NASA Technical Memorandum 83458

Propeller Performance and Weight Predictions Appended to the Navy/NASA Engine Program

(NASA-TM-83458) PROPELLER PERFORMANCE AND WEIGHT PREDICTIONS APPENDED TO THE NAVY/NASA ENGINE PROGRAM (NASA) 41 p HC A03/MF A01 CSCL 21E
N83-32810
Unclas
G3/07 28598

R. M. Plencner, P. Senty,
and T. J. Wickenheiser
Lewis Research Center
Cleveland, Ohio

August 1983

NASA

E-1773

PROPELLER PERFORMANCE AND WEIGHT PREDICTIONS

APPENDED TO THE NAVY/NASA ENGINE PROGRAM

R. M. Plencner, P. Senty, and T. J. Wickenheiser

National Aeronautics and Space Adminstration
Lewis Research Center
Cleveland, Ohio 44135

SUMMARY

The Navy/NASA Engine Program (NNEP) is a general purpose computer program currently employed by government, industry and university personnel to simulate the thermodynamic cycles of turbine engines. NNEP is a modular program which has the ability to evaluate the performance of an arbitrary engine configuration defined by the user.

In 1979, a program to calculate engine weight (WATE-2) was developed by Boeing's Military Division under NASA contract. This program uses a preliminary design approach to determine engine weights and dimensions. Because the thermodynamic and configuration information required by the weight code was available in NNEP, the weight code was appended to NNEP.

Due to increased emphasis on fuel economy, a renewed interest has developed in propellers. This report describes the modifications developed by NASA to both NNEP and WATE-2 to determine the performance, weight and dimensions of propellers and the corresponding gearbox. The propeller performance model has three options, two of which are based on propeller map interpolation. Propeller and gearbox weights are obtained from empirical equations which may easily be modified by the user.

INTRODUCTION

Due to the tremendous cost of turbine engine hardware, the ability to analytically study the feasibility of advanced engine concepts is essential. One tool used to model the thermodynamic cycles of turbine engines is the Navy/NASA Engine Program (NNEP). NNEP has the ability to model arbitrary turbine cycles defined by the user. For design point calculations, the user inputs the engine configuration, component performance and optimization information. For off-design calculations, component performance is determined using component maps. User-defined free variables are iterated to match component gas flows, rotational speeds, and power. For a further discussion of NNEP the user is referred to reference 1.

In 1979, under NASA sponsorship, Boeing developed a computer program called WATE-2 that predicts weights of turbine engines (ref. 2). This program uses a preliminary design approach in which components are sized using geometric and physical constraints along with thermodynamic cycle data. A logical source of this thermodynamic information is NNEP. Therefore, WATE-2 was written to be an integral part of NNEP but can be adapted to any thermodynamic cycle code.

Renewed interest in propellers has grown rapidly due to high fuel prices. While NNEP is a very versatile program, its ability to model propellers is quite limited. The purpose of this paper is to describe modifications made to both NNEP and WATE-2 to determine propeller performance and weight. Three options of modeling propeller performance with varying degrees of sophistication are available. The first option is simply a constant efficiency input. The second option interpolates from a generalized propeller map which is scaled for various integrated design lift coefficients, number of blades, activity factors and power coefficients. The third option interpolates from a user-supplied propeller performance map.

Propeller weight and gearbox weight and dimensions are obtained from empirical relationships. These relationships are defaulted into WATE-2. The user may input his own empirical relations. The method is described later in the section on weight and dimension analysis. In the following discussions it is assumed that the reader has a basic understanding of NNEP and WATE-2.

SYMBOLS

AF	Blade activity factor
B	Number of blades
Cp	Power coefficient $= \dfrac{550\ SHP}{\rho\ n^3\ D^5}$
Ct	Thrust coefficient $= \dfrac{T}{\rho\ n^2\ D^4}$
Cw	Weight of propeller counterweights, lb
D	Propeller diameter, ft
GR	Gear ratio = propeller speed/engine speed
J	Propeller advance ratio = V/nD
M	Airplane Mach number
N	Propeller speed, rpm
n	Propeller speed, rps
PL	Power loading, SHP/D^2
SHP	Shaft horsepower
T	Propeller thrust, lb
TS	Propeller tipspeed, ft/sec
V	Flight velocity - true airspeed, ft/sec
Wt	Component weight, lb
η	Propeller efficiency, J(Ct/Cp)
ρ	Mass density of air, slug/ft^3

PROPELLER PERFORMANCE

The first option of propeller performance modeling uses a fixed propeller efficiency. The efficiency may be input as a different value from case to case but it is not dependent on propeller operating conditions. Propeller thrust is then simply calculated from

$$T = 550 \times SHP \times \eta / V$$

where η is the input propeller efficiency and V is the forward velocity in ft/sec.

For the static case this equation is indeterminate. Therefore, the thrust is calculated from the input static thrust-to-horsepower ratio.

$$T = (\text{thrust}/SHP)_{\text{static}} \times SHP$$

Thus, only the shaft power, the efficiency, and static thrust-to-horsepower ratio are required for this option. However, if the component weight is desired the propeller tip speed, power loading, number of blades, and activity factor are also required.

The second option utilizes a generalized propeller map. This procedure was developed by Hamilton Standard under NASA contract (ref. 3). The advance ratio J and power coefficient Cp are first determined from the following equations:

$$J = \frac{\pi V}{TS}$$

$$Cp = \frac{550\ SHP}{\rho n^3 D^5}$$

where

$$D = \sqrt{SHP/DL}$$

$$n = \frac{TS}{\pi D}$$

These values of Cp and J are used in a generalized map defined for the specified number of blades to obtain a thrust coefficient Ct. The generalized maps have a base activity factor of 150 and a base integrated design lift coefficient of 0.5. Thus, the value of Ct is then modified through the use of adjustment factors to account for the effects of integrated design lift coefficient and activity factor. Finally, the propeller efficiency and thrust are determined by

$$\eta = J\left(\frac{Ct}{Cp}\right)$$

$$T = Ct \rho n^2 D^4$$

For this option it is necessary for the user to know shaft horsepower, propeller power loading, number of blades, integrated design lift coefficient, blade activity factor, and design tip speed.

ORIGINAL PAGE IS
OF POOR QUALITY

The generalized maps were developed to be capable of predicting performance for propellers with 2 to 8 blades, 80 to 200 blade activity factors, and 0.3 to 0.8 integrated design lift coefficients. It should be noted that the original study only covered general aviation powerplants in the size range of 100 to 1500 shaft horsepower. The validity of performance predictions of propellers used for powerplants outside this size range is unknown.

The third option allows the user to input his own propeller map. Propeller maps are input with Ct as a function of Cp and J. The format of the map and a typical map are listed in appendix A. The format is the standard format required for all NNEP maps. The method of analysis is essentially the same for option 3 as it was for option 2. Advance ratio and power coefficient are first determined. These values are then used to obtain Ct from the input map. However, the value of Ct is not corrected for activity factor, number of blades, and integrated lift coefficient since these values are assumed implicitly fixed for a given map. Therefore, inputting these values has no effect on the performance calculated using option 3. The propeller efficiency and thrust are then obtained the same as in option 2.

In both option 2 and option 3 the user has two inputs available which allow the results predicted by the maps to be scaled. One input allows the user to apply a constant scale factor to all thrusts and efficiencies predicted by the maps. The other input allows the user to input a desired design point propeller efficiency. The program then sets up a scale factor which is the ratio of the desired design point efficiency to the map calculated efficiency. This scale factor is then applied to the map predicted performance in subsequent off-design cases. If both of these inputs are specified the constant scale factor overrides the desired design point efficiency.

A list of the user-required inputs for each of the three options is given in table I. A more complete description of all the inputs is given in the User Manual (appendix B).

While using any of the above options the user may desire a specified thrust from the engine. This may be obtained by the use of NNEP controls. To obtain the desired thrust for the design case the user may vary the inlet airflow or the design turbine inlet temperature. For the off-design case the user may vary the turbine inlet temperature to obtain the desired thrust.

PERFORMANCE PROGRAM STRUCTURE

Propeller performance calculations are controlled by a subroutine called NNPROP. NNPROP is called from subroutine FLOCAL after the engine matching is complete. A copy of subroutine NNPROP is given in appendix C.

When a constant efficiency is input (option 1) all propeller performance calculations are performed within NNPROP. Note that for this option NNPROP uses the same three inputs that were originally required by NNEP to run a propeller. Therefore, any data sets which were previously used on NNEP should run with this new version without change.

When the generalized maps are used (option 2) NNPROP makes a call to subroutine PERFM. This subroutine contains the generalized maps and the correction for integrated lift coefficient, number of blades, and activity factor.

The interpolation of the maps is done by two utility subroutines, BIQUAD and UNINT. A copy and a detailed description of these three subroutines are given in reference 4.

For user-supplied maps (option 3), NNPROP calls the NNEP subroutine TLOOK to do the map interpolations.

WEIGHT AND DIMENSION ANALYSIS

WATE-2 was originally constructed to give weights and dimensions of components based on a preliminary design analysis. The routine to determine propeller and gearbox weight and dimensions, however, is based entirely on empirical data.

Generalized propeller weight equations were obtained from Hamilton Standard, reference 5. One equation defines the weight of double-acting propellers as follows:

$$Wt_{PROP} = Kw\left[\left(\frac{D}{10}\right)^2\left(\frac{B}{4}\right)^{0.7}\left(\frac{AF}{100}\right)^{0.75}\left(\frac{ND}{20\ 000}\right)^{0.5}(M+1)^{0.5}\left(\frac{SHP}{10\ D^2}\right)^{0.12}\right],\ \mathrm{lb}$$

where Kw = 355.

A second set of equations predicts the weight for single acting propellers with counterweights:

$$Wt_{PROP} = Kw\left[\left(\frac{D}{10}\right)^2\left(\frac{B}{4}\right)^{0.7}\left(\frac{AF}{100}\right)^{0.7}\left(\frac{ND}{20\ 000}\right)^{0.4}(M+1)^{0.5}\left(\frac{SHP}{10\ D^2}\right)^{0.12}\right]+Cw,\ \mathrm{lb}$$

where Kw = 220 and

$$Cw = 5\left[\left(\frac{D}{10}\right)^2 B\left(\frac{AF}{100}\right)^2\left(\frac{20\ 000}{ND}\right)^{0.3}\right]$$

To provide full flexibility, the propeller weight equation may be defaulted at the above values or the constant Kw and the parameter exponents may be input by the user. However, if the user desires to change any part of the equation, the constant and all the exponents must be input. It is also possible to input a constant value for the counterweights Cw, if desired.

The value of Kw is defaulted at 355 for double-acting propellers and at 220 for single-acting counterweighted propellers. These values imply current technology propellers using solid aluminum blades. The value of 355 represents a higher speed application normally associated with double-acting propellers, than the 220 value. Values of Kw ranging between 160 to 180 may be assumed for advanced technology fiberglass or composite propellers. Other parameters required for the propeller weight equation, such as shaft horsepower, propeller rpm, etc., are obtained from maximum power conditions encountered in NNEP.

There is also a provision for calculating gearbox weight. Figure 1 represents a defaulted curve used to estimate the gearbox weight as a function of output torque for a gear ratio of 0.118. For other gear ratios

$$Wt_{GB} = (Wt_{GB})_{unscaled} \left(\frac{.118}{GR}\right)^{0.5}$$

The user may specify a different equation for weight as a function of torque by inputting the slope and y intercept. This curve must be input at a gear ratio of 0.118.

Dimensions of the gearbox can also be calculated if desired. Defaulted values of these equations are

$$\text{Length} = 0.125\ (SHP)^{0.5}\ (1/GR)^{0.33},\ \text{in.}$$

$$\text{Diameter} = 0.230\ (SHP)^{0.5}\ (1/GR)^{0.33},\ \text{in.}$$

These equations are based on empirical data for in-line gearboxes in the 1000 to 2500 horsepower range. All of the above equations may be modified by the user (see appendix B, User Manual). No provision is currently available for offset gearbox dimensions.

WEIGHT PROGRAM STRUCTURE

The control routine for WATE-2 is called WTEST. WTEST calls a subroutine (PROPWT) which performs the weight and dimension calculations. Copies of both routines are given in appendix C. Changes to the original WTEST are denoted by comment cards.

CONCLUSION

As interest increases in propeller research, analytical tools which accurately model both engine and propeller performance will be required. The modifications described herein to NNEP and WATE-2 provide such models. The program has been kept general enough to permit the user to update models as more detailed information becomes available. The weight and dimension calculations, however, should be used with the understanding that the results are only rough estimates. It is suggested that the user examine these equations and make changes as required.

APPENDIX A

PROPELLER MAP INPUTS

Each propeller map has the following input data card setup.

Card 1 Table reference number (Integer, col 2-5) Table identification label, Col 6-75

Card 2 Z-Identifier (4 Character Symbol, col 1-4). Enter any dummy symbol because Z values are not used.
NZ-Number of Z values (Integer, col 6 & 7). Must enter a value of 1.
Z-Variable values, 7F10., Beginning in Column 11. Must enter a value of 0.

Card 3 Advance ratio identifier (4 Character Symbol, col. 1-4)
NJ-Number of advance ratio values (Integer, col 6 & 7)
XJ-Advance ratio values, 7F10., Beginning in Column 11. If needed, extra cards follow 10X,7F10. Format.
Advance ratio values MUST be in ascending order.

Card 4 Power coefficient identifier (4 Character Symbol, col 1-4)
NCP-Number of Cp values (Integer, col 6 & 7)
Cp values, 7F10., Beginning in Column 11. If needed, extra cards follow 10X,7F10. Format.
Cp values MUST be in ascending order.

Card 5 Thrust coefficient identifier (4 Character Symbol, col 1-4)
NCP-Number of Cp values (Integer, col 6 & 7)
Ct values, 7F10., Beginning in Column 11. If needed, extra cards follow 10X,7F10. Format. These Values correspond to the values on the Cp Identifier Card.

Last Card 3 Character Symbol EOT in col 1-3

The propeller maps must be in the same dataset containing the other maps to be used by NNEP (i.e. turbine maps, compressor maps, etc.)

SAMPLE PROPELLER MAP

ORIGINAL PAGE IS
OF POOR QUALITY

4100 H.S. ADV PROP 4 BLADE 100 AF .55 CLI

Z 1	0.0						
J 16	0.0	0.2	0.4	0.6	0.8	1.0	1.2
	1.4	1.6	1.8	2.0	2.2	2.4	2.6
	2.8	3.0					
CP 10	0.0568	0.0737	0.0926	0.1170	0.1484	0.1866	0.2287
	0.2747	0.3192	0.3558				
CT 10	0.1456	0.1732	0.1965	0.2179	0.2327	0.2456	0.2531
	0.2559	0.2565	0.2541				
CP 10	0.0499	0.0655	0.0841	0.1058	0.1299	0.1614	0.1999
	0.2406	0.2818	0.3200				
CT 10	0.1084	0.1354	0.1626	0.2890	0.2083	0.2300	0.2480
	0.2580	0.2605	0.2600				
CP 11	0.0406	0.0564	0.0769	0.1001	0.1253	0.1547	0.1885
	0.2234	0.2673	0.3840	0.3200			
CT 11	0.0622	0.0931	0.1237	0.1532	0.1812	0.2075	0.2290
	0.2371	0.2470	0.2620	0.2640			
CP 10	0.0369	0.0564	0.0819	0.1110	0.1421	0.1758	0.2133
	0.2561	0.3007	0.3200				
CT 10	0.0362	0.0712	0.1051	0.1376	0.1685	0.1972	0.2229
	0.2439	0.2551	0.2600				
CP 9	0.0482	0.0797	0.1156	0.1541	0.1940	0.2378	0.2850
	0.3347	0.3857					
CT 9	0.0447	0.0828	0.1186	0.1525	0.1844	0.2136	0.2389
	0.2572	0.2649					
CP 10	0.0323	0.0697	0.1141	0.1607	0.2093	0.2597	0.3149
	0.3722	0.4264	0.4821				
CT 10	0.0158	0.0585	0.0985	0.1355	0.1706	0.2025	0.2321
	0.2561	0.2694	0.2696				
CP 8	0.0536	0.1070	0.1639	0.2222	0.2821	0.4117	0.4777
	0.5345						
CT 8	0.0341	0.0785	0.1194	0.1527	0.1930	0.2533	0.2740
	0.2782						
CP 9	0.0337	0.0965	0.1654	0.2346	0.3055	0.3768	0.4542
	0.5307	0.5997					
CT 9	0.0108	0.0600	0.1052	0.1462	0.1850	0.2188	0.2505
	0.2754	0.2883					
CP 9	0.0695	0.1337	0.2011	0.2657	0.3323	0.4006	0.4667
	0.5395	0.6084					
CT 9	0.0339	0.0744	0.1121	0.1466	0.1796	0.2105	0.2364
	0.2624	0.2823					
CP 8	0.0407	0.1134	0.1913	0.2692	0.3464	0.4246	0.5015
	0.5815						
CT 8	0.0094	0.0543	0.0957	0.1338	0.1691	0.2026	0.2318
	0.2588						
CP 6	0.0937	0.1836	0.2749	0.3643	0.4542	0.5855	
CT 6	0.0368	0.0821	0.1238	0.1617	0.1976	0.2308	
CP 6	0.0773	0.1797	0.2853	0.3891	0.4916	0.5951	

CT	6	0.0227	0.0720	0.1170	0.1578	0.1958	0.2312	
CP	7	0.0675	0.1531	0.2433	0.3341	0.4231	0.5103	0.5980
CT	7	0.0130	0.0532	0.0905	0.1254	0.1576	0.1878	0.2167
CP	7	0.0968	0.1794	0.2649	0.3510	0.4361	0.5188	0.6010
CT	7	0.0223	0.0576	0.0904	0.1213	0.1503	0.1773	0.2031
CP	8	0.0759	0.1473	0.2230	0.3005	0.3784	0.4556	0.5312
		0.6046						
CT	8	0.0080	0.0387	0.0674	0.0947	0.1209	0.1457	0.1689
		0.1909						
CP	7	0.0996	0.1810	0.2666	0.3537	0.4410	0.5272	0.6112
CT	7	0.0132	0.0451	0.0754	0.1039	0.1309	0.1565	0.1804

EOT

ORIGINAL PAGE IS OF POOR QUALITY

APPENDIX B

USER MANUAL

The first part of this manual describes the inputs required to run propeller performance calculations, while the later part presents the weight code inputs. It is assumed that the user is familiar with NNEP and its method of input. For instructions the reader is directed to "KONFIG/REKONFIG" (ref. 6) which is an interactive input preprocessing program. A sample input d... ... is given at the end of this Appendix.

The NNEP input string used to read in propeller performance data is the KONFIG statment. All propellers will be read in the following form:

KONFIG(1,N)='LOAD',SPEC(1,N)= (1),(2),....(10)

where N is the component number.

INPUT PROPELLER ARRAY

Spec(1,N) = Shaft horsepower (negative) delivered to the propeller

Spec(2,N) = One of three options:

1) constant efficiency - if a constant efficiency is specified, the value must be $0 < \eta < 1$
2) negative value - any input, negative in sign, will initiate the generalized propeller table interpolation (PERFM). Example: -.1, -100, etc. Input magnitude has no significance.
3) Table number - table number of user input propeller performance table. This value must be < 1.

Note: loads other than propellers may be connected to shafts in NNEP engine configurations; for such loads SPEC(2) must be set to 0.0.

Spec(3,N) = Propeller thrust divided by shaft horsepower (T/SHP) for static thrust calculations.[a]

Spec(4,N) = Design point power loading, PL = SHP/D^2.[b]

Spec(5,N) = Number of blades of the propeller.[c]

Spec(6,N) = Integrated design lift coefficient.[d]

Spec(7,N) = Blade activity factor.[c]

Spec(8,N) = A constant scaling factor which is multiplied by the results taken from the propeller map. This input scales the efficiency, thrust, and coefficient of thrust when using option (2) or option (3) of Spec(2,N). Spec(8,N) will override Spec(9,N) if both are input non-zero.

Spec(9,N) = Desired propeller design point efficiency. This input also determines a scale factor which will be applied to all subsequent off-design cases.
Spec(8,N) will override Spec(9,N) if both are input non-zero.

Spec(10,N) =Design tip speed of propeller (ft/sec).[d]

---

[a]This input is only used for option(1) of Spec(2,N)

[b]This input used to get prop diameter and rpm which are required for options (2) and (3) of Spec(2,N) and for propeller weight calculations.

[c]This input only used for option (2) of spec(2,N) and for propeller weight calculations.

[d]This input only used for option (2) of spec(2,N).

OUTPUT PROPELLER ARRAY

Datout(1,N) = Shaft horsepower delivered to the propeller.

Datout(2,N) = Propeller rotational speed (rpm).

Datout(3,N) = Propeller thrust, (lbf).

Datout(4,N) = Advance ratio, J.

Datout(5,N) = Coefficient of power, Cp.

Datout(6,N) = Coefficient of thrust, Ct.

Datout(7,N) = Propeller tip speed, (ft/sec).

Datout(8,N) = Unadjusted propeller efficiency.

Datout(9,N) = Adjusted propeller efficiency, (using SPEC(8,N) or SPEC(9,N)).

MECHANICAL DESIGN INDICATORS

The mechanical design indicators (IWMEC) are required to determine whether the propeller and gearbox are to be weighed. IWMEC is a two dimensional integer array, and is of the form IWMEC(M,N) where N is the component number of the prop as used in NNEP, and M is the variable identifier as defined below. For a propeller:

IWMEC ARRAY LOCATION M	DESCRIPTION
1	Type of component being weighted. 'PROP' -- indicates propeller.
2	Indicates whether propeller weight is to be calculated. 0 -- Do not calculate propeller weight. 1 -- Calculate propeller weight.
3	Indicates presence of counterweights on propeller. 0 -- No counterweights (double acting prop). 1 -- Prop contains counterweights.
4	Indicates whether gearbox weight is to be calculated. 0 -- Do not calculate gearbox weight. 1 -- Calculate gearbox weight.
5	Indicates whether gearbox dimensions are to be calculated. 0 -- Do not calculate gearbox dimensions. 1 -- Calculate gearbox dimensions.
6	Blank
7	Blank

Notes on IWMEC Inputs

If IWMEC(2,N) is input as 0, a message is printed informing the user that propeller weight will not be calculated. If input is 1, the exponents of the propeller weight equation are alterable. See DESVAL array discussion.

If IWMEC(4,N) is input as 0, a message is printed informing the user that gearbox weight will not be calculated.

If IWMEC(5,N) is input as 0, a message is printed informing the user that gearbox dimensions will not be calculated. It is assumed that the propeller adds no length to the engine beyond that of the gearbox.

ILENG Array

The ILENG array indicates which components will be included in the length summation. The propeller component number should be included to add the length of the gearbox to the engine.

DESIGN VALUES

This section describes mechanical and aerodynamic design data necessary to determine the weight and dimensions of the propeller. The data required is in the floating-point two-dimensional array DESVAL(M,N), where M is the variable array location identifier. N is the component number of the prop as used in NNEP.

The following equations are used to calculate propeller weight and gearbox weight, length, and diameter:

$$Wt_{PROP} = DESVAL(2) \left[\left(\frac{D}{10}\right)^{DESVAL(10)} \left(\frac{B}{4}\right)^{DESVAL(11)} \left(\frac{AF}{100}\right)^{DESVAL(12)} \left(\frac{ND}{20000}\right)^{DESVAL(13)} (DESVAL(1)+1)^{DESVAL(14)} \left(\frac{SHP}{10D^2}\right)^{DESVAL(15)} \right] + Cw \qquad (1B)$$

$$Cw = 5 \left[\left(\frac{D}{10}\right)^2 B \left(\frac{AF}{100}\right)^2 \left(\frac{20000}{ND}\right)^{0.3} \right] \qquad (2B)$$

or if DESVAL(7) > 0.0

$$Cw = DESVAL(7)$$

$$Wt_{GB} = DESVAL(4) * \left[DESVAL(5) * TORQUE + DESVAL(6) \right] \left(\frac{0.118}{GR}\right)^{DESVAL(17)} \qquad (3B)$$

$$LENGTH_{GB} = DESVAL(8) \quad (SHP)^{0.5} \ (1/GR)^{0.33} \qquad (4B)$$

$$DIAMETER_{GB} = DESVAL(9) \quad (SHP)^{0.5} (1/GR)^{0.33} \qquad (5B)$$

ORIGINAL PAGE IS
OF POOR QUALITY

The DESVAL variables used in the above equations are described as follows:

DESVAL ARRAY
LOCATION   DESCRIPTION

(1) Design Mach number of aircraft (see equation 1B). Default input of zero yeilds zero Mach number input.

(2) Kw - Coefficient of propeller weight equation (see equation (1B)). A multiplier representing the level of propeller technology, described in the body of this report. If Kw is input as non-zero, the exponents of the weight equation (DESVAL(10,N)-(15,N)) must be input. If Kw is defaulted (zero), default values are used.

(3) Gear ratio (prop rpm/shaft rpm); see equations 3B,4B, and 5B. OR Shaft RPM. The user can choose whether to specify a gear ratio (usually < 1) or an RPM of the shaft delivering power to the prop. This RPM is independent of the RPM calculated by NNEP. If DESVAL(3,N) > 200. The input is assumed to be RPM and the corresponding gear ratio is calculated and stored in DESVAL(3,N). If DESVAL(3,N) < 200. The input is assumed to be the gear ratio.

(4) Coefficient of gearbox weight equation (see equation 3B). No scaling will occur if 0.0 or 1.0 is input.

(5-6) Slope and Y-Intercept, respectively, of optional user input gearbox weight curve (see equation 3B). If slope is input a Y-Intercept must also be input. Any input curve must be scaled at 0.118 gear ratio. To use default curve described in the body of the report input 0.0 for both values.

(7) Input value of counterweight weight (see equation 2B). The default calculation value may be used by inputting DESVAL(7,N)=0.0.

(8) Scale factor for gearbox length (see equation 4B). No scaling will occur if 0.0 or 1.0 is input.

(9) Scale factor for gearbox diameter (see equation 5B). No scaling will occur if 0.0 or 1.0 is input.

(10-15) Exponents of propeller weight equation (see equation 1B). The default values are used if DESVAL(2,N) = 0.0. If DESVAL(2,N) = 0, all of the exponent values must be input.

(16) Not used.

(17) Exponent on gear ratio term of gearbox weight equation (see equation 3B).

DESVAL DEFAULT VALUES

DESVAL DEFAULT ARRAY LOCATION	VALUE	ABBREVIATED DESCRIPTION
1	0.0	Design Mach number
2	355.0 (220.0)[a]	Kw--prop weight scalar
3	1.0	Gear ratic
4	1.0	Gearbox weight scalar
5	0.0174	Gearbox weight curve slope
6	45.0	Gearbox weight curve Y-Intercept
7	0.0 (equ. 2B)[a]	Weight of counterweights
8	1.0	Gearbox length scalar
9	1.0	Gearbox diameter scalar
10	2.0	Prop weight equation exponents
11	0.7	
12	0.75 (0.70)[a]	
13	0.5 (0.4)[a]	
14	0.5	
15	0.12	
16	---	Not used
17	0.5	Exponent on gear ratio term in equation 3B.

[a]Defaults for propellers with counterweights.

SAMPLE INPUT CASE

ORIGINAL PAGE IS OF POOR QUALITY

```
   THIS IS A TITLE CARD
&D CALBLD=T,AMAC=F &END
&D MODE=1,IWT=1
KONFIG(1,1)=4HINLT,1,0,2,0,SPEC(1,1)=14.3,0,0,0,.23,0,0,0,5000,
KONFIG(1,2)=4HCOMP,2,0,3,0,SPEC(1,2)=1.278,0,1,3707,1,3708,.982,3709,1,0,0,.87,
                                       5.896,.919,
KONFIG(1,3)=4HCOMP,3,0,4,20,SPEC(1,3)=1.803,.05,1,3404,1,3405,.78,3406,1,0,0,
                                       .864,3.887,0.988,
KONFIG(1,4)=4HDUCT,4,0,5,0,SPEC(1,4)=.05,0,0,2710,.985,18300,
KONFIG(1,5)=4HTURB,5,20,6,0,SPEC(1,5)=3.5,.95,1,3801,1,3802,.8881,1,.85,1,.91,
                                       5665,1,08,1,
KONFIG(1,6)=4HTURB,6,20,7,0,SPEC(1,6)=2.49,.05,1,3803,1,3804,.9154,1,1,1,.9156,
                                       5124,1,000000,3,
KONFIG(1,7)=4HNOZZ,7,0,8,0,SPEC(1,7)=30,1,0,0,.95,3*0,1,
KONFIG(1,8)=4HLOAD,SPEC(1,8)=-2686,.985,3.5,9.6,4,.5,180,1.0115,0.00,900.,
KONFIG(1,9)=4HSHFT,2,3,5,0,SPEC(1,9)=38834,4*1,2*1,.99,
KONFIG(1,10)=4HSHFT,8,11,6,0,SPEC(1,10)=18000,.0611,1,1,1,.97,1,1
KONFIG(1,15)=4HCNTL,SPCNTL(1,15)=1,1,4HSTAP,8,2,0
KONFIG(1,11)=4HLOAD,SPEC(1,11)=-68,
KONFIG(1,16)=4HCNTL,SPCNTL(1,16)=1,2,4HSTAP,8,3,0
KONFIG(1,17)=4HCNTL,SPCNTL(1,17)=1,3,4HSTAP,8,5,0
KONFIG(1,18)=4HCNTL,SPCNTL(1,18)=1,5,4HSTAP,8,6,0
KONFIG(1,19)=4HCNTL,SPCNTL(1,19)=1,6,4HSTAP,8,7,0
KONFIG(1,20)=4HCNTL,SPCNTL(1,20)=1,9,4HDOUT,8,9,0
KONFIG(1,21)=4HCNTL,SPCNTL(1,21)=1,8,4HDOUT,8,10,0
KONFIG(1,22)=4HCNTL,SPCNTL(1,22)=2,3,4HPERF,15,0,0,1
KONFIG(1,23)=4HCNTL,SPCNTL(1,23)=10,2,4HDOUT,5,2,15,0,-5,10,
&END
&D CALBLD=F,SPEC(9,22)=0,SPEC(9,15)=1,SPEC(9,16)=1,SPEC(9,17)=1,SPEC(9,18)=1,
IWT=1,SPEC(5,1)=0,SPEC(9,1)=0,SPEC(4,4)=2760,SPEC(9,19)=1,SPEC(9,20)=1,
SPEC(9,21)=1, &END
&D SPEC(5,1)=0.0,IWT=1, &END
&D SPEC(5,1)=.1,IWT=2, &END
&W ISII=F,ISIO=F,IOUTCD=2,DISKWI=1
ILENG=8,2,3,4,5,6,7,
IWMEC(1,2)='LPC',1,1,0,0,0,3
IWMEC(1,3)='HPC',2,0,1,0,0,1
IWMEC(1,4)='PBUR',1
IWMEC(1,5)='HPT',1,2,0,1,0
IWMEC(1,6)='LPT',1,2,0,3,0
IWMEC(1,7)='NOZZ',1,0,0
IWMEC(1,8)='PROP',1,0,1,1
IWMEC(1,9)='SHFT',2,5,0,0,2
IWMEC(1,10)='SHFT',1,6,0,0,8
DESVAL(1,2)=.55,1.9,0.5,1.5,4.0,3.0,0.45,0.0,0.0,1.0,0.0,1.0,1.0,4*0.0,
DESVAL(1,3)=0.4,1.4,0.7,1.5,3.0,1.5,0.3,0.0,0.0,1.0,0.0,1.0,1.0,4*0.0,
DESVAL(1,4)=150.,0.015,13*0.0
```

```
DESVAL(1,5)=0.3,0.28,1.5,1.5,1.5,0.45,125000.,3.0,1.0,6*0.0
DESVAL(1,6)=0.45,0.28,1.5,2.0,4.0,0.55,125000.,3.0,1.0,6*0.0
DESVAL(1,7)=1.3,14*0.0
DESVAL(1,8)=.23,150,.1,4*0.0,1.0,1.0,2.0,0.7,0.75,0.5,0.5,0.12,0.,0.
DESVAL(1,9)=50000.,0.286,0.85,2,3,5,11*0.0
DESVAL(1,10)=50000.,0.286,0.85,8,6,12*0.0
&END
```

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX C

ORIGINAL PAGE IS
OF POOR QUALITY

PROGRAM LISTING

```
C***********************************************************************************
C***********************************************************************************
C
C     SUBROUTINE NNPROP
C     ---------- ------
C     PURPOSE:            THIS ROUTINE CALCULATES PROPELLER PERFORMANCE
C                         USING ONE OF THREE OPTIONS
C
C     DATE OPERATIONAL:   08/13/82
C
C     AUTHOR:             R. M. PLENCNER
C
C     DESCRIPTION:        SUBPROGRAM CALCUALTES PROPELLER THRUST FROM
C                         ONE OF THREE OPTIONS:
C                         1) INPUT CONSTANT EFFICIENCY
C                         2) GENERALIZED HAMILTON STANDARD PROPELLER MAP
C                         3) USER SUPPLIED PROPELLER MAP
C
C***********************************************************************************
C***********************************************************************************
      SUBROUTINE NNPROP (VKTS,ALT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ZMS(2)
      COMMON /DBL/ DATINP(15,60),DATOUT(9,60),WTF(40),TOPRES(40),TOTEMP(-
     140),FAR(40),CORFLO(40),VMACH(40),STATP(40),ERROR(40),TOL,TOLT,TOLT-
     2T,DEPV(20),DTOL(20),PERPF(20),RCH,STOC,TFUEL
      COMMON /SNGL/ JM1,JM2,JP1,JP2,JCX,LOCTBL(9,60),JCOMP(70),IWAY,NIT,-
     1ITAB(70),JCONF(60,4),JTYPE(60),JFLOW(70),IDEDAP(15),KKINDS(14,25),-
     2NCOMP,NOSTAT,NITER,NFINIS,NPASS,JCC,NTBL,NCTS,JCIND(20),JCDEP(20),-
     3JCVIND(20),JCVDEP(20),KDTYP(20),IDONE(60)
      N11=KKINDS(11,1)
      DO 2 IK=1,N11
      JXSHFT=KKINDS(11,IK+1)
      DO 2 IISHFT=1,4
      IF(JCX.EQ.JCONF(JXSHFT,IISHFT))GO TO 5
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  2   CONTINUE
  5   DATINP(15,JCX) = JXSHFT
      PTRPM=DATOUT(2,JXSHFT)
      SHP = -DATOUT(1,JCX)
      PL = DATINP(4,JCX)
      BLADT = DATINP(5,JCX)
      CLI = DATINP(6,JCX)
      ATF = DATINP(7,JCX)
      XFT = DATINP(8,JCX)
      ETADES = DATINP(9,JCX)
      TSDES = DATINP(10,JCX)
      LTAB=0
C --- THIS IS TO IMPLEMENT THE PERFM SUBROUTINE W/ GENERAL MAPS---
      IF(DATINP(2,JCX).LT.0.0)LTAB=-777777.
C --- IF TABLE NO. IS SPECIFIED,SKIP NEXT 4 LINES---
      IF(DATINP(2,JCX).LE.0.0.OR.DATINP(2,JCX).GT.1.0)GO TO 633
      ETA1=DATINP(2,JCX)
      IF(VKTS.EQ.0.0)ETA1=0.0
C --- SET ETA1 EQUAL TO SPECIFIED ETA,CALCULATE THR FOR STATIC CASE ---
      IF(VKTS.EQ.0.0)THR=DATINP(3,JCX)*SHP
  633 IF(LOCTBL(2,JCX).GT.0) LTAB=LOCTBL(2,JCX)
C --- GO TO 10 TO IMPLEMENT SUB. PERFM ---
      IF(LTAB.NE.0.0)GO TO 10
C --- IF DATINP(9) DOES NOT AGREE W/ ETA1,WRITE AN ERROR MESSAGE ---
      IF(NIT.LE.1.AND.ETADES.GT.0.0.AND.ETA1.NE.ETADES)WRITE(10,300)JCX
C --- CALCULATE THRUST FOR NON-STATIC CASE ---
      IF(VKTS.GT.0.0)       THR=550*SHP*ETA1/(VKTS*1.6878)
      THR1=THR
      IF(VKTS.GT.0.0)ETA=ETA1
      IF(PL.EQ.0.0.OR.TSDES.EQ.0.0)GO TO 101
C --- CLACULATIONS BELOW WILL NOT BE REQUIRED FOR SPECIFIED
C     ETA OR TABLE LOOK-UP CASES ----
 10   CALL ICAO (ALT,TP,PRESS,SSFPS,DNSTY)
      DRAT = .002377/DNSTY
      IF (IWAY.EQ.1) DIA = DSQRT(SHP/PL)
      IF (IWAY.EQ.1) TS=TSDES
      IF (IWAY.EQ.1) DESRPM=60.0*TS/(3.14159*DIA)
      IF (IWAY.EQ.1) RPMRAT=DESRPM/PTRPM
      RPM=RPMRAT*PTRPM
      TS = 3.14159*RPM*DIA/60.
      IW = 1
      CP = SHP*10.E10 * DRAT/(2. * TS**3  *  DIA**2 * 6969.26)
      ZJI = 101.27 * VKTS/(RPM * DIA)
      ZMS(1) = VKTS* 1.6878/SSFPS
 20   ZMS(2) = TS/SSFPS
      ZZ=0.0
      IF(LTAB.LT.0.0) CALL PERFM (IW,CP,ZJI,AFT,BLADT,CLI,CT1,ZMS,LIMIT)
      IF(LTAB.GT.0.0) CALL TLOOK (LTAB,CP,ZJI,ZZ,CT1)
      IF(LTAB.NE.0.0)THR1 = CT1* TS**2 * DIA**2 * 2.4084E-4/DRAT
      IF(LTAB.NE.0.0)ETA1 = ZJI*CT1/CP
      IF(NIT.LE.1.AND.ETADES.NE.0.0.AND.ETA1.EQ.0.0) WRITE(10,100)JCX
      IF(NIT.LE.1.AND.ETADES.NE.0.0.AND.XFT.NE.0.0.AND.XFT.NE.1.0)-
     1WRITE(10,200)JCX
      IF(IWAY.NE.1)GO TO 40
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      ETAFIX=ETADES
      IF(ETADES.EQ.0.0)ETAFIX=ETA1
      ETARAT=1.0
      IF(ETA1.NE.0.0)ETARAT=ETAFIX/ETA1
      IF(XFT.NE.1.0.AND.XFT.NE.0.0)ETARAT=XFT
   40 ETA=ETARAT*ETA1
      THR = THR1*ETARAT
      CT = THR*DRAT/(TS**2 * DIA**2 * 2.4084E-4)
  101 DATINP(IISHFT+1,JXSHFT)=RPM/DATOUT(2,JXSHFT)
      DATOUT(IISHFT+2,JXSHFT)=RPM
      DATINP(3,JCX) = THR/SHP
      DATOUT(2,JCX) = RPM
      DATOUT(3,JCX) = THR
      DATOUT(4,JCX) = ZJI
      DATOUT(5,JCX) = CP
      DATOUT(6,JCX) = CT
      DATOUT(7,JCX) = TS
      DATOUT(8,JCX) = ETA1
      DATOUT(9,JCX) = ETA
  100 FORMAT(' ','*** WARNING ***  FOR  LOAD#',I2,'   DATINP(9) IGNORED-
     1 BECAUSE VELOCITY=0 FOR DESIGN CASE')
  200 FORMAT(' ',10X,'*** WARNING ***'/' BOTH DATINP(8) AND (9) HAVE-
     1 BEEN SPECIFIED FOR LOAD #',I2,/' DATINP(8) WILL OVERRIDE DATINP(9)')
  300 FORMAT(' ','*** WARNING ***  FOR  LOAD#',I2,-
     1'    DATINP(9) NOT = DATINP(2) --| DATINP(2) OVERRIDES ')
      RETURN
      END
C
C*******************************************************************************

****
C
C     SUBROUTINE PROPWT
C     ---------- ------
C     PURPOSE:             THIS ROUTINE CALCULATES THE WEIGHT OF A PROPELLER
C                          AS WELL AS WEIGHT OF AN APPROPRIATE GEARBOX.
C
C     DATE OPERATIONAL:    08/13/82
C
C     AUTHOR:              PETER SENTY,NASA SUMMER EMPLOYEE 1982
C
C     DESCRIPTION:         SUBPROGRAM UTILIZES GEOMETRICAL DATA FROM
C                          BOEING "WATE-2" PROGRAM ASWELL AS THERMODYNAMIC
C                          DATA FROM THE NAVY/NASA ENGINE PROGRAM TO PREDICT
C                          THE WEIGHT OF A PROPELLER,WEIGHT OF A COMPATIBLE
C                          GEARBOX AND THE DIMENSIONS OF THAT GEARBOX.
C
C*******************************************************************************

********************************
C*******************************************************************************

********************************
      SUBROUTINE PROPWT(NC)
      REAL*8 DATINP,DATOUT,WTF,TOPRES,TOTEMP,FAR,CORFLO,VMACH,STATP,ERRO-
```

```
      1R,TOL,TOLT,TOLTT,DEPV,DTOL,PERPF,RPMNT,TMTEMP,TMPRES,DATOUM,DATMAC-
      2,DATALT,DESLIM,TNPRES,TNTEMP,CNRFLO,CORFLM,WNTF,DATOUN,FARN,DANINP-
      3,DEBUG,DEPQ,SELAST,DD,TOLOPT
C                *****************
C                * COMMON BLOCKS *
C                *****************
       COMMON /DBL/ DATINP(15,60),DATOUT(9,60),WTF(40),TOPRES(40),TCTEMP(-
      140),FAR(40),CORFLO(40),VMACH(40),STATP(40),ERROR(40),TOL,TOLT,TOLT-
      2T,DEPV(20),DTOL(20),PERPF(20)
       COMMON /SNGL/ JM1,JM2,JP1,JP2,JCX,LOCTBL(9,60),JCOMP(70),IWAY,NIT,-
      1ITAB(70),JCONF(60,4),JTYPE(60),JFLOW(70),IDEDAP(15),KKINDS(14,25),-
      2NCOMP,NOSTAT,NITER,NFINIS,NPASS,JCC,NTBL,NCTS,JCIND(20),JCDEP(20),-
      3JCVIND(20),JCVDEP(20),KDTYP(20),IDONE(60)
       COMMON /DEFAUL/ DEFAUL(15,20),ISCALE(3),SCALE(6)
       COMMON /WMECH/ IWMEC(7,60),WATE(60),ALENG(60),TLENG(40),RI(2,40),R-
      1O(2,40),DESVAL(17,60),DSHAF(5),RPMT(60),IWT,IPLT,IERR,ISII,ISIO,IO-
      2UTCD,NSTAG(60)
       COMMON /CONVER/ CONVER(15)
       COMMON /NEPOPT/ DEBUG,DEPQ,SELAST,DD,TOLOPT,NDSET,NPARTS,IOPTP,NPA-
      1SSO,NVOPT,NJOPT,NOPT
       COMMON /TERMON/ TNPRES(40),TNTEMP(40),CNRFLO(40),WNTF(40),RPMNT(40-
      1),DATOUN(9,60),FARN(40),DANINP(15,60),TMTEMP(40),TMPRES(40),CORFLM-
      2(40),DATOUM(9,60),DATMAC(4,60),DATALT(4,60),DESLIM(15)
       COMMON /CENTER/ CGARM(60)
       COMMON /ARM/ ACCARM
       COMMON /DISKK/ DISKWI,ENGINE
       COMMON /SKIP/ SKIPIT
C
C                ****************************
C                * DATA STORAGE DEFINITION *
C                ****************************
       LOGICAL PINP,IPLT,ISIO,ISII,PLOT,SKIPIT
       INTEGER IDID(60),ILENG(40)
       DIMENSION NUMNUM(17), IRNAME(17), CORFLC(40)
       NAMELIST / W/IWMEC,DESVAL,ACCS,IWT,IPLT,ISII,ISIO,IOUTCD,ILENG,DES-
      1LIM,ISCALE,SCALE,ACCARM,DISKWI,ENGINE,PLOT
       DIMENSION X(17), DEFALT(17)
C                *******************
C                * DATA STATEMENTS *
C                *******************
       DATA IDUC,LSHAF,ENGU,SIU,IVALV/4HDUCT,4HSHAF,4HENGL,4HSIU ,4HVALV/
       DATA ILPC,IHPC,IFAN,IFO,IFI,IHPT,ILPT/3HLPC,3HHPC,3HFAN,2HFO,2HFI,-
      13HHPT,3HLPT/
       DATA IPROP/4HPROP/,DEFALT/0.0,355.,1.0,1.0,0.0174,45.0,0.0,1.0,-
      11.0,2.0,0.7,0.75,0.5,0.5,0.12,220.,0.5/
C ---- TEST CALL TO PROPWT --
       IF(IWMEC(1,NC).NE.IPROP) RETURN
C --- INITIALIZE "X" ARRAY---
       DO 5 I=1,17
       X(I)=0.0
5      CONTINUE
C--- FILL X ARRAY WITH INPUTS OR DEFAULT VALS.---
       DO 10 I=2,4
       IF(DESVAL(I,NC).NE.0.0) X(I)=DESVAL(I,NC)
       IF(DESVAL(I,NC).EQ.0.0) X(I)=DEFALT(I)
10     CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C ******* BEGIN CALC. OF GEAR BOX WT. & DIMENSIONS *****
C --- ATTAIN MAX. VALUE PARAMETERS ---
      SHP=DATOUM(1,NC)
      RPMAX=DATOUM(2,NC)
C --- LOCATE SHAFT ---
      JSHP=DATINP(15,NC)+.01
C --- CHECK FOR GEAR RATIO OR RPM INPUT ---
      IF(DESVAL(3,NC).NE.0.0.AND.DESVAL(3,NC).LT.200.) SGR=DESVAL(3,NC)
      IF(DESVAL(3,NC).NE.0.0.AND.DESVAL(3,NC).GT.200.)-
     1 SGR=RPMAX/DESVAL(3,NC)
      IF(DESVAL(3,NC).NE.0.0)GO TO 35
      RSMAX=RPMAX/SGR
      DO 500 IX=1,4
      IF(NC.EQ.JCONF(JSHP,IX))SGR=DATOUT(2,JSHP)
      WRITE(10,110)SGR,RPMAX,SHP
500   CONTINUE
35    CONTINUE
C --- SEND RPM BACK TO SUB."TURBINE" FOR USE IN DIMENSION CALCS ---
      JSHP=DATOUT(15,NC)
      RPMT(NC)=RSMAX
C--- CALCULATE TORQUE ---
      TORK=SHP*5252./RPMAX
C --- CHECK OPTION TO WEIGH G.B. ---
      IF(IWMEC(4,NC).EQ.0) WRITE(10,210)
      IF(IWMEC(4,NC).EQ.0) GO TO 100
C --- INITIALIZE VALUES TO BE PRINTED OUT ---
      GBWT=0.0
      GBKW=DEFALT(4)
      IF(DESVAL(4,NC).NE.0.0) GBKW=DESVAL(4,NC)
C
C --- CHECK OPTION FOR G.B. WT.CURVE ---
      X(5)=DESVAL(5,NC)
      X(6)=DESVAL(6,NC)
      IF(DESVAL(5,NC).EQ.0.0.AND.DESVAL(6,NC).EQ.0.0) X(5)=DEFALT(5)
      IF(DESVAL(5,NC).EQ.0.0.AND.DESVAL(6,NC).EQ.0.0) X(6)=DEFALT(6)
      GBM=X(5)
      GBB=X(6)
C --- CHECK FOR INPUT GEAR RATIO SCALAR EXPONENT ---
      IF(DESVAL(17,NC).NE.0.0) X(17)=DESVAL(17,NC)
      IF(DESVAL(17,NC).EQ.0.0) X(17)=DEFALT(17)
C--- INTERPOLATE CURVE FOR GEARBOX WEIGHT ---
      GBWT=GBKW*(GBM*TORK+C3B)*((1./SGR)/8.5)**X(17)
100   CONTINUE
C --- CHECK FOR INPUT SCALE FACTORS FOR G.B. DIMENSIONS ---
      GBLN=0.0
      GBDI=0.0
      IF(IWMEC(5,NC).EQ.0) WRITE(10,220)
      IF(IWMEC(5,NC).EQ.0) GO TO 101
      X(8)=DEFALT(8)
      X(9)=DEFALT(9)
      IF(IWMEC(5,NC).NE.0) X(8)=DESVAL(8,NC)
      IF(IWMEC(5,NC).NE.0) X(9)=DESVAL(9,NC)
      GBKL=X(8)
      GBKD=X(9)
```

```
C ---CALCULATE G.B. DIMENSIONS ---
      GBLN=0.005*46.*(SHP**.5)*((1/SGR)**.33)*GBKL
      GBDI=0.005*25.*(SHP**.5)*((1/SGR)**.33)*GBKD
      ILENG(NC)=GBLN
C -- ABOVE CALCS. BASED ON 1980 DDA "STAT" REPORT BASELINE ENG ---
101   CONTINUE
      WRITE(10.120)TORK,GBWT,GBLN,GBDI
C
C
C --- TEST FOR CHOICE TO WEIGH PROP ---
      IF(IWMEC(2,NC).EQ.0) WRITE(10,230)
      IF(IWMEC(2,NC).EQ.0) GO TO 300
C
C ******** BEGIN CALCS OF PROP WT. *******************
C --- CHECK FOR INPUT VALUES OF PROP WT. EQ. EXPONENTS---
      DO 20 I=10,15
      X(I)=DEFALT(I)
      IF(IWMEC(2,NC).NE.0.AND.DESVAL(2,NC).NE.0.0) X(I)=DESVAL(I,NC)
20    CONTINUE
C --- DEFINE COEFFICIENTS OF PROP WT EQ ---
      D=SQRT(DATOUM(1,NC)/DATOUM(3,NC))
      B=DATINP(5,NC)
      AF=DATINP(7,NC)
      PN=DATOUM(2,NC)
      SHP=DATOUM(1,NC)
      PM=DESVAL(1,NC)
      WRITE(10,130)D,B,AF,PN
      IF(DESVAL(2,NC).EQ.0.0) PKW=DEFALT(2)
      IF(DESVAL(2,NC).NE.0.0) PKW=DESVAL(2,NC)
C --- CHECK FOR COUNTERWEIGHTS ---
      IF(IWMEC(3,NC).NE.0) GO TO 200
C
C
C *** BEGIN CALC OF PROP WT. (NO COUNTERWEIGHTS) *****
C
C---- NOTE:PROP WT.EQ. IS BROKEN INTO THREE PARTS FOR CLARITY ----
      PART1=(D/10.)**X(10)*(B/4.0)**X(11)*(AF/100.)**X(12)
      PART2=((PN*D)/20000.)**X(13)*(PM+1.0)**X(14)
      PART3=(SHP/(10.*D**2))**X(15)
      PRPWT=PKW*(PART1*PART2*PART3)
C --- COMBINE PROP WT. W/ G.B. WT. ---
      WATE(NC)=PRPWT+GBWT
      WRITE(10,140)PRPWT,WATE(NC)
C--- SKIP OVER NEXT PROP WT. CALC ---
      GO TO 300
200   CONTINUE
C
C **** BEGIN PROP WT. CALC FOR SINGLE ACTING PROP (W/COUNTERWEIGHTS)
C
      CW=0.0
C --- SET EXPONENT VALUES FOR COUNTERWEIGHT WT. EQ. ---
      X(12)=0.7
      X(13)=0.4
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C --- CHECK FOR INPUT WT. EQ. SCALAR("KW" VALUE) ---
C---- NOTE:PROP WT.EQ. IS BROKEN INTO THREE PARTS FOR CLARITY ----
      PART1=(D/10.)**X(10)*(B/4.0)**X(11)*(AF/100.)**X(12)
      PART2=((PN*D)/20000.)**X(13)*(PM+1.0)**X(14)
      PART3=(SHP/(10.*D**2))**X(15)
      PRPWTC=PKW*(PART1*PART2*PART3)
C --- CHECK FOR INPUT OF CONST. VALUE CW WT. ---
      IF(DESVAL(7,NC).NE.0.0) X(7)=DESVAL(7,NC)
      IF(DESVAL(7,NC).EQ.0.0) -
     1 X(7)=5*((D/10.)**2*B*(AF/100.)**2*(20000./(PN*D))**.3)
      CWT=X(7)
C --- SUM PROP + CW WTS. ---
      PRPWT=PRPWTC+CWT
      WATE(NC)=PRPWT+GBWT
      WRITE(10,150)PRPWT,CWT,WATE(NC)
C
300   CONTINUE
C
C
110   FORMAT('0',2X,'SHAFT-PROP GEAR RATIO=',F10.5,5X,-
     1'MAX RPM=',F15.4,5X,'SHAFT H.P.=',F15.4)
120   FORMAT('0',2X,'TORQUE=',F15.4,11X,'GEAR BOX WT.=',F15.4/2X,
     1'GEAR BOX LENGTH=',F15.4,5X,' GEAR BOX DIA.=',F15.4)
130   FORMAT('0',2X,'PROP DIA.=',F15.4,9X,'NUMBER BLADES=',-
     1F15.4/'ACTIVITY FACTOR=',F15.4,6X,'PROP RPM=',F15.4)
140   FORMAT('0',2X,' PROP WT.=',F15.4,9X,'PROP + GB WT.=',F15.4)
150   FORMAT('0',2X,' PROP WT.=',F15.4,5X,'COUNTERWEIGHT WT.=',-
     1F15.4,2X,'PROP + CW WT.=',F15.4)
210   FORMAT(3X,'*** GEARBOX WEIGHT WILL NOT BE CALCULATED ***')
220   FORMAT(3X,'*** GEARBOX DIMENSIONS WILL NOT BE CALCULATED ***')
230   FORMAT(3X,'*** PROPELLER WEIGHT WILL NOT BE CALCULATED ***')
C
      IDID(NC)=1.0
C
      RETURN
      END


C     SUBROUTINE WTEST
C     ----------------
C
C     PURPOSE
C     -------
C     TO CONTROL THE CALLING OF SUBROUTINES WHICH WILL ESTIMATE THE
C     WEIGHT AND LENGTH OF INDIVIDUAL COMPONENTS
C
C     DESCRIPTION
C     -----------
C     THE OVERALL LENGTH OF THE ENGINE IS CALCULATED BY PROCESSING THE
C     ILENG ARRAY.ALL COMPONENTS EXCEPT DUCTS AND SHAFTS, THEN DUCTS.
C     THE REMAINING COMPONENTS EXCEPT DUCTS AND SHAFTS ARE PROCESSED.
C     THE DUCTS ARE PROCESSED AND FINIALY THE SHAFTS.
C     A BUILT-IN ASSUMPTION IN THE DUCT ROUTINE IS THAT NO DUCT IS
```

```
C     CONNECTED TO ANOTHER DUCT I.E. THE DUCT SIZE IS DETERMINED BY THE
C     ADJOINING COMPONENTS.
C     THEN THE MAXIMUM RADIUS IS FOUND. THEN DEPENDING ON THE PRINT
C     FLAG -IOUTCD- THE REQUIRED PRINTING IS DONE
C     IF THE PLOT CODE FLAG -IPLT- IS TRUE ROUTINE EPLT IS CALLED
C     USAGE
C     -----
C     CALL WTEST
C
C     CALLING ROUTINES
C     ----------------
C     FLOCAL-
C     ZTOPZ -
C
C     REQUIRED SUBROUTINES
C     --------------------
C     COMP   -COMPRESSOR WEIGHT/LENGTH
C     TURB   -TURBINE
C     SHAFT  -SHAFT
C     DUCTW  -DUCT
C     COMBWT -PRIMARY BURNER WEIGHT/LENGTH
C     WTNOZ  -NOZZLE     WEIGHT/LENGTH
C     WMIXR  -MIXER
C     WSPLT  -SPLITTER
C     EPLT   -PRINTER/PLOTTER
C
C     MODIFICATION HISTORY
C     --------------------
C
C       DATE     ID      ANALYST              DESCRIPTION
C     -------- ------ -----------  -----------------------------------
C     MO/DA/YR IDENT  NAME          DESCRIPTION OF CHANGES
C     9/21/81         ROBERT CORBAN  INCORPORATE INTERACTIVE MODE AND GR
C     HICS OPTION
C     8/21/82         PETE SENTY     INCORPORATE PROPELLER WEIGHT MODEL
C
C     AUTHOR/LANGUAGE/DATE
C     --------------------
C     NORMAN PREWITT-BOEING COMPUTER SERV.  /FORTRAN IV    / OCT 10,1976
C
C     GLOSSARY
C     --------
C     NAME    ORIGIN USAGE              DESCRIPTION
C     ----    ------ -----  ------------------------------------------
C     IWMEC   /WMECH/ I     CONTROL INFORMATION
C     WATE    /WMECH/ O     WEIGHT OF EACH COMPONENT
C     ALENG   /WMECH/ O     ACTUAL LENGTH OF EACH COMPONENT
C     TLENG   /WMECH/ O     ACCUMULATED LENGTH TO END OF COMPONENT
C     RI      /WMECH/ O     RADIUS INNER INLET,OUTLET EACH STATION
C     RO      /WMECH/ O     RADIUS OUTER INLET,OUTLET EACH STATION
C     DESVAL  /WMECH/ I     MECHANICAL DESIGN DATA OVERRIDES DEFAUL
C     DSHAF   /WMECH/ O     SHAFT DIAMETER INNER TO OUTER
C     RPMT    /WMECH/ I     ACTUAL COMPONENT RPM
C     IWT     /WMECH/ I     WEIGHT ESTIMATION FLAG TRU=  DO IT
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
C     IPLT     /WMECH/ I     PLOTTER FLAG TRUE= DO IT
C     PLOT             L     GRAPHICS PLOTTING FLAG  TRUE= DO IT
C     IERR     /WMECH/ O     ERROR FLAG
C     ISIO     /WMECH/ I     OUTPUT UNITS O=ENGLISH, O SI
C     ISII     /WMECH/ I     INPUT UNITS O=ENGLISH, O SI
C     IOUTCD   /WMECH/ I     PRINT FLAG O=SUMMARY, 1=GENERAL,2=DIAGNOSTIC
C     ILENG    /WMECH/ I     COMPONENTS CONTRIBUTING TO OVERALL LENGTH
C     IDID             O     FLAG = 0 COMPONENT NOT YET WEIGHED  =1 YES
C     SKIPIT           L     BACKGROUND FLAG YES=SKIP PROMPTS
C
C     ******************************************************************
C
      SUBROUTINE WTEST
      REAL*8 DATINP,DATOUT,WTF,TOPRES,TOTEMP,FAR,CORFLO,VMACH,STATP,ERRO-
     1R,TOL,TOLT,TOLTT,DEPV,DTOL,PERPF,RPMNT,TMTEMP,TMPRES,DATOUM,DATMAC-
     2,DATALT,DESLIM,TNPRES,TNTEMP,CNRFLO,CORFLM,WNTF,DATOUN,FARN,DANINP-
     3,DEBUG,DEPQ,SELAST,DD,TOLOPT,RCH,STOC,TFUEL
C             *****************
C             * COMMON BLOCKS *
C             *****************
      COMMON /DBL/ DATINP(15,60),DATOUT(9,60),WTF(40),TOPRES(40),TOTEMP(-
     140),FAR(40),CORFLO(40),VMACH(40),STATP(40),ERROR(40),TOL,TOLT,TOLT-
     2T,DEPV(20),DTOL(20),PERPF(20),RCH.STOC,TFUEL
      COMMON /SNGL/ JM1,JM2,JP1,JP2,JCX,LOCTBL(9,60),JCOMP(70),IWAY,NIT,-
     1ITAB(70),JCONF(60,4),JTYPE(60),JFLOW(70),IDEDAP(15),KKINDS(14,25),-
     2NCOMP,NOSTAT,NITER,NFINIS,NPASS,JCC,NFBL,NCTS,JCIND(20),JCDEP(20),-
     3JCVIND(20),JCVDEP(20),KDTYP(20),IDONE(60)
      COMMON /DEFAUL/ DEFAUL(15,20),ISCALE(3),SCALE(6)
      COMMON /WMECH/ IWMEC(7,60),WATE(60),ALENG(60),TLENG(40),RI(2,40),R-
     1O(2,40),DESVAL(17,60),DSHAF(5),RPMT(60),IWT,IPLT,IERR,ISII,ISIO,IO-
     2UTCD,NSTAG(60)
      COMMON /CONVER/ CONVER(15)
      COMMON /NEPOPT/ DEBUG,DEPQ,SELAST,DD,TOLOPT,NDSET,NPARTS,IOPTP,NPA-
     1SSO,NVOPT,NJOPT,NOPT
      COMMON /TERMON/ TNPRES(40),TNTEMP(40),CNRFLO(40),WNTF(40),RPMNT(40-
     1),DATOUN(9,60),FARN(40),DANINP(15,60),TMTEMP(40),TMPRES(40),CORFLM-
     2(40),DATOUM(9,60),DATMAC(4,60),DATALT(4,60),DESLIM(15)
      COMMON /CENTER/ CGARM(60)
      COMMON /ARM/ ACCARM
      COMMON /DISKK/ DISKWI,ENGINE
      COMMON /SKIP/ SKIPIT
      COMMON /WTLDAT/ WATENG,WATACC
C
C             ****************************
C             * DATA STORAGE DEFINITION *
C             ****************************
      LOGICAL PINP,IPLT,ISIO,ISII,PLOT,SKIPIT
      INTEGER IDID(60),ILENG(40)
      DIMENSION NUMNUM(17), IRNAME(17), CORFLC(40)
      NAMELIST /W/ IWMEC,DESVAL,ACCS,IWT,IPLT,ISII,ISIO,IOUTCD,ILENG,DES-
     1LIM,ISCALE,SCALE,ACCARM,DISKWI,ENGINE,PLOT,IDID
C             *******************
C             * DATA STATEMENTS *
C             *******************
```

```
      DATA IDUC,LSHAF,ENGU,STU,IVALV/4HDUCT,4HSHAF,4HENGL,4HSTU ,4HVALV/
      DATA ILPC,IHPC,IFAN,IFO,IFI,IHPT,ILPT/3HLPC,3HHPC,3HFAN,2HFO,2HFI,-
     13HHPT,3HLPT/
      DATA PINP,YES/.TRUE.,4HY   /,PLOT/.FALSE./
C
C     ---- TEST WTEST FLAG
      IF (IWT.EQ.0) GO TO 620
      IF (IWAY.GE.0) GO TO 60
      DO 40 I=1,40
      WNTF(I)=WTF(I)
      FARN(I)=FAR(I)
      INPRES(I)=IOPRES(I)
      INTEMP(I)=IOTEMP(I)
      CNRFLO(I)=CORFLO(I)
      DATOUN(1,I)=DATOUT(1,I)
      DATOUN(2,I)=DATOUT(2,I)
      DATOUN(8,I)=DATOUT(8,I)
      DATOUN(9,I)=DATOUT(9,I)
      IF (I.LE.20) GO TO 10
      K=I+20
      DATOUN(1,K)=DATOUT(1,K)
      DATOUN(2,K)=DATOUT(2,K)
      DATOUN(8,K)=DATOUT(8,K)
      DATOUN(9,K)=DATOUT(9,K)
   10 IMPRES(I)=0
      IMTEMP(I)=0
      CORFLM(I)=0
      DO 20 KK=1,4
      DATMAC(KK,I)=0
   20 DATALT(KK,I)=0
      DO 30 KK=1,9
      DATOUM(KK,I)=0
      IF (I.LE.20) GO TO 30
      K=I+20
      DATOUM(KK,K)=0
   30 CONTINUE
   40 CONTINUE
      DO 50 I=1,900
   50 DANINP(I,1)=DATINP(I,1)
   60 DO 80 I=1,40
C     ****************************************************** 07/12/82 **
C     *****
      IF (JTYPE(I).NE.10) GO TO 65
      IF (DATINP(2,I).EQ.0.0) GO TO 65
      IF (DATOUM(1,I).LT.DABS(DATOUT(1,I))) DATOUM(3,I)=-DATOUT(1,I)*9.8-
     16968*DATOUT(2,I)**2/(3600.*DATOUT(7,I)**2)
      IF (DATOUM(1,I).LT.DABS(DATOUT(1,I))) DATOUM(2,I)=DATOUT(2,I)
      IF (DATOUM(1,I).LT.DABS(DATOUT(1,I))) DATOUM(1,I)=DABS(DATOUT(1,I)-
     1)
   65 CONTINUE
C     ****************************************************** 07/12/82 **
C     *****
      CORFLC(I)=CORFLO(I)/1.5497255
      IF (JTYPE(I).EQ.1) II=I
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF (TOPRES(I).GT.TMPRES(I)) DATMAC(1,I)=DATINP(5,II)
      IF (TOPRES(I).GT.TMPRES(I)) DATALT(1,I)=DATINP(9,II)
      IF (TOTEMP(I).GT.TMTEMP(I)) DATMAC(2,I)=DATINP(5,II)
      IF (TOTEMP(I).GT.TMTEMP(I)) DATALT(2,I)=DATINP(9,II)
      IF (DATOUT(2,I).GT.DATOUM(2,I)) DATMAC(3,I)-DATINP(5,II)
      IF (DATOUT(2,I).GT.DATOUM(2,I)) DATALT(3,I)=DATINP(9,II)
      IF (CORFLC(I).GT.CORFLM(I)) DATMAC(4,I)=DATINP(5,II)
      IF (CORFLC(I).GT.CORFLM(I)) DATALT(4,I)=DATINP(9,II)
      IF (TOPRES(I).GT.TMPRES(I)) TMPRES(I)=TOPRES(I)
      IF (TOTEMP(I).GT.TMTEMP(I)) TMTEMP(I)=TOTEMP(I)
      IF (CORFLC(I).GT.CORFLM(I)) CORFLM(I)=CORFLC(I)
      IF (DATOUT(2,I).GT.DATOUM(2,I)) DATOUM(2,I)=DATOUT(2,I)
      K=I+20
      IF (I.LE.20) GO TO 70
      IF (DATOUT(2,K).GT.DATOUM(2,K)) DATOUM(2,K)=DATOUT(2,K)
   70 IF (JTYPE(I).NE.5) GO TO 80
      DATTRQ=DATOUT(1,I)/DATOUT(2,I)
      IF (DATTRQ.GT.DATOUM(1,I)) DATOUM(3,I)=DATOUT(2,I)
      IF (DATTRQ.GT.DATOUM(1,I)) DATOUM(1,I)=DATTRQ
      IF (I.LE.20.OR.JTYPE(K).NE.5) GO TO 80
      DATTRQ=DATOUT(1,K)/DATOUT(2,K)
      IF (DATTRQ.GT.DATOUM(1,K)) DATOUM(3,K)=DATOUT(2,K)
      IF (DATTRQ.GT.DATOUM(1,K)) DATOUM(1,K)=DATTRQ
C     THE FOLLOWING MAY BE USEFUL FOR DEBUGGING AND HAS BEEN LEFT IN
C     WRITE(10,992)JTYPE(I)
C     992 FORMAT(//' COMP TYPE'I3)
C     WRITE(10,997)
C     997 FORMAT('  DES TOTPRES    N  D    TOTEMP')
C     WRITE(10,999)TOPRES(I),TNPRES(I),TOTEMP(I),TNTEMP(I)
C     WRITE(10,996)
C     996 FORMAT('  DES CORFLO N  D  M   WTF')
C     WRITE(10,999)CORFLO(I),CNRFLO(I),CORFLM(I),WNTF(I)
C     WRITE(10,993)
C     993 FORMAT('  DES EFF        N  D    PR')
C     WRITE(10,999)DATOUT(8,I),DATOUN(8,I),DATOUT(9,I),DATOUN(9,I)
C     WRITE(10,995)
C     995 FORMAT('  MAX TOTPRES    N  M    TOTEMP')
C     WRITE(10,999)TOPRES(I),TMPRES(I),TOTEMP(I),TMTEMP(I)
C     WRITE(10,994)
C     994 FORMAT('  MAX RPM        N  M  D')
C     WRITE(10,999)DATOUT(2,I),DATOUM(2,I),DATOUN(2,I)
C     WRITE(10,991)
C     991 FORMAT(' MTRQ  N M  TRPM MTRPM ')
C     WRITE(10,999)DATTRQ,DATOUM(1,I),DATOUT(2,I),DATOUM(3,I)
C     999 FORMAT(4F12.3)
C     WRITE(10,999)DATMAC(1,I),DATMAC(2,I),DATMAC(3,I)
C     WRITE(10,999)DATALT(1,I),DATALT(2,I),DATALT(3,I)
   80 CONTINUE
      IF (IWT.GE.2) GO TO 90
C     ---- ZERO OUT OUTPUT ARRAYS
      GO TO 620
   90 JSCALE=0
      ISAVE=IWT
      IF (IWT.NE.4) GO TO 130
```

```
  100 JSCALE=JSCALE+1
      IF (JSCALE.GT.ISCALE(2)) GO TO 620
      IF (JSCALE.GT.1.AND.ISCALE(1).EQ.2) IOUTCD=1
      SCALEF=SCALE(JSCALE)
      IF (JSCALE.GT.1) SCALEF=SCALE(JSCALE)/SCALE(JSCALE-1)
  110 IF (REVISE.EQ.YES) SCALEF=1.0
      DO 120 I=1,40
      WNTF(I)=WNTF(I)*SCALEF
      CNRFLO(I)=CNRFLO(I)*SCALEF
      CORFLM(I)=CORFLM(I)*SCALEF
      IDID(I)=0
      DATOUN(1,I)=DATOUN(1,I)*SCALEF
  120 CONTINUE
      IF (JSCALE.GT.1.AND.IWT.EQ.4) GO TO 190
      IF (REVISE.EQ.YES) GO TO 180
  130 DO 140 I=1,5
  140 DSHAF(I)=0.
      DO 150 I=1,60
      WATE(I)=0
      NSTAG(I)=0
      IDID(I)=0
      RPMT(I)=0.
  150 ALENG(I)=0
      DO 160 I=1,40
      TLENG(I)=0
      RI(1,I)=0
      ILENG(I)=0
      RI(2,I)=0
      RO(1,I)=0
  160 RO(2,I)=0
C
C     ---- NAMELIST READ OF WTEST DATA
C
      CALL NAMEPR (9,10,8,PINP)
      READ (8,W)
      DO 170 I=1,60
      DO 170 K=1,17
      IF (IWMEC(1,I).EQ.NUMNUM(K)) IWMEC(1,I)=IRNAME(K)
  170 CONTINUE
      GO TO 190
  180 WRITE (20,760)
      CALL NAMEPR (20,10,8,PINP)
      READ (8,W)
C
C     ---- PROCESS LENGTH CONTRIBUTING VECTOR EXCEPT DUCTS AND SHAFTS
C
  190 DO 310 I=1,40
      NC=ILENG(I)
C     WRITE(10,7777)NC,ILENG(I),JTYPE(NC)
      IF (NC.EQ.0) GO TO 320
      JT=JTYPE(NC)
      GO TO (310,230,270,200,210,280,260,220,250,295,310,310,310,310), J-
     1T
C     ---- COMPRESSOR
```

ORIGINAL PAGE IS OF POOR QUALITY

```
  200 CALL COMP (NC)
      GO TO 300
C     ---- TURBINE
  210 CALL TURB (NC)
      GO TO 300
C     ---- MIXER
  220 CALL WMIXR (NC)
      GO TO 300
C     ---- PRIMARY BURNER
  230 IF (IWMEC(1,NC).EQ.IDUC) GO TO 240
      IF (IWMEC(1,NC).EQ.IVALV) GO TO 290
      CALL COMBWT (NC)
      GO TO 300
C     ---  DUCTS
  240 CALL DUCTW (NC)
      GO TO 300
C     ----  NOZZLES
  250 CALL WTNOZ (NC)
      GO TO 300
C     ---- SPLITTER
  260 CALL WSPLT (NC)
      GO TO 300
C     TRANSFER DIMENSIONS FOR WATER INJECTION
  270 CALL DUMMY (NC)
      GO TO 300
C     HEAT EXCHANGER WEIGHT
  280 CALL HMEC (NC)
      GO TO 300
C     VALVES
  290 CALL VALVWT (NC)
C     ********************** 07/15/82 *****
      GO TO 300
  295 IF (DATINP(2,NC).EQ.0.0) GO TO 310
      CALL PROPWT (NC)
      IUP=JCONF(1,1)
      IDN=JCONF(1,3)
      GO TO 305
C     ********************** 07/15/82 *****
C     ----  ACCUME LENGTH
  300 IUP=JCONF(NC,1)
      IDN=JCONF(NC,3)
  305 TLENG(IDN)=TLENG(IUP)+ALENG(NC)
      IF (JT.EQ.6) TLENG(IDN)=TLENG(IUP)
      ID2=JCONF(NC,4)
      IF (ID2.GT.0) TLENG(ID2)=TLENG(IUP)+ALENG(NC)
      IF (JT.EQ.6) TLENG(ID2)=TLENG(IUP)
      IU2=JCONF(NC,2)
      IF (IU2.GT.0) TLENG(IU2)=TLENG(IUP)
      IDID(NC)=1
  310 CONTINUE
C     ---- LAST COMPONENT WAS A NOZZLE SET ENGINE MAXIMUM LENGTH
  320 ENGLEN=TLENG(IDN)
C
```

```
C     PROCESS REMAINING COMPONENTS
C
      DO 410 I=1,60
C     WRITE(10,7777)NC,I,IDID(I)
      IF (IDID(I).EQ.1) GO TO 410
C
C     --   PROCESS COMPRESSORS,TURBINES,MIXERS,BURNERS,SPLITTERS
      NC=JTYPE(I)
      IF (NC.LE.0) GO TO 410
      GO TO (410,360,400,330,340,390,380,350,410,410,410,410,410,410), N-
     1C
C     --   COMPRESSOR
  330 CALL COMP (I)
      GO TO 410
C     --   TURBINES
  340 CALL TURB (I)
      GO TO 410
C     --   MIXER
  350 CALL WMIXR (I)
      GO TO 410
C     --   BURNERS
  360 IF (IWMEC(1,I).EQ.IDUC) GO TO 410
      IF (IWMEC(1,I).EQ.IVALV) GO TO 370
      CALL COMBWT (I)
      GO TO 410
C     VALVES
  370 CALL VALVWT (I)
      GO TO 410
C     ---  SPLITTER
  380 CALL WSPLT (I)
      GO TO 410
C     HEAT EXCHANGERS
  390 CALL HMEC (I)
      GO TO 410
C     TRANSFER DIMENSIONS FOR WATER INJECTION
  400 CALL DUMMY (I)
  410 CONTINUE
C     WRITE(10,7771)JTYPE
C
C     ---- PROCESS DUCTS
      DO 420 I=1,60
      IF (IDID(I).EQ.1) GO TO 420
      NC=JTYPE(I)
      IF (NC.NE.2) GO TO 420
      IF (IWMEC(1,I).NE.IDUC) GO TO 420
      CALL DUCTW (I)
  420 CONTINUE
C     WRITE(10,7771)JTYPE
C     ---- PROCESS NOZZLES
      DO 430 I=1,60
      IF (IDID(I).EQ.1) GO TO 430
      NC=JTYPE(I)
      IF (NC.NE.9) GO TO 430
      CALL WTNOZ (I)
  430 CONTINUE
```

ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
C     WRITE(10,7771)JTYPE
C     ---- ACCUME LENGTH
      DO 450 I=1,40
C     WRITE(10,7777)NC,I,IDID(I)
      IF (JTYPE(I).LE.0) GO TO 440
      IF (IDID(I).EQ.1) GO TO 450
      NC=JTYPE(I)
      GO TO (450,440,440,440,440,440,440,440,440,450,450,450,450,450), N-
     1C
  440 IUP=JCONF(I,1)
      IU2=JCONF(I,2)
      IDN=JCONF(I,3)
      ID2=JCONF(I,4)
      TLENG(IDN)=TLENG(IUP)+ALENG(I)
      IF (NC.EQ.6) TLENG(IDN)=TLENG(IUP)
      IF (IU2.GT.0) TLENG(IU2)=TLENG(IUP)
      IF (ID2.GT.0) TLENG(ID2)=TLENG(IDN)
      IDID(I)=1
  450 CONTINUE
C     ---- PROCESS SHAFTS
      DO 470 J=1,5
      DO 460 I=1,25
      NC=KKINDS(11,I)
      IF (NC.LE.0) GO TO 470
      IF (IWMEC(1,NC).NE.LSHAF) GO TO 460
      IF (IWMEC(2,NC).EQ.J) CALL SHAFT (NC)
  460 CONTINUE
  470 CONTINUE
C
C     ---- FIND ENGINE MAXIMUM RADIUS
      XR=0
      DO 490 I=1,NOSTAT
      IF (XR.GE.RO(1,I)) GO TO 480
      XR=RO(1,I)
  480 IF (XR.GE.RO(2,I)) GO TO 490
      XR=RO(2,I)
  490 CONTINUE
C
C     ---- GET ENGINE TOTAL WEIGHT AND  ALENG CONVERSION
      WATENG=0
      IF (ACCS.EQ.0) ACCS=.1
      WAT=0.
      DO 500 I=1,60
      IF (JTYPE(I).EQ.9) GO TO 500
      WAT=WATE(I)+WAT
  500 CONTINUE
      WATACC=ACCS*WAT
      IF (IOUTCD.GT.1) WRITE (10,660) IOUTCD
      IF (ISIO) WATACC=WATACC*CONVER(3)
      IF (IOUTCD.GT.1) WRITE (10,690) WATACC
      DO 510 I=1,60
      WFACTR=1.
      IF (DESVAL(15,I).NE.0.) WFACTR=DESVAL(15,I)
      WATE(I)=WATE(I)*WFACTR
```

```
      IF ((JTYPE(I).EQ.4.OR.JTYPE(I).EQ.5).AND.WATE(I).EQ.0.) WATE(I)=+1-
     1.E6
      IF (.NOT.ISIO) GO TO 510
      WATE(I)=WATE(I)*CONVER(3)
      ALENG(I)=ALENG(I)*CONVER(1)
  510 WATENG=WATENG+WATE(I)
C
C     ---- CONVERT RADIAL DIMENSIONS AND TLENG
      IF (.NOT.ISIO) GO TO 530
      DO 520 I=1,NOSTAT
      RI(1,I)=RI(1,I)*CONVER(1)
      RI(2,I)=RI(2,I)*CONVER(1)
      RO(1,I)=RO(1,I)*CONVER(1)
      RO(2,I)=RO(2,I)*CONVER(1)
      TLENG(I)=TLENG(I)*CONVER(1)
  520 CONTINUE
C
C     ---- WRITE COMPONENT WEIGHT INFO
  530 UNITSI=ENGU
      IF (ISII) UNITSI=SIU
      UNITSO=ENGU
      IF (ISIO) UNITSO=SIU
      WRITE (10,670) UNITSI,UNITSO
      IF (IWT.EQ.4) WRITE (10,630)
      IF (IWT.EQ.4) WRITE (10,640)
      IF (IWT.EQ.4) WRITE (10,650) SCALE(JSCALE)
      IF (IWT.EQ.4) WRITE (10,640)
      IF (IOUTCD.LT.1) GO TO 580
      WRITE (10,680)
      HLENG=0.
      CGLENG=0.
      CGWATE=0.
      CGTOTM=0.
      CGCOMP=0.
      DO 570 I=1,60
      NC=JTYPE(I)
      IF (NC.LE.0) GO TO 570
C     WRITE(10,7777)JT
      GO TO (540,540,540,540,540,540,540,540,540,570,540,570,570,570), N-
     1C
C     **************** 07/15/82 *********
  540 IF (NC.EQ.10.AND.DATINP(2,I).EQ.0.0) GO TO 570
      IUP1=JCONF(I,1)
      IUP2=JCONF(I,2)
      IDN1=JCONF(I,3)
      IDN2=JCONF(I,4)
      IF (NC.EQ.10) IUP1=JCONF(I,1)
      IF (NC.EQ.10) IUP2=JCONF(I,2)
      IF (NC.EQ.10) IDN1=JCONF(I,3)
      IF (NC.EQ.10) IDN2=JCONF(I,4)
C     ********** 07/15/82 ******
      WRITE (10,700) I,WATE(I),ALENG(I),TLENG(IDN1),RI(1,IUP1),RO(1,IUP1-
     1),RI(1,IUP2),RO(1,IUP2),RI(2,IDN1),RO(2,IDN1),RI(2,IDN2),RO(2,IDN2-
     2),NSTAG(I)
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      IF (ALENG(I).EQ.O.) GO TO 550
      HLENG=ALENG(I)/2.
      IF (IWMEC(1,I).EQ.ILPC.OR.IWMEC(1,I).EQ.IHPC) CGX=ALENG(I)-CGARM(I-
     1)
      IF (IWMEC(1,I).EQ.IFAN.OR.IWMEC(1,I).EQ.IFO) CGX=ALENG(I)-CGARM(I)
      IF (IWMEC(1,I).EQ.IFI.OR.IWMEC(1,I).EQ.IHPT) CGX=ALENG(I)-CGARM(I)
      IF (IWMEC(1,I).EQ.ILPT) CGX=ALENG(I)-CGARM(I)
      IF (IWMEC(1,I).EQ.ILPC.OR.IWMEC(1,I).EQ.IHPC) HLENG=CGX
      IF (IWMEC(1,I).EQ.IFAN.OR.IWMEC(1,I).EQ.IFO) HLENG=CGX
      IF (IWMEC(1,I).EQ.IFI.OR.IWMEC(1,I).EQ.IHPT) HLENG=CGX
      IF (IWMEC(1,I).EQ.ILPT) HLENG=CGX
      CGLENG=TLENG(IDN1)-HLENG
      GO TO 560
  550 IF (IWMEC(1,I).EQ.ISHAF) CGLENG=CGARM(I)
      IF (IWMEC(1,I).NE.ISHAF) GO TO 570
  560 CGWATE=WATE(I)
      CGCOMP=CGWATE*CGLENG
      CGTOTM=CGTOTM+CGCOMP
  570 CONTINUE
      CENGRA=(CGTOTM+(WATACC*ACCARM))/WATENG
C
C     ---- MAKE SUMMARY PRINT
  580 IF (.NOT.ISIO) GO TO 590
      XR=XR*CONVER(1)
      ENGLEN=ENGLEN*CONVER(1)
      CENGRA=CENGRA*CONVER(1)
  590 WRITE (10,710) WATENG,WATACC,ENGLEN,XR
      IF (SCALE(JSCALE).EQ.1.) SEXPO1=WATENG
      SEXPO2=1.
      IF (SCALE(JSCALE).NE.1..AND.IWT.EQ.4) SEXPOE=ALOG(WATENG/SEXPO1)/A-
     1LOG(SCALE(JSCALE)/SEXPO2)
      IF (SCALE(JSCALE).EQ.1.) SEXPOE=1.
      WRITE (10,720) CENGRA
      IF (IWT.EQ.4) WRITE (10,730) SEXPOE
      IF (JSCALE.GT.1.AND.ISCALE(1).EQ.2) GO TO 610
      IF (ENGINE.EQ.2.) GO TO 610
      IF (IPLT) CALL ENGPLT (ENGLEN,XR)
      IF (PLOT) GO TO 600
      GO TO 610
  600 IF (SKIPIT) CALL EGPLOT (ENGLEN,WATENG)
      IF (SKIPIT) GO TO 610
      WRITE (20,740)
      READ (20,770) ANSWER
      IF (ANSWER.NE.YES) GO TO 610
      CALL EGPLOT (ENGLEN,WATENG)
      WRITE (20,750)
      READ (20,770) REVISE
      IF (REVISE.EQ.YES) IWT=2
  610 ENGINE=1.
      IF (REVISE.EQ.YES) GO TO 110
      IWT=ISAVE
      IF (IWT.EQ.4) GO TO 100
  620 IWT=0
      IOUTCD=2
      RETURN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C
  630 FORMAT (1H /24H ENGINE SCALING DATA     )
  640 FORMAT (20H *******************)
  650 FORMAT (14H SCALE FACTOR ,F5.2)
  660 FORMAT (1H /14H *************/14H *           */,14H *  ACCS WT  *-
     1/14H *           */13H ************,I1)
  670 FORMAT (1H1,26H    WEIGHT INPUT DATA IN ,A4,6H UNITS/27H     WEIG-
     1HT OUTPUT DATA IN ,A4,6H UNITS//)
  680 FORMAT (69H   COMP   WT   COMP  ACCU   UPSTREAM RADIUS    DOWNS-
     1TREAM RADIUS  /77H    NO   EST    LEN   LEN   RI   RO   RI   RO  -
     2 RI   RO   RI   RO    NSTAGE/)
  690 FORMAT (/,11H   ACCS WT=,F8.3)
  700 FORMAT (I7,F6.0,F7.0,F6.0,4F5.0,F6.0,3F5.0,I8)
  710 FORMAT (/,27H  TOTAL BARE ENGINE WEIGHT=,F6.0,2X,12HACCESSORIES=,F-
     17.2,2X,23HESTIMATED TOTAL LENGTH=,F6.0,2X,25HESTIMATED MAXIMUM RAD-
     2IUS=,F5.0)
  720 FORMAT (30H  ESTIMATED CENTER OF GRAVITY=,F6.0)
  730 FORMAT (39H  ESTIMATED AIRFLOW SCALING EXPONENT IS,F6.3)
  740 FORMAT (46H  DO YOU WISH A GRAPHICS PICTURE? YES=Y ; NO=N)
  750 FORMAT (54H  DO YOU WISH TO MAKE CHANGES TO THE INPUT? YES=Y;NO=N)
  760 FORMAT (42H  ENTER DESIRED CHANGES FROM TERMINAL,I.E./39H   &W  DE-
     1SVAL(1,6)=****(DATA)****  &END)
  770 FORMAT (A4)
      END
C*********************************************************************************

*****
```

REFERENCES

1. Fishbach, L. H.; and Caddy, M. J.: NNEP - The NAVY/NASA Engine Program. NASA TM X-71857, Dec. 1975.

2. Onat, E.; and Klees, G. W.: A Method to Estimate Weight and Dimensions of Large and Small Gas Turbine Engines. NASA CR-159481, Jan. 1979.

3. Worobel, R.: and Mayo, M. G.: Advanced General Aviation Propeller Study. NASA CR-114289, Apr. 1971.

4. Worobel, R.: Computer Program Users Manual for Advanced General Aviation Propeller Study. NASA CR-2066, May 1972.

5. Smith, C. E.; Hirschkron, R.; and Warren, R. E.: Propulsion System Study for Small Transport Aircraft Technology (STAT). (R80AEG068, General Electric Company; NASA Contract NAS 3-21996.) NASA CR-165330, May 1981.

6. Fishbach, L. H.: KONFIG and RECONFIG - Two Interactive Preprocessing Programs to the NAVY/NASA Engine Program (NNEP). NASA TM-82636, May 1981.

TABLE I. - USER INPUT FOR PROPELLER PERFORMANCE

SPEC	Option 1, fixed efficiency	Option 2, generalized map	Option 3, user-input map
1	(-) Shaft horsepower	(-) Shaft horsepower	(-) Shaft horsepower
2	Efficiency	Any negative number	Map number
3	Static T/SHP	---	---
4	Power loading[a]	Power loading	Power loading
5	Number of blades[a]	Number of blades	Number of blades[a]
6	---	Lift coefficient	---
7	Activity factor[a]	Activity factor	Activity factor[a]
8	---	Optional scaling factor	Optional scaling factor
9	---	Optional desired design efficiency	Optional desired design efficiency
10	Tip speed[a]	Tip speed	Tip speed

[a]Required only for weight calculation, not for performance.

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

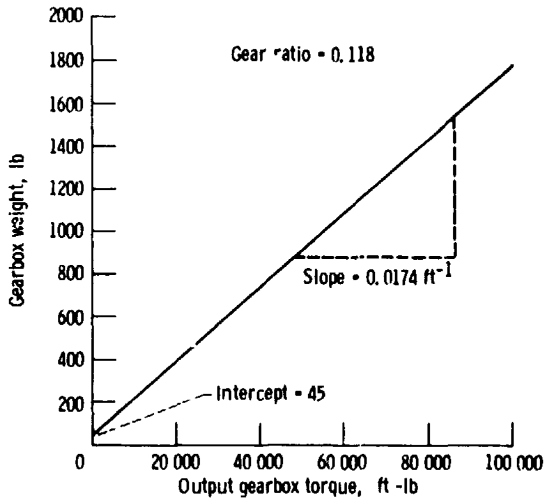

Figure 1. - Baseline gearbox weight as a function of torque for gear ratio = 0.118.