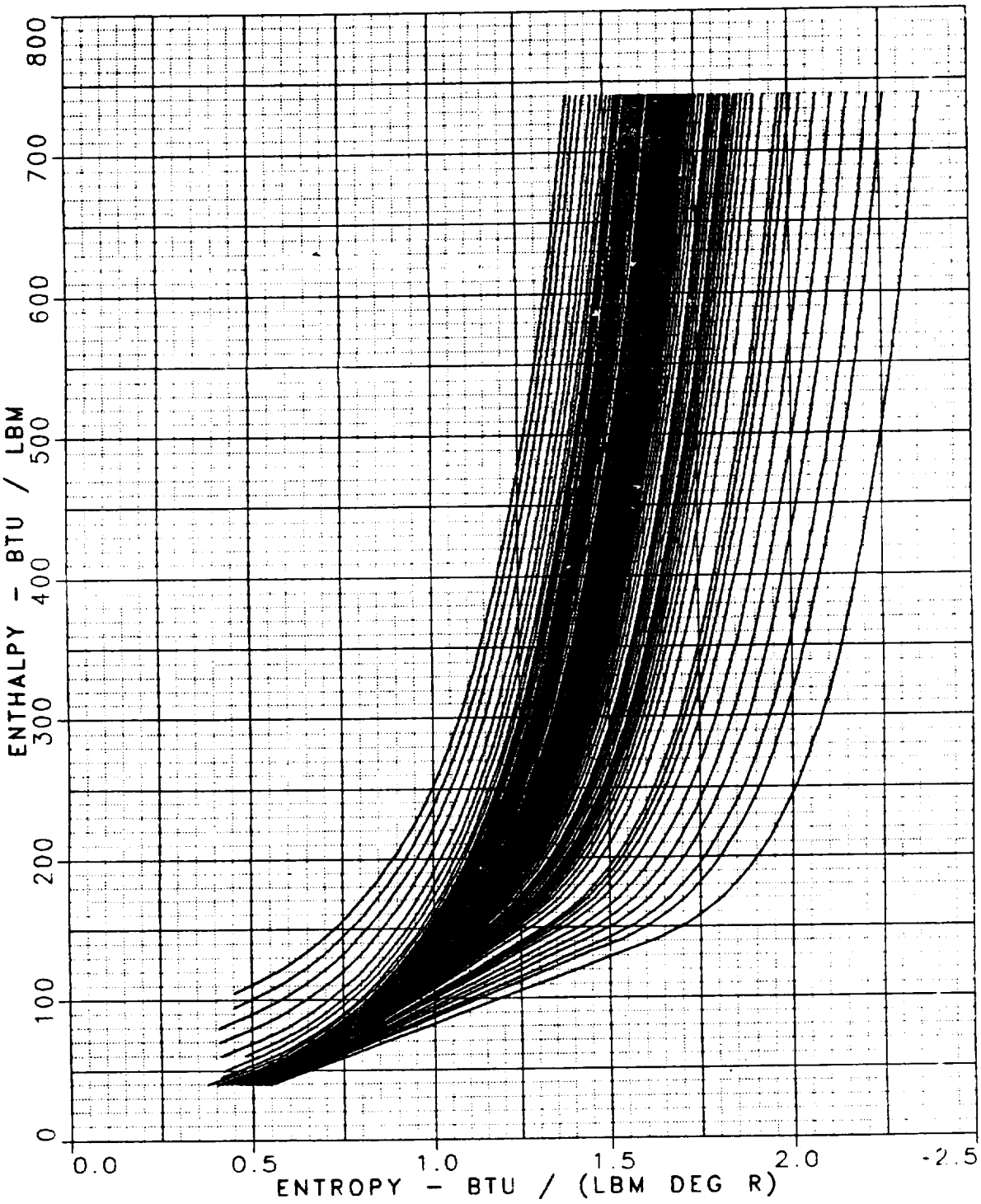


FIGURE 3.4.2.4.12

PRATT & WHITNEY

METHANE PROPERTY MAP

DENSITY VS PRESSURE AT CONSTANT ENTHALPY

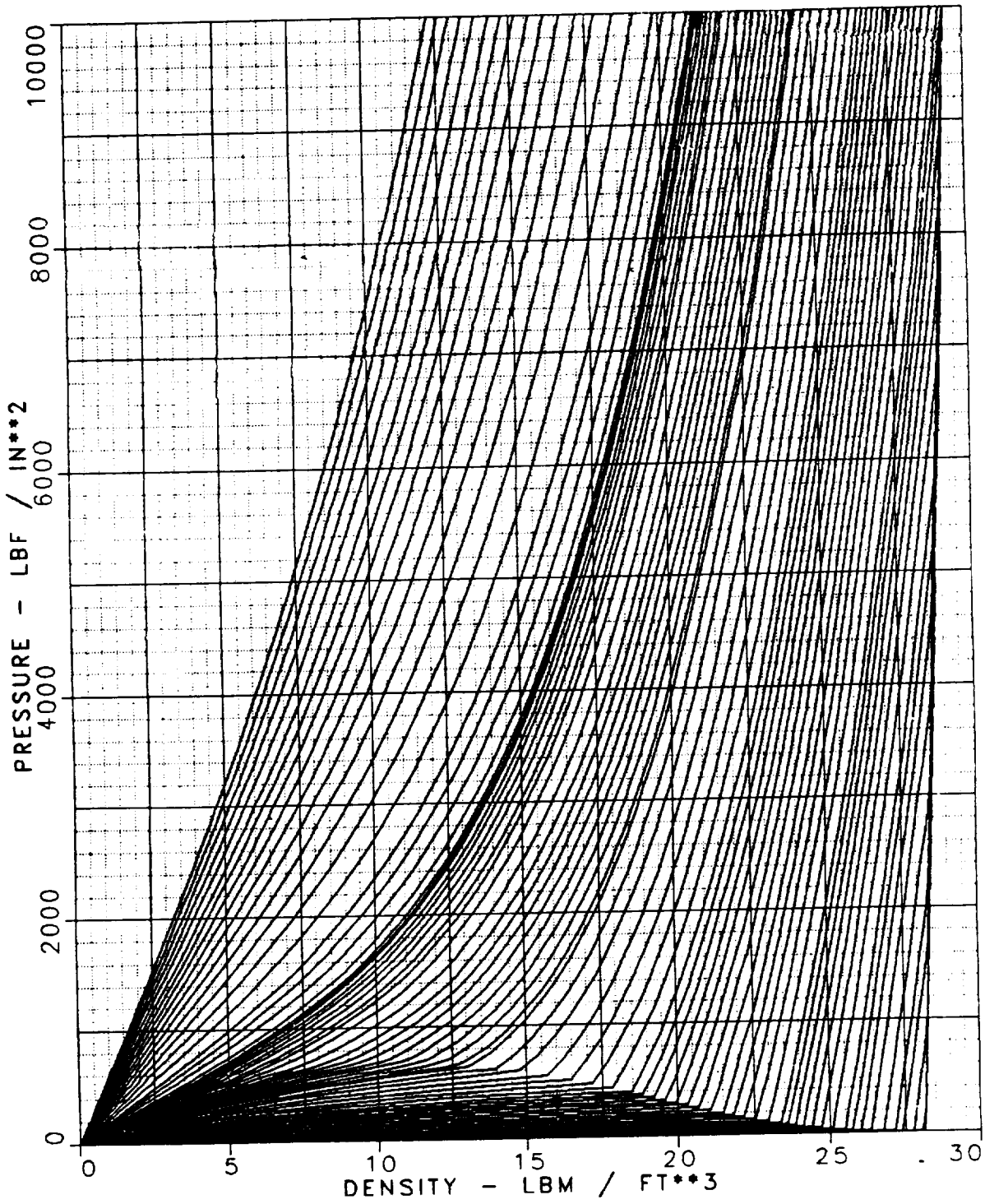


FIGURE 3.4.2.4.13

PRATT & WHITNEY

METHANE PROPERTY MAP

ENTHALPY VS TEMPERATURE AT CONSTANT PRESSURE

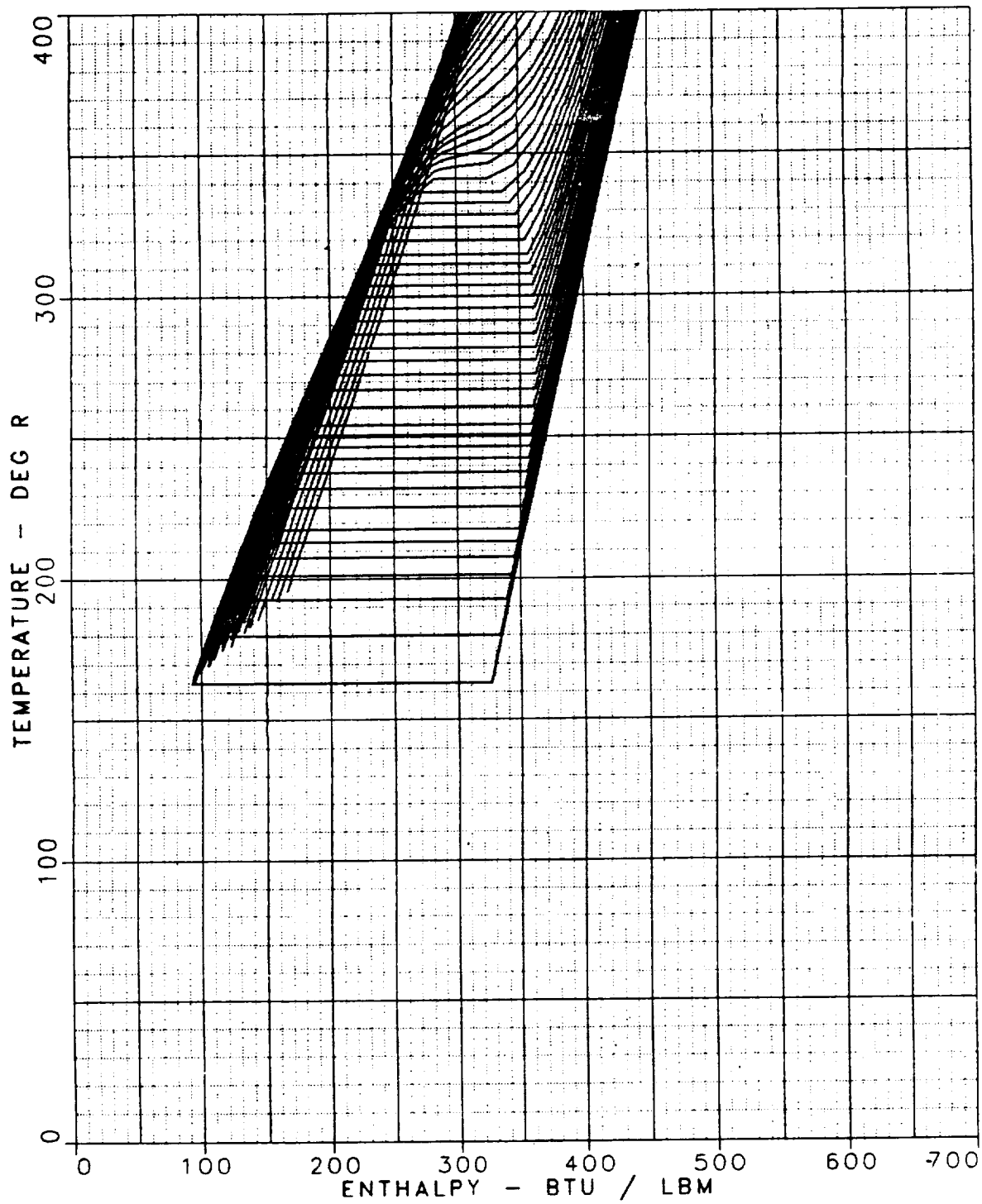


FIGURE 3.4.2.4.14

PRATT & WHITNEY

METHANE PROPERTY MAP

ENTHALPY VS TEMPERATURE AT CONSTANT PRESSURE

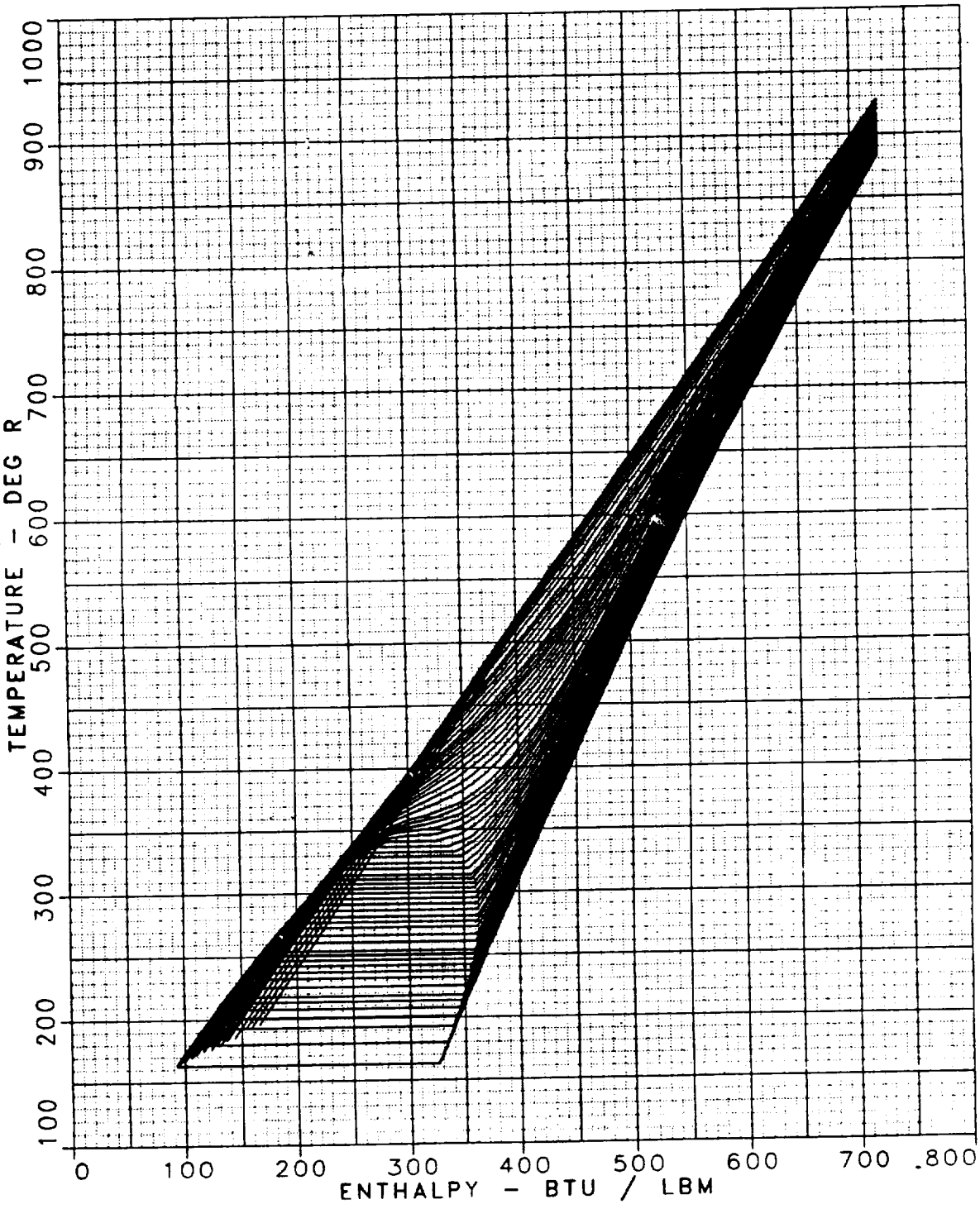


FIGURE 3.4.2.4.15

PRATT & WHITNEY

METHANE PROPERTY MAP

ENTROPY VS ENTHALPY AT CONSTANT PRESSURE

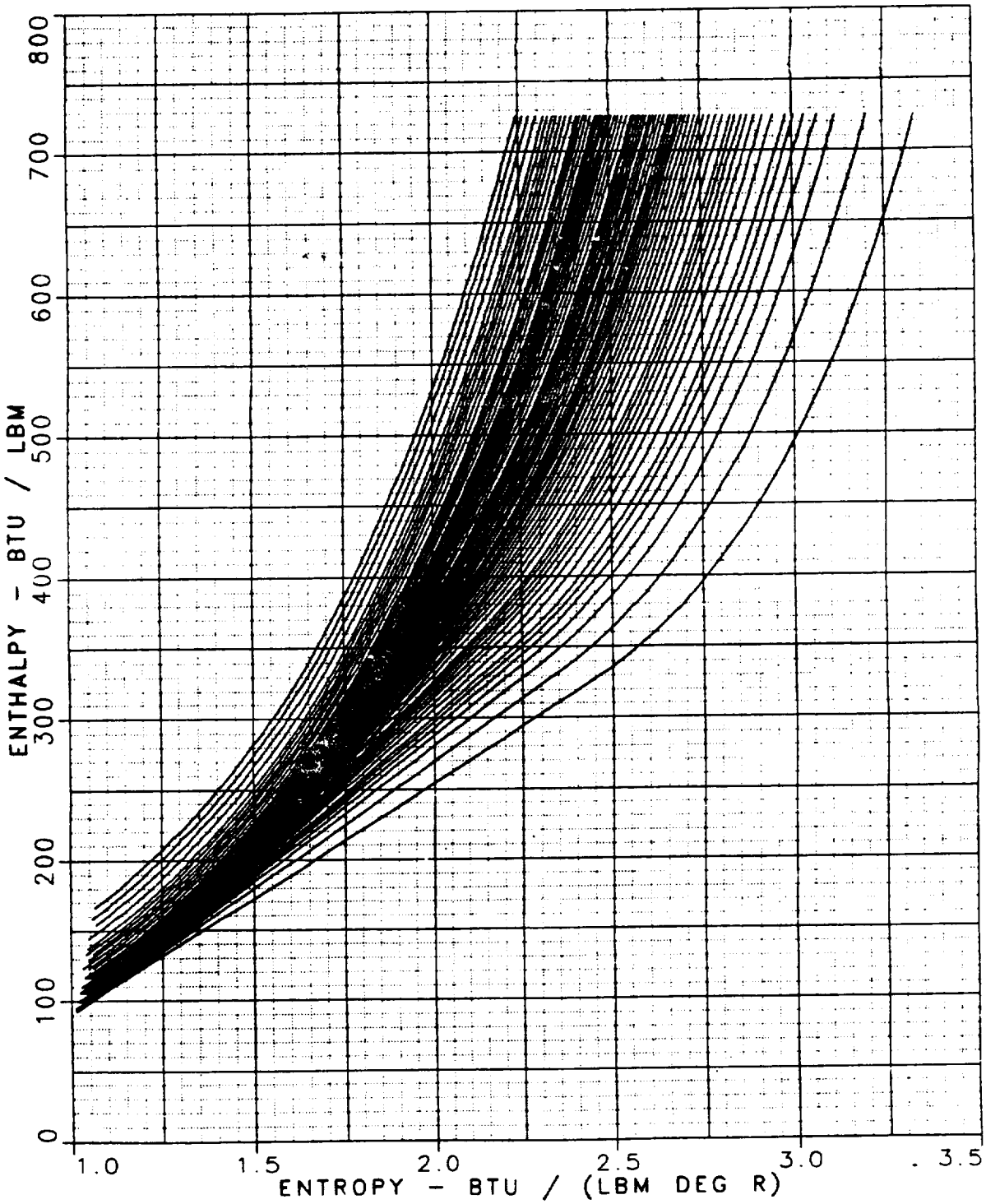


FIGURE 3.4.2.4.16

PRATT & WHITNEY

METHANE PROPERTY MAP

TEMPERATURE VS CV

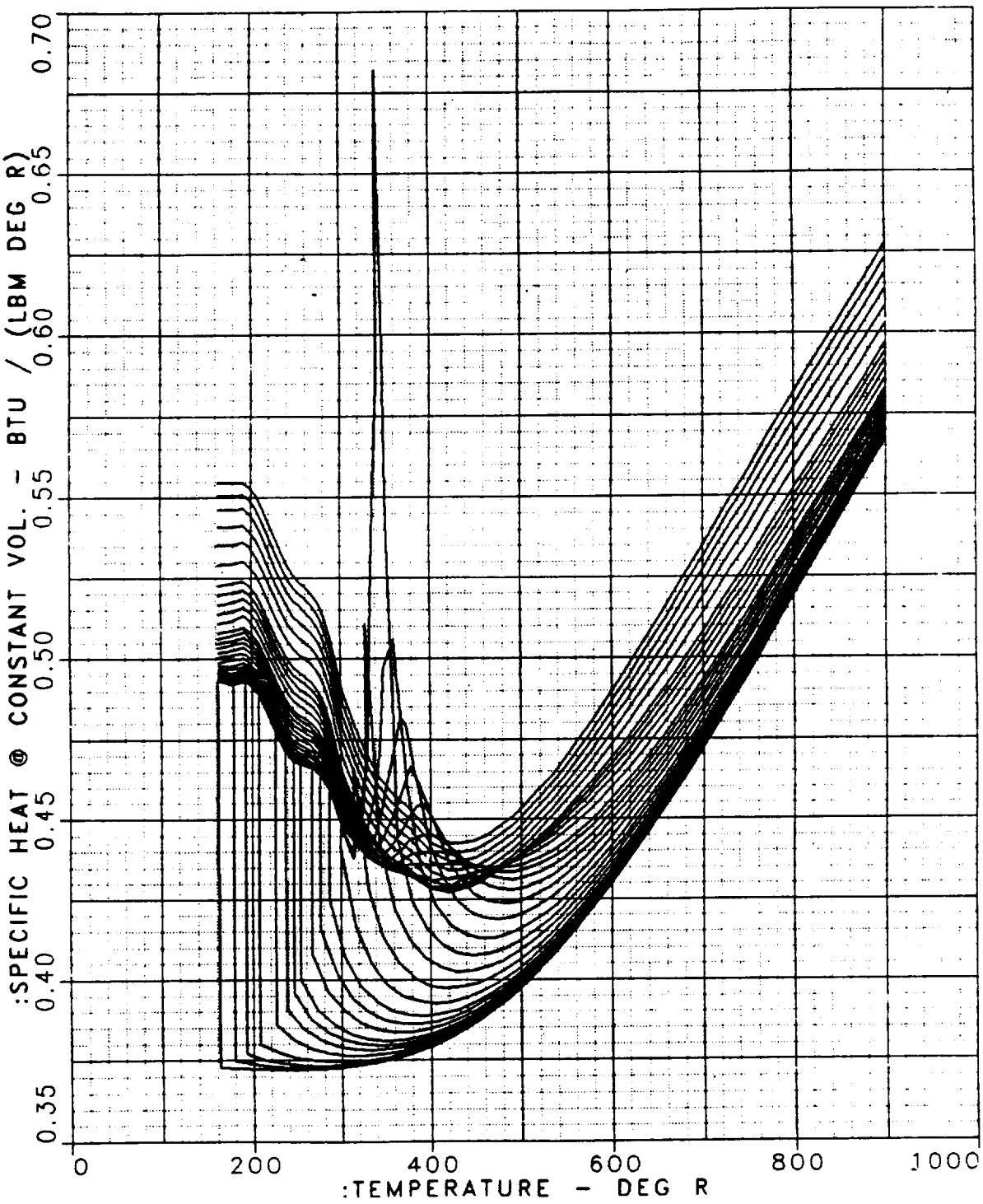


FIGURE 3.4.2.4.17

PRATT & WHITNEY

METHANE PROPERTY MAP

TEMPERATURE VS CP

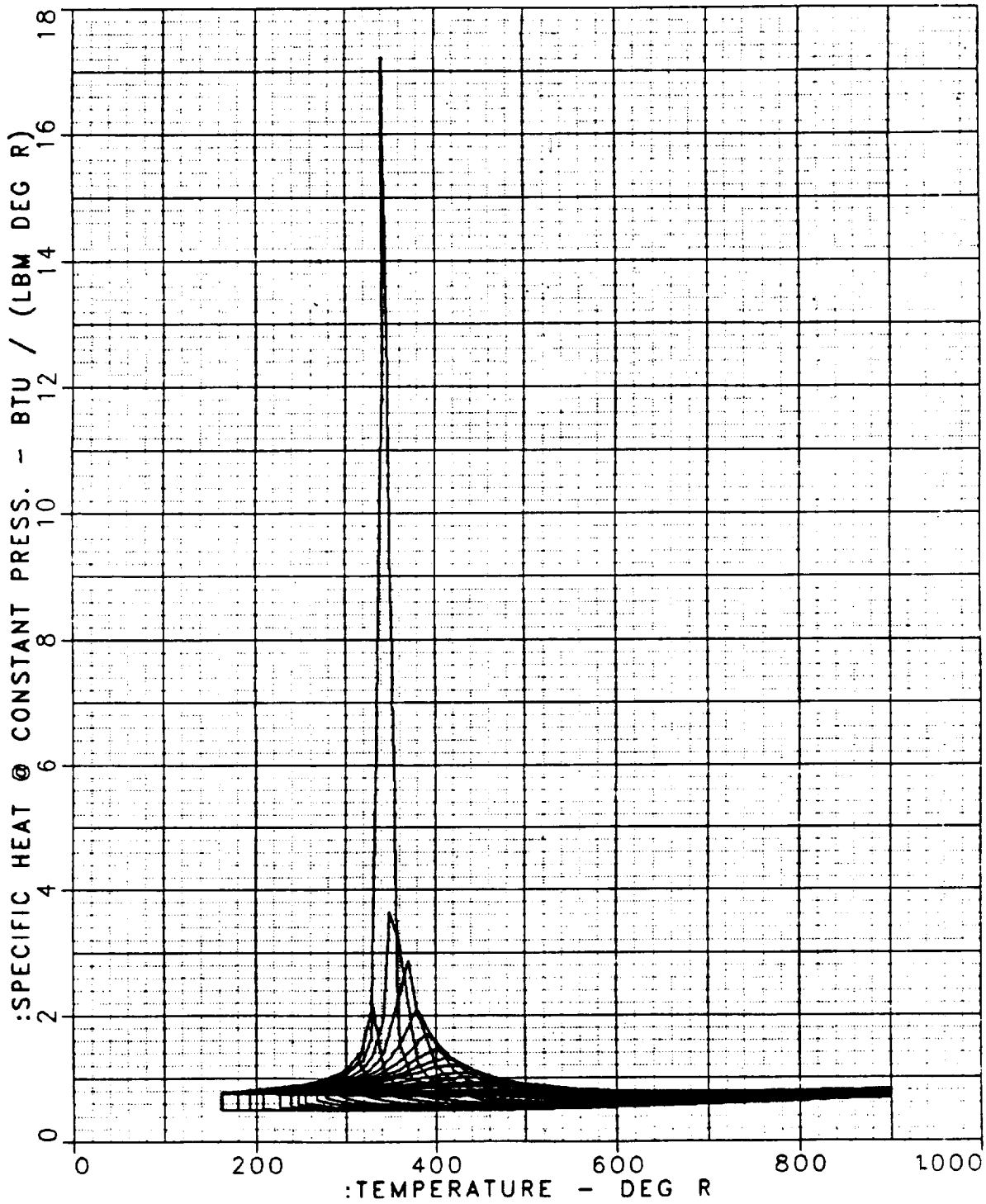


FIGURE 3.4.2.4.18

PRATT & WHITNEY

NITROGEN PROPERTY MAP

DENSITY VS PRESSURE AT CONSTANT ENTHALPY

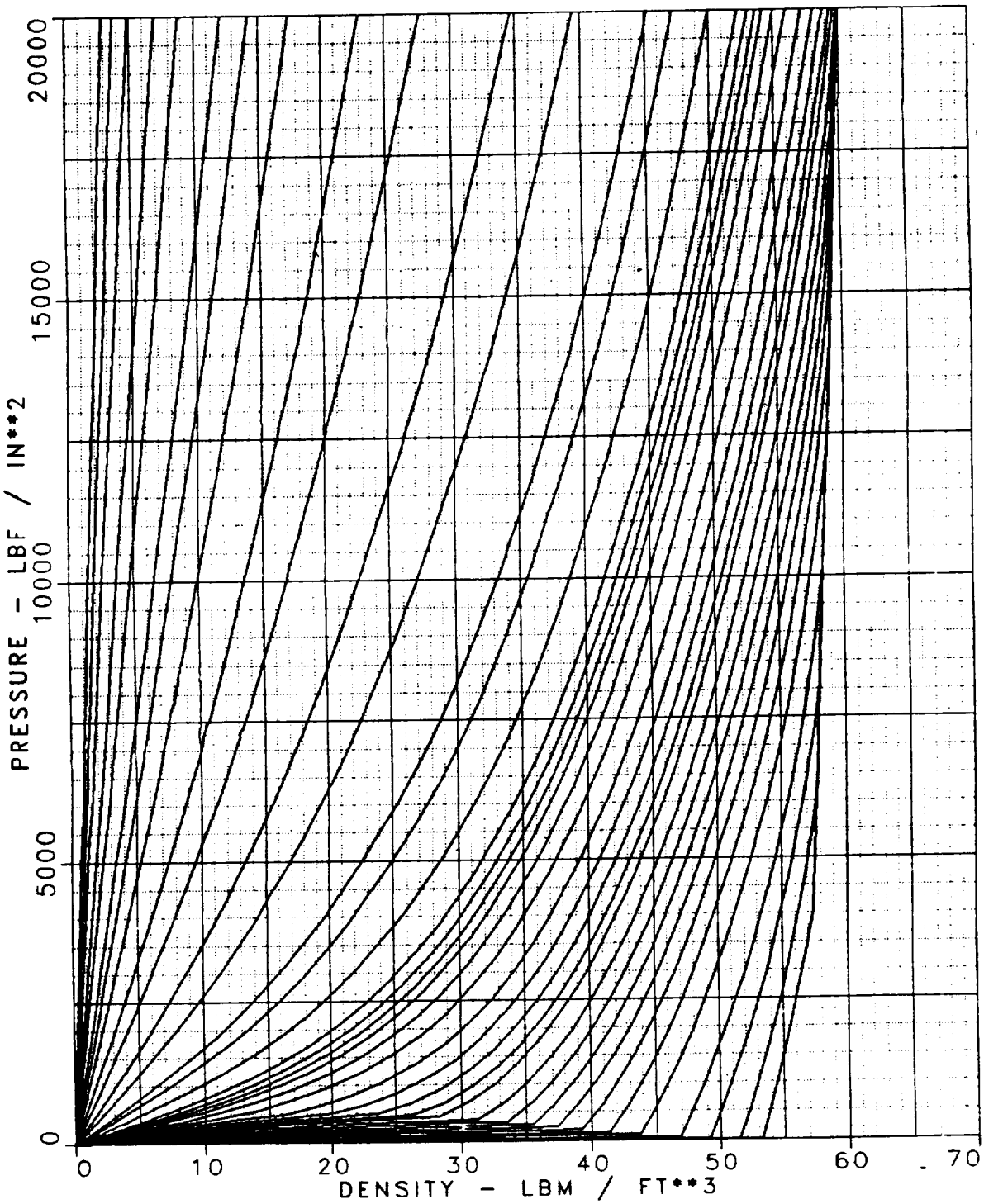


FIGURE 3.4.2.4.19

PRATT & WHITNEY

NITROGEN PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT ENTHALPY

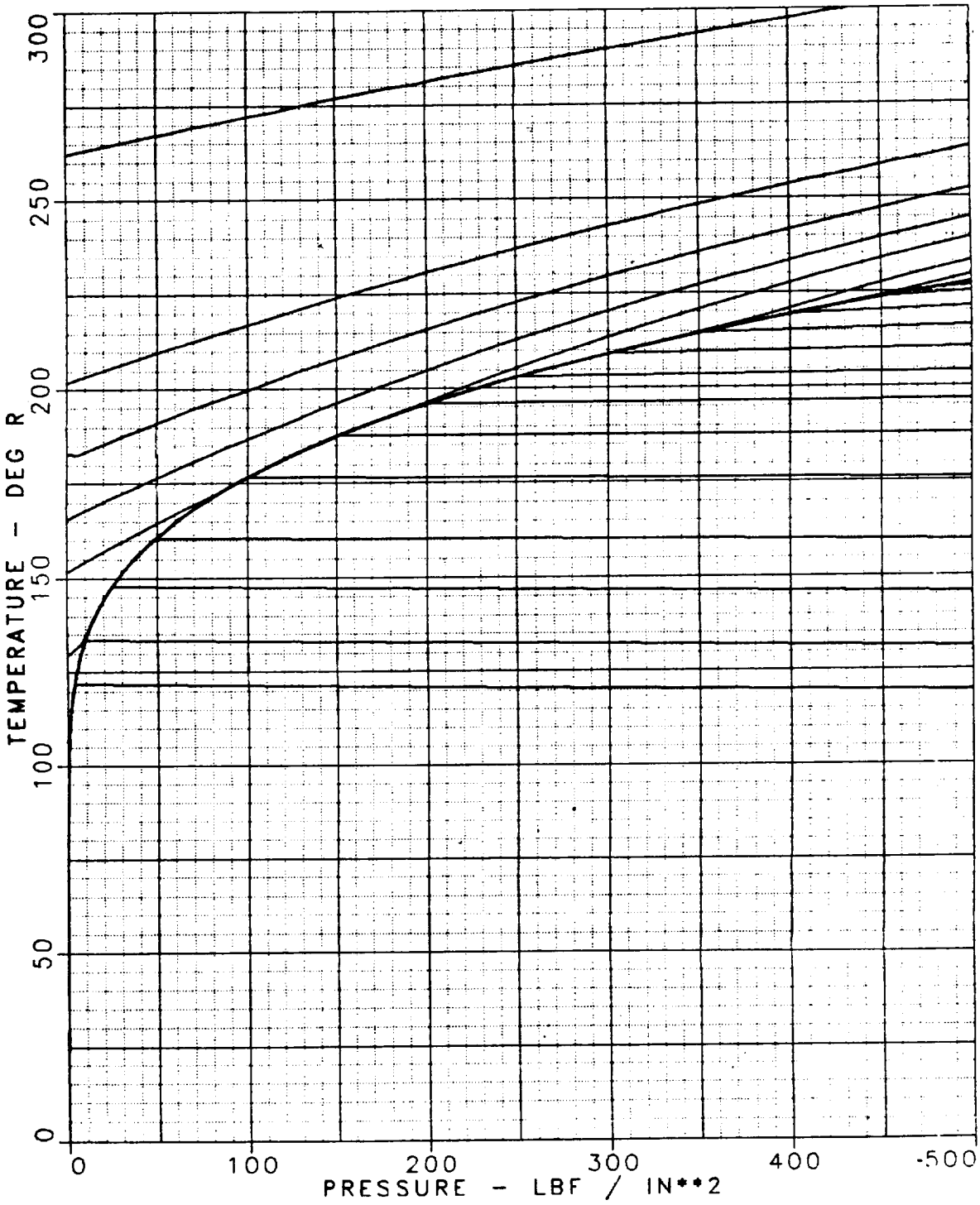


FIGURE 3.4.2.4.20

PRATT & WHITNEY

NITROGEN PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT ENTHALPY

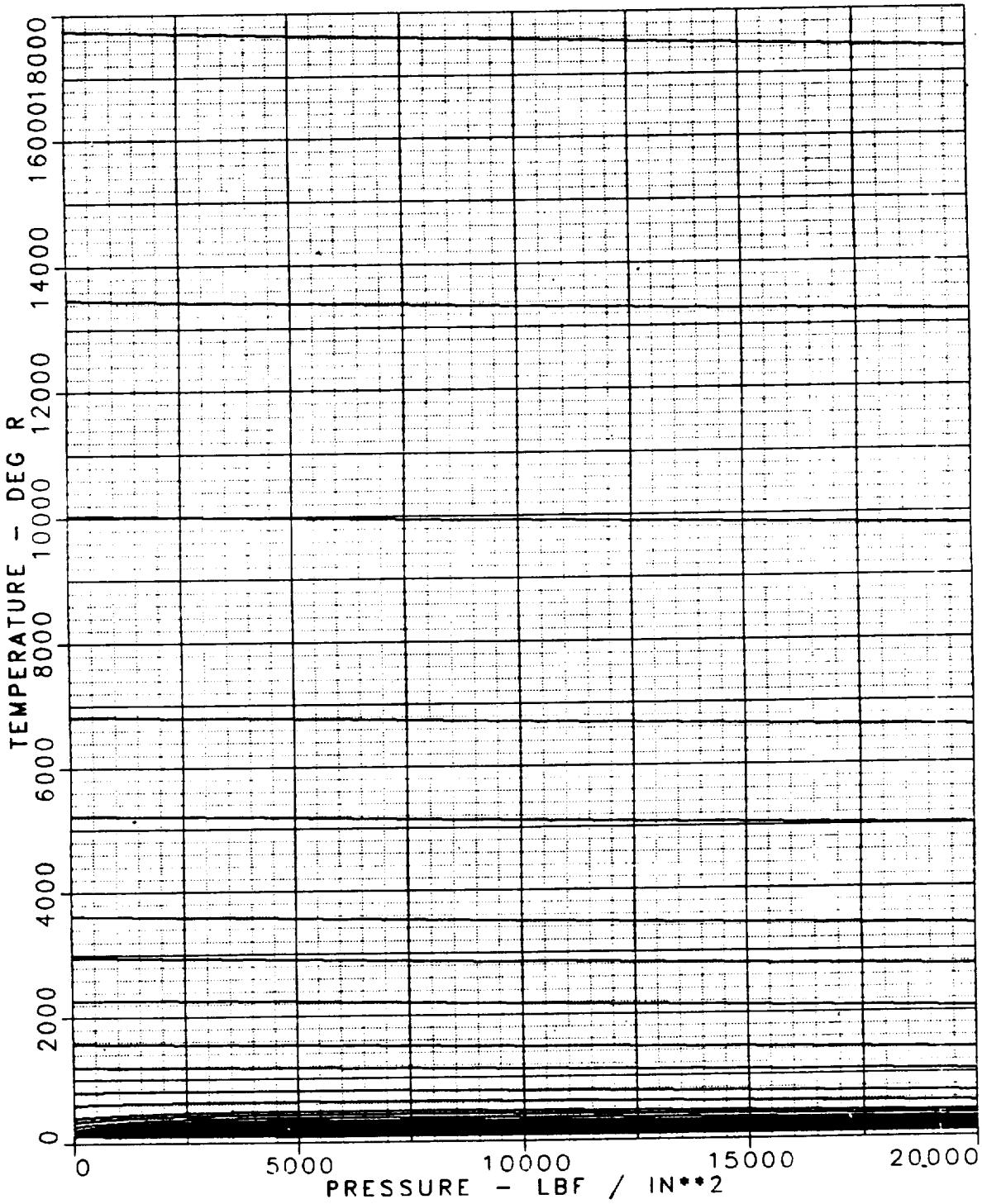


FIGURE 3.4.2.4.21

PRATT & WHITNEY

NITROGEN PROPERTY MAP

ENTROPY VS ENTHALPY AT CONSTANT PRESSURE

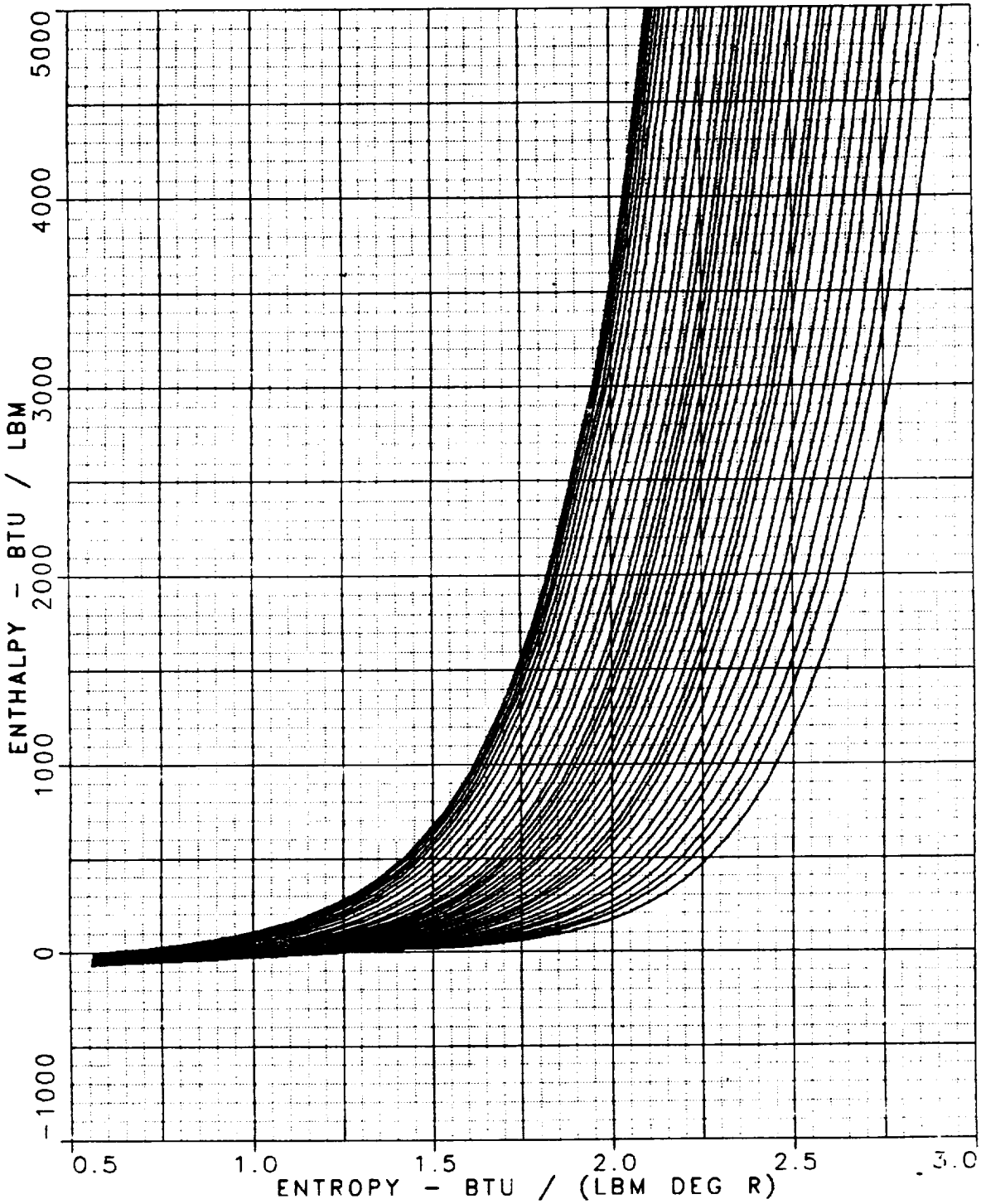


FIGURE 3.4.2.4.22

PRATT & WHITNEY

HELIUM PROPERTY MAP

DENSITY VS PRESSURE AT CONSTANT ENTHALPY

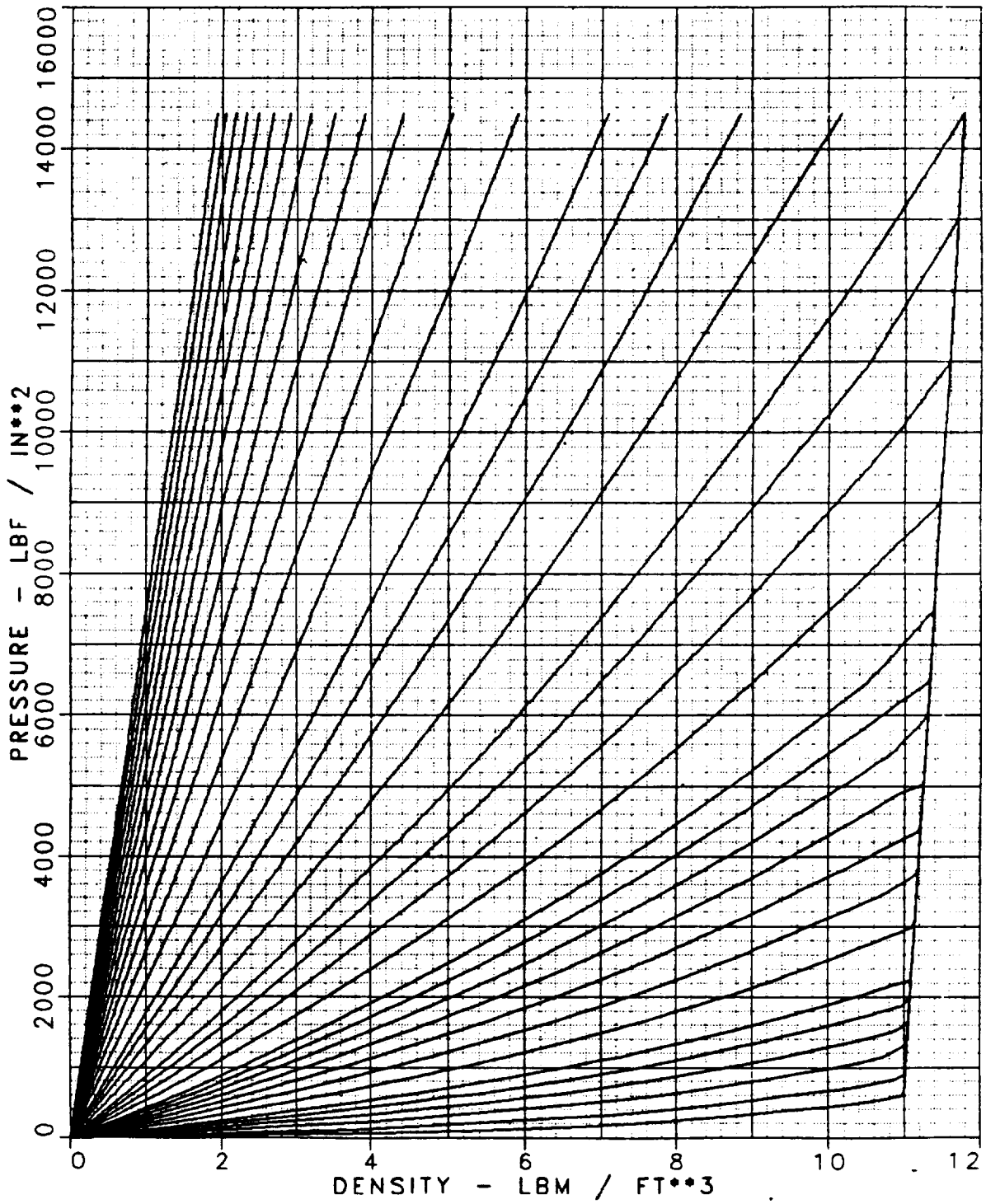


FIGURE 3.4.2.4.23

PRATT & WHITNEY

HELIUM PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT ENTHALPY

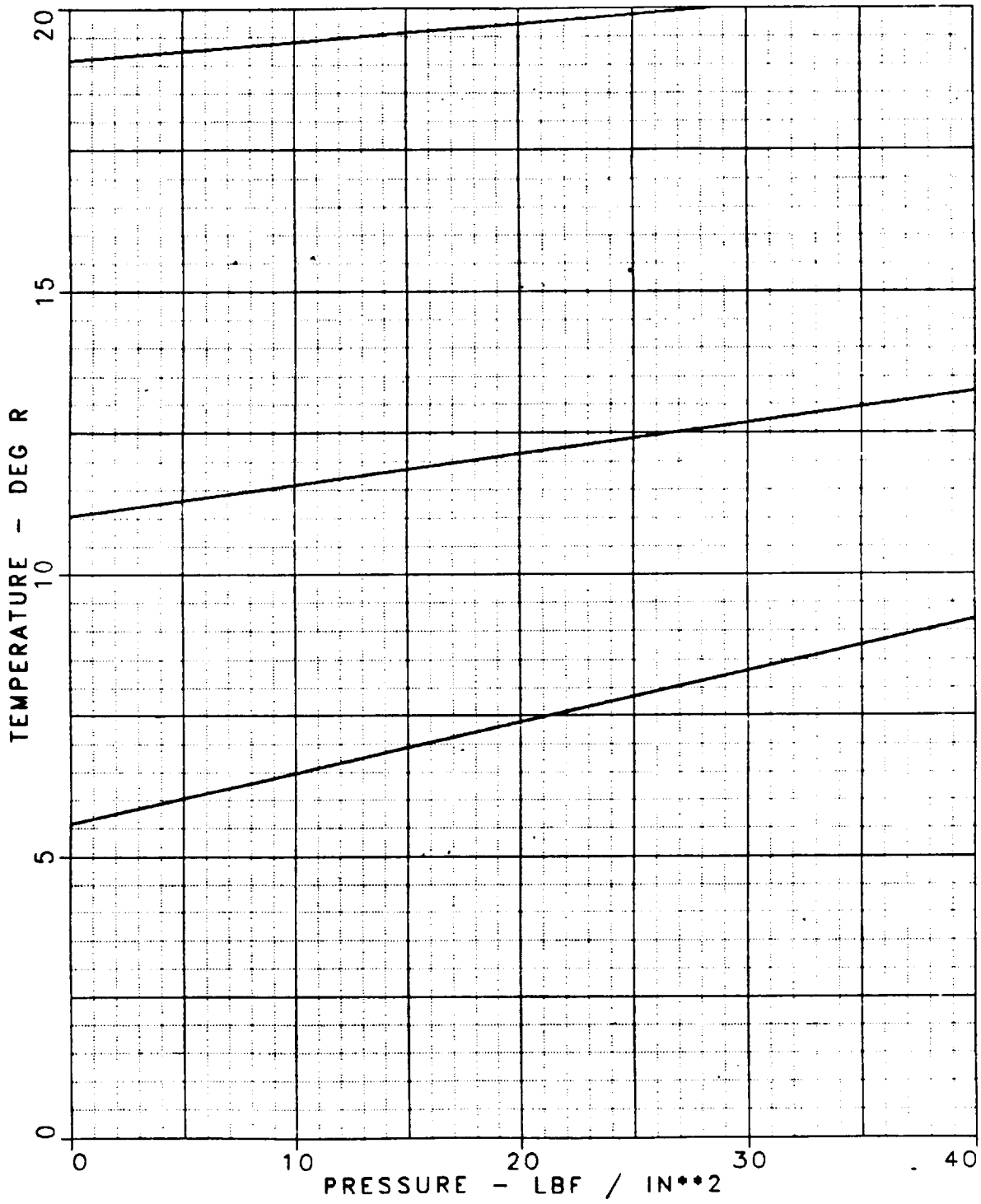


FIGURE 3.4.2.4.24

PRATT & WHITNEY

HELIUM PROPERTY MAP

PRESSURE VS TEMPERATURE AT CONSTANT ENTHALPY

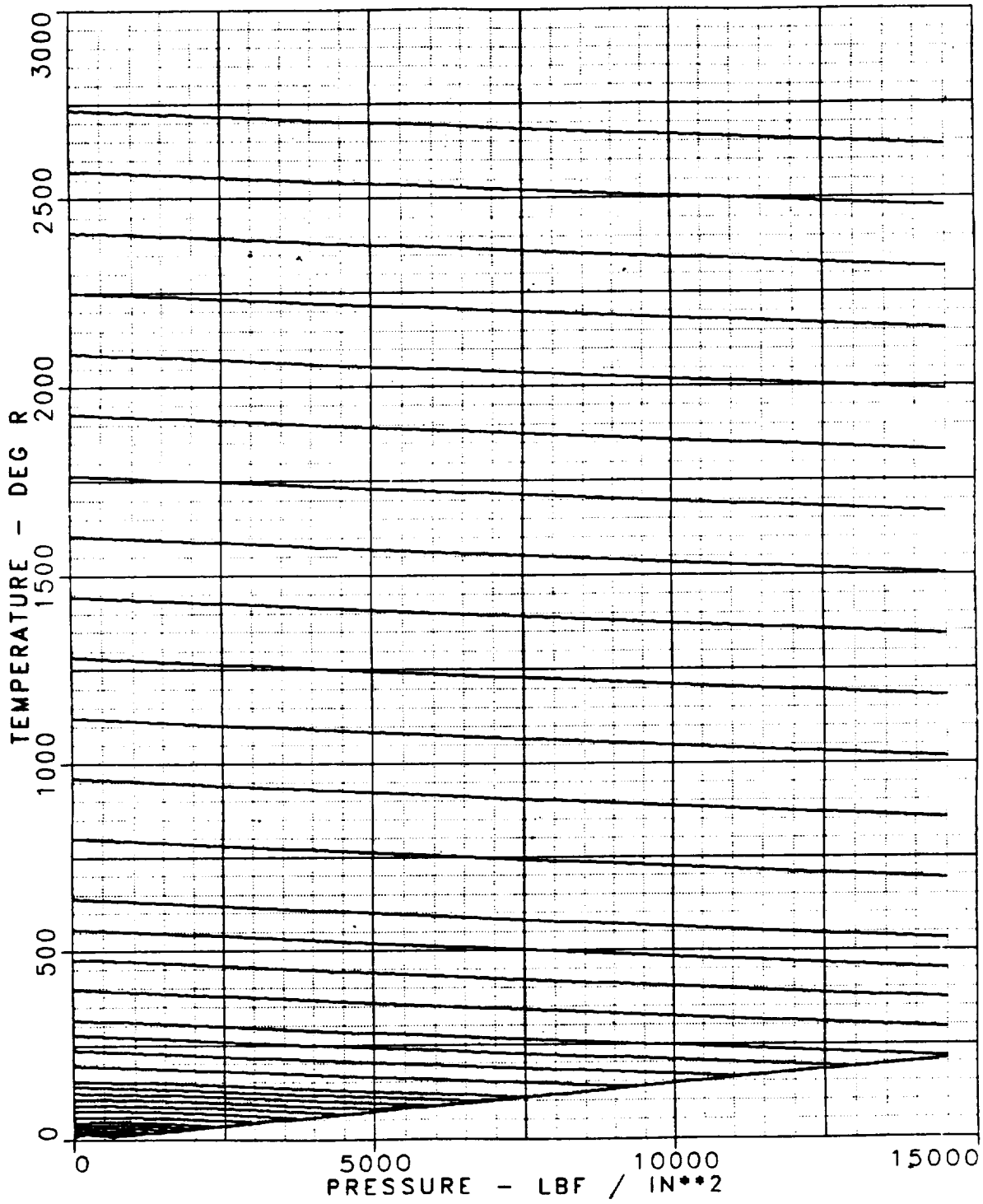


FIGURE 3.4.2.4.25

PRATT & WHITNEY

HELIUM PROPERTY MAP

ENTROPY VS ENTHALPY AT CONSTANT PRESSURE

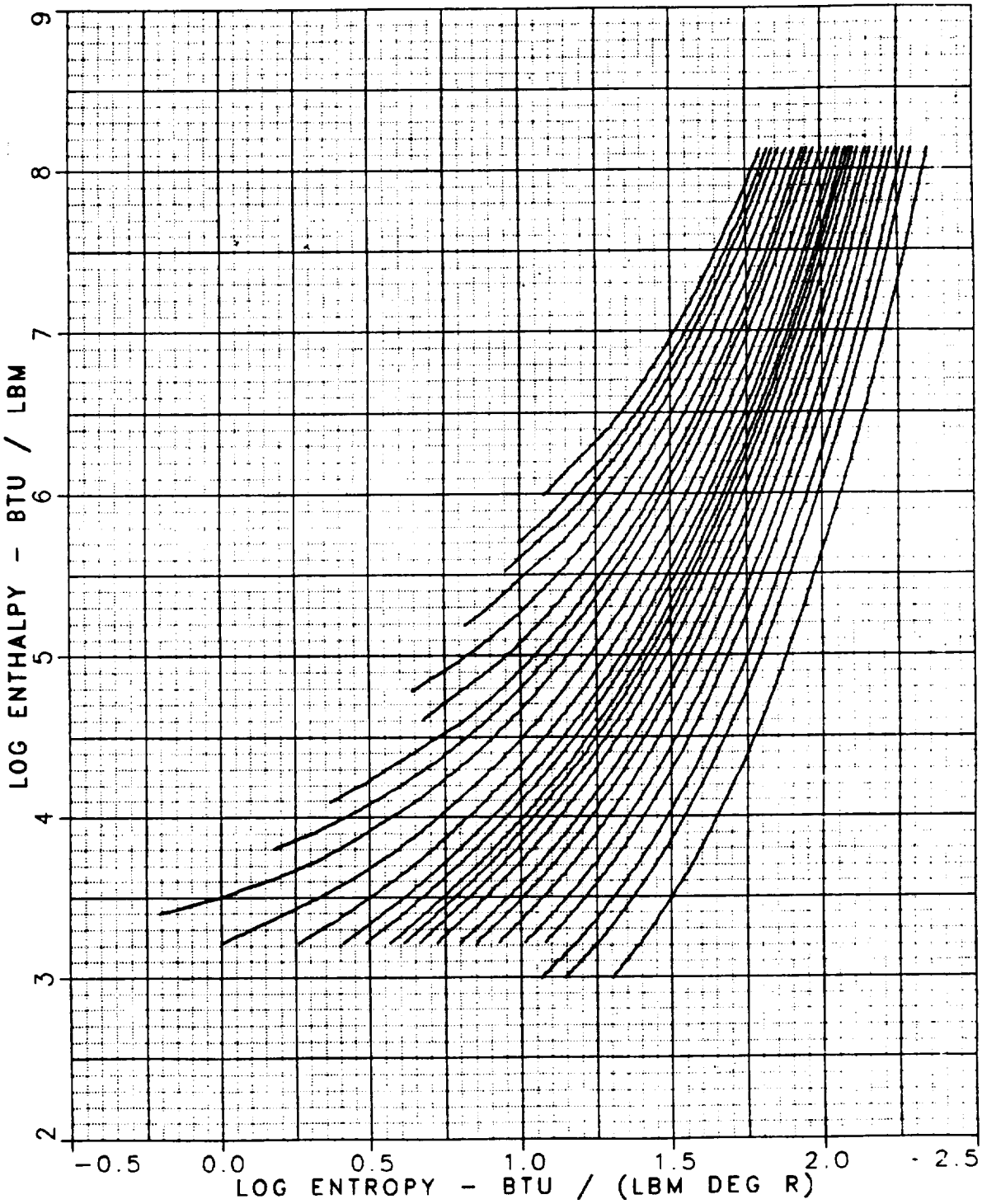


FIGURE 3.4.2.4.26

PRATT & WHITNEY
COMBUSTION PROPERTY MAP
COMPRESSIBLITY VS PRESSURE WITH LINES OF TEMPERATURE

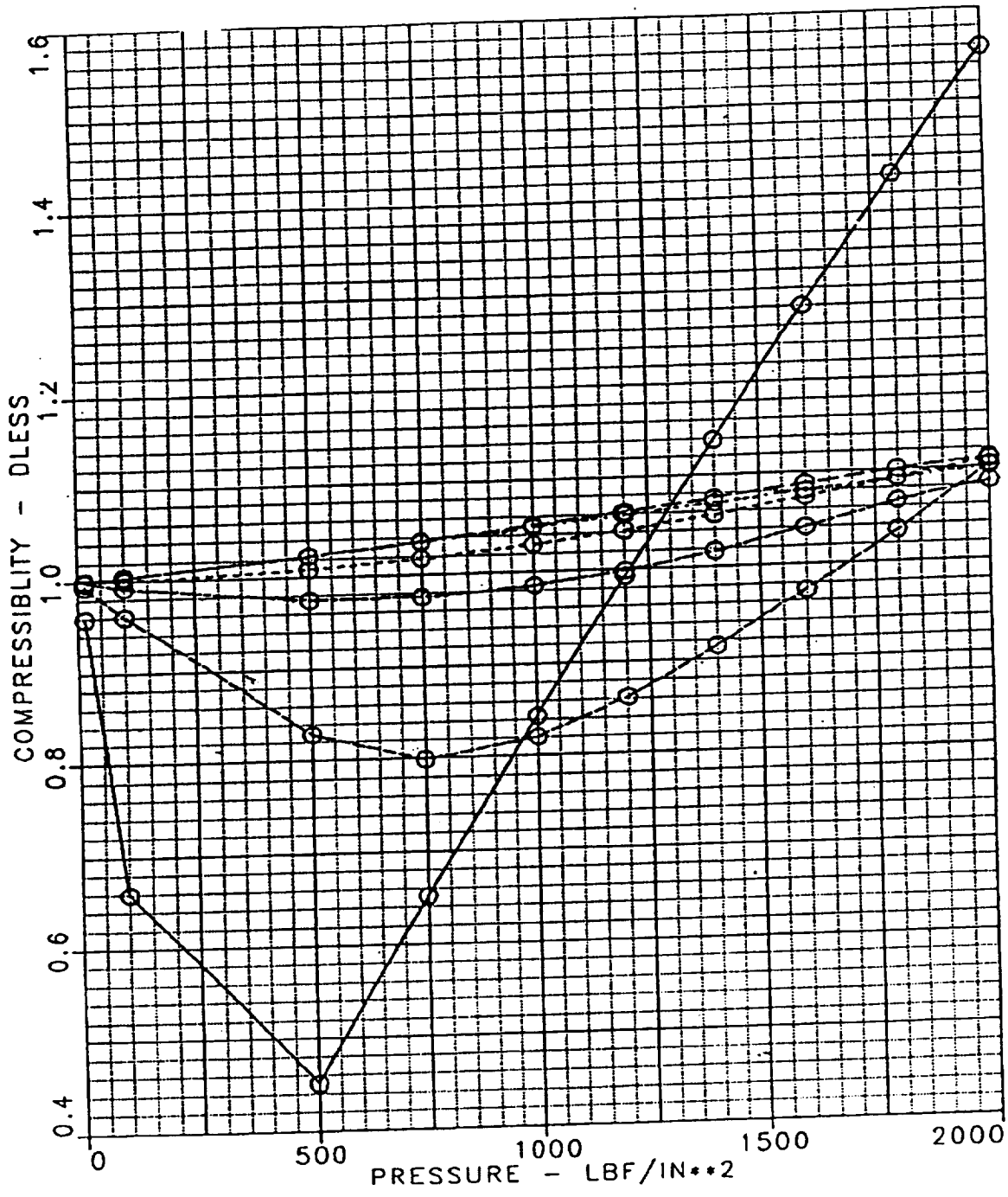


Figure 3.4.2.4.27

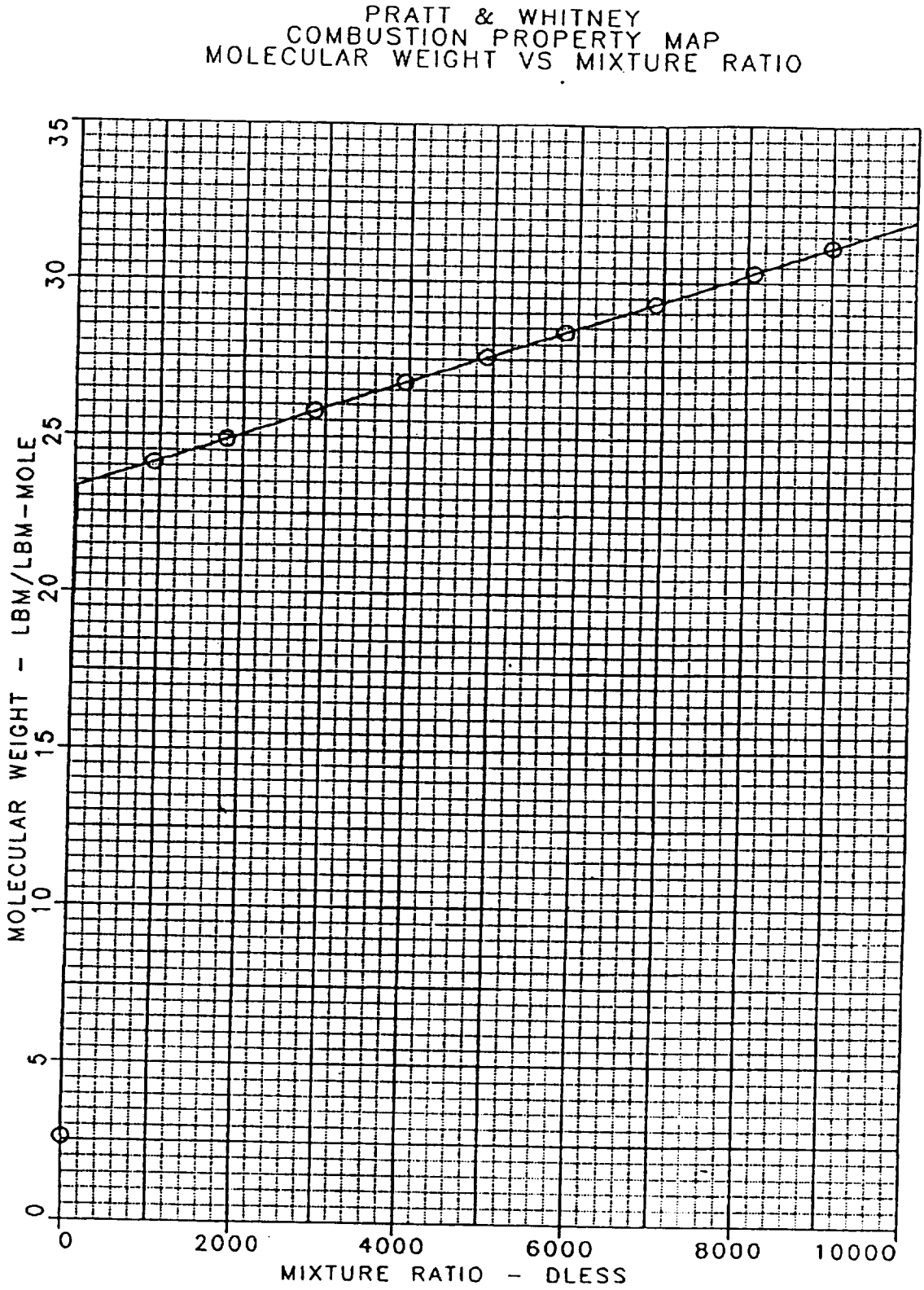


Figure 3.4.2.4.28

PRATT & WHITNEY
COMBUSTION PROPERTY MAPS
SPECIFIC HEAT VS TEMP W/LINES OF PRESSURE

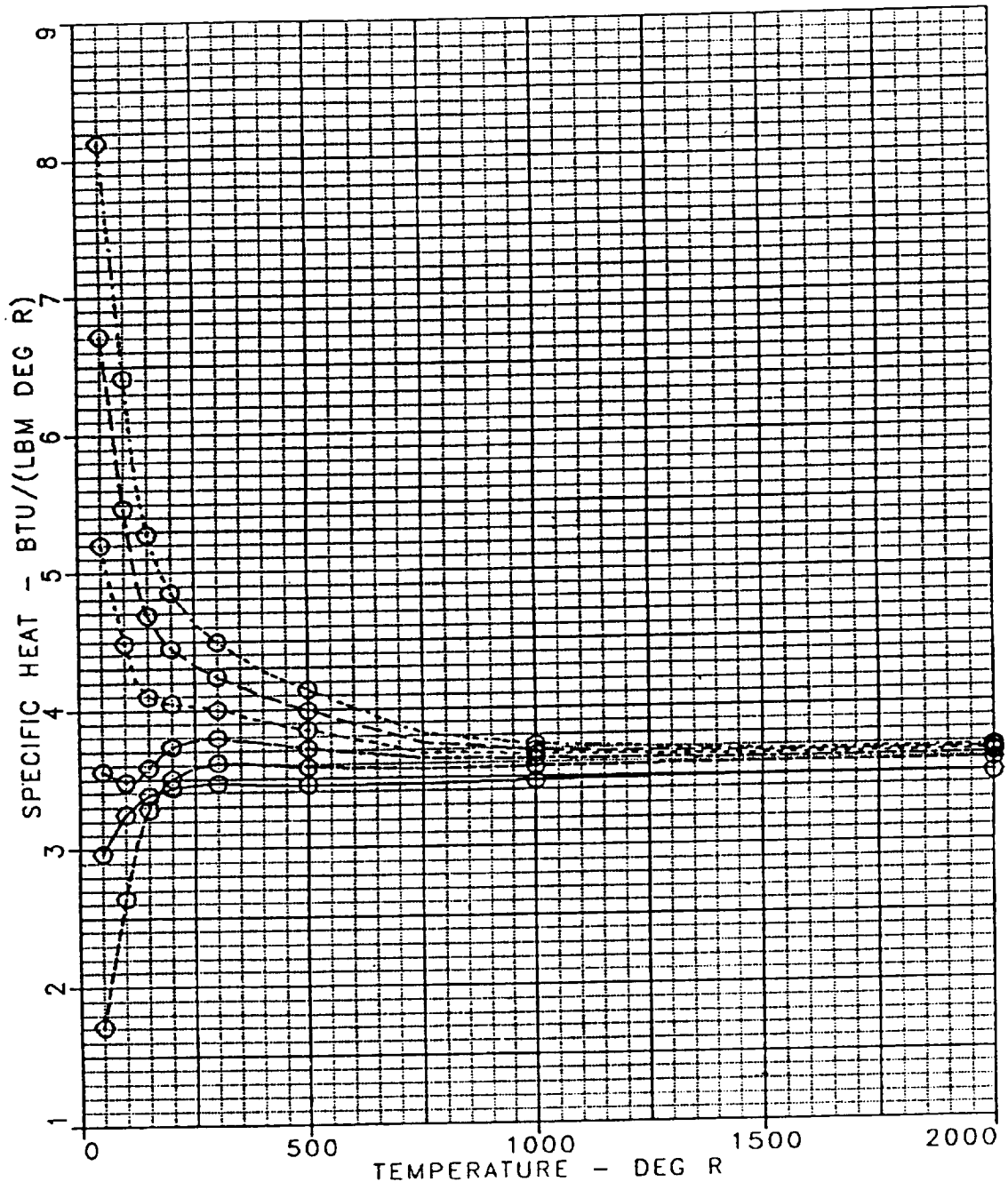


Figure 3.4.2.4.29

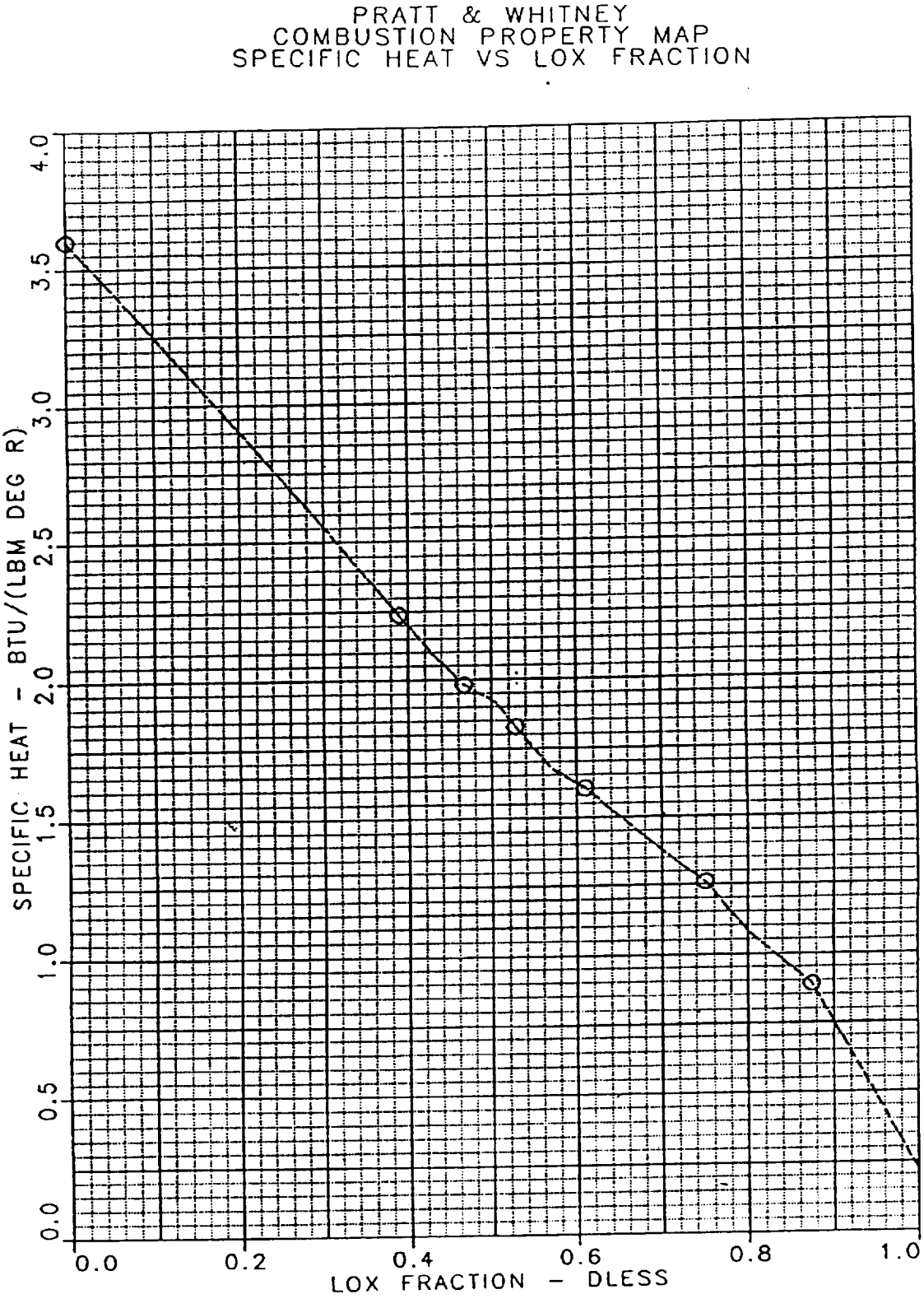


Figure 3.4.2.4.30

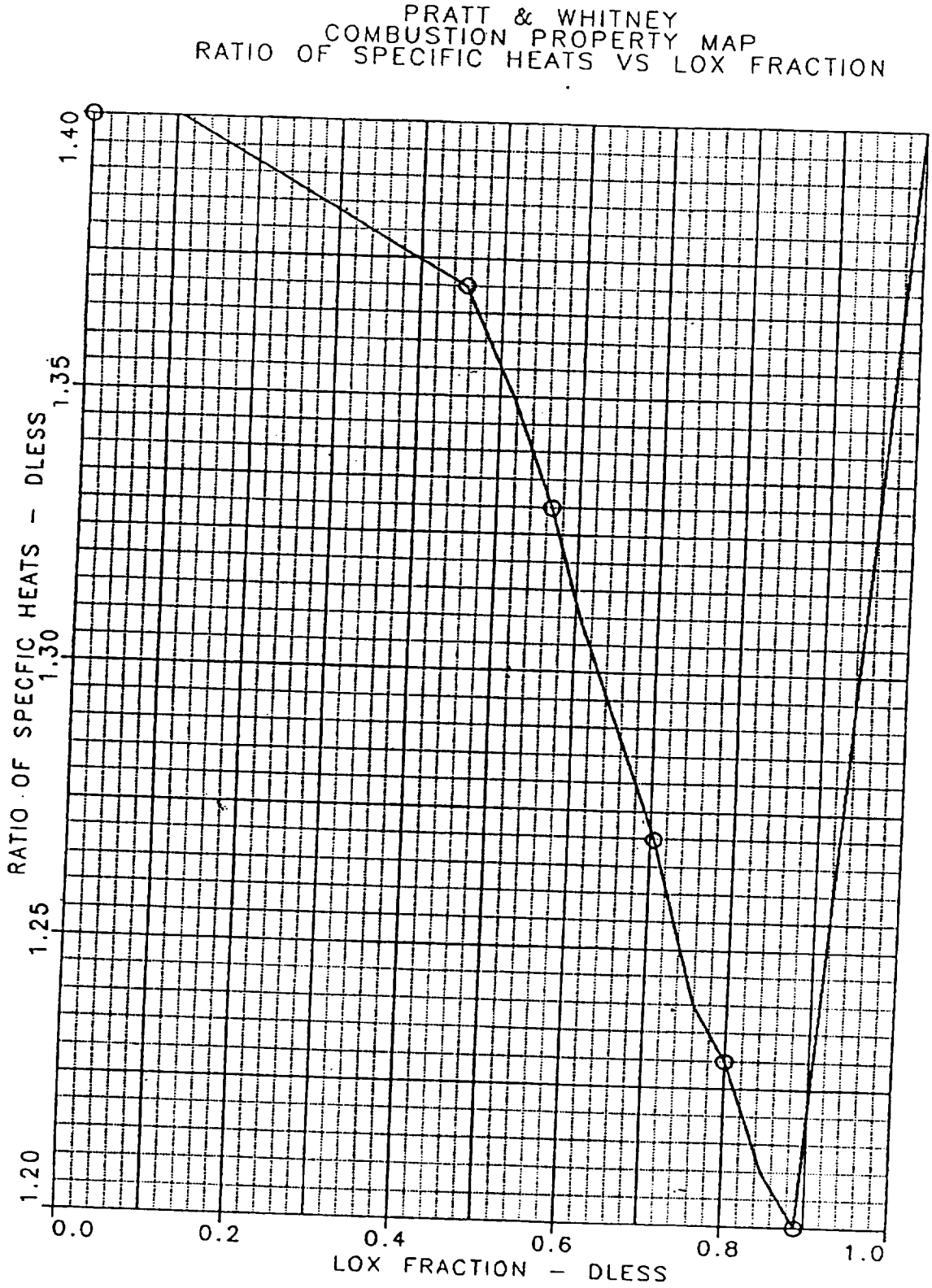


Figure 3.4.2.4.31

PMAPXX - Pump Characteristic MAP Format 1

INPUTS:

IOPT - MAP Option
ϱ - MAP Density
N - Pump Speed
W - Flow
SFSP - Speed Scale Factor
SFW - Flow Scale Factor
SFHD - Head Scale Factor
SFTQ - Torque Scale Factor

OUTPUTS:

HD - Pump Head
TORQ - Pump Torque

DESCRIPTION OF MAP OPTION:

(I) (IOPT = -1) : Values returned in call list are design values (100% RPL).

(II) (IOPT=Ø) : Scale factors are calculated from the input values. (HD and TORQ are inputs when this option is chosen).

$SFSP = N_{DES}/N$ $SFW = W_{DES}/W$
$SFHD = HD_{DES}/HD$ $SFTQ = TORQ_{DES}/TORQ$

(III) (IOPT = 1) : MAP is read (curve fit) by using input values and given scale factors from call list.

$N_{MAP} = N/SFSP$ (MAP Speed)
$W_{MAP} = W/SFW$ (MAP Speed)

CALCULATIONS:

Speed (in radians): $N_R = N_{MAP} \Big/ \frac{2\pi}{60}$

Tip speed is defined as: $U = \frac{D}{2}\left(\frac{2\pi}{60}\right)N_{MAP}$

Flow coefficient (Φ) is defined as: $\Phi = \frac{Q}{AU}$ *(shown below)*

- Dimensional analysis:

By relating the speed of rotation, pump diameter and volumetric flow rate the following equation results:

$$\frac{Q}{ND^3} = CONSTANT$$

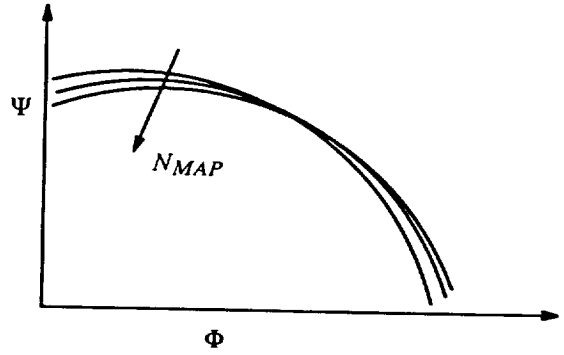
$$= \left(\frac{Q}{D^2}\right)\left(\frac{1}{ND}\right)$$

We know that: $U \; a \; ND$ and $A \; a \; D^2$

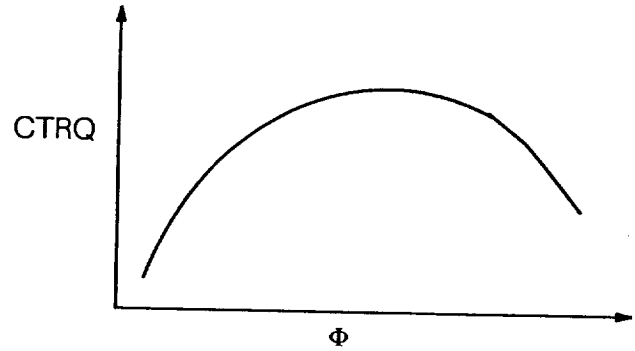
Substituting yields: $\frac{Q}{A}\left(\frac{1}{U}\right) = CONSTANT = \Phi$

$$\boxed{\frac{Q}{AU} = \Phi}$$

Head rise coefficient can now be read from the MAP using flow coefficient and speed as inputs:



Torque coefficient can be read from the MAP using flow coefficient as an input:



NOTE: For option III (IOPT=1) scale factors are applied to the inputs before reading the MAPS as defined under Option II above.

PUMP HEAD:

Dimensional analysis for pump volumetric flow rate, pressure loss, density and diameter yields:

$$\frac{Q}{D^2}\frac{\sqrt{\Delta P}}{\varrho} = CONSTANT$$

Squaring yields $\frac{Q^2}{D^2}\frac{\Delta P}{\varrho} = CONSTANT$ (1)

Pump head is defined as: $H_P = \frac{\Delta P}{\varrho g_r}$ (2)

We know that: $Q \; a \; UD^2$ (3)

Substituting (2) and (3) into (1) : $\frac{U^2D^4}{D^4 H_P G_r} = CONSTANT$

$$H_{MAP} = \frac{U^2}{g_r}\Psi$$

PUMP TORQUE:

Dimensional analysis of pump area, diameter, tip speed and density related to pump torque yields the following:

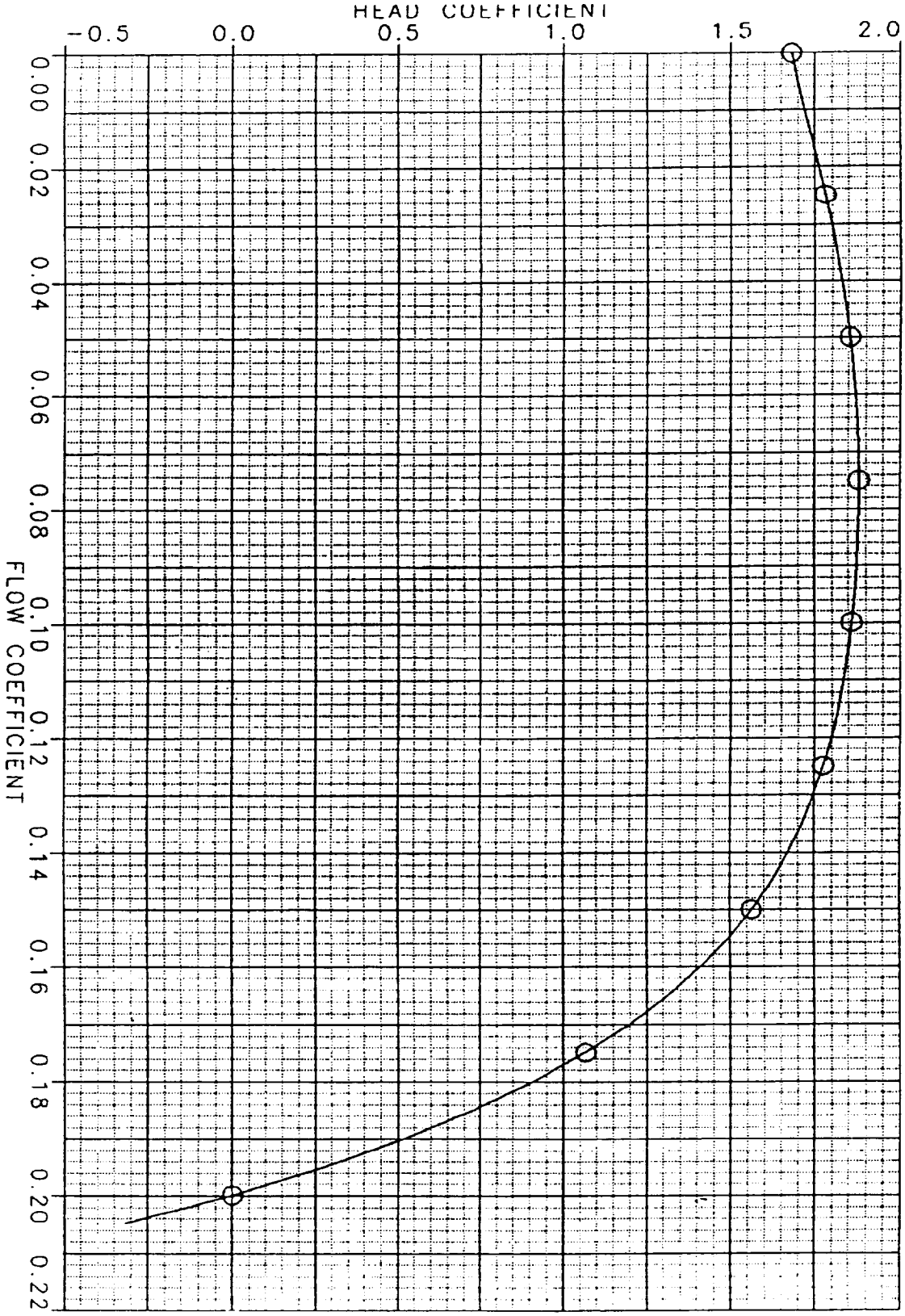
$$T_{MAP} = \frac{\varrho A D U^2}{g_c} \; x \; CTRQ$$

For Option III (IOPT = 1) scale factors are applied to the head and torque.

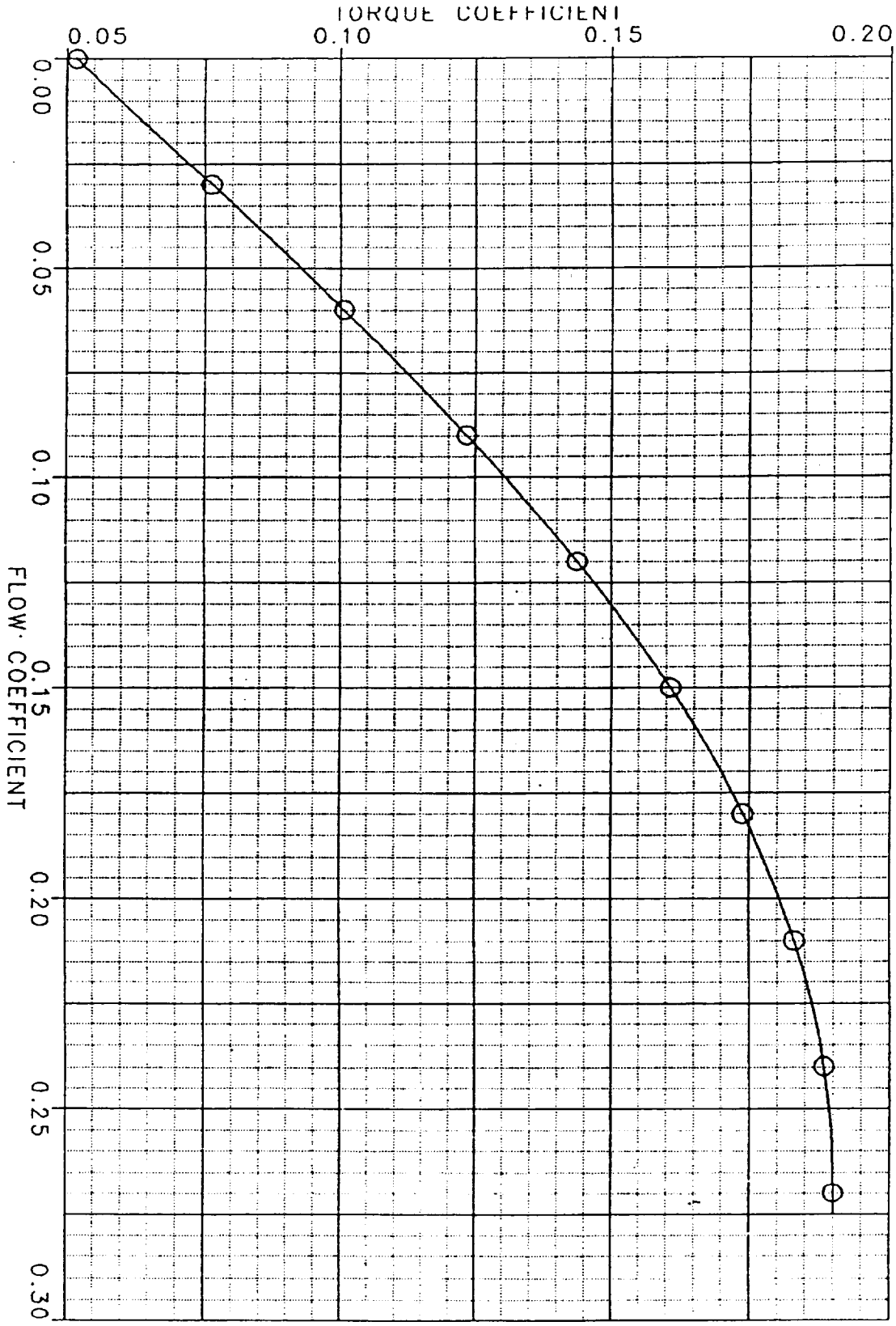
$H_P = H_{MAP} \; x \; SFHD$ (pump head)

$T_P = T_{MAP} \; x \; SFTQ$ (pump torque)

PMAP01 - Pratt & Whitney SSME High Pressure Fuel Pump
Pump Map Format 1

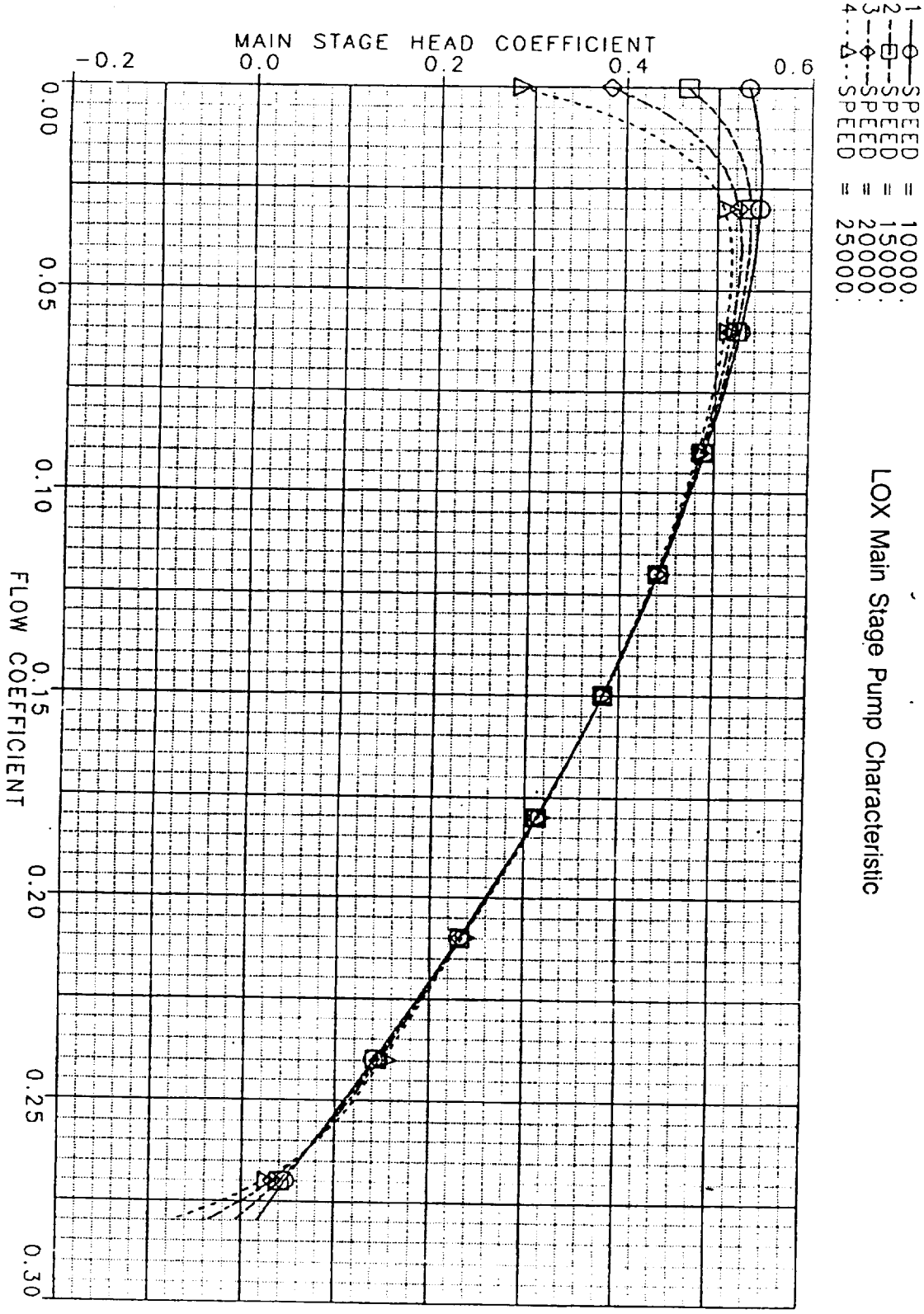


HIGH PRESSURE FUEL PUMP CHARACTERISTIC

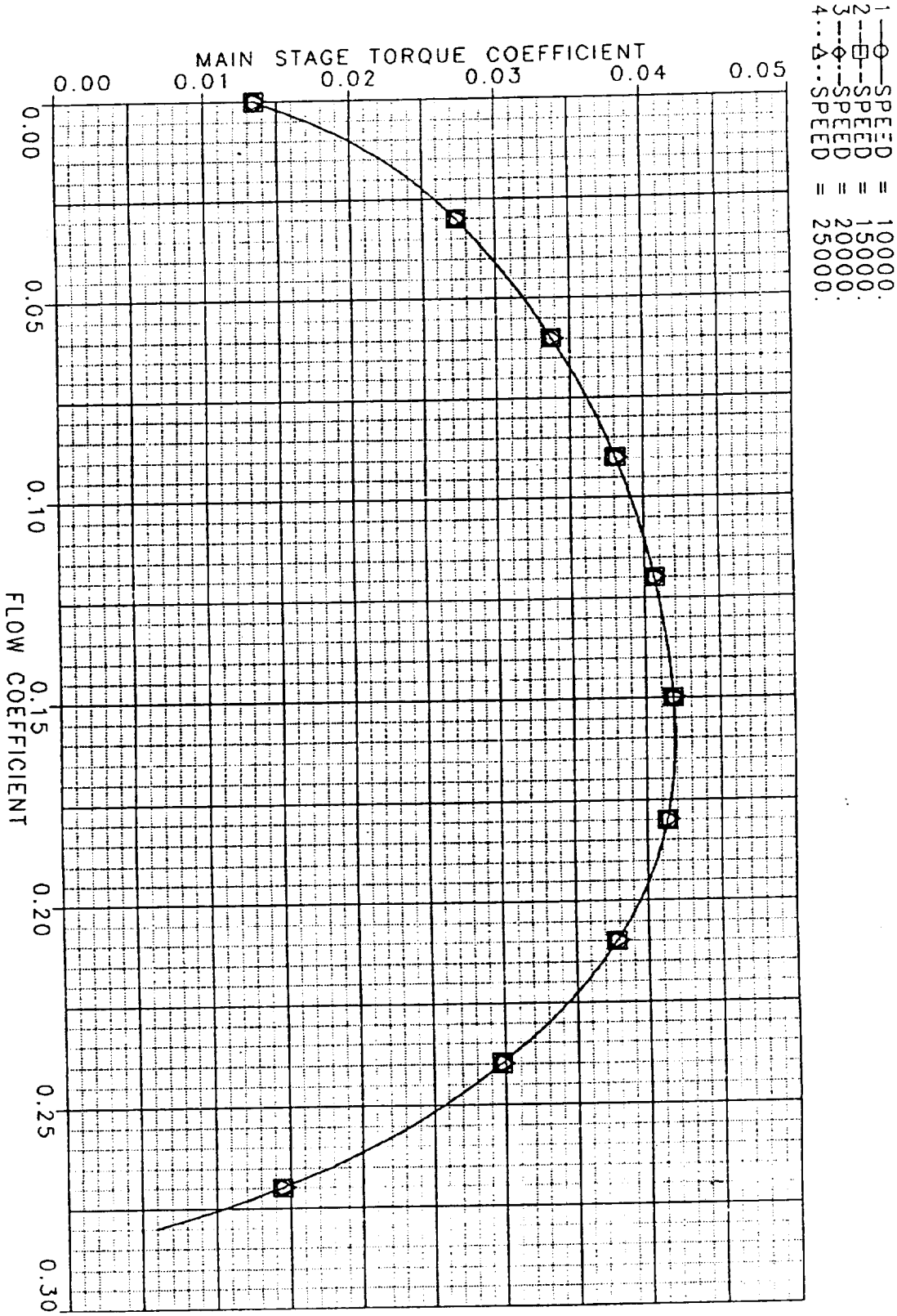


HIGH PRESSURE FUEL PUMP CHARACTERISTIC

PMAP02 - Pratt & Whitney SSME High Pressure LOX Pump (Main Stage) Pump Map Format 1

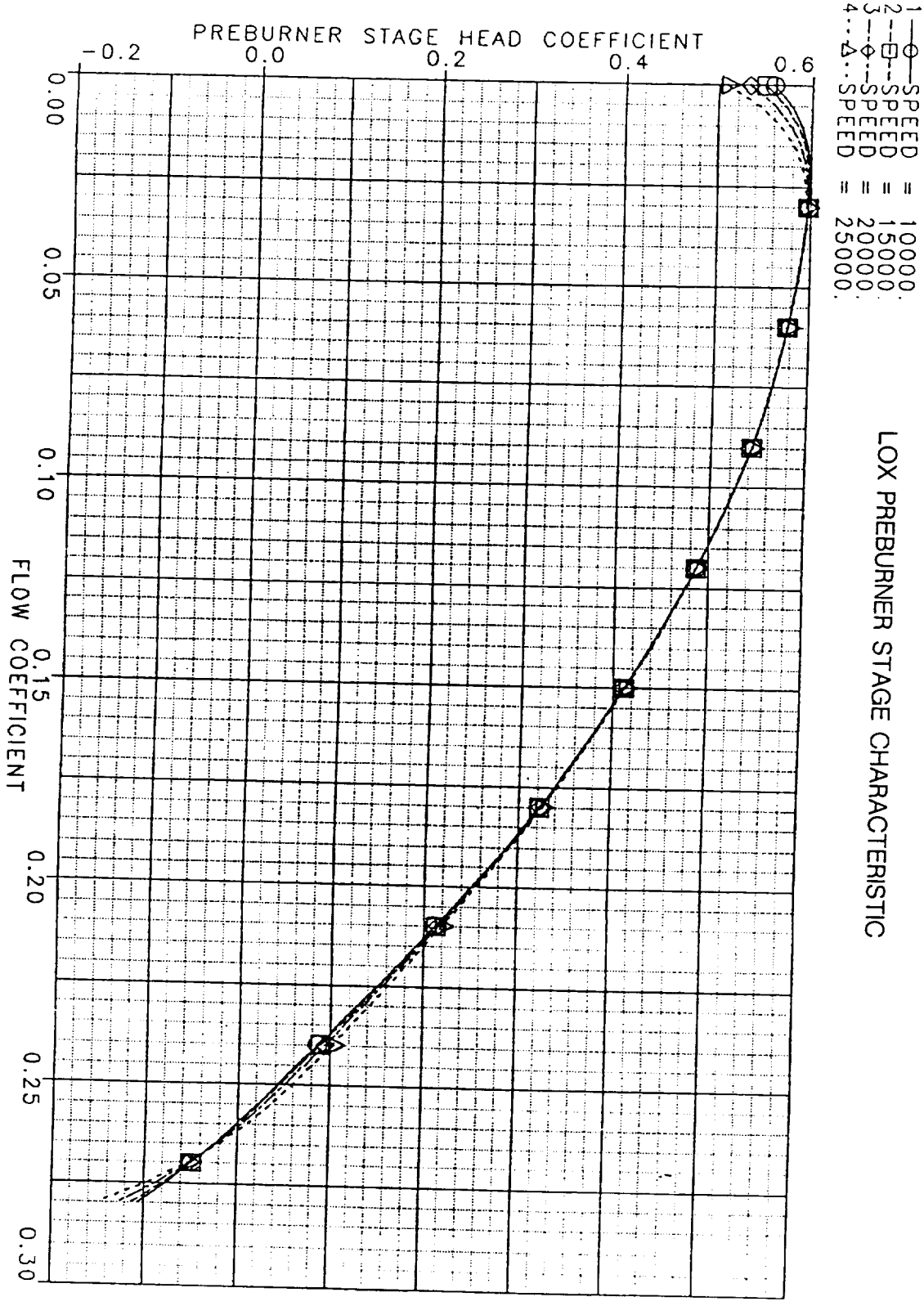


LOX Main Stage Pump Characteristic

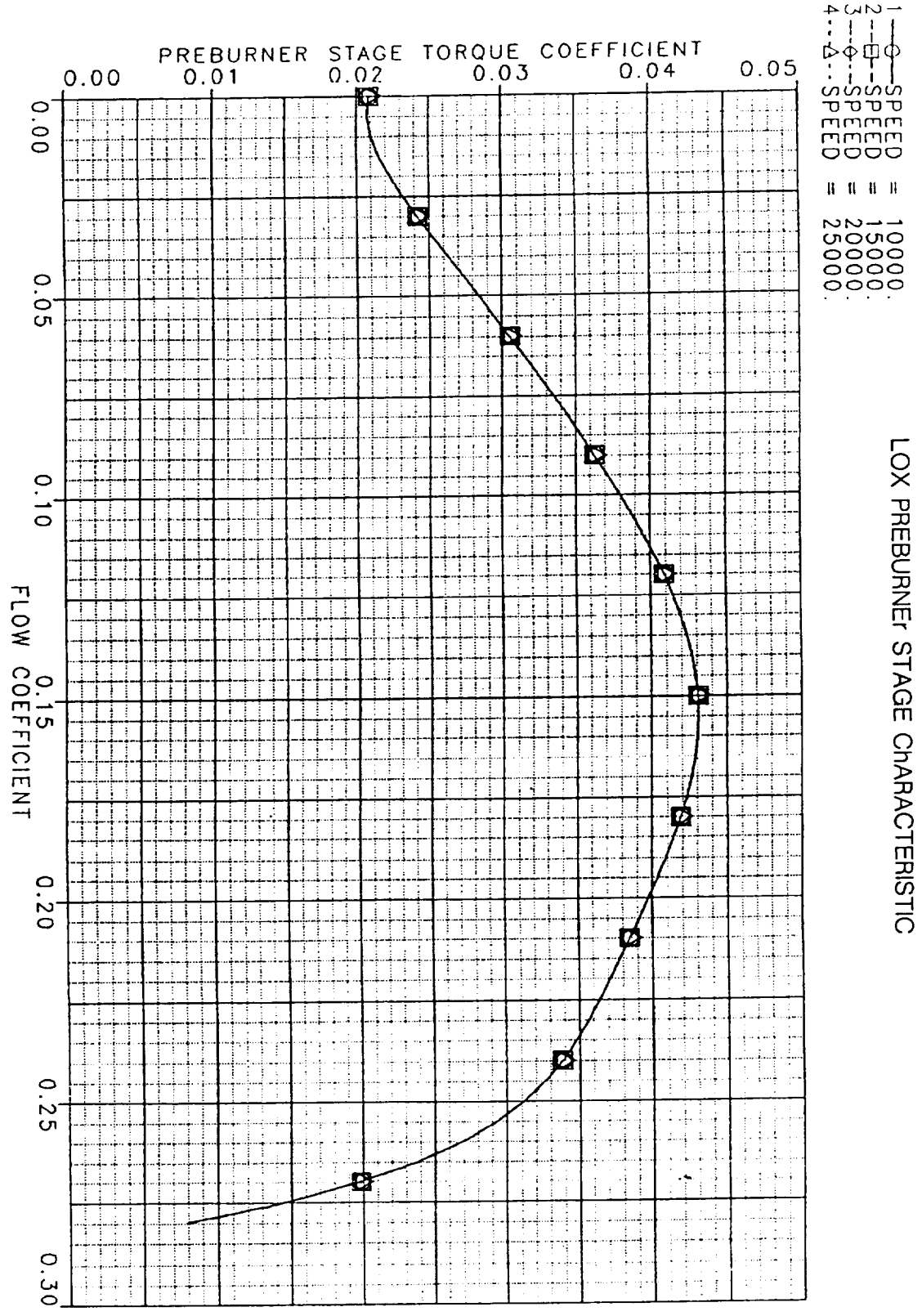


LOX Main Stage Pump Characteristic

PMAP03 - Pratt & Whitney SSME High Pressure LOX Pump (Preburner Stage) Pump Map Format 1



LOX PREBURNER STAGE CHARACTERISTIC



LOX PREBURNEr STAGE ChARACTERISTIC

SUBMODULE TBMPXX - Turbine Characteristic MAP Format 1

INPUTS:

IOPT	-	MAP Option
N	-	Rotational Speed
T_{IN}	-	Inlet Temperature
P_{IN}	-	Inlet Pressure
P_{EX}	-	Exit Pressure
γ	-	Specific Heat Ratio
R	-	Gas Constant
SFSPD	-	Speed Scale Factor
SFPR	-	Pressure Ratio Scale Factor
SFPSI	-	PSI Scale Factor
SFETA	-	Design Point Efficiency Adder

OUTPUTS:

FP	-	Flow Parameter
η	-	Efficiency
$\Delta h'$	-	Ideal Delta h

OPTION ONE (IOPT = -1)

This option will return design conditions (100% RPL) for all arguments in the call list.

OPTION TWO (IOPT = 0)

This option will calculate the scale factors based on the input conditions and the design conditions as follows:

$$SFSPD = N_{Des}/N$$

$$SFPR = \frac{(PR_{DES} - 1)/PR_{DES}}{(PR - 1)/PR}$$

$$SFPSI = PSI_{DES} / PSI$$

$$SFETA = (\eta - \eta_{DES})(SFSPD)^2$$

OPTION THREE (IOPT = 1)

This option reads the MAP using the input parameters and input scale factors to calculate the MAP parameters (actual numbers used when reading the maps).

CALCULATIONS :

Speed parameter: $$Npm = \frac{N}{\sqrt{R \; X \; T_{IN}}}$$

IDEAL Δh :

$$\Delta h' = Cp(T_{IN} - T_{OUT})$$

$$\Delta h' = -Cp \; T_{IN}\left(\frac{T_{OUT}}{T_{IN}} - 1\right) \qquad (1)$$

$$\frac{T_{OUT}}{T_{IN}} = \left(\frac{P_{OUT}}{P_{IN}}\right)^{\frac{\gamma-1}{\gamma}} \qquad (2)$$

$$Cp = \frac{\gamma \; R}{\gamma - 1} \qquad (3)$$

Substitute (2) and (3) into (1)

$$\boxed{\Delta h' = \frac{-T_{IN} \; x \; R}{\gamma - 1}\left[\left(\frac{1}{PR}\right)^{\frac{\gamma-1}{\gamma}} - 1\right]}$$

Tip Speed:

$$U = \frac{D}{2} \; N_r = \frac{D}{2} \; x \; \left(\frac{2\pi}{60}\right)(N)$$

$$\boxed{U = \left(\frac{D}{2}\right)\left(\frac{2\pi}{60}\right)(N)}$$

Axial velocity: Axial velocity, C is obtained from the kinetic energy term in the energy equation such that:

$$\Delta h' = \frac{C^2}{2gc}$$

$$\boxed{C = \sqrt{2gc \; \Delta h'}}$$

Define PSI: PSI = velocity ratio

$$\boxed{PSI = U/C}$$

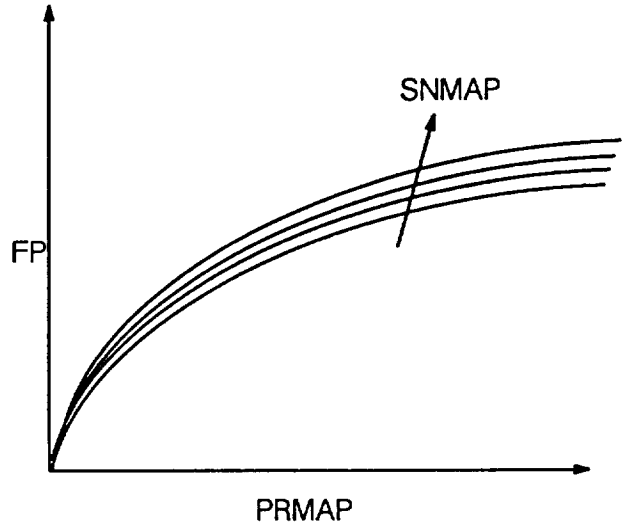
For option 3 (IOPT = 1) MAP parameters are calculated:

$$PRMAP = PR/[PR(1 - SFPR) + SFPR]$$

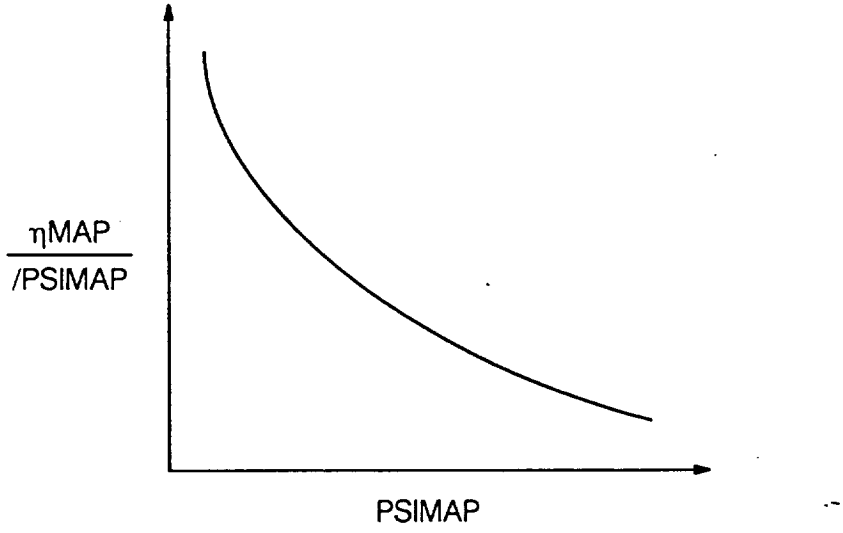
$$SNMAP = N_{pm} \times SFSPD$$

$$PSIMAP = PSI \times SFPSI$$

Flow parameter can now be read from the MAP using pressure ratio and speed parameter as inputs:

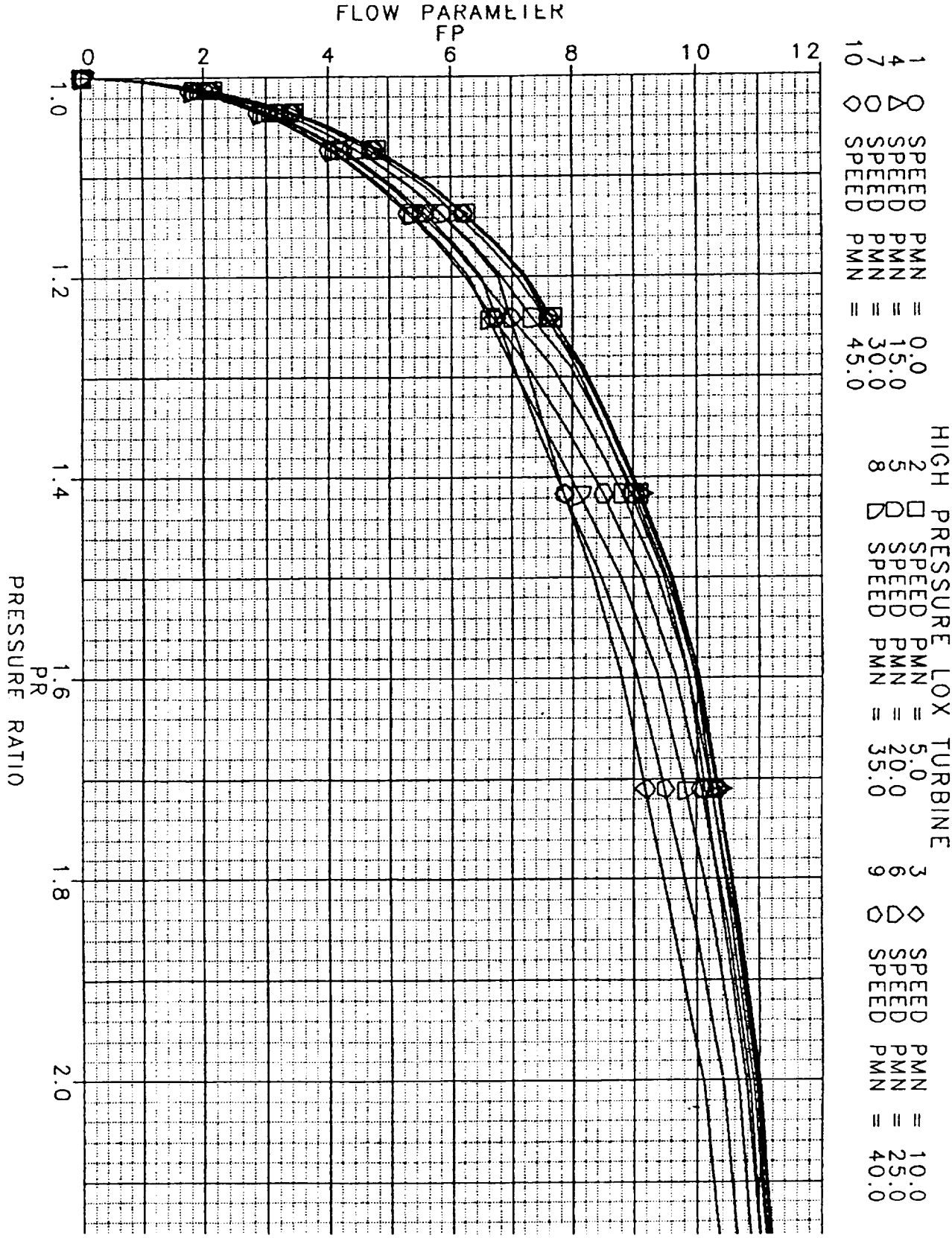


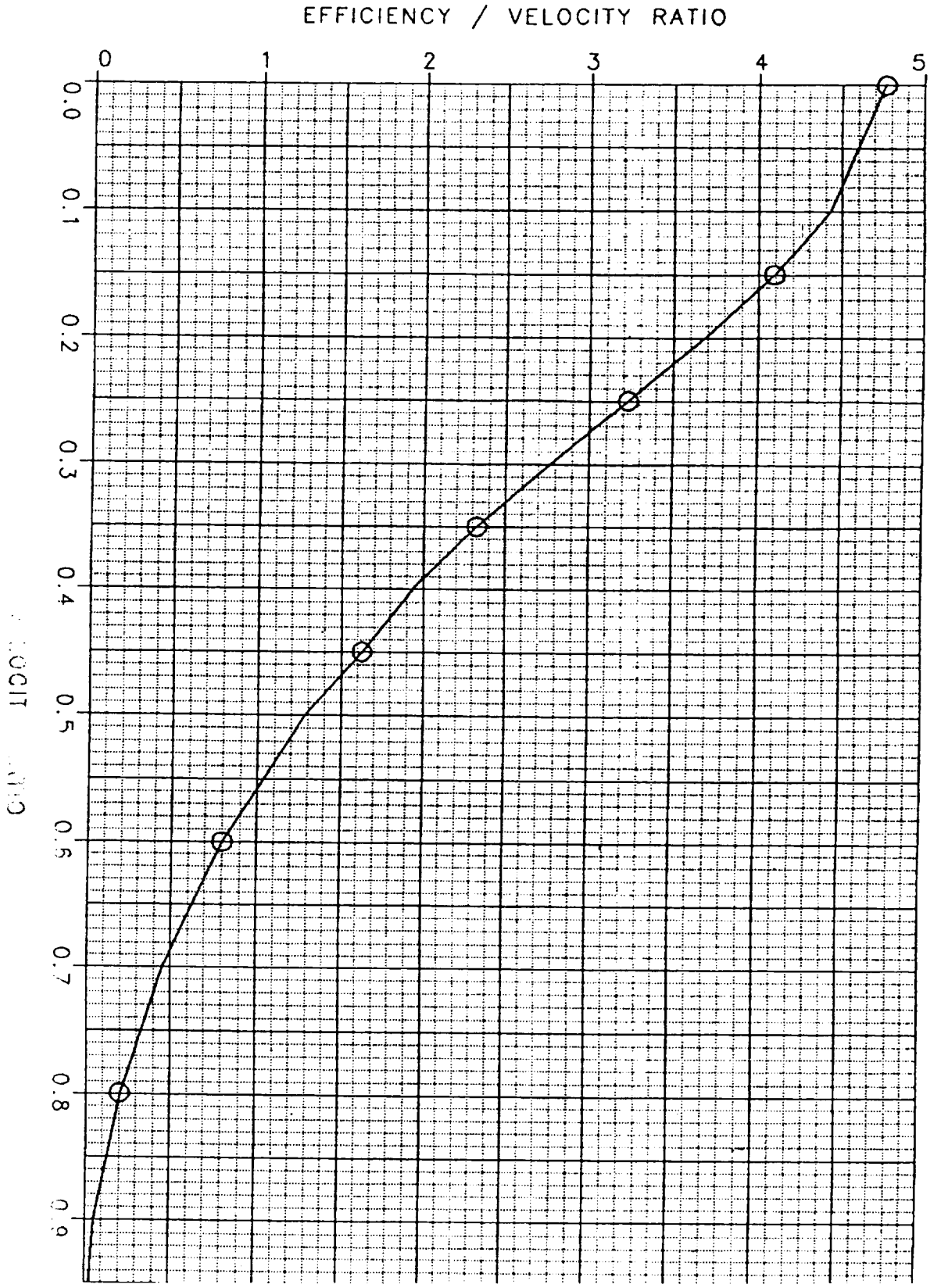
Efficiency can be determined by reading ηMAP/PSIMAP as a function of PSIMAP



EFFICIENCY: $\eta = \eta_{MAP} + SFETA \times (SNMAP / SNDES)^2$

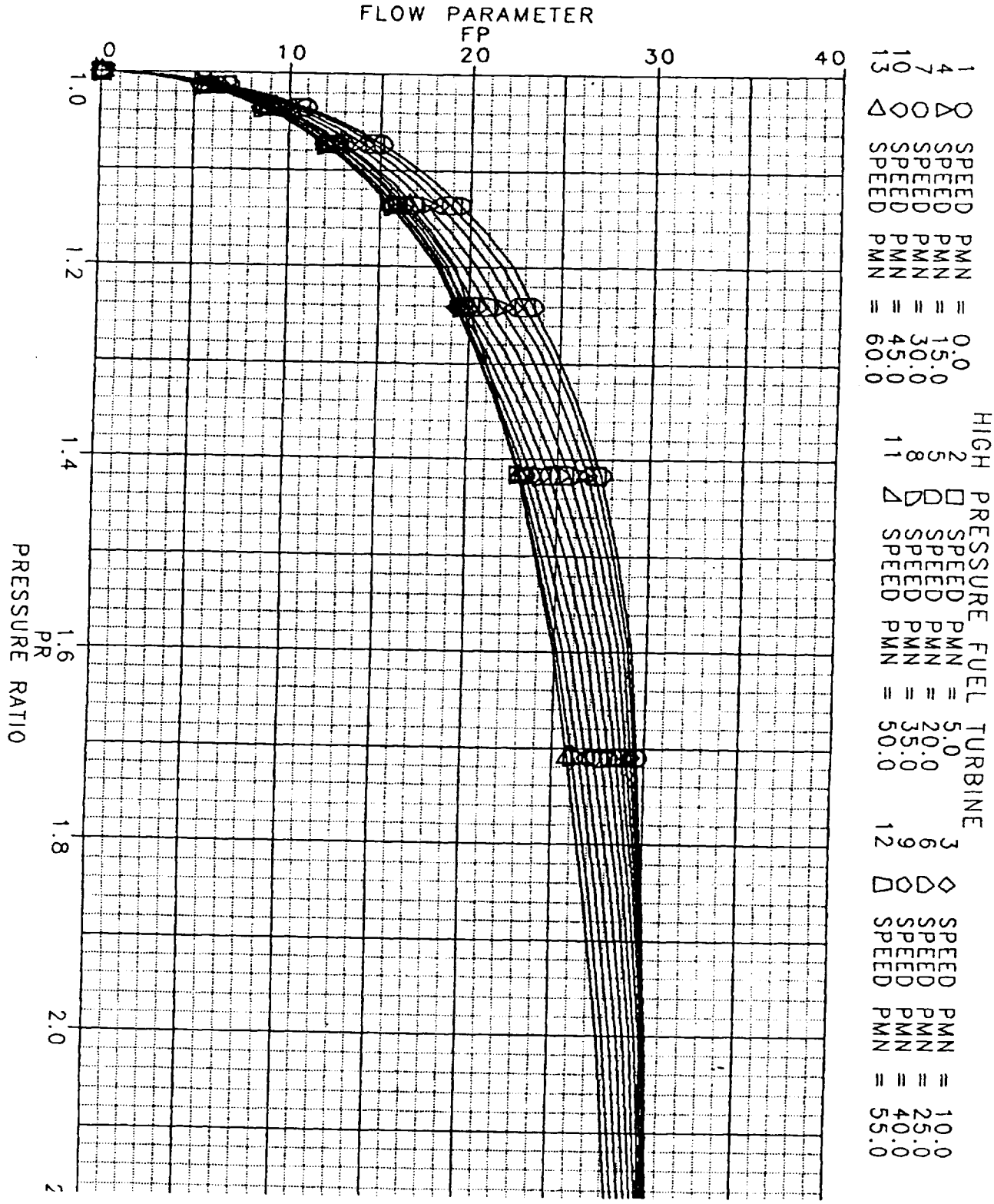
TBMP01 - Pratt & Whitney SSME High Pressure Oxidizer Turbine Turbine Map Format 1.

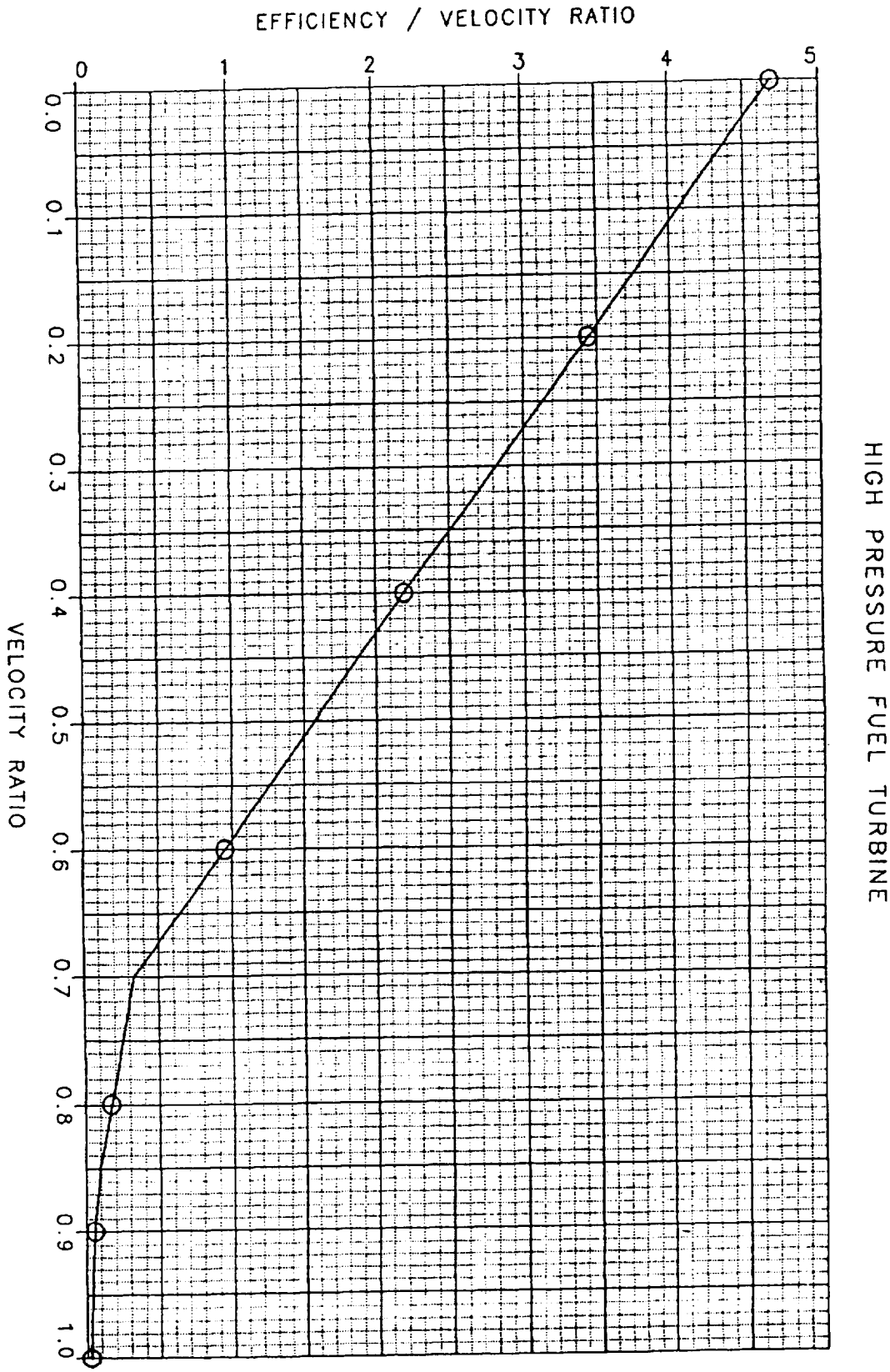

FLOW PARAMETER
FP
0
2
4
6
8
10
12
1.0
1.2
1.4
1.6
1.8
2.0
PR
PRESSURE RATIO
HIGH PRESSURE LOX TURBINE
1 SPEED PMN = 0.0
4 SPEED PMN = 15.0
7 SPEED PMN = 30.0
10 SPEED PMN = 45.0
2 SPEED PMN = 5.0
5 SPEED PMN = 20.0
8 SPEED PMN = 35.0
3 SPEED PMN = 10.0
6 SPEED PMN = 25.0
9 SPEED PMN = 40.0



HIGH PRESSURE LOX TURBINE

TBMP02 – Pratt & Whitney SSME High Pressure Fuel Turbine Turbine Map Format 1.


FLOW PARAMETER
FP
0
10
20
30
40
PRESSURE RATIO
PR
1.0
1.2
1.4
1.6
1.8
2.0
HIGH PRESSURE FUEL TURBINE
1 SPEED PMN = 0.0
4 SPEED PMN = 15.0
7 SPEED PMN = 30.0
10 SPEED PMN = 45.0
13 SPEED PMN = 60.0
2 SPEED PMN = 5.0
5 SPEED PMN = 20.0
8 SPEED PMN = 35.0
11 SPEED PMN = 50.0
3 SPEED PMN = 10.0
6 SPEED PMN = 25.0
9 SPEED PMN = 40.0
12 SPEED PMN = 55.0


EFFICIENCY / VELOCITY RATIO
0
1
2
3
4
5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
VELOCITY RATIO
HIGH PRESSURE FUEL TURBINE

SUBMODULE SMIT01 - Multivariable Iteration Routine

Utility submodule SMIT01 is a multi-variable iteration routine using a modified Newton-Raphson technique. The matrix equation used in SMIT01 is as follows:

$$\Delta Y = J\Delta X$$

Where deltaY is the amount that the errors, or dependent variables, need to change to be zero; and deltaX is the associated change in the independent variable. J, the solver Jacobian, is a matrix of partial derivative generated with the model. SMIT01 makes a number of passes equal to the number of iteration variables plus 1 through the model to generate the Jacobian. The first pass is kept as a base point and the each of the next n variable passes generates a column of the J matrix by perturbing one of the independent variables and observing the changes in the dependent variables from the base point. The perturbation size for each of the independent variables is set through the SMIT01 inpt array DELTAX. Each element of DELTAX is the perturbation magnitude of the corresponding independent variable. The J matrix is then inverted, and changes in the independent variable values, deltaX, are calculated as follows:

$$\Delta X = J^{-1}\ \Delta Y$$

The SMIT01 input array DXALOW is used to limit the change in the independent variables along the same direction as the deltaX vector. This is done to prevent excessive changes in independent variables on a single pass. This process is iterated until the dependent variables are within a specific tolerance. The maximum number of iterations allowed is input to SMIT01 as the variable MXPASS. The tolerance for each variable is set in the SMIT01 input array TOL.

The arrays XVAR and YVAR contain the independent and dependent variables respectively. XVAR is input to SMIT01 and changed as above until each YVAR is within the specified tolerance. The number of variables is input as NVAR.

To improve the efficiency of SMIT01, Broyden's algorithm for updating the inverse J matrix J^{-1}, is incorporated. Broyden's method updates the inverse J matrix without evaluating or inverting a new matrix providing a large savings in number of total passes through the model. The matrix is updated with a secant method and can be performed during a convergence attempt. Broyden's algorithm is applied whenever the reduction in the sum of the squared errors is less than the SMIT01 input SSEBRD. It may still be necessary to evaluate a new J matrix if SMIT01 fails to converge at an acceptable rate. If the improvement in the sum of the squared errors is less than SSEJAC for greater than MXFAIL

consecutive passes, a new J matrix is evaluated. SSEJAC and MXFAIL are SMIT01 inputs.

Other SMIT01 inputs are defined as follows:

1. IPNT = Print flag
   0 = No print
   1 = SMIT01 statistics output at completion

2. KFILE = Logical unit number for print output

3. ILOOP = Loop number for nesting SMIT01. Two SMIT01 loops can be nested; ILOOP is used to distinguish internal variable for each loop.

4. ISAMAT = Flag to use previous J matrix
   0 = Do not use stored J matrix
   1 = Used stored J matrix

5. IDTYPE = Pertubation type for J matrix calculation
   1 = Backward difference
   0 = Central difference
   +1 = Forward difference

The flag ISIG is used to control the program flow. It is set as 0 on the first call to SMIT01 and is set by SMIT01 on subsequent passes as follows:

1 = New X's returned in an attempt to converge
2 = Perturbation pass to create Jacobian matrix
3 = Iteration has converged
4 = Jacobian is singular
5 = Maximum iterations failed to converge

Additional parameters are output as statistics on SMIT01 operation as follows:

PCTRDN = Percent reduction in the sum of squares error for convergence attempt

KOUNT = Convergence attempt counter.

KTOTAL = Total pass counter (including J matrix passes).

KJACOB = Number of J matrices generated

KBROYD = Number of Boyden updates

SUBMODULE SUNB01 -Univariant or bi-variant sunbar-type map reader with option to read map in any direction and either extrapolate or return corner values.

INPUTS

A = Array containing map
X = Absicssa value. Normally first independent variable
Y = Family value. Normally second independent variable
(Not used for univariant curve)
T = Ordinate value. Normally dependent variable

IOPT = OPTION CODE

1 = X and Y are input, T is returned
(X only for univariant)
2 = T and Y are input, X is returned
(T only for univariant)
3 = X and T are input, Y is returned
(Not applicable for univariant)

OUTPUTS:

IERRX = Error code describing the absicssa

0 = Normal return
1 = X value too small. Lowest X used.
2 = X value too small. Extrapolated value used.
3 = X value too large. Largest X used.
4 = X value too large. Extrapolated value used.

IERRY = Error code describing the family

0 = Normal return
1 = Y value too small. Lowest Y used.
2 = Y value too small. Extrapolated value used.
3 = Y value too large. Largest Y used.
4 = Y value too large. Extrapolated value used.

Method of setting up the A array

Absicsssa = X
Ordinate = T
Family = Y

A (1) = Storage space for X position indicator -- initially zero
A (2) = Storage space for Y position indicator -- initially zero, or curve identification number
A (3) = Number of X values defining each line

A (4) = Number of Y values (must be zeero for univariant curve) then, beginning with A(5), the map is input as:

A (5) - 4(4+NX) = X values in ascending order
A (5+NX) - A(4+NX+NY) = Y values in ascending order
A (5+NX+NY) = T for X1,Y1
A (6+NX+NY) = T for X1,Y2

A(2*NY+NX+4) = T for X1,YNY
A(2*NY+NX+5) = T for X2,Y1
A(2*NY+NX+6) = T for X2, Y2

A(3*NY+NX+4) = T for X2,YNY

A(NX+NY*NX+NY+4) = T for XNX,YNY

NOTE: In the above setup, NX=number of X'S, NY=number of Y'S for bivariant curve. Curve for univariant, A(5) - A(4+NX) contains the X values, A(5+NX) - A(2*NX+4) contains the T values..

COMMENTS:

1. For normal reading (IOPT=1), X is bracketed first, followed by Y bracketing. Since X and Y are bracketed independently, care must be exercised when setting up maps to ensure correct values will result. If maps are not fairly flat, a corresponding point map reader should be used.

2. For reading X as a function of T and DY (IOPT=2), Y is bracketed first, followed by T bracketing. When corner values are returned, they look as if X and T were swapped.

3. When reading Y as a function of T and X (IOPT=3), X is bracketed first, followed by T bracketing. This option is invalid for univariant curves.

4. Univariant and Bivariant curves are calculated similarly but there are no Y-calculations.

5. Multiple valued functions (more than 1 T for given X and Y) should not be read with Option 2.