Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

ODIN – AN UNDERWATER WARFARE SIMULATION ENVIRONMENT

Terence Robinson

QinetiQ
Offshore & Acoustics Department
Bincleaves,
Weymouth, Dorset, U.K.

ABSTRACT

This paper describes the capability, design and application of the generic underwater warfare simulation environment ODIN. The model was developed by QinetiQ, previously DERA (Defence Evaluation and Research Agency), to model the detailed underwater interaction between surface ship/submarine/UUV (Unmanned Underwater Vehicle) platforms, torpedoes and countermeasures. It was originally developed out of a need to model the effectiveness of advanced countermeasure concepts and uses innovative techniques to model multi-static signal acoustics. The environment provides a 'whole system' integrated approach to modelling using multiple levels of fidelity to support a wide range of applications, from high-level Monte Carlo assessment to algorithmic design and evaluation.

1 INTRODUCTION

To cope with the demands of the modern Navy, underwater weapons and countermeasures must become increasingly sophisticated. The operational emphasis has shifted from deep water to the more demanding littoral environment, and the threat has become more diverse. In order to determine future requirements for underwater warfare systems, and to design and evaluate those systems, an ability to model advanced concepts and novel techniques is essential. This paper describes the new underwater warfare simulation environment, ODIN, which has been designed to fulfil this need.

ODIN has been developed on behalf of the U.K. MoD(N) by the Offshore & Acoustics Department of QinetiQ at Bincleaves to provide a 'whole system' integrated approach to underwater modelling. It is designed to support a range of applications and customers whose requirements include:

- Support to the decision making process for future equipment procurements
- Balance of Investment studies to underpin and guide torpedo and countermeasure research
- Design and evaluation of torpedo & countermeasure systems
- Performance assessment of future torpedo and countermeasure concepts
- Threat assessment
- Tactical development.

Development of ODIN began in January 1996 with the construction of a generic framework. Whilst its primary focus lies with underwater warfare, the framework could in principle be used to model other scenarios such as above water warfare, traffic flow, or other situations involving interaction between multiple bodies using multiple sensors. A distinguishing feature of the model is its ability to correctly handle multi-static signal acoustics.

2 TECHNICAL CAPABILITY

ODIN is a generic simulation tool of flexible modular design. It was designed to simulate the detailed interaction between underwater objects, whilst retaining a sufficiently fast running speed to enable multiple engagement Monte Carlo effectiveness studies to be performed. It offers an advanced capability beyond other U.K. models, such as THOR, which are rapidly being superseded.

At the outset, ODIN was designed to model the interaction between multiple underwater bodies without artificial limitations, such as the constraint of having to define entities as 'targets' or 'attackers', which is common in other similar models. Consequently, it is ideally suited to interactive assessments between opposing forces. Applications include:

- Submarine & surface ship torpedo defence studies
- Torpedo system performance studies
- Advanced countermeasure signal design

- Design & evaluation of torpedo homing & guidance algorithms
- Anti-torpedo torpedo modelling.

Using a network of software 'objects', of varying fidelities, ODIN is able to model a 'complete' torpedo engagement e.g. from initial contact between two opposing submarine platforms through weapon launch, search and target acquisition to actual physical hit. High fidelity algorithms are used to model the homing trajectory of the torpedo to determine the point of impact upon the hull. By combining the impact point with a built-in representation of platform vulnerability, overall weapon effectiveness is determined. The use of consistent algorithms and data through each stage of the engagement, ensures an integrated approach to performance assessment. Previous techniques that were used to assess 'complete' performance, suffered from a 'piecemeal' approach requiring several different models to be run independently and later combined off-line. Figure 1 shows the position of ODIN within the U.K. models hierarchy; a brief description of each model can be found in the Appendix.

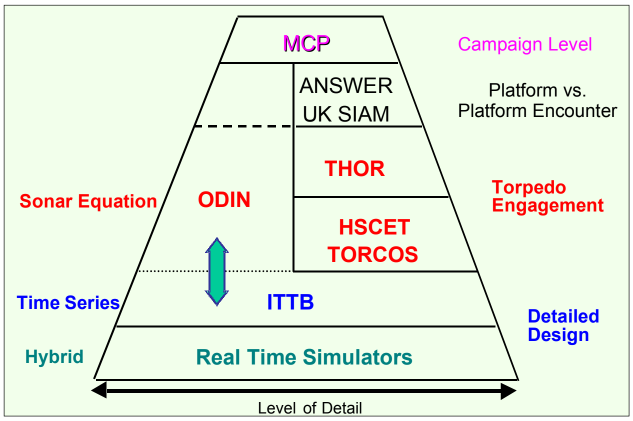


Figure 1: Hierarchy of Models

The base of the 'triangle' corresponds to high fidelity detailed design tools, that use time-series or hybrid (hardware-in-the-loop) techniques. At the apex, sit the high-level tools: MCP (Maritime Campaign Model) and ANSWER & SIAM platform encounter models. ODIN resides in the middle ground, providing a broad flexible capability. The design is primarily focussed towards torpedo engagement modelling, but the framework could equally be populated with 'higher-level' objects to model platform vs. platform encounters. This potential for growth is indicated by the dashed line in Figure 1.

A key feature of the model is its ability to link to more detailed models to enhance fidelity as needed. For example, via the link to the ITTB (Integrated Torpedo TestBed), ODIN can be used to stimulate and evaluate advanced torpedo homing & countermeasure algorithms / techniques that have been developed using in-water data, thus allowing 'real' code to be exercised within an operational context.

One of ODIN's key strengths is that the software applications, which populate the framework, are developed and run by the U.K. torpedo and countermeasure technology groups. This integrated approach promotes synergy across the teams in QinetiQ and provides an underpinning capability to the U.K. research programme.

3 MODEL DESIGN

The design takes advantage of the benefits afforded by Object-Oriented methodology. The model comprises a set of interconnected software modules or 'objects' each performing a specific function within the simulation. The design comprises two parts - the central core or 'framework' and derived applications. The framework provides the user with a generic simulation environment, in which detailed applications relating to underwater warfare systems can be constructed.

3.1 Framework

The framework provides a means of controlling & synchronising events and passing information between model entities. It is both flexible and modular allowing new object and environmental behaviours to be developed in an efficient manner. The UML (Unified Modelling Language) class design is shown, in simplified form, in Figure 2. The advantages of Object-Oriented (OO) design include code re-use, via inheritance, and greater flexibility and robustness through the self-contained nature of each object.

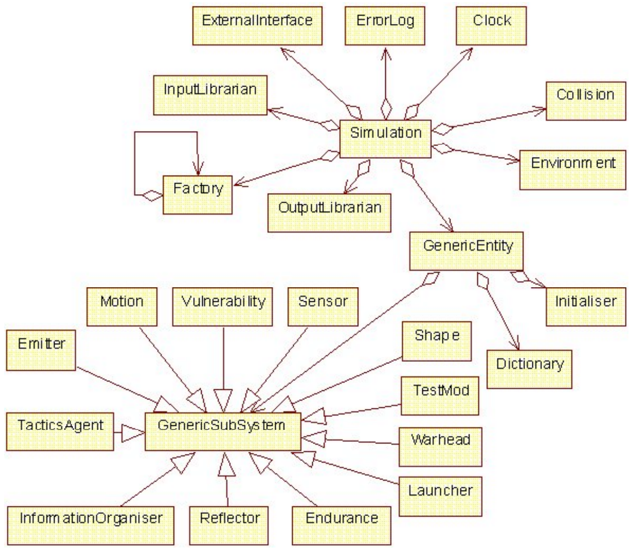


Figure 2: ODIN Framework

Each object encapsulates all data and algorithms associated with its function and it communicates with other objects through real world processes, e.g. sonar. The framework makes no assumptions about the behaviour of any object, but allows the objects to remotely interact solely through their sensors and tactics. For example, the model has no built-in concept of attacker or target.

As the objects are self-contained, adding or removing objects within the simulation is straightforward. This flexibility allows the object set to be re-configured at runtime to meet the desired application requirements. There is no limit to the number of objects within the simulation e.g. numbers of platforms within a task force, or torpedoes fired within an engagement. Provided that the interface between objects is not altered, the method of encapsulation allows objects to be modified independently, without affecting others within the model. This degree of isolation between software modules improves the robustness of future developments.

3.2 Application Development

From within the framework, the user is able to construct real-world entities, such as submarines or torpedoes, using the set of generic sub-systems shown in Figure 2. As illustrated in Figure 3, a submarine may be created as a generic entity that has shape, motion & endurance; can reflect sonar transmissions and emit radiated noise; can detect and transmit (sonar) signals using a sensor; and can launch other objects such as torpedoes and countermeasures. The command and control function within the entity is provided by the tactics agent. The data dictionary provides a means of storing data for each entity, which is accessible by each sub-system.

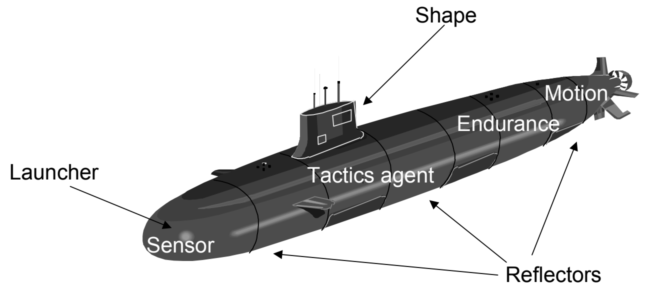


Figure 3: Submarine Entity

The sub-systems may be ‘inherited' directly from the framework, or enhanced to create additional functionality. For example, the basic hull-mounted sonar sensor, shown above, may be modified to model a one dimensional line array of any given length which can be physically offset astern of the vessel and connected via a tow cable. By giving the array a motion sub-system, that models the hydrodynamics of the tow, a towed array can be created.

In addition to the more familiar objects, the user may construct less conventional entities; for example, an iceberg may be created by building an entity that has a shape, can reflect sonar transmissions, and has some elementary drift motion. In a similar fashion, a set of ‘reflective’ objects may be spatially positioned within the environment to recreate false contacts that are often present at the output of an active sonar system. Via this innovative approach the construction of new entities becomes intuitive.

3.3 External interfaces

ODIN has two types of external interface to allow the user to connect to other models:

- Software ‘sockets’, which can be used from anywhere in ODIN and are effective across different computer platforms; the ITTB, for example, is connected via sockets
- A Distributed Interactive Simulation (DIS) interface, which has been used to link to the “Virtual Ship” developed by the Future Systems Technology Group in QinetiQ. A prototype system was successfully demonstrated at IMDEX 1997, where the platform launched a “virtual” torpedo generated by ODIN.

Future design enhancements are planned to make the system HLA (High Level Architecture) compliant to improve model portability.

4 MODEL DESCRIPTION

This section describes the key features of ODIN.

4.1 Simulation engine

Execution and control of ODIN is carried by the Simulation class which exists at the highest level within the model. Its tasks include the extraction of data from input files, model execution, termination of the simulation and destruction of objects and production of the model output. Its position within the framework is shown in Figure 2; its functional role is described in the following paragraphs.

The Simulation class instigates the generation of the physical entities e.g. ships, submarines, torpedoes, countermeasures, etc., which are constructed using the generic sub-systems. During model execution, the Simulation class performs three main functions:

- Updating the status and position of each entity via associated time and event stepping
- Control of the direct or indirect interaction between entities by passing messages between entities (e.g. acoustic signals)
- Detection of collisions between entities, using data provided by the Shape & Motion objects of each entity.

With reference to the first function, the Simulation class deals in two types of event. A state event is defined as a request to all physical entities to update their position, velocity, acceleration and orientation within the local topographic simulation framework. State events are requested by the Simulation object whenever the elapsed simulation time exceeds a user defined threshold. Time events are details of internal time related activities that a physical entity wishes to perform, e.g. a course change in 10 seconds time. Time events are sorted and processed in a 'next-nearest' chronological order by the Simulation object. Time is calculated using the Clock object. The increment time-step is taken to be the smaller of the user defined time step, or, the difference between the current time and the time to the next scheduled time event.

4.2 Kinematics

There are currently three levels of fidelity available to model the motion of an entity, each of increasing complexity. Simple motion, as the name implies, allows a body to move in a straightforward manner in straight lines or arcs of circles. Complex motion is suitable for modelling the dynamic behaviour of submarine and surface ships and has been successfully compared against platform trials and more detailed simulations run by the hydrodynamics group within QinetiQ. An accurate model of dynamic behaviour, using such parameters as system latency, acceleration and turn rate, is essential to correctly determine the likelihood of being able to outrun an attacking torpedo. Body motion is a six-degrees-of freedom model suitable for modelling torpedo dynamics, where the system is controlled via a highly responsive feedback system acting on demands from the on-board autopilot computer. This level of fidelity is particularly relevant to highly dynamic scenarios such as those encountered by the anti-torpedo torpedo, where, for example, changes to the steer vector of sonar beam that are encountered during a rapid turn, need to be modelled accurately. By inheriting from the base Motion class - or any of the above – the developer may create their own specialist motion model.

4.3 Acoustic Signal Modelling

This section describes how ODIN handles the acoustic interaction between underwater bodies, both in terms the capability of the model as well as its implementation as acoustic ‘messages’ between entities.

To recap, an entity may be a submarine, a torpedo, a countermeasure, etc. which has physical size and shape and which may have a number acoustic sensors (sonar), a number of acoustic reflectors and a number of radiated noise emitters distributed along its hull. One of the innovative features of ODIN is its ability to correctly handle multi-static acoustics within the underwater environment, using multiple sonars and reflectors. Consider, for example, the scenario whereby an acoustic homing torpedo is using its active sonar to echo-range a surface ship target, as illustrated in Figure 4. In this instance, the ship has been assigned a set of four reflectors positioned along the hull to model its multiple highlight active signature. The nearby surface ships are given a simplified two reflectors signature, but are not within the acoustic beam width of the torpedo and hence receive the torpedo’s transmission at reduced level. When the torpedo sonar transmits, all entities that possess an active signature will return echoes with correct time delay and position of origin. In this scenario, the torpedo will receive direct echoes back from each platform. However, since the modelling of signals is generic, ODIN also represents reflections of the active ping from each (red) reflector, to all other receivers and reflectors in the engagement (i.e. the generalised bi-static case). Hence, these platforms will also receive indirect returns via the bi-static paths from neighbouring surface vessels, and the torpedo sonar could in principle receive returns with any number of reflections from one upwards The proliferation of multiple bounce echoes can be suppressed by a software switch in the model.

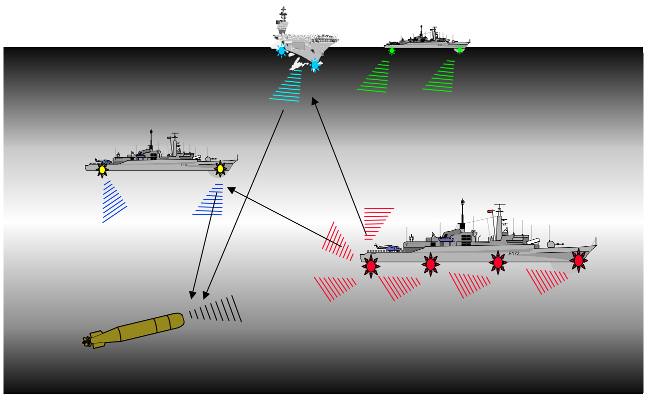


Figure 4: Acoustic Signal Modelling

Following reception of a signal, the weapon is equipped with detection, tracking, association and classification algorithms to select the true target from the background, which includes self and ambient noise plus reverberation components. False contacts may be easily introduced as a cluster of objects each having a reflective signature. This feature allows torpedo tracking and terminal homing algorithms to be developed and assessed within ODIN.

Within the ‘base’ model, signal excess and hence detection performance is modelled using the Detection Threshold (DT) term within the familiar Sonar Equations. However, via the link to the ITTB the detection process can be modelled using a suite of time series algorithms allowing the user extended fidelity. Any type of sonar signal can be represented in ODIN, but the user must specify the behaviour of the signal processor within the receiver.

In software terms, the sonar transmission is treated as a instance of the generic signal (or message) class, which is transmitted between entities via the event scheduling process within the Simulation object. At the required time of transmission, the simulation object broadcasts the signal message into the environment to all entities within the simulation that possess a sonar receiver or reflectors. The message contains the characteristics of the signal i.e. frequency (Doppler Shift corrected), signal bandwidth, transmission type, etc. plus a history of the signal's passage through the environment enabling any receiver to calculate the returned signal level. As the message class is generic, there is no technical reason why the signal should not be extended to model radar transmissions. Hence ODIN has the potential to model above water warfare scenarios, although some of the timing issues that are important in water are insignificant with the much faster speed of propagation of radar.

4.4 Shape modelling

To model collisions between entities, for example to model torpedo impact against a submarine target, each entity is given a shape. In this respect, ODIN is extremely flexible allowing the user to create almost limitless designs using a concept known as sphere trees. The technique allows the construction of a 3-D shape using any number of spheres of different sizes, located at user-defined positions relative to the centre of the entity. There is no limit to the complexity of the object in terms of the numbers of spheres, sphere radii or sphere co-ordinates, which can all be varied to control the accuracy of the design. Figure 5 reveals how a complex submarine shape can be constructed from around 180 spheres. The outer circle represents an outer bounding sphere, which is defined for all entities and is used to improve the computational efficiency of collision detection.

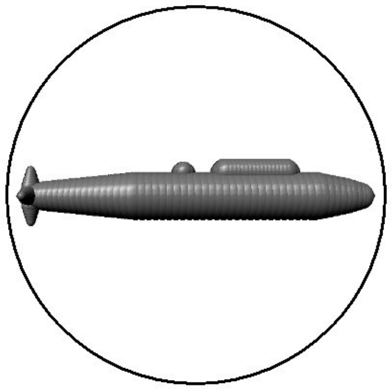


Figure 5: Submarine Shape with Bounding Sphere

Collision detection occurs in two stages. Throughout an engagement the Simulation object monitors the positions of all entities – only the Simulation object has access to the positions of all entities. This 'low' fidelity mode continues until the bounding spheres of two entities overlap. This triggers the 'high' fidelity mode where the simulation update rate increases, taking account of the relative closure speed of each entity, to compute the inter-distances between spheres within respective shape models. Following a collision, impact messages are created and broadcast within the simulation, and either body may be damaged or explode if a warhead object and associated fuze have been implemented.

4.5 Tactical language

ODIN has a High Level Language (HLL) that allows the user to specify the Command and Control function of a platform, or the weapon's tactical control logic. HLL provides a rapid, flexible tool for tactical development and is ideally suited to prototyping. The language is interpretative and hence 'tactics' can be modified and tested without recompilation. The language comprises a series of 'English-like' statements consisting of tactical actions and circumstances, which are grouped into phases akin to subroutines. At run-time, the statements are converted into a series of numerical codes that control object behaviour. Once proven, any tactical phase may be coded directly into C++. Alternatively, the user may choose to code detailed tactical algorithms using C++ at the outset. To improve coding efficiency, user-defined multiple call functions may be written; these are termed 'command aids'. An example of an HLL module is shown below.

```
IF (PING_COUNTER EQ 150)
    SET PING_COUNTER TO 0
        IF (LEG EQ 0)
            SET LEG TO 1
            CHANGE_COURSE_PORT 179.0 DEGREES
        ELSE
            SET LEG TO 0
            CHANGE_COURSE_STBD 179.0 DEGREES
        END_IF
    ELSE
        SET PING_COUNTER TO (PING_COUNTER + 1)
END_IF
END
```

4.6 Acoustic environment

ODIN has an acoustic environment, which handles the transmission and reception of signals between entities. The environment calculates the propagation loss between transmitter and receivers, boundary layer reverberation from surface and seabed, plus interference from background noise sources. At its present stage of development, propagation loss is calculated using a simple spherical spreading law, with absorption coefficient specified as a function of frequency. All acoustic rays are assumed to travel in straight lines. Over the next 3 years, QinetiQ plan to develop a realistic shallow water acoustic environment; this will model important features such as curved ray paths and depth/range dependent propagation.

An important feature within ODIN is the ability to model ship wakes. The bubbly wake is modelled as a series of rectangular sections 'deployed' sequentially behind the ship into the environment. Each section has defined dimensions and 'persistence' permitting the age and shape of the wake to be defined as a function of ship type and speed. Acoustic signals passing through the wake objects suffer both reflection and attenuation. Via this technique the effect of a wake on the acoustic signature of a surface platform can be modelled.

5 ODIN SOFTWARE SUITE

The ODIN Software Suite is shown in Figure 6.

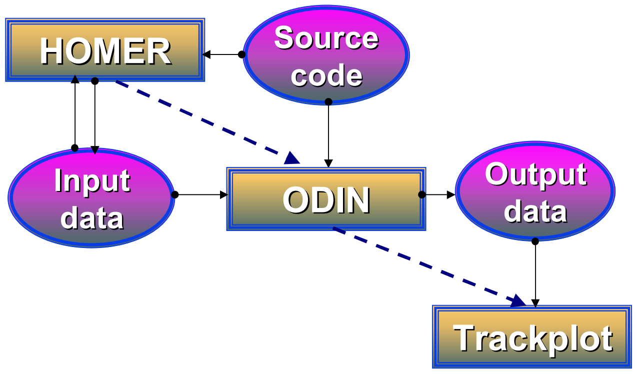


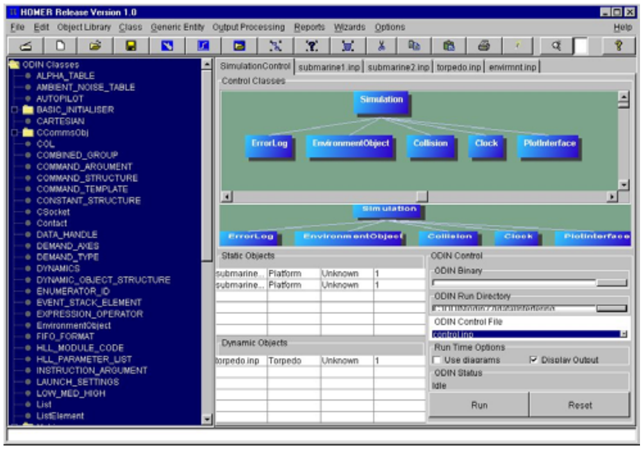
Figure 6: ODIN Software Suite

ODIN is coded in C++ and runs on either a SUN Sparc workstation under Solaris, or on a PC, under Windows NT or LINUX. The user interface is via a Graphical User Interface (GUI) written in Java and known as HOMER. Output can be visualised either at run time or post run, using a Java plotting package, TRACKPLOT.

To supplement HOMER and the on-line documentation, an ODIN user guide and training package is also available.

5.1 HOMER

HOMER, the Hierarchical ODIN Modelling EnviRonment, is a front-end GUI which is used to prepare and edit ODIN input files, run the ODIN simulation model and view the on-line software documentation & source code. HOMER has the advantage that it requires no manual configuration - it scans the source files and automatically determines the current class structure within ODIN. Figure 7 shows a screen shot of the user interface.



Static Objects

submarine...	Platform	Unknown	1
submarine...	Platform	Unknown	1

Dynamic Objects

torpedo.inp	Torpedo	Unknown	1

Figure 7: HOMER Screenshot

HOMER provides the user with a visualisation of the ODIN class structure, as defined by the source code, and the ODIN data structure, as defined by the input file configuration. The GUI includes the following functionality:

- An edit function, which allows the user to cut, copy and paste objects to modify or add new objects to the configuration
- Input data, such as sonar beam width can be visualised and modified in a user-friendly manner.
- The shape of an entity can visualised and rotated in a 3-D form
- An HLL tactics library is provided to ensure key tactical words are entered using the correct syntax and structure.

Future enhancements are expected to included an object repository, or database, to facilitate storage of software objects and record key assumptions and data used within studies, together with a dedicated package for analysis of Monte Carlo output. Monte Carlo analysis is currently undertaken using the Microsoft Excel programme.

6 APPLICATIONS

This section illustrates example applications for ODIN using screen shots provided by the TRACKPLOT analysis tool.

6.1 Torpedo Engagement

Figure 8 illustrates how ODIN may be used to study the effectiveness of a salvo of two heavyweight torpedoes against a submarine target.

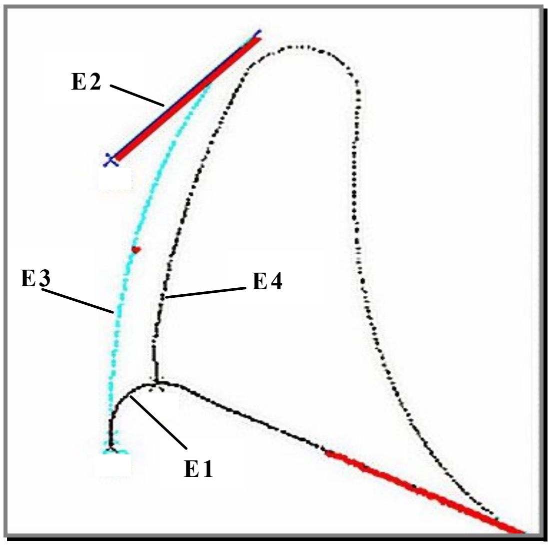


Figure 8: Torpedo Salvo

At time 0, submarine E1 launches the first torpedo (E3) against submarine E2. In this scenario, it is assumed that E2 does not detect the incoming torpedo, hence cannot take evasive action. Having launched the torpedo, E1 turns to starboard and after a delay of one minute launches a second weapon (E4). Both torpedoes are modelled using different sonar frequencies to avoid mutual interference. The engagement proceeds and the first weapon hits the target and destroys E2; in this case 100% lethality is assumed and the entity is removed from the simulation. The removal of E1, immediately causes the second torpedo to lose sonar contact causing it to turn to search for a new contact, and as a consequence acquires the launch platform (E1) which it proceeds to hit.

In reality, of course, a safety box would be imposed around the launch platform to avoid the weapon turning to attack friendly forces, however, the example has been chosen to illustrate the generic nature of ODIN ie. that in the 'eyes' of the torpedo the 'attacker' and 'target' are treated identically. It should be noted that the safety box could, of course, be implemented in ODIN.

As an illustration of the versatility of ODIN, the effects of mutual interference (between torpedoes) may easily be investigated by simply aligning their respective sonar frequencies.

6.2 Countermeasure Studies

Figure 9 illustrates how countermeasure effectiveness may be studied.

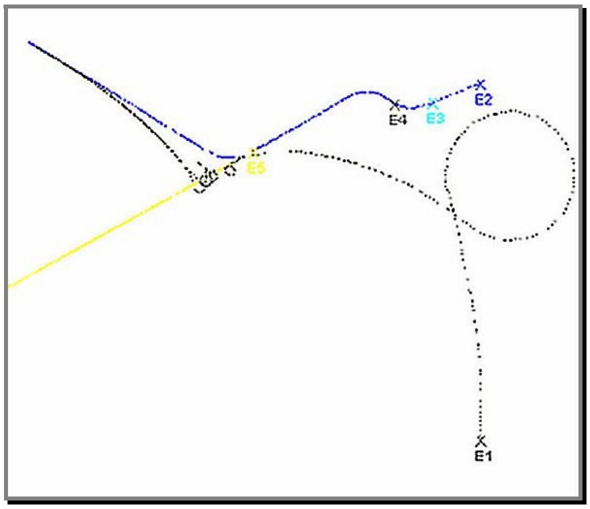


Figure 9: Countermeasure Studies

In response to the attacking torpedo (E1), the submarine (E2) launches a series of static and mobile decoy countermeasures, and executes a turning manoeuvre. The first pair of decoys force the weapon to lose sonar contact, and the torpedo executes a circular search to try to re-acquire the target. Despite being lured towards, and attacking the final (yellow) mobile decoy, the weapon overruns the position of the decoy, detects the target and proceeds to hit. By varying system parameters such as: decoy performance, number of decoys deployed, as well as the characteristics and timing of the evasion manoeuvre, ODIN may be used to optimise naval tactics to maximise platform survivability.

7 DEVELOPMENT STATUS

Within QinetiQ, ODIN is being actively developed by both torpedo and countermeasure research groups under MoD funding. The current emphasis is to support the future U.K. Spearfish and torpedo defence programmes as well as to pursue novel countermeasure and torpedo techniques.

To ensure the model is 'fit for purpose', each new model development includes a package of work to validate the representations created. The validation process includes comparison with actual torpedo firings where possible. The validation evidence is summarised and collated in the ODIN Validation Logbook which is reviewed and updated as necessary to align with the needs of the U.K. research & procurement programmes. ODIN was recently used to support the Business Case for the U.K. Surface Ship Torpedo Defence (SSTD) procurement programme.

To improve development and gain wider acceptance of the model, QinetiQ are actively seeking to create an ODIN User Group, where users can either license the model, or, contribute to the model development by providing additional software modules. The OO design of ODIN, with its 'core' framework and applications, makes the model ideally suited to multi-user operation and development; each user runs an identical framework, but develops / runs customised applications to suit national requirements. Within this context, the model has attracted considerable interest within the U.K. and world-wide, and has been the subject of international collaborative discussions with the U.S. and the Netherlands (N.L.).

The model is currently in use with the U.K. Maritime Warfare Centre and the U.S. Naval Undersea Warfare Center and it is anticipated that ODIN will form part of the future joint U.K. / N.L. torpedo defence testbed, which is currently being pursued by QinetiQ with TNO of the Netherlands.

8 POINT OF CONTACT

Dilys Grant, is the project manager for ODIN and point of contact for all technical enquiries. She is responsible for the development and maintenance of the ODIN framework, and the co-ordination of all related application developments. Her email address is <dgrant@QinetiQ.com>.

ACKNOWLEDGMENTS

The author would like to thank all staff who continue to contribute to the development of ODIN and hence this paper, with particular reference to Philip Bardswell and Dilys Grant of QinetiQ.

APPENDIX: U.K. MODELS

1. MCP: Maritime Campaign Program, used for analysis of maritime warfare at the campaign / theatre level. It is a symmetric two-sided model.
2. ANSWER: ANti-Submarine-Warfare-Realisation model, used to model the effectiveness of antisubmarine warfare within a multi-platform scenario. It is essentially a (few) submarine platform vs. (many) surface ship platform model.
3. (U.K.) SIAM: Submarine Interactive Attack Model, used to model submarine vs. submarine engagement. It is an interactive one-on-one model and includes a lower fidelity (cf. THOR) torpedo model.
4. ODIN: Generic underwater warfare simulation environment, the subject of this paper.
5. THOR: Generic torpedo engagement model used to assess torpedo and countermeasure effectiveness; a previous generation model to ODIN.
6. TORCOS: Torpedo engagement model, originally developed to model U.K. Sting Ray torpedo. It is of higher fidelity than THOR, but less capable than ODIN.
7. HSCET: High Speed Concept Evaluation Tool, used to model the effectiveness of an anti-torpedo torpedo.
8. ITTB: A suite of software modules comprising sonar beamformers, signal processors, target trackers etc. which may be interconnected to model the component parts of a homing system within a modern torpedo. The model is used to develop homing algorithms and may be stimulated by synthetic or trials data.

REFERENCE

ODIN User Guide, Model version 2.2 incorporating HOMER version 1 & TRACKPLOT version 3, Offshore & Acoustics Department, QinetiQ.

AUTHOR BIOGRAPHY

TERENCE ROBINSON is the Technical Leader for Underwater Torpedo and Countermeasure Studies at QinetiQ, Bincleaves. He received his B.Sc. from the University of Birmingham in 1974 and joined the Admiralty Underwater Warfare Establishment (a predecessor of QinetiQ) in the same year. His research experience has included the practical study of target and environmental acoustics relating to torpedo performance, and the design of torpedo signal processing systems. His email address is <trobinson@QinetiQ.com>.