



DMOS – Bellhop Extension

*C. Calnan
xwave*

*xwave
36 Solutions Drive
Halifax, NS B3S 1N2*

Project Manager: J. Theriault, 902-426-3100 ext 376

Contract Number: W7707-05-2934

Contract Scientific Authority: J. Theriault, 902-426-3100 ext 376

xwave Contract Number: 1009782

Defence R&D Canada – Atlantic

Contract Report
DRDC Atlantic CR 2006-005
February 2006

This page intentionally left blank.

DMOS – Bellhop Extension

C. Calnan
xwave

xwave
36 Solutions Drive
Halifax, Nova Scotia
B3S 1N2

Project Manager: J. Theriault 902-426-3100 ext 376

Contract Number: W7707-05-2934

Contract Scientific Authority: J. Theriault 902-426-3100 ext 376

xwave Contract Number: 1009782

Defence R&D Canada - Atlantic

Contract Report
DRDC Atlantic CR 2006-005
February 2006

Colin Calnan

C. Calnan

Approved by

J. Theriault

J. Theriault
Contract Scientific Authority

Approved for release by

A. Reid

for K. Foster

Chair, Document Review Panel (DRP)

Terms of release: The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Terms of release: The information contained herein is proprietary to Her Majesty and is provided to the recipient on the understanding that it will be used for information and evaluation purposes only. Any commercial use including use for manufacture is prohibited. Release to third parties of this publication or information contained herein is prohibited without the prior written consent of Defence R&D Canada.

Abstract

The current contract contains requirements to enhance the sonar analysis suite *DMOS* to incorporate the use of the ray trace program **Bellhop**, and then to produce an updated version of the *DMOS* User's Guide.

The current document is a report on the project in general, describing the enhancement of *DMOS* and indicating changes made to existing members of that analysis suite necessitated by the addition of **Bellhop**. An updated User's Guide is provided in a separate document.

Résumé

Le présent contrat contient les exigences nécessaires pour améliorer le système *DMOS* de l'ensemble d'analyses sonar afin d'y inclure l'utilisation du programme de traçage de rayon **Bellhop** et de produire par la suite une version mise à jour du guide de l'utilisateur du *DMOS*.

Le présent document est un rapport sur le projet en général; il décrit l'amélioration de l'ensemble *DMOS* et indique les modifications apportées aux éléments actuels de cet ensemble d'analyses en raison de l'ajout du programme **Bellhop**. Un guide de l'utilisateur mis à jour est fourni à part.

This page intentionally left blank./Page laissée en blanc intentionnellement

Executive Summary

Introduction

The DRDC Atlantic Model Operating System (*DMOS*) is an evolution of the *SWAMI* (Shallow Water Active-sonar Modelling Initiative) suite of programs in use at DRDC Atlantic that enables a user to produce modelled reverberation, transmission loss, signal excess, and probability of detection for an active sonar.

Originally the suite was based on normal-mode theory, with one of *SWAMI*'s routines producing normal-mode data files used by later programs in the operational sequence. However, normal-mode theory is best suited to shallow water and low acoustic frequencies, and users occasionally need to model reverberation and other parameters under other conditions.

Results

This contracted effort was to enhance *DMOS* to include a Gaussian-beam acoustic propagation model. Though there had been some previous efforts in this direction, a focused effort was required to complete the task. This contract integrated the open-source model, BellHop, into *DMOS* as an alternate propagation engine. Rather than start with the publicly available version, a DRDC-extended version that included a range-dependant capability was chosen.

Significance

The completion of the model enhancements means that *DMOS* may now be used to model both active and passive sonars in shallow or deep water.

Future plans

The *DMOS* structure allows for an active source and receiver to be separated in space (known as bistatic active sonar). However, an initial effort to include bistatic, or multistatic, models had not progressed to completion. After more testing, *DMOS* will have the multistatic components extended and reintegrated into the suite of tools.

Calnan, C. 2006. *DMOS – Bellhop Extension*. DRDC Atlantic CR 2006-005. Defence R&D Canada – Atlantic.

Sommaire

Introduction

Le système d'exploitation de modèle de RDDC Atlantique (*DMOS*) est une évolution de l'ensemble de programmes *SWAMI* (pour Shallow Water Active-sonar Modelling Initiative, ou initiative de modélisation de sonar actif en eau peu profonde) en usage à RDDC Atlantique qui permet à l'utilisateur de produire un modèle de la réverbération, des pertes de transmission, de l'excédent de signal et de la probabilité de détection pour un sonar actif.

À l'origine, l'ensemble était basé sur la théorie du mode normal, une des routines *SWAMI* produisant des fichiers de données en mode normal utilisés par des programmes subséquents dans la séquence opérationnelle. Cependant, la théorie du mode normal convient le mieux à l'eau peu profonde et aux basses fréquences acoustiques, alors que, à l'occasion, les utilisateurs ont besoin de modéliser la réverbération et d'autres paramètres dans d'autres conditions.

Résultats

Ce travail à contrat avait pour but d'améliorer le système *DMOS* en y incluant un modèle de propagation acoustique de faisceau gaussien. Bien qu'il y ait eu quelques efforts antérieurs dans cette direction, il fallait un effort concentré pour achever la tâche. Ce contrat a intégré le modèle à source ouverte BellHop au système *DMOS* en tant que moteur de propagation de remplacement. Plutôt que de commencer avec la version disponible au public, on a choisi une version étendue par RDDC qui comprenait une fonction dépendant de la distance.

Portée

La réalisation des améliorations du modèle implique que le système *DMOS* peut désormais être utilisé pour modéliser tant des sonars actifs que des sonars passifs en eau profonde et en eau peu profonde.

Recherches futures

La structure du système *DMOS* prévoit une source active et un récepteur séparés physiquement (connu sous le nom de sonar actif bistatique). Cependant, un effort initial pour inclure des modèles bistatiques ou multistatiques n'avait pas connu son achèvement. Après d'autres essais, les éléments multistatiques du système *DMOS* seront étendus et réintégrés à l'ensemble d'outils.

Calnan, C. 2006. *DMOS – Bellhop Extension* (DMOS – Ajout du programme Bellhop). RDDC Atlantique CR 2006-005. R & D pour la défense Canada – Atlantique.

Table of Contents

Abstract.....	i
Résumé	i
Executive Summary.....	iii
Sommaire.....	iv
Table of Contents	v
List of Tables.....	vi
1. Introduction	1
2. The Evolution of <i>SWAMI</i>	2
3. Modified and New Routines.....	4
3.1 Bellhop Changes.....	4
3.2 BellhopDMOS.....	6
3.3 ref_tl	7
4. <i>DMOS</i> Programs Changed for Bellhop	8
5. File Locations	9
6. <i>DMOS</i> Eigenray File Format.....	10
7. Suggestions for Further Work	14
Bibliography	15
Initialisms and Acronyms.....	16
Distribution List.....	17

List of Tables

Table 1. <i>DMOS</i> Components.....	3
Table 2. Former and Current Bellhop Output.....	6
Table 3. GSM Eigenray File Header	10
Table 4. GSM Eigenray File Data Layout.....	11
Table 5. Bellhop Eigenray File Header.....	12
Table 6. Bellhop Eigenray File Data Layout	13

1. Introduction

The DRDC Atlantic Model Operating System (*DMOS*) is an evolution of the *SWAMI* (Shallow Water Active-sonar Modelling Initiative) suite of programs in use at DRDC Atlantic that enables a user to produce modelled reverberation, transmission loss, signal excess, and probability of detection for an active sonar.

Originally the suite was based on normal-mode theory, with one of *SWAMI*'s routines producing normal-mode data files used by later programs in the operational sequence. However, normal-mode theory is best suited to shallow water and low acoustic frequencies, and users occasionally need to model reverberation under other conditions.

Accordingly the current contract was issued in order to enhance the capabilities of *DMOS* by adding deep-water and high-frequency capabilities. There are a number of programs that can perform calculations for these conditions, and one of them **Bellhop**, was chosen for its abilities as a Gaussian-beam model.

The following section describes the evolution of *DMOS*, and includes information on how it developed and how the addition of **Bellhop** fits into the suite's developmental philosophy.

Some of the material in this report has been copied to [1], the *DMOS User's Guide*. This was done to allow someone reading the User's Guide to gain some extra information on *DMOS* that would be useful in using the suite.

Besides common English typographic conventions, the following conventions are used in this document:

- **bold text** is used for filenames (e.g. **test.pro** or **/local/files/test.pro**)
- **bold italics text** is used for directories (e.g. ***/usr/tmp***)
- *italics text* is used for computer and program suite names (e.g. *Tessie* and *SWAMI*)
- **Bold Arial text** is used to indicate program names (e.g. **Bellhop**)
- **Arial text** is used to indicate function and subroutine names (e.g. **PPR_SETUP**)
- **italic Arial text** is used for variable names in computer programs, the operating system, or an environment (e.g. ***IDL_PATH***)
- **Courier text** is used for text to be typed on the keyboard, code or text file lines, etc. (e.g. enter "idl")

2. The Evolution of *SWAMI*

Much of the information in this section was taken from references [2] and [3], which contain more detailed information on the history of *DMOS*, its components, and its operation at the time that those documents were written.

DMOS is a logical extension of the original *SWAMI* model. *SWAMI* was originally developed to model the performance of low-frequency active-sonar systems in shallow water. By that time many of the required components of such a model had been developed, but the framework to model the detection performance had not been constructed. The Shallow-Water Active-sonar Modelling Initiative (*SWAMI*) was intended to address the deficiency. The Initiative had two facets: a development project and a research project. The initial goal was to create the ability to predict the sonar performance (signal excess) for a monostatic low-frequency active sonar in a 3D-dependent shallow-water environment.

DMOS was devised along the lines of what Etter [4] calls a *Model Operating System*. This is a modular approach to developing and using a sonar model. The general design scheme is that the modelling process is divided into more-or-less independent sections of manageable size that are tied together with a Graphical User Interface (GUI). Because of this modularity the model is easily extendable.

Since its original development, and due to various reasons, the overall GUI wrapper for *SWAMI* has been dropped, returning it to a group of related programs. As a replacement for the GUI, shell scripts have been written to run the desired *DMOS* components. This technique is much more amenable to batch processing and program dispersal among different types of computer systems than is the use of GUIs, which typically require continual user input and tend to be machine or operating system dependant.

The modular approach has been taken advantage of a number of times. The current reverberation model **MONOGO** has replaced earlier models, and the current contract was issued to add ray-tracing capabilities to *DMOS* as a user-selectable alternative to its original normal-mode routine.

Instead of replacing the normal-mode model in *DMOS* by a different version, the current contract resulted in users being allowed to choose an alternate processing path. In this way the suite's previous capabilities were left untouched in case a user wishes to run the normal-mode analysis method should it be more appropriate to the case under study.

The following table lists the current components of *DMOS*; older components that have been replaced are not referenced. The first column in the table, **Order of Use**, lists the order in which the components are to be used. In general a given *DMOS* program requires files produced by earlier members as input.

Table 1. *DMOS* Components

Order of Use	Program Name	Function
1	INTEGRATE	Generates beam pattern and other data files to be used by later modules.
2	PMODES	Range-independent normal-mode transmission-loss program that also produces mode details used by later programs.
2	BellhopDMOS / Bellhop	<p>Bellhop is a range-dependent ray theory model that can produce transmission loss, ray trace, or arrival structure results and is an alternative to PMODES.</p> <p>Bellhop was developed independently of <i>DMOS</i>, so in order to help it fit into <i>DMOS</i>'s operational scheme the program BellhopDMOS was written.</p> <p>BellhopDMOS creates all the input files needed by Bellhop from other <i>DMOS</i> input files, runs Bellhop, and then converts Bellhop's output arrival file data into an eigenray file that can be read by later <i>DMOS</i> programs. In this document this pair of programs is simply referred to as Bellhop, the more important of the pair.</p>
3	MONOGO	Uses output from earlier programs to produce a reverberation time series file for a specified radial.
4	CMBRAD	Combines MONOGO 's reverberation output files to produce an azimuthally symmetrical reverberation file. If more than one reverberation file is used, when being combined they are weighted according to their spacing.
5	EXCESS1	Calculates transmission loss, signal excess, and echo level from CMBRAD output and either PMODES or Bellhop output.
6	PROBDET	Calculates probability of detection from signal excess files produced by EXCESS1 .
7	TSPREAD1	Calculates time-spreading using PMODES output.

As can be seen from column 1 in the above table, users have the option of using either **PMODES** or **Bellhop** depending on whether a user desires normal-mode or ray theory analyses. The only limitation put on **Bellhop** in *DMOS* is that a *DMOS* run made using **Bellhop** doesn't allow a user to run the time-spreading module **TSPREAD1**. This was not addressed at this time since there was no requirement for time-spreading analysis; however, this may be addressed in the future.

3. Modified and New Routines

The version of **Bellhop** currently being used at DRDC Atlantic was developed by Dr. Diana McCammon, who continues to work on the model. More specifically, the **Bellhop** version used for the *DMOS* expansion was named **BellhopDRDC_S** by Dr. McCammon and was generated from the code contained in the Zip file **BellhopDRDC - 20 July 2005.zip**.

Because Dr. McCammon is performing most of the original work on modifying **Bellhop**, the hope was that her code could be used without modification. Then as she released newer versions of the program, they could be easily slotted into *DMOS*. Some of her work on **Bellhop** is presented in [5] and [6].

Initially only a few, very minimal changes were made to **Bellhop**. These were necessitated by Dr. McCammon using a different Fortran 90 compiler than the one that is being used at DRDC Atlantic, and the necessity of changes of this sort was not completely unexpected. It is believed that Dr. McCammon is performing her **Bellhop** work on a Windows computer with a Microsoft Fortran 90 compiler, while the current contract work was done on the Linux computer *Tessie*. On this computer **Bellhop** and **BellhopDMOS** were compiled with the Portland Group's Fortran 90 compiler.

More changes to **Bellhop** were needed later, however, in order to impose consistency with the results produced by the **PMODES** operation of *DMOS*. These changes are described in the following section.

Once a more thorough testing of the **Bellhop**-enhanced *DMOS* is completed, the changes made to **Bellhop** during the expansion will be made known to Dr. McCammon so that she will be able to modify her baseline version of the code so as to contain these needed changes. In this way it is hoped that various "species" of **Bellhop** won't begin to evolve at DRDC. At that time, copies of the **Bellhop** code modified for this contract will also be sent to DRDC Atlantic personnel who use *DMOS* or **Bellhop**.

3.1 Bellhop Changes

As received from Dr. McCammon, **Bellhop** can produce four different ASCII output files, depending on the user's input choices. These input files are given the fixed names:

- Bellhop.log** - This file contains some basic data pertaining to a program run, including the frequency used, type of requested calculation, number of rays and their horizontal angular limits, and the distance step size. As well, any error messages or warning are written to this file.
- Arrival.txt** - This is a file containing "arrival" data for eigenrays reaching a

number of user-selected depth and range combinations, the theoretical locations of a number of receivers. The data are arranged in columns and consist of receiver depth, receiver range, acoustic pressure (a.k.a. amplitude), phase, elapsed time, source angle, receiver angle, number of reflections from the surface, and number of reflections from the bottom.

- Rays.txt** - This file contains the points that describe the tracings of the rays. The file's header contains a number of items that include the number of rays in the file. Following this information are the data for each ray, consisting of the ray's launch angle and the number of points in the trace, and these points as pairs of range and depth values.
- TL.txt** - This file contains transmission-loss results from the program following the receiver depths and ranges.

The first file was always produced, but only one of the other three was produced depending on the user's input choices.

As part of the testing procedure *DMOS* was made to use **Bellhop** arrival-time data to produce transmission-loss data. These values were compared to transmission-loss values produced by **Bellhop** with the same input data. The two sets of data were very different. It was eventually discovered that **Arrival.txt** contained pressure data calculated from ray trace parameters and modified them by adding Thorpe attenuation and cylindrical spreading.

However in calculating transmission loss, **Bellhop** started with these values, then (at least in the case of incoherent transmission loss) modified these values by taking into account Gaussian decay. These modified values were then used to calculate transmission loss. **Bellhop** didn't store either the Gaussian decay parameters or the data needed to calculate them in any of its output files, making it impossible for an external program to recreate the **Bellhop** transmission-loss results.

Bellhop was changed so that when transmission-loss output is requested the program also produces arrival data. However the pressure data put into the arrival data file consisted of the amplitudes as modified for the type of transmission loss that was calculated, i.e. coherent, incoherent, or semi-coherent.

Another change was in the naming of the output files depending on user requests. The following table compares the original and current output of **Bellhop**, depending on the specific type of results that were requested in the main **Bellhop** input file. The column header "Run Choice" contains the letter code for that row's option as well as a description of the code's meaning.

Table 2. Former and Current Bellhop Output

Run Choice	Former Action	Current Action
“A” - Arrival data	File Arrival.txt was created.	Same.
“C” - Coherent transmission-loss data	File TL.txt was created.	Files Arrival_CTL.txt and CTL.txt are created.
“S” - Semi-coherent transmission-loss data	File TL.txt was created.	Files Arrival_STL.txt and STL.txt are created.
“I” - Incoherent transmission-loss data	File TL.txt was created.	Files Arrival_ITL.txt and ITL.txt are created.
“R” - Ray trace data	File Rays.txt was created.	Same.
All options	File Bellhop.log was created.	Same.

As can be seen, the production of transmission-loss data now results in the production of an arrival-data file containing the pressure data used to produce the type of requested transmission loss. As well, both the output transmission loss and arrival-data files are given names that reflect the type of data they contain.

3.2 BellhopDMOS

Bellhop requires four ASCII input files:

- Runinput.inp** - Contains pulse frequency, source depth, receiver depths, receiver ranges, wind speed, desired output, some ray parameters, and a smoothing parameter for coherent transmission-loss output.
- Speed.inp** - Contains sound speed profiles.
- Bathy.inp** - Contains the bathymetry along the radial for which the program is being run.
- Bottomloss.inp** - Contains range-dependant bottom-loss parameters as an MGS province number.

Much of the data contained in the above files are also needed by various *DMOS* modules, although spread among different files and in different formats. One of the main differences between the *DMOS* modules and **Bellhop** is that the former perform a significant amount of parameter value validity, consistency checking, and array overflow checking, whereas the latter does none of this.

The author was tempted to add this input data checking to **Bellhop**, but didn't due to the desire to keep **Bellhop** changes to a minimum. However the lack of these data checks, combined with the need to ensure that both **Bellhop** and the other *DMOS* routines use the same data led to the decision to write a new program that could perform both of these tasks.

Accordingly, **BellhopDMOS** was designed and written. This program has one input file, **BD.inp**, which contains most of the data in **Runinput.inp** and the name of a **MONOGO** stdin input file. This new program was written under the assumption that the output of **Bellhop** would most often be used first by **MONOGO**, although it contains input parameters that allow the simple addition of other cases.

MONOGO's stdin input file contains the names of data files (either directly or within files named in it) that contain both the bathymetry and sound speed profile data needed for the current *DMOS* run. **BellhopDMOS** reads its own input file and the **MONOGO** input files, and then creates the input files needed by **Bellhop**, thus ensuring consistency for the run. It also ensures that none of **Bellhop's** fixed-size arrays will overflow.

The only data needed by **Bellhop** that are not directly part of *DMOS* are the bottom loss data, meaning that the only **Bellhop** file that must be created by a user is **Bottomloss.inp**.

After checking the validity of the data in **BD.inp** and creating the **Bellhop** input data files, **BellhopDMOS** then runs **Bellhop**. If a user requests that **Bellhop** be run for a number of different frequencies (as is not currently needed by *DMOS*) then **BellhopDMOS** will make the required runs.

3.3 ref_tl

All of **Bellhop's** transmission-loss files are ASCII files created in the same format, regardless of their contents. The files have a header that contains the number of ranges and depths, followed by the range values, the depth values, and finally all the transmission-loss values. Although this format does save space, it is not particularly easy to work with.

The program **ref_tl** (reformat transmission loss) was written to read in a **Bellhop** transmission-loss file's contents and rewrite it in a more user-friendly format. This new format consists of a number of groups of data, one per depth.

In a group for each depth there is a line that gives the depth associated with the group in the form: "Depth = 12.67 m". This line is followed by two columns of numbers: the range in km and the transmission loss in dB. This allows the transmission losses for a given depth to be easily extracted and compared to the output from the existing *DMOS* output files. In particular, the reformatted data could be easily put into **SAPLOT** input files to allow plotted comparisons.

When run, **ref_tl** asks for the names of the input and output files.

4. **DMOS Programs Changed for Bellhop**

Although **Bellhop** and **BellhopDMOS** are Fortran 90 programs, the remaining *DMOS* routines were written in FORTRAN 77 and were compiled with the GNU **f77** compiler.

Only two of the programs that were members of *DMOS* prior to the work performed under the current contract had to be modified in order to make use of **Bellhop** output; these were **MONOGO** and **EXCESS1**. Both programs required new code to read **Bellhop** eigenray files and process that data type. As well, both programs' primary input parameter files were changed so that they could indicate whether **PMODES** or **Bellhop** produced the input data. The programs were changed to be able to process the modified input file format.

The other programs weren't modified because:

- **INTEGRATE** is run before any other routine and is independent of all other *DMOS* routines.
- **PMODES** is run as an alternative to **Bellhop** and neither is dependant in any way on the workings of the other.
- **CMBRAD** combines individual radials' files produced by **MONOGO**.
- **PROBDET** uses signal excess files produced by **EXCESS1**.
- **TSPREAD1** did not have to be made compatible with **Bellhop** at this time.

5. File Locations

Users should be aware that the program **Bellhop** is actually called **BellhopDRDC_S**, and the name **Bellhop** is only used for convenience. However, since in the usual course of events **Bellhop** (or **BellhopDRDC_S**) is only run by **BellhopDMOS**, this misnomer should not affect processing in any way.

Using **BellhopDMOS** is the safest and easiest way to run **Bellhop** since **BellhopDMOS**:

- Performs all necessary input data checking.
- Creates most of the input files needed by **Bellhop**.
- Runs **Bellhop** as often as is required.
- Renames **Bellhop** output files appropriately.
- Converts **Bellhop** arrivals files into eigenray files useable by **MONOGO** and **EXCESS1**.

The executables of the *DMOS* programs mentioned in this Contractor Report are currently located in the directory *~calnan/projects/RevInv/dmos/bin* on the computer *Tessie*. However once sufficient testing has been performed and the author and Scientific Authority are satisfied that *DMOS* is working properly the executables will be moved to a subdirectory of */local/models/DMOS* on *Tessie*.

Instructions on how to run *DMOS* are located in [1].

6. *DMOS* Eigenray File Format

Page 3-4 of [7] describes the format of the Naval Underwater Systems Center's Generic Sonar Model (GSM) eigenray file, which contains eigenrays indexed by range and frequency.

The model **Bellhop** is able to produce eigenray data indexed by range, frequency, and depth. It was realized that it would be useful to define a binary file format similar to GSM's eigenray file in which to store **Bellhop's** eigenrays for use in *DMOS*. In fact, a successful file format could store frequency/range/depth indexed eigenrays regardless of their source.

The first format considered was GSM's. This format contains a header with data that relate to the eigenrays' calculation, followed by the values specific to the eigenrays.

The following two tables present the GSM eigenray file header and a description of the data stored for each eigenray.

Table 3. GSM Eigenray File Header

Bytes	FORTTRAN Data Type	Contents
1-6	CHARACTER*6	The string "EIGRAY", which indicates that the file contains eigenray data
7-10	REAL*4	range minimum (km)
11-14	REAL*4	range maximum (km)
15-18	REAL*4	range increment (km)
19-22	REAL*4	frequency minimum (Hz)
23-26	REAL*4	frequency maximum (Hz)
27-30	REAL*4	frequency increment (Hz)
31-34	REAL*4	frequency factor
35-38	REAL*4	source depth (km)
39-42	REAL*4	target depth (km)
43-46	REAL*4	bottom depth (km)
47-64	CHARACTER*18	not used
65-70	CHARACTER*6	"RANGE " if the eigenray file is sorted by range "FREQUE" if the eigenray file is sorted by frequency
71-76	CHARACTER*6	eigenray model name

The rest of the records in the file all contain 128 eigenrays worth of data. Each eigenray is described by nine words. The following table describes one eigenray's worth of data.

Table 4. GSM Eigenray File Data Layout

Word	FORTRAN Data Type	Contents	
1	INTEGER*4	<i>IRNG</i>	range index; <i>IRNG</i> < 1 indicates end of the file's data
2	INTEGER*4	<i>IFRQ</i>	<i>IFRQ</i> = frequency index; if <i>IFRQ</i> < 0 semi-coherent addition is being used
3	INTEGER*4	<i>NSRF</i>	<i>NSRF</i> = number of upper reversals; if <i>NSRF</i> > 0 → surface intersection if <i>NSRF</i> < 0 → upper vertex
4	INTEGER*4	<i>NBTM</i>	<i>NBTM</i> = number of lower reversals; if <i>NBTM</i> > 0 → bottom intersection if <i>NBTM</i> < 0 → lower vertex
5	REAL*4	<i>SRCANG</i>	source angle (rad)
6	REAL*4	<i>TRGANG</i>	target angle (rad)
7	REAL*4	<i>TIMEIG</i>	travel time (s)
8	REAL*4	<i>AMPEIG</i>	relative pressure amplitude referred to the range
9	REAL*4	<i>PHSEIG</i>	pressure phase (rad)

An examination of the above tables reveals that the format does not allow the environment to change with depth, something required for **Bellhop**. Regardless, an attempt was made to see if **Bellhop** output could be shoehorned into GSM's eigenray file format.

The conclusion: almost. Since GSM's eigenrays are range and frequency indexed and **Bellhop** runs are for a single frequency, the possibility of replacing frequency data with depth data (minimum, maximum, and step size) was considered. However it was decided that this scheme was unsatisfactory as it resulted in the loss of the frequency data, thus making the format unsuitable for models such as **WATTCH**. It would also present problems if a future expansion of *DMOS* were to make use of multiple frequencies.

An examination of the GSM eigenray file header did reveal a number of unused bytes that could be appropriated for the depth data. Unfortunately no low-hazard method could be devised to store the depth indexing in the space allocated for each GSM eigenray data. Reluctantly it was decided to change the format, only "breaking" it slightly. The changes involved making use of the unused bytes in the header without changing its size, and adding one new parameter to each eigenray's data.

The latter change would definitely cause breakage if any programs currently used to read GSM output were used on one of the new files. However, by specifying the file's type in its first six bytes with an identifier unknown to GSM, and with the assumption that all programs that read GSM output first verify the files' types by checking this identifier against valid identifiers, it was felt that the danger was minimized.

Accordingly, the following “FRDEIG” (frequency/range/depth eigenray) file format was devised, based on the GSM eigenray format. By keeping it as close as possible to the GSM format it was hoped that any routines that currently read GSM eigenray data files would only require minimal changes in order to read the new format.

The format is presented in the next two tables. The first table describes the header of the new format, which consists of 76 bytes of data like the GSM eigenray file header.

Table 5. Bellhop Eigenray File Header

Bytes	FORTRAN Data Type	Contents
1-6	CHARACTER*6	The string “FRDEIG”, which indicates f requency/ r ange/ d epth indexed e igenrays
7-10	REAL*4	range minimum (km)
11-14	REAL*4	range maximum (km)
15-18	REAL*4	range increment (km)
19-22	REAL*4	frequency minimum (Hz)
23-26	REAL*4	frequency maximum (Hz)
27-30	REAL*4	frequency increment (Hz)
31-34	REAL*4	frequency factor
35-38	REAL*4	source depth (km)
39-42	REAL*4	target depth (km)
43-46	REAL*4	maximum bottom depth (km)
47-50	REAL*4	depth minimum (km)
51-54	REAL*4	depth maximum (km)
55-58	REAL*4	depth increment (km)
59-64	CHARACTER*6	not used
65-70	CHARACTER*6	sorting order used to write the file (“RN”, “FR”, “DE” indicate, respectively, range, frequency, depth): “FRRNDE” – frequency, range, and depth; “DERNFR” – depth, range, and frequency; etc.
71-76	CHARACTER*6	eigenray model (“BELLHOP” for B ellhop data)

The rest of the records in the file all contain 128 eigenrays worth of data. Each eigenray is described by 10 words. The following table describes one eigenray’s worth of data.

Table 6. Bellhop Eigenray File Data Layout

Word	FORTRAN Data Type	Contents	
1	INTEGER*4	<i>IDEP</i>	depth index; <i>IDEP</i> < 1 indicates end of the file's data
2	INTEGER*4	<i>IRNG</i>	range index; <i>IRNG</i> < 1 indicates that this eigenray's data are not to be used
3	INTEGER*4	<i>IFRQ</i>	frequency index
4	INTEGER*4	<i>NSRF</i>	number of upper reversals
5	INTEGER*4	<i>NBTM</i>	number of lower reversals
6	REAL*4	<i>SRCANG</i>	source angle (rad)
7	REAL*4	<i>TRGANG</i>	target angle (rad)
8	REAL*4	<i>TIMEIG</i>	travel time (s)
9	REAL*4	<i>AMPEIG</i>	relative pressure amplitude referred to the range
10	REAL*4	<i>PHSEIG</i>	pressure phase (rad)

The differences between the **Bellhop** and GSM formats are:

- header bytes 47-58 are used for **Bellhop**,
- header bytes 65-70 have different sorting strings,
- **Bellhop** eigenray data word 1, *IDEP*, is new,
- **Bellhop** eigenray data word 2, *IRNG*, is allowed to indicate that the record is not to be used, and
- the values for *IFRQ*, *NSRF*, and *NBTM* only contain positive integers for **Bellhop** since that program doesn't note the differentiations that GSM does.

7. Suggestions for Further Work

The only *DMOS* module that hasn't been modified to use **Bellhop** eigenray data is **TSPREAD1**, which requires **PMODES** normal-mode data. To complete the expansion of *DMOS* with regards to eigenray input, **TSPREAD1** could be modified to also use eigenray data and calculate time spreading from these input files.

Bibliography

- [1] Theriault, J.A. and Calnan, C. (2006). *DMOS User's Guide*, in preparation.
- [2] Theriault, J.A. (1996). *SWAMI: Shallow Water Active-Sonar Modelling Initiative Nx2D Sonar System Performance Model*, Informal User's Guide, December 11, 1996.
- [3] Theriault, J.A. (1995). [.../dmos/docs/DMOS.html](#) – this file is located on several computers
- [4] Etter, P.C. (1991). *Underwater Acoustic Modeling, Principles, Techniques and Applications*, Elsevier Science Publishers, Essex, U.K.
- [5] McCammon, D.F. (2004). *Investigating the Bellhop Acoustic Prediction Model*, DRDC Atlantic CR 2004-285, in review.
- [6] McCammon, D.F. (2005). *Active Acoustics using Bellhop-DRDC*, DRDC Atlantic CR 2005-087, May 2005.
- [7] Weinberg, H. (1985). *Generic Sonar Model*, NUSC Technical Document 5971D, 6 June 1985.

Initialisms and Acronyms

ASCII	American Standard Code for Information Interchange
DMOS	DRDC Atlantic Model Operating System
DRDC	Defence Research and Development Canada
GSM	Generic Sonar Model
SWAMI	Shallow-Water Active-sonar Modelling Initiative

Distribution List

Internal Distribution

- 2 DRDC Atlantic LIBRARY FILE COPIES
- 3 DRDC Atlantic LIBRARY (SPARES)

- 3 J. Theriault
- 1 S. Pecknold
- 1 J. Osler
- 1 D. Ellis
- 1 P. Hines
- 1 M. LeFrancois
- 1 B. Roger
- 1 G. Mellema

External Distribution

- 1 C. Calnan
xwave
36 Solutions Drive
Halifax, NS B3S 1N2
- 1 D. McGammon
McCammon Acoustical Consulting
475 Baseline Road
Waterville, NS B0P 1V0
- 1 LCdr W. Renault,
SSO METOC,
MARLANT
CFB Halifax
- 1 M. Prior,
NATO Undersea Research Centre
Anti-Submarine Warfare Department
Viale San Bartolomeo 400
I-19138 La Spezia
- 1 NDHQ/ CRAD/ DRDKIM 3

Total 20 copies

This page intentionally left blank.

UNCLASSIFIED

DOCUMENT CONTROL DATA (Security classification of the 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. Centre sponsoring a contractor's document, or tasking agency, are entered in section 8.) Publishing: DRDC Atlantic Performing: xwave, 36 Solutions, Drive, Halifax, NS Monitoring: Contracting:		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 is indicated by the appropriate abbreviation (S, C, R, or U) in parenthesis at the end of the title) DMOS – Bellhop Extension (U)		
4. AUTHORS (First name, middle initial and last name. If military, show rank, e.g. Maj. John E. Doe.) Colin Calnan		
5. DATE OF PUBLICATION (Month and year of publication of document.) February 2006	6a NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.) 32	6b. NO. OF REFS (Total cited in document.) 7
7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.) Contract Report		
8. SPONSORING ACTIVITY (The names of the department project office or laboratory sponsoring the research and development – include address.) Sponsoring: Tasking:		
9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant under which the document was written. Please specify whether project or grant.) 11cs	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.) W7707-05-2934	
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)	10b. OTHER DOCUMENT NO(s). (Any other numbers under which may be assigned this document either by the originator or by the sponsor.) DRDC Atlantic CR 2006-005	
11. DOCUMENT AVAILABILITY (Any limitations on the dissemination of the document, other than those imposed by security classification.) Unlimited distribution		
12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, when further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.)) Unlimited announcement		

UNCLASSIFIED

UNCLASSIFIED

DOCUMENT CONTROL DATA

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

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.)

(U) The current contract contains requirements to enhance the sonar analysis suite DMOS to incorporate the use of the ray trace program Bellhop, and then to produce an updated version of the DMOS User's Guide.

The current document is a report on the project in general, describing the enhancement of DMOS and indicating changes made to existing members of that analysis suite necessitated by the addition of Bellhop. An updated User's Guide is provided in a separate document.

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 is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

(U) Active sonar,
Model,
Reverberation,
Transmission Loss,
Sonar Performance

UNCLASSIFIED

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale



www.drdc-rddc.gc.ca