Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 153582
UNCLASSIFIED	

TITLE

ACOUSTIC TRANSDUCER CALIBRATION SOFTWARE PROGRAMMER'S GUIDE FOR THE LABVIEW 3.1 ENVIRONMENT

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to:

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

DREA CR/95/410

ACOUSTIC TRANSDUCER CALIBRATION SOFTWARE **PROGRAMMER'S GUIDE** FOR THE LABVIEW 3.1 ENVIRONMENT

by
James S. Pyra

MacLAREN PLANSEARCH (1991) Limited
#200 5657 Spring Garden Road
Halifax, Nova Scotia, Canada
B3J 3K1

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

DEFENSIO PER SCIENTIAM

Canada

PLEASE DIRECT ENQUIRIES TO:

THE CHIEF,
DEFENCE RESEARCH ESTABLISHMENT ATLANTIC,
P.O. BOX 1012,
DARTMOUTH, NOVA SCOTIA, CANADA
B2Y 3Z7

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

DREA CR/95/410

ACOUSTIC TRANSDUCER CALIBRATION SOFTWARE **PROGRAMMER'S GUIDE** FOR THE LABVIEW 3.1 ENVIRONMENT

by
James S. Pyra

MacLAREN PLANSEARCH (1991) Limited
#200 5657 Spring Garden Road
Halifax, Nova Scotia, Canada
B3J 3K1

Scientific Authority Yolande R. Bonin

W7707-3-2635/01-OSC
Contract Number

March 1995

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

DEFENSIO PER SCIENTIAM

Canada

ABSTRACT

Software for the calibration of acoustic transducers has been developed in the Labview 3.1 programming system on Macintosh Quadra 900 computer. This document presents notes for the programmer who may be enhancing or maintaining the software, or who is undertaking additional development.

RÉSUMÉ

Le logiciel pour l'étallonage des transducteurs a été développé dans le système de programmation Labview 3.1 avec un Macintosh Quadra 900. Ce document présente des notes pour le programmeur qui augmente ou entretient le logiciel ou pour le programmeur qui entreprend du développement additionel du logiciel.

1. Introduction

Software for the calibration of acoustic transducers has been developed in the Labview programming system on the Macintosh Quadra environment. The tasks for Hydrophone Calibration using Tone Burst or CW Signals, Projector Calibration using Tone burst or CW Signals, Calibration of Hydrophones Using Pseudo-Random Noise, Transducer Directivity Pattern, and Calibration of Hydrophones Using the Reciprocity Method have been implemented using the National Instruments NuBus hardware and Labview software, version 3.1 Some of the tasks were completed in Labview 2.2.1, and ported into the Labview 3.1 environment using the automatic compatibility library provided by National Instruments.

This document presents notes for the programmer who may be enhancing or maintaining the software, or who is undertaking additional development. Throughout this document, the reader is assumed to have access to the software and the Labview development system documentation. This document is not a tutorial on Labview development.

The following information is presented in this Programmers Guide:

Labview Notes

Documentation Features

Organization of the Software

Compatibility Library Issues

VI Reuse

Clusters and Global Values

Shift Registers and Loops

Data Types

NuBus Hardware Issues

Memory and System Speed

Timing Control and Interfacing

Equipment Under GPIB Remote Control

2. Labview Notes

Labview programs are developed as a hierarchical group of Virtual Instruments (VIs) that consist of a front panel and a wiring diagram for each component of the system. Labview is an interpreted system so that any changes made can be put into place by simply running the program again. The flow of data can also be traced through the system and probes can be placed on the wiring diagrams to indicate specific interim values during actual execution.

The Labview programming environment offers the following advantages over conventional procedural programming systems:

- easy to learn;
- diagrams rather than program code are more easily understood;
- visual interface;
- interpreted system rather than compiled makes changes on the fly easier;
- built in hardware support for the NuBus hardware;
- ability to link in external code segments developed in conventional languages;
- large collection of analysis and control VIs available for use in new systems.

The system also presents the following challenges to the programmer:

- high memory requirements, particularly in development mode;
- slow to load and run programs;
- interpreted system rather than compiled makes it easy to accidentally change programs;
- limited documentation.

The maintenance and minor modification of the system can be undertaken by a technical person familiar with the requirements of the system and with Labview particulars as long as they ensure that they back up their work and proceed carefully. Development of extensive applications requiring custom VI development such as the transducer calibration software is not a trivial exercise.

3. Documentation Features

The wiring diagrams that form a Labview program represent the best documentation of that program by visually depicting the flow of information through the various VIs of the system. However, in a complex system, the volume of VIs involved becomes staggering and an interactive examination of the system structure on the computer is preferable to an attempt to produce paper versions of the system hierarchy.

Labview provides a *Help Window* that indicates what all the connections are for the inputs and outputs of each VI pointed to on the screen. Various labels and comments are also provided on the wiring diagrams. Finally, labels used to prompt the user on front panels serve to identify data objects in user terms.

The documentation provided with the Labview development system does not explain the internal mathematics of some of the standard calculation VIs in detail, and documentation regarding the production of linkages to conventional code segments is limited.

The documentation for the compatibility library is also scant, failing to define the many VIs that must be rewired to be executable under the new version of Labview.

There is a Labview runtime module available from National Instruments that permits the creation of executable Labview programs that cannot be modified by the user.

4. Organization of the Software

Labview applications are hierarchical groups of VIs. Each VI consists of a front panel and a wiring diagram. Each front panel allows for user interaction, input, value setting and modification, etc. Each wiring diagram allows for the flow of information (inputs, items declared on the front panel, and outputs) to be directed through other VIs, control mechanisms and data structures. Data types are indicated by the colour of the wires connecting the various objects. Any individual wire or object can have a temporary probe attached in order to display its value during program execution and the flow of all data can be watched in a debugging display mode.

Ordinary execution involves running the software and making inputs via front panels, with results presented to front panels for display. Each parameter value on a front panel is either declared as a global variable (accessable as a VI to other VIs) or is used locally within the wiring diagram attached to the panel. There are various ways to pass data between VIs discussed below. Devices are controlled by means of the standard VIs that support specific functions of the National Instruments Nubus boards.

A single front panel generally has a separate VI for each major function to be launched from that panel. Each of those VIs may contain several other VIs, and so on in a hierarchical organization from the top down. Modifying the software requires opening the panels and diagrams of the affected VIs, and because of the graphical representation of the system, requires a great deal of the system memory. Trying to run other programs while modifying a Labview component deep within the hierarchy is difficult if not impossible due to the memory requirements.

5. Compatibility Library Issues

In order to quickly get previous versions of Labview running under the new version, Labview provides a compatibility library feature to ease the port of the application. The compatibility library documentation indicates that the majority of VIs will be automatically converted for use in the new Labview system, however the experience of the Transducer calibration tasks was that only those VIs with identical names in the old and new versions of Labview were so converted. The compatibility library allows some of the previous version of Labview VIs to be used, however many other VIs have new versions and the old are no longer supported. To convert these unsupported VIs, a manual rewiring process to reconnect the components of the system was required.

The conversion process itself removed obsolete wiring so a dual process of viewing the old wiring and then replacing the new wiring to match it was required. Changing Labview versions and attempting to port old applications is thus a time consuming effort. Furthermore, there may be performance pitfalls in continuing to use the compatibility library, as it is continuing to use older VIs that may have been made more efficient in the newest Labview.

6. VI Reuse

When several similar applications require the same VIs, the VIs can be reused, however there are some complications. If any changes need to be made to particular VIs for the new application, merely copying to a new location is not sufficient to ensure separation of the two VIs. Labview stores file locations as part of its interpreter information, and so when the new program is run, there may be references to the old VI location from each VI that calls or uses the modified one. Labview will prompt if changes are to be saved when a VI is modified but if any old references exist that look for the VI in the old location, the previous version of the VI may be overwritten and lost. To safeguard the old VI, a copy of the entire application should be made in a new directory and the path to the old code temporarily changed so that when the new application is loaded, it does not use any of the old VIs and instead uses only the copy. You may need to indicate the location of some VIs to Labview when it tries to load the new application the first time. After the system is working, the old application can be placed in its proper path.

Careful planning of what VIs must be modified can make this procedure unnecessary, but in large and complicated Labview applications, it is worth making backups of similar applications and removing them from the hard disk during the development process to avoid the automatic search feature of Labview overwriting the older applications with new modifications. It is also wise to ensure that a modified version of a VI has a unique name so that any references to that VI cannot confuse which version should be used.

7. Clusters and Global Values

The first three tasks (Hydrophone Calibration and Projector Calibration Using Tone Burst or CW Signals and Hydrophone Calibration Using Pseudo-Random Noise) make extensive use of *clusters* to pass parameter values to lower level VIs from the panels on which they are set. The Reciprocity Task, although drawn from the first two tasks, was converted to not rely on clusters. We have found that clusters are difficult to use because of the layered nature of Labview programs, requiring extensive modification throughout the software if any single element of the original panel that has been incorporated into a cluster must be changed. Clusters simulate "parameter passing" between VIs but due to their layered, hierarchical nature, the cluster is a read only local copy of the original data value. We therefore embraced the use of global values that are defined as separate objects shown on the panel where they are set by the user and then may be accessed in any VI within the program by simply placing their icons in the wiring diagram as if they were standard VIs. By creating them as separate objects, the global variables can be more freely altered and used for different VIs.

Also, a cluster is a separate data structure requiring memory to be used in each VI where it appears, whereas access to a global variable requires less system memory. The price of global variables is that programmers must remember that every VI that accesses a global variable can potentially alter that variable's value. There should only be modification/definition of global variables on control panels, etc. where the user can consciously choose to set or alter the value in use.

8. Shift Registers, Array Indexing and Loops

Shift registers are used as automatic counters to control the execution of iterative loops within a Labview VI. Iterative counters and array indexing start from 0 (as in the C Programming Language). Shift registers automatically increment by the factor specified at each iteration of the loop. Shift registers can also be used as values within the VI in which they appear for calculation purposes, however, the programmer must be careful not to alter the value used as a shift register to ensure that loop control is not tampered with.

Shift registers are also not automatically initialized, so the programmer must ensure that they are set to an appropriate starting value before they are used.

9. Data Types

Many of the Labview VIs automatically convert data types if the input does not match the expected type (e.g. integer vs. real). However, we found that automatic conversion of data types was suspect (inaccuracies would be introduced into calculations that relied on automatic type conversion). Therefore the programmer is advised to explicitly convert the type of any data object prior to wiring it to a VI that expects a different data type as input.

10. NuBus Hardware Issues

The individual slot numbers where NuBus boards are installed in the Quadra system are typically wired into the Labview programs. Trying to run a Labview program on a different machine can lead to automatic messages about board configurations differing or the system's inability to locate the necessary boards. If this occurs, then merely reset the board numbers in the Labview software to match the particular configuration on which you are running.

GPIB (IEEE 488) interface VIs often report an error message stating that there is no error to report upon first initializing the GPIB interface. There does not appear to be a way to disable or ignore this status message. Also, the Labview VI to send a message to the GPIB does not appear to accurately use the device address information for all

GPIB devices. Some (newer) devices require a distinct address, but other equipment is consistently addressed with the same value without an error.

If the Monitor/keyboard/mouse is placed too close to other equipment, interference is noticeable in the generation and display of noise other than the desired signals. Care must be taken to ensure that the equipment is adequately positioned to avoid such noise.

11. Memory and System Speed

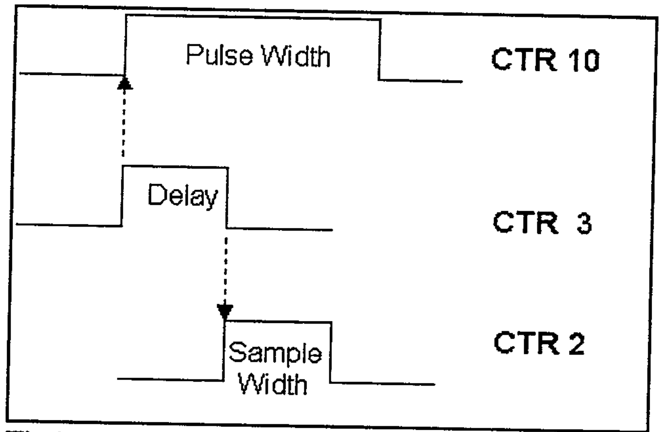
As has been mentioned, the system requires a geat deal of memory to run, and trying to run other software while several VIs are open is difficult. This includes running the Labview application under development itself while modifying deeply nested VIs. There is also a noticeable difference in speed between the Quadra connected to the DREA Ethernet and that which is not. Network card activity may place an unacceptable further drain on system resources.

12. Timing Control and Interfacing

The most critical component of each of the transducer calibration applications is the VI that controls the clocks in order to trigger the devices and the sampling of signals. For each of the software tasks, the use of the clocks within the system is presented below in a summary, logic diagram, and VI flow.

12.1 Calibration of Hydrophones Using Tone Burst or CW Signals

Counter CTR 3 on the TIO-10 board is software triggered, and remains high for the user specified delay. It triggers CTR 10 on its rise to start the Wavetek Function Generator, and triggers CTR 2 on its fall to perform sampling.



Timing for Task 1 (Pull Trigger VI)

The sequence of VI events to achieve this is as follows:

1. CTR_RESET sets output state to low for CTR 10 and CTR 2.
2. CTR_CONFIG sets CTR 10 gatemode to 3 (rising edge).
   CTR_CONFIG sets CTR 2 gatemode to 4 (falling edge).
3. CTR_PULSE sets CTR 10 pulsewidth = user Pulse Width.
   CTR_PULSE sets CTR 2 pulsewidth = Sample Window.
   Both CTR 2 and CTR 10 use delay of 3 and timebase of 2.
4. The timevalue is obtained.
5. CTR_PULSE sets CTR 3 pulsewidth = Sample Delay, delay of 3 and timebase of 2.
6. Wait for the default delay.

The necessary wiring is documented in the Users Guide.

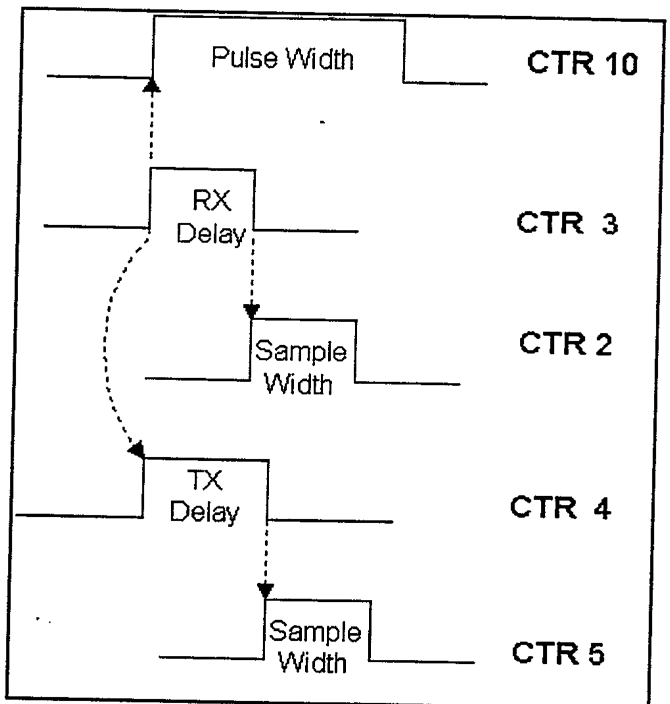
12.2 Calibration of Projectors Using Tone Burst or CW Signals

CTR 3 on the TIO-10 board is software triggered, and remains high for the user specified receive (RX) delay. It triggers CTR 10 on its rise to start the Wavetek Function Generator and the CTR 4 (Transmitting TX Delay), and triggers CTR 2 on its fall to perform sampling. The CTR 4 (TX Delay) triggers CTR 5 on its fall to perform TX sampling.

The sequence of VI events to achieve this is as follows:

1. CTR_CONFIG sets CTR 10 gatemode to 3 (rising edge).
   CTR_CONFIG sets CTR 2 gatemode to 4 (falling edge).
   CTR_CONFIG sets CTR 4 gatemode to 3 (rising edge).
   CTR_CONFIG sets CTR 5 gatemode to 4 (falling edge).
2. CTR_PULSE sets CTR 10 pulsewidth = user Pulse Width.
   CTR_PULSE sets CTR 2 and CTR 5 pulsewidth = Sample Window.
   CTR_PULSE sets CR 4 pulsewidth = TX Sample Delay
   All use delay of 3 and timebase of 1.
3. The timevalue is obtained.
4. CTR_PULSE sets CTR 3 pulsewidth = RX Sample Delay, delay of 3 and timebase of 1.
6. Wait for the default delay.

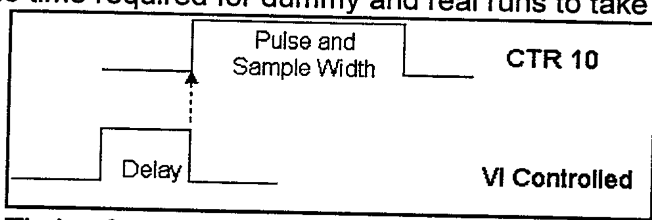
The necessary wiring is documented in the Users Guide.



Timing for Task 2 (Pull Trigger TD VI)

12.3 Calibration of Hydrophones Using Pseudo-Random Noise

The Wait VI is used to delay sampling as specified by the user specified delay. CTR 10 on the TIO-10 board is software triggered, and remains high for the user specified pulse width (the sum of the time required for dummy and real runs to take place).



Timing for Task 3 (Pull Trigger Noise)

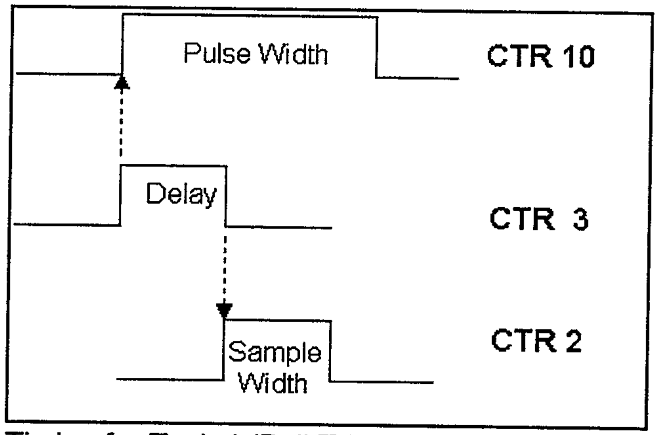
The sequence of VI events to achieve this is as follows:

1. CTR_RESET sets output state to low for CTR 10.
2. Wait for the user specified period.
3. CTR_CONFIG sets CTR 10 gatemode to 0 (no gating).
4. CTR_PULSE sets CTR 10 pulsewidth = (total samples /sampling frequency) using delay of 3 and timebase of 1.
5. The timevalue is obtained.
6. Wait for the default delay.

The necessary wiring is documented in the Users Guide.

12.4 Transducer Directivity Measurements

CTR 3 on the TIO-10 board is triggered by the shaft encoder, and remains high for the user specified delay. It triggers CTR 10 on its rise to start the Wavetek Function Generator, and triggers CTR 2 on its fall to perform sampling.



Timing for Task 4 (Pull Trigger VI)

The sequence of VI events to achieve this is as follows:

1. CTR_RESET sets output state to low for CTR 10 and CTR 2.
2. CTR_CONFIG sets CTR 10 gatemode to 3 (rising edge).
   CTR_CONFIG sets CTR 2 gatemode to 4 (falling edge).
3. CTR_PULSE sets CTR 10 pulsewidth = user Pulse Width.
   CTR_PULSE sets CTR 2 pulsewidth = Sample Window.
   Both CTR 2 and CTR 10 use delay of 3 and timebase of 2.
4. The timevalue is obtained.
5. CTR_PULSE sets CTR 3 pulsewidth = delay for the proportion of the time window to be achieved, delay of 3 and timebase of 2.
6. Wait for the default delay.

The necessary wiring is documented in the Users Guide.

The trigger from the shaft encoder must allow for the possibility of the shaft encoder reversing its direction of rotation. Furthermore, the lack of a "Data Ready Line" on the TIO-10 board means that the software must poll the signals from the shaft encoder to determine the presence of appropriate signals to trigger the sampling.

The logic of such a polling operation is as follows:

A counter, array index pointer, and flags to indicate that measured rotation has started, and the direction of intended rotation must be initialized. The shaft encoder lines (ZERO, CW, CCW) as well as a combined line representing the presence of any shaft encoder signal (the existing NOR line -- active low) must be wired to the Digital I/O lines and the hardware appropriately configured. The NOR line must be read in a loop (polled). When the NOR line indicates the presence of data, then if the measured rotation has not yet begun, the ZERO line must be polled until it goes high. Otherwise, if the measured rotation is underway (that is after the ZERO line has gone high), the CW and CCW lines must be polled. If a CW rotation is underway and the CW line goes high, then the counter must be incremented. If the counter is higher than the array pointer, then sampling should be triggered, and the array pointer incremented. If a CW rotation is underway and the CCW line goes high, then the counter should be decremented. The same activities (reversed) must be used for a CCW rotation. The procedure must be repeated until the rotation is complete and the total (3600) number of samples has been achieved.

The following represents pseudocode for the same logic:

1. **Initialize**:
```
COUNTER = 0
ARRAY_POINTER = 0
STARTED_ROTATION = FALSE
DIRECTION = CW (or CCW)
```

2. **Configure**:
DIG_LineConfig VI configures four input lines (ZERO, CW, CCW, NOR).

3. **Poll**:
```
while COUNTER < 3600
        DIG_In_Line VI reads NOR
        If STATE = TRUE then
                while STARTED_ROTATION = FALSE
                        DIG_In_Line VI reads ZERO
                        if STATE = TRUE then
                                STARTED ROTATION = TRUE
                endwhile
                if STARTED_ROTATION = TRUE
                        if DIRECTION = CW then
                                DIG_In_Line reads CW
                                if STATE = TRUE then
```

```
                        advance
                else DIG_In_Line reads CCW
                if STATE = TRUE then
                        retreat
        elseif DIRECTION = CCW then
                DIG_In_Line reads CCW
                if STATE = TRUE then
                        advance
                else DIG_In_Line reads CW
                if STATE = TRUE then
                        retreat
endwhile
```

Where the following operations are defined as sub VIs:

advance:

```
        increment COUNTER
        if COUNTER > ARRAY_POINTER
                increment array pointer
                trigger sampling
```

retreat:

```
        decrement COUNTER
```

12.5 Calibration of Transducers Using the Reciprocity Method

Clock control is the same as that of section 12.1 for the first two passes and the same as section 12.2 for the second two passes.

13. Equipment Under GPIB Remote Control

The program sequence strings transmitted via the GPIB interface is indicated below for each piece of equipment so controlled. All tasks use the following command strings for the Preamplifier and Brickwall filters:

RTS 3510B PreAmplifier:

?*c*KC ?*g*IA ?*mux* ?*imp*

Where *c* is the channel, *g* is the gain, *mux* is a switch that must be **M** if enabled and **E** if not, and *imp* is a switch that must be **B** for low and **J** for high.

Wavetek 753A Brickwall Filter:

CHF*highfrequency*G*gh*
CLF*lowfrequency*G*gl*

Where *highfrequency* and *lowfrequency* are represented as *nn*M*x* and *gh* and *gl* are the gain values to use. The frequency format represents values using *x* as a power of 10 multiplier (e.g. 20000 Hz is represented as 02M4).

The following are the specific strings used by each task to control the Wavetek 295 Waveform Generator.

13.1 Calibration of Hydrophones Using Tone Burst or CW Signals

Wavetek 295 Waveform Generator:

(Pulsed) :INIT:CONT OFF;:TRIG:GATE ON;:TRIG:SOURCE:START EXTERNAL;:SOURCE:FREQ:CW *freq*:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

(CW) :INIT:CONT ON;:SOURCE:FREQ:CW *freq*:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

Where *freq* is the frequency and *volt* is the voltage amplitude being output.

13.2 Calibration of Projectors Using Tone Burst or CW Signals

Wavetek 295 Waveform Generator:

(Pulsed) :INIT:CONT OFF;:TRIG:GATE ON;:TRIG:SOURCE:START EXTERNAL;:SOURCE:FREQ:CW *freq*;:SOURCE:FREQ:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

(CW) :INIT:CONT ON;:SOURCE:FREQ:CW *freq*:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

Where *freq* is the frequency and *volt* is the voltage amplitude being output.

13.3 Calibration of Hydrophones Using Pseudo-Random Noise

Wavetek 295 Waveform Generator:

(Pulsed) :INIT:CONT ON;:SOURCE:FREQ:CW *freq*;:INIT:CONT OFF;:TRIG:GATE ON;:TRIG:SOURCE:START EXTERNAL;:SOURCE:FUNC:SHAPE PRNOISE;:SOURCE:FREQ:MODE:CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

(CW) :INIT:CONT ON;:SOURCE:FREQ:CW *freq*;:SOURCE:FREQ:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

Where *freq* is the frequency and *volt* is the voltage amplitude being output.

13.4 Transducer Directivity Measurements

Wavetek 295 Waveform Generator:

(Pulsed) :INIT:CONT OFF;:TRIG:GATE ON;:TRIG:SOURCE:START EXTERNAL;:SOURCE:FREQ:CW *freq*:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

(CW) :INIT:CONT ON;:SOURCE:FREQ:CW *freq*:SOURCE:FREQ:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

Where *freq* is the frequency and *volt* is the voltage amplitude being output.

13.5 Calibration of Transducers Using the Reciprocity Method

Wavetek 295 Waveform Generator:

(Pulsed) :INIT:CONT OFF;:TRIG:GATE ON;:TRIG:SOURCE:START EXTERNAL;:SOURCE:FREQ:CW *freq*:MODE CW;:OUTP:STATE

ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

(CW) :INIT:CONT ON;:SOURCE:FREQ:CW *freq*:MODE CW;:OUTP:STATE ON;:SOURCE:VOLTAGE LEVEL:AMPL *volt*;

Where *freq* is the frequency and *volt* is the voltage amplitude being output.

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

MacLaren Plansearch (1991) Limited
#200 5657 Spring Garden Road
Halifax, N.S. B3J 3R4

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

unclassified

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Acoustic Transducer Calibration Software Programmers Guide In The Labview 3.1 Environment

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Pyra, James S.

5. DATE OF PUBLICATION (month and year of publication of in document)

March 1995

6a. NO OF PAGES (total containing information Include Annexes, Appendices, etc).

16

6b. NO. OF REFS (total cited document)

0

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address).

DREA, P.O. Box 1012, Dartmouth, N.S. B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).

6AH, DRDM-11:WU11

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

W7707-3-2635/01-OSC

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document).

MacLaren: 006938/DREA

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor).

DREA CR 95/410

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Software for the calibration of acoustic transducers has been developed in the Labview 3.1 programming system on Macintosh Quadra 900 computer. This document presents notes for the programmer who may be enhancing or maintaining the software, or who is undertaking additional development.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Underwater Acoustics
Transducers
Calibration
Software

153581