NRC·CNRC

NRC Publications Archive
Archives des publications du CNRC

Uncertainty analysis for gas turbine performance testing / Analyse d'incertitude pour la vérification des performances d'une turbine à gaz

Lutes, D.

For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:
https://doi.org/10.4224/40003892

Laboratory Memorandum (National Research Council Canada. Institute for Mechanical Engineering. Machinery and Engine Technology); no. IME-MET-LM-006, 1992-02

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
https://nrc-publications.canada.ca/eng/view/object/?id=f9be4e8b-59ff-4570-a76d-d13cb981bb28
https://publications-cnrc.canada.ca/fra/voir/objet/?id=f9be4e8b-59ff-4570-a76d-d13cb981bb28

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada | Conseil national de recherches Canada

Canada

National Research Council Canada

Institute for Mechanical Engineering

Machinery and Engine Technology

Conseil national de recherches Canada

Institut de génie mécanique

Technologie des machines et des moteurs

NICCKiC

Uncertainty Analysis for Gas Turbine Performance Testing

D. Lutes

DECLASSIFIED
DÉCLASSIFIÉ

Laboratory Memorandum

Mémoire de laboratoire

1992/02

IME-MET-LM-006

CONTROLLED
UNCLASSIFIED

CONTRÔLÉE
NON CLASSIFIÉE

Canada

INSTITUTE FOR MECHANICAL ENGINEERING PUBLICATIONS

DM **(Institute for Mechanical Engineering Report)**
Scientific and technical information considered important, complete and a lasting contribution to existing knowledge.

TR **(Technical Report)**
Information less broad in scope, but a substantial contribution to existing knowledge.

CTR **(Controlled Distribution or Security Classified Technical Report)**
A technical report with controlled distribution for national security industrial, proprietary or other reasons.

LM **(Laboratory Memorandum)**
Preliminary or exploratory information with controlled distribution.

CAT **(Calibration Analysis and Test Report)**
Information on minor laboratory projects or services.

PUBLICATIONS DE L'INSTITUT DE GÉNIE MÉCANIQUE

DM **(Rapport de l'institut de génie mécanique)**
Informations scientifiques et techniques jugées importantes, complètes et susceptibles de contribuer de façon durable à l'avancement des connaissances courantes.

TR **(Rapport technique)**
Informations de moindre importance, mais pouvant contribuer substantiellement à l'avancement des connaissances actuelles.

CTR **(Rapport technique à distribution contrôlée ou classification sécuritaire)**
Rapport technique à distribution contrôlée pour des raisons de sécurité nationale, de propriété intellectuelle et autres.

LM **(Mémoire de laboratoire)**
Informations préliminaires ou de nature exploratoire à distribution contrôlée.

CAT **(Rapport d'étalonnage d'analyse et d'essai)**
Informations sur de petits projets ou des services de laboratoire.

CONTROLLED
UNCLASSIFIED

CONTRÔLÉE
NON CLASSIFIÉE

UNCERTAINTY ANALYSIS FOR GAS TURBINE PERFORMANCE TESTING

ANALYSE D'INCERTITUDE POUR LA VÉRIFICATION DES PERFORMANCES D'UNE TURBINE À GAZ

D. Lutes

This memorandum is issued to furnish internal or preliminary information. It is subject to revision and updating.

Le présent mémoire est mis en circulation dans le but de fournir des renseignements internes ou préliminaires. Il sera sujet à des révisions et des mises-à-jour.

Institute for Mechanical Engineering
Laboratory Memorandum

1992/02

Institut de génie mécanique
Mémoire de laboratoire

IME-MET-LM-006

I.R.G. Lowe
Interim Director General/
Directeur général intérimaire

ABSTRACT

Uncertainty estimations are an important part of gas turbine performance assessment. A program has been developed to calculate the uncertainties of basic performance measurements and of derived parameters. Functional requirements and software specifications are presented, accompanied by sample calculations. Recommendations are made for future work in this area.

RÉSUMÉ

L'évaluation du degré d'incertitude constitue une part importante de l'évaluation des performances d'une turbine à gaz. Un programme a été élaboré pour calculer l'incertitude des mesures de performances fondamentales et des paramètres dérivés. Les critères de fonctionnement et les caractéristiques des logiciels sont précisés et sont accompagnés d'exemples de calculs. Des recommandations sur des travaux ultérieurs dans ce domaines sont également faites.

CONTENTS

Page

ABSTRACT i

RÉSUMÉ ii

LIST OF APPENDICES iv

LIST OF TABLES iv

LIST OF FIGURES iv

1.0 INTRODUCTION 1

2.0 OBJECTIVE 1

3.0 LITERATURE SURVEY 3

4.0 PHASE I DESIGN 5
4.1 Introduction 5
4.2 Modular Structure 6
4.3 Recommendations for Phase II 8

5.0 PHASE II DESIGN 9
5.1 Introduction 9
5.2 Modular Structure 9

6.0 PROGRAM VALIDATION 12

7.0 RECOMMENDATIONS FOR FUTURE WORK 13

8.0 REFERENCES 14

DOCUMENTATION PAGE

CONTENTS (Cont'd)

LIST OF APPENDICES

APPENDIX		Page
A	DESCRIPTION OF PHASE I MODULES	A-l
B	ARRAY SIZE LIMITATIONS FOR PHASE II	B-l
C	DESCRIPTION OF PHASE II MODULES	C-l
D	SAMPLE CALCULATIONS	D-l

LIST OF TABLES

TABLE		Page
1	P3 elemental error sheet	15
2	Pl elemental error sheet	15
DI	Uncertainty model: RSS	D-9
D2	Uncertainty model: additive	D-10

LIST OF FIGURES

FIGURE		Page
1	Phase I modular structure	16
2	Phase II modular structure	17

UNCERTAINTY ANALYSIS FOR GAS TURBINE PERFORMANCE TESTING

1.0 INTRODUCTION

A versatile software tool was required for uncertainty assessments of gas turbine engine performance values. The program was to be compatible with gas turbines being tested at the Engine Laboratory, NRC, such as the F404, T56 and J85 engines, using the NEFF data acquisition system. The program was implemented on the laboratory MicroVAX with menu-driven interactions.

2.0 OBJECTIVE

The objective of the program was the development and implementation of an uncertainty assessment system for gas turbine performance parameters. The following is a list of functional requirements that were to be met in a three-phase plan. This report will present the design and application of Phases I and II to a J57 experimental setup. Phase III has not been completed as yet and will await further development.

Phase I

a) to allow user input/edit of elemental error sources through a menu-driven interface;

b) to allow the user to type bias and precision error sources on the screen or load previously saved input data files;

c) to allow the user to express input data in terms of % reading or % full scale;

d) to input the target or reference value at which the performance parameters are to be evaluated;

e) to allow the user to input/edit influence coefficients or load previously saved files of coefficients;

f) to propagate the elemental errors in the calculation of overall bias, precision, and uncertainty for each performance parameter.

The objective for the first phase was to develop a working model that performed the basic statistical calculations and propagated identifiable errors to the performance parameters, providing an overall measure of uncertainty. Values of all elemental error sources were assumed. These values were taken from Table C2 of the J85 NEFF System Uncertainty Appendix A to provide typical bias and precision estimates. Influence coefficients for the elemental sources were also assumed, and used as weighting factors in accurately propagating the errors.

Phase II

a) to allow the user to select input data displays for each measurement of force, pressure, temperature, fuel flow, speed, and area;

b) to incorporate an option menu that will allow the definition of the elemental error sources, the sensors used in the calculations, the performance parameters, and the type of uncertainty model (additive or root-sum-square model);

c) to facilitate the association of a sensor label with a measurement filename; (Therefore, when, for example, a temperature file is created, the user can provide a sensor label to the program to identify the type of temperature file.)

d) to allow the association of a performance parameter label with an influence coefficient filename;

e) to provide a student "t" table for t95 values, according to input degrees of freedom;

f) to facilitate automatic loading of all measurement and influence coefficient files;

g) to display the individual results of bias, precision, uncertainty, and reference value for each measurement;

h) to print the error elements of the various performance parameters.

The second phase involved designing menus for each measurement of force, pressure, temperature, fuel flow, speed, and area. The format for the menus closely follows that of the error audit sheets from AEDC (Appendix B). Each measurement display contains a complete breakdown of all elemental error sources (bias and precision) associated with calibration, the sensor system, and the data acquisition system.

The option menu is intended to increase the flexibility of the package, allowing the user to define performance parameters having any combination of dependent sensors. Therefore, if, traditionally, a performance parameter is dependent on variables W, X and Y, it could be tested for its response to additional variables (say W,X,Y,Z), or deleted variables (say W and Y).

Phase III

a) to allow elemental error sources (bias and precision) to be estimated from sensor data;

b) to incorporate I/O functions to retrieve incoming sensor data from an RS232 interface;

c) to provide curve-fitting functions for calibrating individual sensors, using incoming data and known measurement values;

d) to calculate the bias and precision errors of the curve fits corresponding to each sensor, and to use the results for estimating elemental error sources;

e) to calculate influence coefficients for each performance parameter;

f) to determine curve slope effects on performance parameters arising from errors in the reference value reading;

g) to provide graphic output for bias, precision, and uncertainty results, i.e. bar graphs, pie charts, line graphs, etc.

The goal of the final phase is to reduce the need for the user to input all of the elemental error estimates. Many of these input values can be derived from sensor data. By supplying the sensor and data acquisition systems with known input measurements and voltages, the output can be recorded, and some of the elemental estimates can be calculated. Thus, the Phase III program will help reduce set-up times when performing the error analysis on different engines and sensor configurations.

Influence coefficients will be calculated by applying a one percent increase to each elemental measurement, and observing the relative change of the performance parameter. Curve slope effects are to be determined by propagating errors in measurements to the independent and dependent performance parameters used in graphical and functional analysis (Ascough 1985). Graphic output will be obtained by linking the program to a graphics library within the VAX system.

3.0 LITERATURE SURVEY

Although much has been written on the subject of uncertainty, few articles deal with its automated implementation. Abernethy et al. (1980) have developed uncertainty methods which are now considered to be an international standard for gas turbine measurement analysis. However, a standard approach to coding computer algorithms has not been defined.

Boedicker (1986) has developed and implemented a computer subroutine which basically performs the same error calculations as Phases I and II of this project. Elemental errors are input by the user and influence coefficients are calculated by applying a one percent increase to the measurement values and observing the change in the performance parameter. The errors are then combined in a root-sum-square equation, weighted by the influence coefficients. Input for degrees of freedom have been ignored since they are assumed to be greater than 30 for all elemental errors, therefore a "t" value of 2 is always used to calculate uncertainty.

Boedicker stresses the point that the influence coefficients are only valid for one operating condition. The coefficients will change when different reference points are selected, and

thus there will be a change in the performance parameter uncertainties. Furthermore, the selection of the reference point should be such that sensors are not operating in the low end of their range where uncertainty can be quite high.

Existing computer or algorithmic implementations incorporating the functional requirements of Phase III have not been developed as yet. Taylor (1988) describes a method for calculating the elemental bias and precision errors of the data acquisition system (DAS). This is done by applying known voltages at m number of points and making n observations at each point. To estimate the DAS bias, Taylor suggests selecting the largest elemental bias, rather than calculating the arithmetic average of n observations at each voltage point and taking the root-sum-square (RSS) of m points. DAS precision error is estimated by calculating the standard deviation of n observations at each of m points, then taking the RSS of m standard deviations.

Errors associated with the sensor system are more difficult to isolate. Error estimates derived from sensor data typically are a combination of several elemental errors. For example, Abdelwahab, Biesiadny, and Silver (1987) write that force calibration data can be used to estimate sensor bias and precision errors. However, the resulting estimate is a combined value of hysteresis, repeatability, nonlinearity, pressure, and temperature effects on the test cell and service lines. Isolating each of these elemental errors would be quite difficult in an automated process. At this point, the question concerning how far to isolate the elemental errors should be given consideration. It is not useful trying to estimate errors from sensor data if, in the end, a complete elemental breakdown is required. Perhaps the best one can achieve is a "combined-elemental" error estimate, consisting of more than one influencing error source.

A detailed uncertainty analysis methodology is given by Thompson, Kimzey, and Boals (1987). They recognized that combined-elemental or overall measurement errors can be calculated from measurement system end-to-end checks, where known inputs are compared to observed outputs. For example, known voltages can be input to the data acquisition system and the processed, digitized output can be recorded. In this way a combined-elemental error estimate of data processing and data sampling can be automatically calculated. However, some engineering judgement is required to establish the amount that each of these individual errors contributes to this overall combined error. It is, therefore, often the case that individual errors are estimated through special tests and engineering judgement, and automated end-to-end checks are later used to validate the elemental estimates.

Vleghert (1987) writes that accurate comparisons between tests require equally accurate estimates of reference value uncertainty and curve slope effects. Any uncertainty in the chosen basic of reference parameter Az translates into a discrepancy in the performance parameter Ay. Uncertainty in the reference value is due to the reference value being dependent on some other variable, say x. Therefore an influence coefficient ôz/ôx, establishing the effect of x on the reference value z, can be determined. As a result, a shift on the performance parameter y-axis is equal to the product of the change in x, its influence

on the reference value z, and the slope of the y-z performance parameter curve. This yields the following equation:

$$Ay - Ax \cdot \frac{\hat{o}z}{\hat{o}x} \cdot \frac{dy}{dz} \quad \textbf{(1)}$$

Equation (1), however, is incomplete since the variable x may also directly affect the performance parameter y, causing an additional y-axis shift. Therefore, an additional influence coefficient ôy/ôx must be determined to establish the direct effect of x on the performance parameter y. The following equation incorporates the two simultaneous shifts on the y-axis:

$$\Delta y = \Delta x \frac{\delta y}{\hat{o}x} - \frac{\delta z}{\hat{o}x} \cdot \frac{dy}{dz} \quad \textbf{(2)}$$

Therefore, knowing the uncertainty of the variable x, its influence on the reference value and the performance parameter, and the slope of the performance parameter curve y-z, the resulting uncertainty of the performance parameter y, can be calculated.

Ascough (1985) writes that in addition to the three main categories of error, viz. Calibration Hierarchy, Measurement System, and Data Acquisition, an additional category, called Real Effects, exists. This category consists of three error sources: Profiles, Fluctuations, and Mechanics. Profile errors are mainly bias errors associated with the variance of temperature or pressure profiles across a duct. Since only a finite number of probes can be used to determine the profile, a curve fit must be made which, in turn, is likely to incorporate some fixed error. Fluctuation errors are mainly precision errors caused by the engine control system, not being completely steady, or are induced by the installation of the particular test facility. Mechanic errors apply mostly to the engine test frame. Ascough writes that this source of error is much more important than that of the instrumentation, and that day-to-day variations in thrust calibrations are due mainly to test frame mechanics.

Since the mechanics and control system of each test facility are never exactly the same, Poti and Rabe (1988) write that, during tests, performance parameter data should be repeated at each of the basic operating conditions. Comparisons between the repeated data sets not only evaluate the measurement capability of the data system, but also account for the facility's capability for accurately controlling and repeating an operating point.

4.0 PHASE I DESIGN

4.1 Introduction

During the preliminary design, the functional requirements for the Phase I design were outlined as follows:

a) to allow user input/edit of elemental error sources;

b) to allow the user to express input data in terms of % reading or % full scale;

c) to input target or reference value at which the performance parameters are to be evaluated;

d) to input bias, precision, and degrees of freedom for each elemental error source;

e) to allow user input/edit of influence coefficients;

f) to propagate the elemental errors in the calculation of the overall bias, precision, and uncertainty for each performance parameter.

All of the above requirements have been incorporated into the completed Phase I design. While coding the program in VAX Fortran, it was found that a menu-driven interface, using screen displays, would be very helpful since the user is required to input a large number of data values. It was also found that Phase I was in need of a file storage and retrieval facility to eliminate the need of re-typing all of the input data when performing successive runs. These functions were originally going to be incorporated in Phase II, however it was quickly realized that the user interface and file facility were fundamental to the modular structure.

The program contains two main input displays: the measurement error source display and the influence coefficient display. The user has the choice of either calling up new input displays or updating old ones. Once the selected display is on the screen, the user can control the cursor with the arrow keys to type data values in at the appropriate screen locations. The contents of the display can then be saved by selecting the SAVE/LOAD menu option and providing a filename. When the display is saved, the data values are loaded into the system at the same time. Once both, measurement error source and influence coefficient displays, have been saved, the program is ready to run. The data from each display are then combined in the proper bias, precision, and uncertainty calculations for each performance parameter. The results are presented in the output display.

4.2 Modular Structure

The modules, or subroutines, have been organized as shown in Figure 1. The goal of the structure design was to isolate program functions as much as possible and to eliminate redundant coding. This simplifies the design considerably and makes the task of modifying the program much easier. For example, one module performs the editing for all the displays. Upon completing Phase I, it was noticed that some improvements in modular structure could be made. These improvements will be discussed in Section 5. A short discussion of each module is given in Appendix A. The modules of Phase I are:

a) MAIN_MENU

The MAIN_MENU subroutine creates the main menu display (VD_MAIN) that presents four menu options: FILE, RUN, OPTIONS, and EXIT.

b) FILE_MENU

The FILE_MENU subroutine performs all file input/output operations of measurement and influence coefficients.

c) NEW_ERROR_DISPLAY

This subroutine creates a display of new measurement errors, which contains zeros at the screen locations where the user can type in nominal bias and precision values, and degrees of freedom for each sensor.

d) NEW_IC_DISPLAY

In NEW_IC_DISPLAY a new influence coefficient display is created, showing zeros at the screen locations where the user can type in the coefficient values for each variable.

e) SCREEN_EDIT

The SCREEN EDIT subroutine performs the editing for both the error and influence coefficient displays.

f) SAVE LOAD

The SAVE_LOAD subroutine creates a filename display (VD_FILENAME) that allows the user to type/edit a name for the file.

g) RETRIEVE

The RETRIEVE subroutine creates the display VD_RETRIEVE, which prompts the user to indicate the type of file they wish to retrieve (i.e. error display or influence coefficient display).

h) RETRIEVE_ERROR_DISPLAY

In RETRIEVE-ERROR DISPLAY the existing error measurement file is opened and all data values in the file are read into a character array for presentation on the screen.

i) RETRIEVE_IC_DISPLAY

In subroutine RETRIEVE_IC_DISPLAY the existing influence coefficient file is opened and all data values in the file are read into a character array for presentation on the screen.

j) READ_ERROR_DISPLAY

Once called, the READ_ERROR_DISPLAY subroutine will read all data values from their assigned screen locations into a character array and store them in a sequential file.

k) READ IC-DISPLAY:

The READ_IC_DISPLAY subroutine performs the same function as the READ_ERROR_DISPLAY module, except the read commands are set for the VD_IC display.

l) RUN_MENU

The RUN_MENU subroutine performs all the bias, precision, and uncertainty calculations for each performance parameter.

4.3 Recommendations for PHASE II

While coding the Phase I program, some possible improvements were noticed, which would increase the structure of the design. The following is a list of recommendations for the Phase II development:

a) to initialize (create) all displays in one module so they can simply be pasted and unpasted when necessary, rather than re-creating them each time their routines are called;

b) to incorporate all editing functions for all displays in one module. This can be easily accomplished by numbering the displays and using a computed GOTO control structure that selects the correct cursor control option, according to the value of CURRENT-DISPLAY. Therefore, when editing for any display is required, simply call the one existing edit routine.

c) to design the input data displays such that all data values are spaced at regular intervals on the screen. This will allow easy reading of the data from, and writing to, the screen, using a WHILE loop, rather than having to position the cursor with separate SMG$READ_FROM_DISPLAY and SMG$PUT_CHARS statements for each data value.

These changes were made during Phase II.

5.0 PHASE n DESIGN

5.1 Introduction

The functional requirements for Phase II were outlined as follows:

a) to allow the user to select input data displays for each measurement of force, pressure, temperature, fuel flow, speed and area;

b) to incorporate an option menu that will allow the user to define the elemental error sources, the sensors used in the calculations, the performance parameters, and the type of uncertainty model (additive or root-sum-square model);

c) to provide the facility which allows the user to associate a sensor label with a measurement filename. Therefore when, for example, a temperature file is created, the user can provide a sensor label for the program to identify the type of temperature file it will be.

d) to provide the user with the capability to associate a performance parameter label with an influence coefficient filename;

e) to incorporate a Student "t" table for t95 values according to input degrees of freedom;

f) to facilitate the automatic loading of all measurement and influence coefficient files; The user can create and edit a list of measurement files, then save the file list under a user defined name. Thus, an entire file set can be loaded by specifying the file set name.

g) to provide a series of displays showing the individual results of bias, precision, uncertainty, and reference value for each measurement. The displays are pasted on top of each other such that only the measurement name is visible. By positioning the cursor on the highlighted area of any display, the user can pop the chosen display into view and move it to any area on the screen. This allows him to select and compare randomly the individual error results of each measurement file.

h) to provide a hardcopy output of error results for elemental measurements and performance parameters.

5.2 Modular Structure

The modular structure of Phase II is shown in Figure 2. The design has changed considerably from that of Phase I, mainly due to the development of the OPTIONS_MENU and the automatic file loading facility LOAD_FILE_SET. The RUN module of Phase I is no longer present in the MAIN_MENU. The automatic loading facility, contained in the FILE_MENU, merely calls the CALCULATE and PROPAGATE_ERRORS subroutines

as it is loading in each of the input data files. It should be noted that the Phase II design is much more flexible than the one of Phase I, allowing the user to define error types, performance parameters, and a list of all the sensors involved in the error analysis. Appendix B lists the maximum number of error types, performance parameters, and sensors that the user can define. A short discussion of each of the Phase II subroutines is given in Appendix C. The subroutines covered are:

a) INITIALIZEDISPLAYS

The INITIALIZEDISPLAYS subroutine creates most of the displays used in the program as well as updates any displays which have been modified by the OPTIONS_MENU.

b) MAIN_MENU

The first three main selections are displayed in MAIN_MENU: FILE_MENU, OPTIONS_MENU, and EXIT.

c) OPTIONS_MENU

The OPTIONS_MENU module allows the user to select which setup parameters (errors, sensors, performance parameters or uncertainty model) he wishes to define.

d) DEFINEERRORS

Elemental error names for each measurement file, such as temp, press, force, etc., can be typed in and are saved in a file used to update the corresponding display.

e) DEFINESENSORS

The DEFINE-SENSORS subroutine allows the user to define a list of all sensors or elemental measurements that are involved in the error analysis.

f) DEFINE-PERFORMANCE PARAMETERS

All performance parameters are defined in the DEFINEPERFORMANCEPARAMETERS subroutine.

g) DEFINE-UNCERT-MODEL

The user may select either the root-sum-square uncertainty model or the additive model by choosing the DEFINE_UNCERT_MODEL option.

h) CALCULATE_IC

The CALCULATE_IC module is currently empty and will be developed in Phase III.

i) FILE_MENU

The FILE_MENU subroutine accesses all file input/output operations as well as the automatic loading and error propagation routines.

j) RETRIEVE-DISPLAY

All types of measurement files, such as temp, press, force, etc., as well as influence coefficient files, are selected from the RETRIEVE-DISPLAY.

k) SAVE/LOAD DISPLAY

The SAVE/LOAD-DISPLAY subroutine calls the READ ERRORS or READ_IC module depending on which display is currently displayed.

l) EDIT_CURRENT_DISPLAY

Selecting this option from the FILE_MENU will allow the user to jump into the display currently showing on the screen, with editing control.

m) PRINT_RESULTS

Selecting this option from the FILE_MENU allows the user to obtain a hardcopy of the error results.

n) LOADFILESET

The LOAD_FILE_SET subroutine performs automatic loading and error calculations for each measurement and influence coefficient file.

o) EDIT

The EDIT subroutine performs all editing functions for all displays in the program.

p) RETRIEVE ERRORS

Existing measurement files are uploaded by the RETRIEVE ERRORS subroutine from memory or new files are created through it.

q) RETRIEVEJC

Existing influence coefficient files are uploaded by the RETRIEVE IC module from memory or new IC files are created by it.

r) READ ERRORS

The READ_ERRORS subroutine facilitates the reading of all error values from their correct screen locations and stores them in a sequential file.

s) READ IC

All influence coefficient values are read from their screen locations through the READ_IC subroutine and stored in a sequential file.

t) CALCULATE

The CALCULATE subroutine performs all error calculations for each of the measurement files.

u) PROPAGATE ERRORS

Error results calculated from the CALCULATE subroutine are combined with influence coefficients to propagate each of the measurement errors to the performance parameters.

v) INDIVIDUAL RESULTS

The results from the CALCULATE module are written by the INDIVIDUAL_RESULTS subroutine to small, individual displays, one for each sensor, and are presented on the screen.

w) MOVE_DISPLAY

The MOVE_DISPLAY subroutine allows the user to move each of the individual displays to any location on the screen and to compare error results between different sensors.

x) STUDENT_t_TABLE

Given input degrees of freedom, the correct t95 value is returned by the STUDENT_t_TABLE subroutine.

6.0 PROGRAM VALIDATION

The following is a discussion of how the elemental error estimates are processed and combined with the influence coefficients to obtain the performance parameter errors. Sample calculations are provided to verify the correctness of the program.

Each measurement of temp, press, etc., has three main sections containing elemental errors.

These sections are:

a) Calibration Hierarchy
b) Measurement System
c) Data Acquisition

The CALCULATE module of the program calculates a first bias, precision, uncertainty, and overall degrees of freedom estimate for each of these three main sections. Root-sum-square equations calculate the bias and precision errors, while the Welch-Satterthwaite formula is used for determining the overall degrees of freedom for each section. The degrees of freedom then obtain a t95 values from a Student T table. Uncertainty is calculated by either the additive or the root-sum-square model, as specified by the user.

The results from each of the three sections are then combined to obtain an overall estimate of bias, precision, uncertainty, and degrees of freedom for the sensor. These calculations are performed using the same method as described for the three main sections of measurement error.

Once the individual error results for each sensor have been calculated, the program is ready to propagate these errors to the performance parameters, using the influence coefficients. Each dependent variable, i.e. each influencing sensor, of the performance parameter is multiplied by its corresponding influence coefficient in a root-sum-square equation. This is done for both bias and precision errors. The overall degrees of freedom for each dependent variable of the performance parameter are multiplied by their influence coefficients in the Welch-Satterthwaite equation. Once again, a t95 value is obtained from the Student 'f Table. Uncertainty is calculated either by the root-sum-square approach or the additive one, as specified by the user.

For the sample calculations, the performance parameter P3Q1 will be used. In test cell No.5, P3Q1 is typically influenced by the sensors P3 and Pl, having influence coefficients of 0.85 and 0.15, respectively. Elemental error sheets containing fictional data are shown in Tables 1 and 2. Hand calculations for this set of input data are shown in Appendix D.

7.0 RECOMMENDATIONS FOR FUTURE WORK

Although Phases I and II were quite challenging, as far as developing the file input/output, flexibility, user friendliness, and calculating routines are concerned, more theoretical work will be required in Phase III. Since the goal of Phase III is to develop an on-line system that will help calculate elemental errors from curve-fits of test data and from measurement responses to known inputs, it will be difficult to isolate errors that are due to a change of, say, only one variable. The automated process will likely lump many elemental errors together and will therefore only be capable of evaluating a combined elemental error, which is a function of perhaps three variables. If one is trying to determine a curve slope effect, and the uncertainty of the variable x affecting the reference value y, is a function of more than one variable, the calculations become confusing very quickly. Therefore, it is

advantageous to try and isolate errors as much as possible, rather than following the more convenient method of grouping them together.

As a final note, care should be taken in passing parameters between modules, when common blocks are used. It should be ensured that the blocks are identical and typed in the same order, wherever they are used in the program. To be safe, all variables in the common block should also be declared at the beginning of the routine, whether they are used in the routine or not.

8.0 REFERENCES

Abernethy, R.B. 1980. Handbook, Uncertainty in Gas Turbine Measurements, Arnold Engineering Development Center, Arnold Air Force Station, TE, NTIS AEDC-TR-73-5.

Abdelwahab, Biesiadny, and Silver. 1987. Measurement Uncertainty for the Uniform Engine Testing Program Conducted at NASA Lewis Research Center, Technical Memorandum 88943.

Ascough, J.C. 1985. Prediction Synthesis of Error Limits for J57 UETP Tests in Cell 3, Tech Memo P1069, Pyestock, RAE, September 1985.

Boedicker, C.A. 1986. A Unique, Rapid Uncertainty Analysis for Turbine Engine Testing, ASME 86-GT-280.

Poti and Rabe. 1988. Verification of Compressor Data Accuracy by Uncertainty Analysis and Testing Methods, Journal of Turbomachinery, Vol. 110, pp. 265-269, April 1988.

Taylor J.L. 1988. Fundamentals of Measurement Error, Neff Instrument Corporation, pp. 74-80.

Thompson, Kinzey, and Boats. 1987. Overview of the Application of Standard Measurement Uncertainty Methodology to Propulsion Testing, AGARD CP 429, AGARD Fluid Dynamics Panel.

Vleghert J.P.K. 1987. Investigation of Measuring Uncertainty within the Uniform Engine Test Program, Report of the Sub-Group on Uncertainty Assessment, Draft AGARDograph 307.

Table 1. P3 elemental error sheet
REF: 81.00 psi

			BIAS	PRECISION	EF
I	1-----	CALIBRATION HIERARCHY	0.267	0.001	30
	1___		0.000	0.000	00
	1-----	INSTALLATION EFFECTS	0.020	0.000	30
	|	SENSOR HYSTERESIS	0.080	0.004	20
	|	NON-LINEARITY	0.040	0.050	20
II	1	REPEATABILITY	0.060	0.050	30
	|	TEMPERATURE EFFECTS	0.034	0.050	30
	|	VIBRATION EFFECTS	0.020	0.002	30
	|	PRESSURE EFFECTS	0.010	0.000	30
	|		0.000	0.000	00
	1		0.000	0.000	00
	1___		0.000	0.000	00
	1-----	SAMPLING ERROR	0.040	0.004	30
III	|	MEASUREMENT CHANNEL ERROR	0.040	0.004	30
	|	DATA PROCESSING ERROR	0.040	0.004	30
	1___		0.000	0.000	00

Table 2. Pl elemental error sheet
REF: 14.73 psi

			BIAS	PRECISION	EF
I	1-----	CALIBRATION HIERARCHY	0.057	0.003	20
	1___		0.000	0.000	00
	1-----	INSTALLATION EFFECTS	0.030	0.050	20
	|	SENSOR HYSTERESIS	0.030	0.060	20
	|	NON-LINEARITY	0.001	0.050	20
II	1	REPEATABILITY	0.002	0.006	20
	|	TEMPERATURE EFFECTS	0.010	0.006	20
	|	VIBRATION EFFECTS	0.020	0.070	20
	|	PRESSURE EFFECTS	0.060	0.070	20
	|		0.000	0.000	00
	|		0.000	0.000	00
	1___		0.000	0.000	00
	1-----	SAMPLING ERROR	0.200	0.000	20
HI	1	MEASUREMENT CHANNEL ERROR	0.050	0.400	20
	1	DATA PROCESSING ERROR	0.300	0.050	20
	1___		0.000	0.000	00

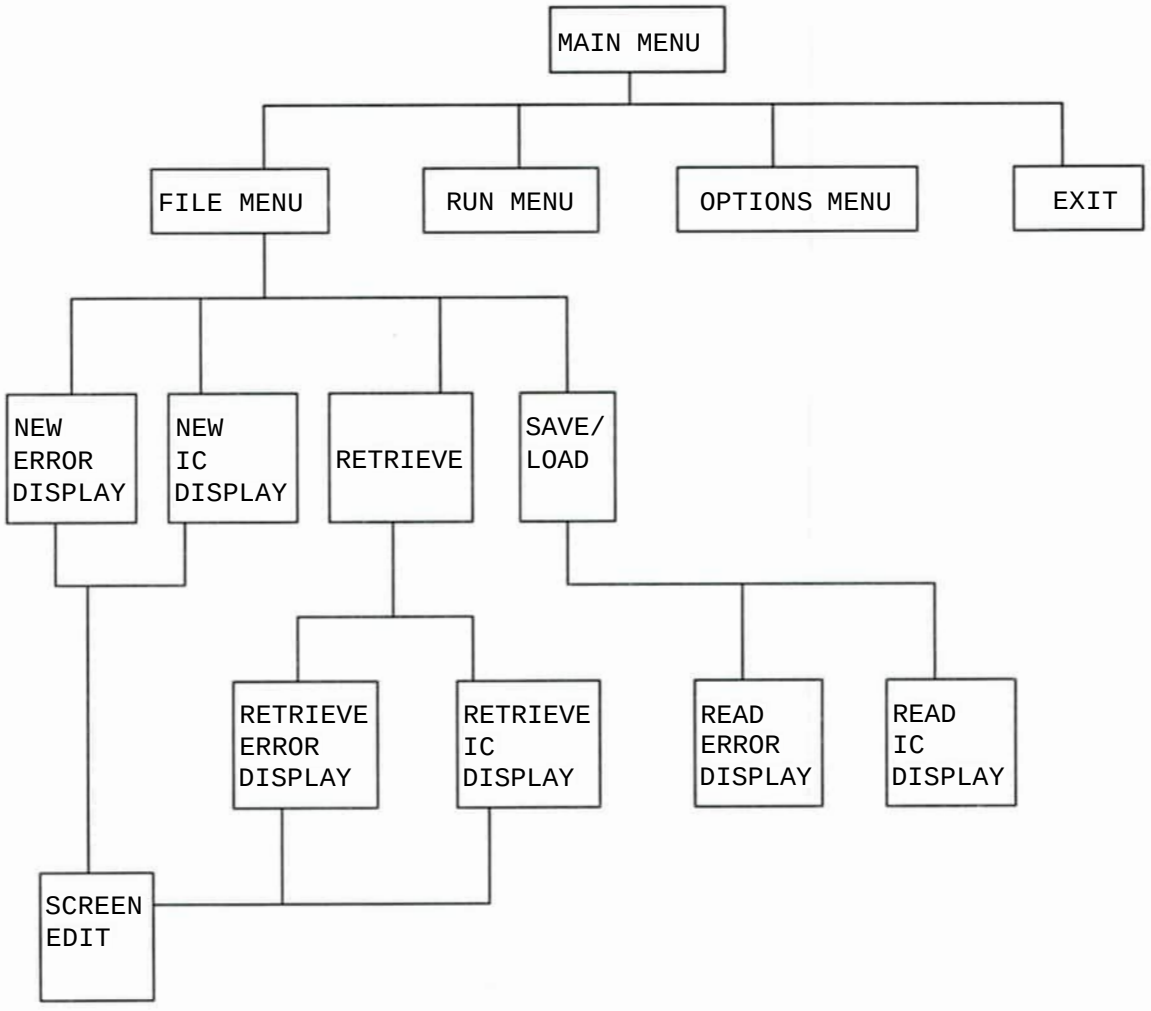

Figure 1. Phase I modular structure

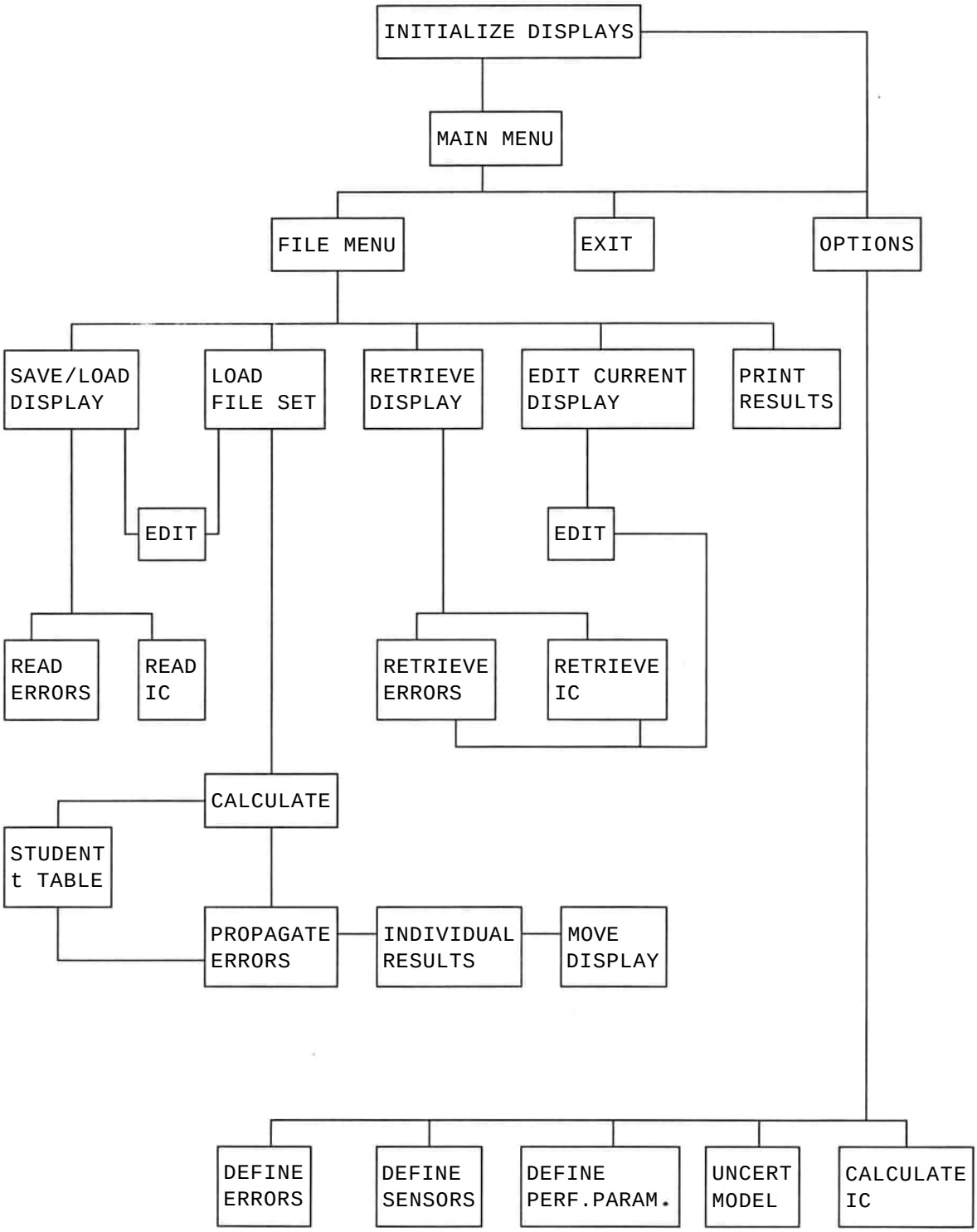

Figure 2. Phase II modular structure

Appendix A

DESCRIPTION OF PHASE I MODULES

DESCRIPTION OF PHASE I MODULES

a) MAIN_MENU

The MAIN_MENU subroutine creates the main menu display (VD_MAIN) that presents four menu options: FILE, RUN, OPTIONS, and EXIT. Most of the work in Phase I has been concentrated in the FILE option. This option, as shown in Figure 1, contains all the file input/output and editing facilities. The RUN module is simply a routine that performs bias, precision, and uncertainty calculations, and displays the results on the screen. The OPTIONS module has not been coded yet. This is to be left for Phases II and III where the user will be able to select output displays or indicate the manner in which he wants to run the program, eg. if he wishes to link the program to certain modules that will calculate influence coefficients, curve fit sensor data, etc. The EXIT option simply terminates the program.

The VMS Screen Management Library has been used throughout the program to create and control the pop-up displays. Before modifying the code, one should become familiar with this relatively easy-to-use set of functions. (See Section 8.2 of the VMS Programmers Manual, and Section RTL-146 to RTL_236 of the VMS Programming Support Manual.) The Screen Management Library also allows the programmer to create a virtual keyboard that enables the user to select menus by pressing specific keys. (See RTL-218 of the VMS Programming Support Manual for a list of corresponding key codes.)

All global variables have been defined in common blocks, such as COMMON /DISP/ and COMMON /DISP2/. The MAIN_MENU subroutine cycles in a continuous loop, performing a read operation each cycle to determine the selected key. Once the key is selected, the corresponding module is called.

b) FILE_MENU

If the FILE_MENU is selected, four options are presented: NEWERRORDISPLAY, NEW IC-DISPLAY, SAVE/LOAD, and RETRIEVE. The first two options display blank measurement error and influence coefficient sheets. The data values in these displays appear as zeros and are to be filled in by the user. The SAVE/LOAD option will read the contents of the current display and load the data into their corresponding variables. The RETRIEVE option will display an existing error file or influence coefficient file on the screen. Once again, a loop cycles continuously until a key selection is pressed.

c) NEWERRORDISPLAY

The NEWERRORDISPLAY subroutine creates a new measurement error display, which contains zeros at the screen locations where the user can type in nominal, bias, precision values, and degrees of freedom for each sensor. The global variable CURRENT-DISPLAY is set to "one" to indicate the currently active display. Once the display is on the screen, the SCREEN_EDIT module is called to enable screen editing of the display.

d) NEWJC-DISPLAY

A new influence coefficient display is created in the NEW_IC_DISPLAY module. Each performance parameter is listed with its dependent variables. Again, zeros appear at the screen locations where the user can type in the coefficient values for each variable. The global variable CURRENT_DISPLAY is set to "two". The SCREEN_EDIT module is then called to enable editing of the display.

e) SCREEN_EDIT

The SCREENEDIT subroutine is a continuous loop that cycles until the user decides to save the contents of the screen by pressing <F17>. The read statement which is performed on each cycle places the keyboard character in the variable INPUT_STRING. The codes for the arrow keys are held in the variable TERMINATOR. By checking the integer value of TERMINATOR on each loop cycle, the cursor can be moved in the appropriate direction using the SMG$SET_CURSOR_REL command. If any key other than the arrow keys has been pressed, the value of KEY will equal "one", causing the pressed character to be displayed on the screen. This subroutine performs the editing for both the error and influence coefficient displays.

f) SAVE LOAD

Once a display has been filled with the desired data values, the FILE key can be selected, <F17>, and the SAVE/LOAD key pressed, <PF3>. The SAVE_LOAD routine creates a filename display, VD_FILENAME, that allows the user to type/edit a name for the file. After pressing <RETURN>, either the READ_ERROR_DISPLAY or the READ_IC_DISPLAY routines will be called, depending on the value of CURRENT-DISPLAY.

g) RETRIEVE

The RETRIEVE module creates the display VD_RETRIEVE, which prompts the user to indicate the type of file he wishes to retrieve, i.e. error display or influence coefficient display. Once the type of file is selected, the filename display, VD_FILENAME is presented on the screen, prompting the user for a filename. After pressing <RETURN>, the INQUIRE command is used to check if the file exists. If the file does not exist, the VD_NOEXIST display is created and flashed on the screen for one second. Control is then passed back to the beginning of the editing loop for the VD FILENAME display, again prompting the user for an existing filename. Once an existing file has been found, either the RETRIEVE_ERROR_DISPLAY or RETRIEVE_IC_DISPLAY modules are called, depending on the type of file which was selected in the VD_RETRIEVE display.

h) RETRIEVE_ERROR_DISPLAY:

In the RETRIEVE ERROR DISPLAY subroutine, the existing file is opened and all data values in the file are read into the character array MENU_STRIN(72). The error display, VD_ID, is then presented on the screen, followed by a loop which inserts all the stored data values at their appropriate locations, using the command SMG$PUT_CHARS. The SCREEN_EDIT module is then called to enable editing of the retrieved file.

i) RETRIEVE_IC_DISPLAY

The RETRIEVEICDISPLAY subroutine performs the same function as described in paragraph h) above, except that the influence coefficient display VD_IC is presented on the screen. Since the VD_IC display has its data values placed at irregular screen locations, it was difficult to insert the MENU_STRIN(39) at the correct screen locations, using a loop, as was done in the RETRIEVE_ERROR_DISPLAY module. Thus, the SMG$PUT_CHARS command is used for each stored data value. This could be overcome by re-designing the VD IC display to ensure regular spacing of data values.

j) READ ERROR DISPLAY

The common block COMMON/DATA/ is created in the READ_ERROR_DISPLAY module to hold all the read data variables, which will later be passed to the RUN module for the calculations. As mentioned earlier, the READ_ERROR_DISPLAY subroutine is called from the SAVE_LOAD module. Once called, the subroutine will read all data values from their assigned screen locations into the character array MENU_STRIN(72), using the SMG$READ_FROM_DISPLAY command. This is accomplished by a loop similar to that of the RETRIEVE_ERROR_DISPLAY subroutine. The character values now held in MENU_STRIN(72) must be converted into real data types in order to perform the calculations. Therefore, a READ statement is used to convert each value in MENU_STRIN(72), and assign it to a real variable which can be passed to the RUN module. Once the data have been read into the variable names, the values held in MENU_STRIN(72) are written to a sequential file.

k) READ IC DISPLAY

The READICDISPLAY subroutine performs the same function as the READ ERROR DISPLAY module, except that the SMG$READ_FROM_DISPLAY commands are set for the VD_IC display, and the real variable names are different. Again, since the data values are irregularly spaced, a "brute force" method of reading the display is used.

l) RUN_MENU

The RUN_MENU module performs all the bias, precision, and uncertainty calculations for each performance parameter. The common block COMMON/DATA/ is declared at the

beginning of the routine to access all the previously loaded data values. The measurement error estimates, weighted by the influence coefficients, are then combined in a root-sum-square calculation of the bias and precision errors. Uncertainty is obtained by adding the bias and total estimated scatter. (Note: total estimated scatter = 2 x precision). It has been assumed that a minimum of 30 degrees of freedom are required to merit the use of the "t" factor = 2.

Appendix B

ARRAY SIZE LIMITATIONS FOR PHASE II

ARRAY SIZE LIMITATIONS FOR PHASE H

INPUT	LIMIT
Number of different sensors for each measurement:	
Temperature	10
Pressure	10
Force	5
Speed	5
Fuel Flow	5
Area	5
Number of different performance parameters	20
Number of elemental error sources for each measurement section (for all measurements):	
Calibration Hierarchy	2
Measurement System	10
Data Acquisition	4
Number of virtual displays handled by EDIT routine	40
Number of characters used for a filename	10
Number of characters used for sensor and performance parameter labels	4

s?naaoi\ n asvM jo NOiidiHDsaa

3 xipuaddy

DESCRIPTION OF PHASE H MODULES

a) INITIALIZE DISPLAYS

The INITIALIZE_DISPLAYS module creates most of the displays used in the program, as well as updates any displays which have been modified by the OPTIONS_MENU. Displays created here are stored in common blocks so they can be passed to other modules and simply be pasted when needed. When the program is started, this routine will upload the current .OPT files which were previously defined in the OPTIONS_MENU. There are eight .OPT files, six files for error definitions of temp, press, force, speed, fuel flow, and area, and two files for the sensor list and the performance parameter list. They are named as follows:

1) T ERRORS.OPT
2) PERRORS.OPT
3) FERRORS.OPT
4) S_ERRORS.OPT
5) FF ERRORS.OPT
6) AERRORS.OPT
7) SENSOR_LIST.OPT
8) PP LIST.OPT

b) MAIN_MENU

The first three main selections are displayed in the MAIN_MENU module: FILE MENU, OPTIONS_MENU, and EXIT. This is a continuous loop that cycles until the <F20> key is pressed. A SMG$READ_STRING command is executed on each pass of the loop to obtain the TERMINATOR value of the user's keystroke.

c) OPTIONS MENU

The OPTIONS_MENU module allows the user to select the setup parameters to be defined, i.e.errors, sensors, performance parameters or uncertainty model. Once all setup parameters have been defined, the user may quit the menu with the <Do> key. The INITIALIZE_DISPLAYS routine is then called to update the error displays, sensor list, and performance parameter list.

d) DEFINE ERRORS

Elemental error names for each measurement file (temp, press, force, etc.) can be typed in and are saved in a file to update the corresponding display. This subroutine is called from OPTIONS_MENU. The new error names can be typed on the screen, and are then saved in the appropriate X ERRORS.OPT file, when the <RETURN> is pressed. To backup to the previous menu, press <Do>.

e) DEFINESENSORS

The DEFINESENSORS subroutine allows the user to define a list of all sensors that are involved in the error analysis. In this routine, a display is pasted to the screen which allows the user to type in the labels for all sensors to be used. The sensor list can then be saved in the SENSORLIST.OPT file by pressing <RETURN>. To backup to the previous menu, press <Do>.

f) DEFINE PERF PARAM

All performance parameters are defined in the DEFINE_PERF_PARAM module. A display is pasted to the screen, allowing the typing in of all the performance parameters used in the analysis. The labels can be saved in the PPLIST.OPT file by pressing <RETURN>. To backup to the previous menu, press <Do> .

g) DEFINEUNCERTMODEL

The user may select either the root_sum_square (RSS) uncertainty model or the additive model through the DEFINE_UNCERT_MODEL option. Each time the <PF4> key is pressed, the name of the current model setting (additive or RSS) will be displayed on the screen.

h) CALCULATEJC

The CALCULATE_IC module is currently empty and will be developed in Phase III.

i) FILE_MENU

The FILE_MENU subroutine accesses all file input/output operations, as well as the automatic loading and error propagating routines. To quit this menu, press <Do> .

j) RETRIEVE DISPLAY

All types of measurement files (temp, press, force, etc.), as well as influence coefficient files are selected from the RETRIEVE-DISPLAY. This subroutine is called from FILE_MENU. It provides a list of seven selections, viz.six measurement displays and one influence coefficient display.

When a measurement file is selected, a small display is pasted, prompting the user for a filename. Once a name has been typed in, and <RETURN> is pressed, the program checks to see if the file exists, and if it does, it will simply be recalled from memory and pasted to the screen. If the file does not exist, a second display is pasted, prompting the user for a sensor label. This is done so the program identifies the new file is, for example, a temperature file for the sensor PSI. If the sensor label typed in is not found in the sensor list, previously defined in the DEFINE SENSORS routine, a message will flash on the

screen, telling the user the label was not found. If the user has forgotten the labels defined previously, he may backup to the MAIN_MENU with the <Do> key and call up the OPTIONS_MENU to check the sensor list. Once a legal sensor label is typed in, a new measurement file for that particular sensor is created and displayed on the screen.

When an influence coefficient file is selected, the filename display is once again pasted. If the file exists, it is uploaded and displayed. If the file does not exist, a second display is pasted, prompting the user for a performance parameter label. If the typed label is not found in the performance parameter list previously defined in DEFINEPERFPARAM module, a warning message will flash on the screen.

k) SAVE/LOADDISPLAY

The SAVE/LOAD DISPLAY subroutine calls the READ_ERRORS or READ_IC module, depending on which display is currently on the screen.

l) EDIT CURRENT DISPLAY

Selecting the EDIT_CURRENT_DISPLAY option from the FILE_MENU will allow the user to jump into the display currently showing on the screen, with editing control. The global variable CURRENT_DISPLAY is checked to see which display is on the screen, and the EDIT routine is called.

m) PRINT RESULTS

Selecting the PRINT_RESULTS option from the FILE_MENU provides the capability of obtaining a hardcopy of the error results. The error results are those from the most current run of the program. Individual results from each sensor, as well as the bias, precision, and uncertainty of each performance parameter, are printed. The user has the option of obtaining a hardcopy from either the laser or the dot matrix printer.

n) LOAD FILE SET

The LOAD_FILE_SET subroutine performs automatic loading and error calculations for each measurement and influence coefficient file. When called from the FILE_MENU, a small display is pasted, prompting the user for a data set name. If the file does not exist, a new file is created under the filename. The sensor and performance parameter lists are then written adjacent to the screen location, where the user will type in the filenames of all the measurement and influence coefficient files of the analysis.

The user must type in an 'END' statement after the last influence coefficient file, otherwise a warning message will flash. Also, all filenames must be matched with a sensor or performance parameter label, or a warning message will flash. Finally, if one of the filenames typed does not exist, a "does not exist" message flashes.

Once all legal filenames are typed, < RETURN > may be pressed to begin the automatic loading. The RETRIEVE ERRORS module is called to write the first sensor file to the screen. The READ_ERRORS and CALCULATE modules are then called to evaluate the sensor errors. Once all sensor files have been loaded and calculated, the RETRIEVE_IC and READ_IC routines are called to load the influence coefficients. Finally, the PROPAGATE_ERRORS routine is called to evaluate errors for the performance parameters.

o) EDIT

The EDIT subroutine performs all the editing functions for all displays in the program. It uses a computed GOTO statement and the value of CURRENT DISPLAY to provide editing control over the display currently on the screen.

p) RETRIEVE ERRORS

Existing measurement files are uploaded from memory or new files are created by the RETRIEVE_ERRORS module. New files are displayed on the screen with zeros printed at the input locations. Where error names have not been defined, the printed zeros can be simply left at their screen locations. All measurement files are written to the screen in this routine, using a computed GOTO statement, which uses the integer value of the global variable CURRENT_DISPLAY to write characters to the correct display. Once the file is on the screen, the EDIT subroutine is called, enabling the user to type in data. After the data are typed, the file key <F17> may be pressed to call up the FILE_MENU and to select the SAVE_LOAD_DISPLAY key to save the file.

q) RETRIEVE IC

Existing influence coefficient files are uploaded from memory or new IC files are created by the RETRIEVE_IC subroutine. When an influence coefficient (IC) file is uploaded, the sensor list is also uploaded. Each IC file has its corresponding performance parameter printed at the top of the screen. The sensor list is written to the screen beside all the user input locations. Therefore, each IC file is for one performance parameter, allowing the user to type in IC values next to any influencing sensor. Sensors, having no effect on the performance parameter, are simply left as zeros. Once the IC values are typed in, the editing and save functions operate in the same manner as described in paragraph p).

r) READ ERRORS

All elemental error values are read from their correct screen locations and are stored in a sequential file in READ ERRORS. There must be at least one non-zero value in each of the three main sections, Calibration Hierarchy, Measurement System, and Data Acquisition, of the measurement file, otherwise a message will flash on the screen telling the user there are insufficient data.

s) READIC

All influence coefficient values are read from their screen locations by the READ_IC module and stored in a sequential file. There must not be any numerical values typed at screen locations not having an adjacent sensor label. If this condition is detected by the program, a message will flash explaining that there is an unpaired numerical value and sensor label.

t) CALCULATE

The CALCULATE subroutine is used to perform the individual calculations for all the sensors. The sensor list is first searched to find the correct array position for the calculations that follow. The routine calculates a bias, precision, uncertainty, and overall degrees of freedom estimate for each of the three main sections. Root-sum-square equations calculate the bias and precision errors, while the Welch-Satterthwaite formula is used to calculate the overall degrees of freedom for each section. The degrees of freedom then obtain a t95 values from a Student 't' table. Uncertainty is calculated using either the additive or root-sum-square model, as specified by the user. The results from each of the three sections are then combined to obtain an overall estimate of bias, precision, uncertainty, and degrees of freedom for the sensor. These calculations are performed by the same method as described for the three main sections of measurement error.

u) PROPAGATE_ERRORS

Error results calculated from the CALCULATE routine are combined with influence coefficients in the PROPAGATE_ERRORS module to propagate each of the measurement errors to the performance parameters. This subroutine contains three nested DO WHILE loops. The outer loop cycles through all of the performance parameters in the performance parameter list, the middle loop marks whether the calculation is for propagation of bias, precision, or degrees of freedom, and the inner loop cycles through all of the sensors in the sensor list. Before stepping to the next performance parameter, a t95 value is obtained from a STUDENT_t_TABLE and the uncertainty is calculated, using either the additive or RSS model. After the calculations are completed, the results are displayed on the screen.

v) INDIVIDUAL-RESULTS

Once the performance parameter error results are written to the screen, the user has the option to view the individual error results of each sensor by means of the INDIVIDUAL-RESULTS module, or quit to the FILE_MENU. When viewing the individual results, the bias, precision, and uncertainty of all three main sections, as well as the total and reference values, will be pasted to the screen in small displays, one for each sensor.

w) MOVE DISPLAY

The MOVE_DISPLAY subroutine allows the user to move each of the individual displays to any location on the screen and compare error results between different sensors. By positioning the cursor on the highlighted area of any display, the < 4 > , < 8 >, < 6 > , and < 2 > keys on the keypad may be used to pop the display up into view and move it around the screen.

Appendix D

SAMPLE CALCULATIONS

SAMPLE CALCULATIONS

This section provides a sample calculation to illustrate the algorithms which are implemented in this package. The complete uncertainty calculation is shown for the compressor pressure ratio, P3/P1 (P3Q1). It involves calculations for P3 and Pl initially, and then propagation through the influence coefficients to the bias, precision and uncertainty for the parameter, P3/P1. Throughout, results are given for both additive and root sum square (RSS) error models. Details are also included of the degrees of freedom and the associated 95th percentile point for the two-tailed Students "t" distribution (t95).

The nomenclature for the intermediate results is:

Vstation_class_type

where *station* = Pl for Pl
= P3 for P3

class = CALI for the calibration hierarchy error class
= MEAS for the measurement methods error class
= DATA for the data reduction error class
= TOTAL for the total error from all classes

type = BIAS for the bias contribution
= PREC for the precision contribution
= DF for the degrees of freedom
= UNCERTA for the total uncertainty with the additive error model
= UNCERTR for the total uncertainty with the root sum square error model

The influence coefficients for P3Q1 are P1_IC and P3_IC with respect to Pl and P3, respectively. For the P3Q1 parameter, the nomenclature is:

P3QI *_type*

where the type follows the nomenclature for the intermediate results.

Sample Calculations

1) P3

a) Calibration Hierarchy

P3_CALI_BIAS - 0.267

P3_CALI_PREC - 0.001

P3_DF - 30, Therefore: t95 - 2.0

P3_CALI_UNCERTA - 0.267 + (2.00x0.001)

- 0.269 (additive)

$$P3_CALI_UNCERTR - ((0.267)^2 + (2.00x0.001)^2)^{\frac{2}{2}}$$

- 0.267 (RSS)

b) Measurement System

$$P3_MEAS_BIAS - (0.02^2 + 0.08^2 + 0.04^2 + 0.06^2 + 0.34^2 + 0.02^2 + 0.01^2)^{\frac{2}{2}}$$

- 0.117

$$P3_MEAS_PREC - (0.004^2 + O.O5^2 + 0.05^2 + 0.05^2 + 0.002^2)^{\frac{2}{2}}$$

- 0.087

$$P3_MEAS_DF - \frac{(0.004^2 + 0.05^2 + 0.05^2 + 0.05^2 + 0.002^2)^2}{\frac{(0.004)^4}{20} + \frac{(0.05)^4}{20} + \frac{(0.05)^4}{30} + \frac{(0.05)^4}{30} + \frac{(0.002)^4}{30}}$$

- 260, therefore: t95 - 2.0

P3_MEAS_UNCERTA - 0.117 + (2.00x0.087)

- 0.291 (additive)

$$P3_MEAS_UNCERTR - (0.117^2 + (2.00x0.087)^2)^{\frac{2}{2}}$$

- 0.209

c) Data Acquisition System

$$P3_DATA_BIAS - (0.04^2 + 0.04^2 + 0.04^2)^{\frac{2}{2}}$$

- 0.069

$$P3_DATA_PREC = (0.004^2 + 0.004^2 + 0.004^2)^{\frac{2}{2}}$$

$$= 0.007$$

$$P3_DATA_DF = \frac{(0004^2 \pm \frac{0.004^2}{(0.004)^4} \pm \frac{0.004^2)^2}{(0.004)^4}}{30 \quad + \quad 30 \quad + \quad 30}$$

$$= 89, \quad \text{therefore: } t95 = 2.0$$

$$P3_DATA_UNCERTA = 0.069 + (2.00 \times 0.007)$$

$$= 0.083 \quad \text{(additive)}$$

$$P3_DATA_UNCERTR = ((0.069)^2 + (2.00 \times 0.007)^2)^{\frac{2}{2}}$$

$$= 0.070 \quad \text{(RSS)}$$

d) Total Sensor Error

$$P3_TOTAL_BIAS = (P3_CALI_BIAS^2 + (P3_MEAS_BIAS^2 + P3_DATAJBIAS^2)^{\frac{2}{2}}$$

$$= (0.267^2 + 0.117^2 + 0.069^2)^{\frac{2}{2}}$$

$$= 0.300$$

$$P3_TOTAL_PREC = (P3_CALI_PREC^2 + (P3_MEAS_PREC^2 + P3_DATA_PREC^2)^{\frac{2}{2}}$$

$$= (0.001^2 + 0.087^2 + 0.007^2)^{\frac{2}{2}}$$

$$= 0.087$$

$$P3_TOTAL_DF = \frac{(P3_CALI_PREC^2 + P3_MEAS_PREC^2 \leftarrow P3_DATA_PREC^2)^2}{\frac{(P3_CALI_PREC)^4}{P3_CALI_DF} + \frac{P3_MEAS_PREC)^4}{P3_MEAS_DF} + \frac{(P3_DATA_PREC)^4}{P3_DATA_DF}}$$

$$\text{P3 TOTAL DF} - \frac{(0\text{-}001^2 + 0\text{-}087^2 + 0.007^2)^2}{\frac{(0.001)^4}{30} + \frac{(0.087)^4}{260} + \frac{(0.007)^4}{89}}$$

- 263, therefore: t95 - 2.0

P3_TOTAL_UNCERTA - 0.300 + (2.00x0.087)

- 0.474 (additive)

$$\text{P3_TOTAL_UNCERTR} - ((0.300)^2 + (2.00\text{x}0.087)^2)^{\frac{2}{2}}$$

- 0.347 (RSS)

2) Pl

a) Calibration Hierarchy

P1CALIBIAS - 0.057

P1_CALI_PREC - 0.003

P1_CALI_DF - 20, Therefore: t95 - 2.086

P1_CALI_UNCERTA - 0.057 + (2.086x0.003)

- 0.063 (additive)

$$\text{P1_CALI_UNCERTR} - ((0.057)^2 + (2.086\text{x}0.001)^2)^{\frac{2}{2}}$$

- 0.059 (RSS)

b) Measurement System

$$\text{P1_MEAS_BIAS} - (0.03^2 + 0.03^2 + 0.001^2 + 0.002^2 + 0.010^2 + 0.020^2 + 0.060^2)^{\frac{2}{2}}$$

- 0.077

$$P1_MEAS_PREC = (0.05^2 + 0.06^2 + 0.05^2 + 0.006^2 + 0.006^2 + 0.006^2 + 0.07^2 + 0.07^2)^{\frac{2}{2}}$$

$$= 0.136$$

$$P1_MEAS_DF = \frac{(0.05^2 + 0.06^2 + 0.05^2 + 0.006^2 + 0.006^2 + 0.07^2 + 0.07^2)^2}{\frac{(0.05)^4}{20} + \frac{(0.06)^4}{20} + \frac{(0.05)^4}{20} + \frac{(0.006)^4}{20} + \frac{(0.006)^4}{20} + \frac{(0.07)^4}{20} \frac{(0.07)^4}{20}}$$

$$= 130, \quad \text{therefore: } t95 = 2.0$$

$$Pl.MEASJJNCERTA = 0.077 + (2.00x0.136)$$

$$= 0.349 \quad \text{(additive)}$$

$$P1_MEAS_UNCERTR = (0.077^2 + (2.00x0.136)^2)^{\frac{2}{2}}$$

$$= 0.283 \quad \text{(RSS)}$$

c) Data Acquisition System

$$P1_DATA_BIAS = (0.200^2 + 0.05^2 + 0.30^2)^{\frac{2}{2}}$$

$$= 0.364$$

$$Pl\text{-}DATA\text{-}PREC = (0.04^2 + 0.05^2)^{\frac{2}{2}}$$

$$= 0.064$$

$$P1_DATA_DF = \frac{(0\ 042 + \circ_052)2}{\frac{(0.04)^4}{20} + \frac{(0.05)^4}{20}}$$

$$= 42, \quad \text{therefore: } t95 = 2.0$$

$$P1_DATA_UNCERTA = 0.364 + (2.00x0.064)$$

$$= 0.492 \quad \text{(additive)}$$

$$P1_DATA_UNCERTR = ((0.364)^2 + (2.00 \times 0.064)^2)^{\frac{2}{2}}$$

$$= 0.386 \quad (RSS)$$

d) Total Sensor Error

$$P1_TOTAL_BIAS = (P1_CALI_BIAS^2 + (P1_MEAS_BIAS^2 + P1_DATA_BIAS^2)$$

$$= (0.057^2 + 0.077^2 + 0.364^2)^{\frac{2}{2}}$$

$$= 0.376$$

$$P1_TOTAL_PREC = (P1_CALI_PREC^2 + (P1_MEAS_PREC^2 + P1_DATA_PREC^2)^{\frac{2}{2}}$$

$$= (0.003^2 + 0.136^2 + 0.064^2)^{\frac{2}{2}}$$

$$= 0.150$$

$$Pl\ TOTAL\ DF = \frac{(P1\text{-}^{CALI}\text{-}PREC^{2} + \frac{P1_MEAS_PREC^2 + P1_DATA_PREC^2)^2}{(P1_CALI_PREC)^4}}{P1_CALI_DF} + \frac{}{P1_MEAS_DF} + \frac{(P1_DATA_PREC)^4}{P1_DATA_DF}$$

$$P1_TOTAL_DF = \frac{\frac{(0.0032 + \circ\text{-}.1362 + 0.0642)^2}{(0.003)^4}}{20} + \frac{}{130} + \frac{(0.064)^4}{42}$$

$$= 169, \quad \text{therefore: } t95 = 2.0$$

$$P1_TOTAL_UNCERTA = 0.376 + (2.00 \times 0.150)$$

$$= 0.676 \quad (additive)$$

$$P1_TOTAL_UNCERTR = ((0.376^2 + (2.00 \times 0.150)^2)^{\frac{2}{2}}$$

$$= 0.481 \quad (RSS)$$

3) Propagation to P3Q1

influence coefficients: P3_IC = 0.85

P1_IC = 0.15

$$P3Q1.BIAS - [(P3_ICxP3_TOTAL_BIAS)^2 + (PlJCxPl_TOTAL_BIAS)^2]$$

$$-[(O.85xO.3OO)^2 + (0.15x0.376)^2]^2$$

$$- 0.261$$

$$P3Q1_PREC - [(P3JCxP3_TOTAL_PREC)^2 + (Pl_ICxPl_TOTAL_PREC)^2]$$

$$-[(0.85x0.087)^2 + (0.15x0.150)^2]^2$$

$$- 0.077$$

$$P3Q1\ DF - \frac{[(^{p3} - ^{ICxP3} - TOTAL_PREC)^2 + (Pl_ICxPl_TOTAL_PREC)^2]^2}{\frac{(P3_ICxP3_TOTAL_PREC)^4}{P3_TOTAL_DF} + \frac{(Pl_ICxPl_TOTAL_PREC)^4}{P1_TOTAL_DF}}$$

$$P3Q1DF - \frac{[(^{\circ}-^{85} *^{0}-^{087})^{2} + (O\text{-}lSxO.15)^2]^2}{\frac{(0.85x0.087)^4}{263} + \frac{(0.15x0,15)^4}{130}}$$

$$- 18004 \quad \text{therefore:} \quad t95 - 2.0$$

$$P3Q1JJNCERTA - 0.261 + (2.00x0.077)$$

$$- 0.415 \quad (\text{additive})$$

$$P3Q1.UNCERTR - ((0.261^2 + (2.00x0.077)^2)^2)^2$$

$$- 0.303 \quad (RSS)$$

Table DI. Uncertainty model: RSS

Pl ERRORS Reference Pl = 14.73 psi			
	BIAS	PRECISION	UNCERTAINTY
CALIBRATION HIERARCHY	0.057	0.003	0.057
MEASUREMENT SYSTEM	0.077	0.136	0.282
DATA ACQUISITION	0.364	0.064	0.386
TOTAL	0.376	0.150	0.482

P3 ERRORS Reference P3 = 81.00 psi			
	BIAS	PRECISION	UNCERTAINTY
CALIBRATION HIERARCHY	0.267	0.001	0.267
MEASUREMENT SYSTEM	0.117	0.087	0.209
DATA ACQUISITION	0.069	0.007	0.071
TOTAL	0.300	0.087	0.346

	BIAS	PRECISION	UNCERTAINTY
P3Q1	0.261	0.077	0.303

Table D2. Uncertainty model: additive

Pl ERRORS Reference Pl = 14.73 psi			
	BIAS	PRECISION	UNCERTAINTY
CALIBRATION HIERARCHY	0.057	0.003	0.063
MEASUREMENT SYSTEM	0.077	0.136	0.349
DATA ACQUISITION	0.364	0.064	0.492
TOTAL	0.376	0.150	0.677

P3 ERRORS Reference P3 = 81.00 psi			
	BIAS	PRECISION	UNCERTAINTY
CALIBRATION HIERARCHY	0.267	0.001	0.269
MEASUREMENT SYSTEM	0.117	0.087	0.290
DATA ACQUISITION	0.069	0.007	0.083
TOTAL	0.300	0.087	0.474

	BIAS	PRECISION	UNCERTAINTY
P3Q1	0.261	0.077	0.415

REPORT DOCUMENTATION FORM/FORMULAIRE DE DOCUMENTATION DE RAPPORT

REPORT No./ NO DU RAPPORT 1 IME-MET-LM-006	REPORT No./ NO DU RAPPORT 2	SECURITY CLASSIFICATION/ CLASSIFICATION DE SÉCURITÉ 3 _ Top Secret/Très secret _ Secret _ Confidential/Confidentiel _ Protected/Protégée _X Unclassified/Non classifiée
DISTRIBUTION/DIFFUSION _X Controlled/Contrôlée 4 _ Unlimited/Illimitée		

DECLASSIFICATION: DATE OR REASON/DÉCLASSEMENT: DATE OU RAISON
5

TITLE, SUBTITLE/TITRE, SOUS-TITRE

6 Uncertainty Analysis for Gas Turbine Performance Testing

AUTHOR(S)/AUTEUR(S)

7 D. Lutes

SERIES/SÉRIE

8

CORPORATE AUTHOR/PERFORMING ORGANIZATION/
AUTEUR D'ENTREPRISE/AGENCE D'EXÉCUTION

National Research Council of Canada
9 Institute for Mechanical Engineering

SPONSORING OR PARTICIPATING AGENCY/AGENCE DE SUBVENTION OU PARTICIPATION

10

DATE	FILE/DOSSIER	SPECIAL CODE/ CODE SPÉCIAL	PAGES	FIGURES	REFERENCES
11 92/02	12	13 EEPOO	14 46	15 2	16 8

NOTES

17

DESCRIPTORS (KEY WORDS)/MOTS-CLÉS

18 gas turbine, uncertainty, analysis, software

SUMMARY/SOMMAIRE

19 The design and implementation of an analysis tool is described. Organization, calculation and display is possible for uncertainty estimates for measurements during gas turbine testing.

ADDRESS/ADRESSE
Dr. G. Krishnappa
Engine Laboratory
Institute for Mechanical Engineering
Montreal Road, Ottawa, Canada
20 K1AOR6 (613)993-2425