Copy No. _____

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

A Recursive Engagement Simulation Tree (REST) For Use in Maritime Defence

Brian Mury
Bao Nguyen

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2006-096
October 2007

Canada

This page intentionally left blank.

A Recursive Engagement Simulation Tree (REST) For Use in Maritime Defence

Brian Mury and Bao Nguyen

Defence R&D Canada - Atlantic

Technical Memorandum

DRDC Atlantic TM 2006-096

October 2007

Authors

Original signed by Dr. Bao Nguyen

SLt Brian Mury and Dr. Bao Nguyen

Approved by

Original signed by Paul Saunders

Mr. Paul Saunders

Acting Maritime OR Section Head

Approved for release by

Original signed by J.L. Kennedy

Kirk Foster

DRP Chair

Abstract

In support of the Integrated Signature Management (ISM) program at DRDC Atlantic, the Operational Research group is developing a model to assess the benefits of signature reduction of a defending ship. Within this mandate and as a first step, we have built a stochastic simulation to determine the Measures Of Effectiveness (MOEs) of a warfare scenario where a ship defends against multiple threats with multiple defensive systems. The simulation is designed with the aim of comparing the MOEs of a Canadian Patrol Frigate (CPF) as a function of distinct signatures. This simulation simulates the outcomes of a battle and requires inputs such as the probability of detection, the probability of track initiation, the probability of seduction and the probability of a hit/miss etc. The underlying framework of the simulation can be used to analyze torpedo defence, mine defence, maritime air defence as well as soft kill and hard kill capabilities. In addition, it can assess the trade off in terms of vulnerability of the defence between the reduction of one of its signatures and the rise of another. This report describes the implementation of the underlying algorithms which make use of the concept of recursivity and the notion of a decision tree. To demonstrate the applicability of this simulation, we provide examples where a ship defends against multiple threats with multiple interceptors.

Résumé

Appuyant le programme de gestion intégrée des signatures (*Integrated signature management* ou *ISM*) à RDDC Atlantique, le groupe de recherche opérationnelle est à mettre au point un modèle pour évaluer les avantages que présente la réduction des signatures d’un navire à la défensive. Nous avons réalisé dans le cadre de ce mandat une simulation stochastique visant à déterminer les critères d’efficacité d’un scénario de guerre où un navire se défend contre des menaces multiples avec des systèmes défensifs multiples. La simulation est conçue avec l’idée de considérer les critères d’efficacité d’une frégate canadienne de patrouille (FCP) comme fonction de signatures distinctes. Elle peut servir à analyser les capacités de défense par torpilles, de protection anti-mines, de défense aérienne des opérations maritimes ainsi que les capacités de neutralisation par déroutement et de destruction. De plus, elle peut évaluer, compte tenu de la faiblesse de la défense, le choix entre réduire une signature et augmenter une autre. Le présent rapport décrit la mise en œuvre des algorithmes sous-jacents qui utilisent le concept de récursivité et la notion d’un arbre de décision. Pour démontrer les possibilités d’application de cette simulation, nous fournissons des exemples où le navire se défend contre des menaces multiples avec des intercepteurs multiples.

This page intentionally left blank.

Executive summary

Introduction

The ISM (Integrated Signature Management) group is a recent addition to DRDC Atlantic that looks for ways to reduce the signatures of a naval platform to minimize its susceptibility. Typically, there are four types of signatures: the radar signature, the infrared (IR) signature, the acoustic signature and the electromagnetic (EM) signature. Each of these has distinct characteristics. Associated with each signature are threats that exploit them. Anti-ship missiles usually carry radar or IR seekers. Torpedoes exploit acoustic signatures. EM or acoustic signatures can trigger mines. In general, the higher the signature strength is, the more vulnerable the ship is.

Results and Significance

In support of the ISM group, the Operational Research group at DRDC Atlantic is developing a model to assess the benefits of signature reduction of a Canadian Patrol Frigate (CPF). Within this mandate and as a first step, we have built a stochastic simulation which simulates the outcomes of a warfare scenario. This simulation is designed in a way such that it provides a natural means to compare the Measures of Effectiveness (MOEs) among the signatures. The MOEs include the probability of neutralizing all threats and the expected number of defensive weapons expended as a function of signature type and strength. The simulation also records the engagement timeline and models tactics. For example, a simple tactic can be one in which the defence engages a threat with soft kill and hard kill weapons until the threat is neutralized, or the defence runs out of time or resources. We demonstrate the applicability of the simulation by examining scenarios where ships defend themselves against multiple missile threats with multiple interceptors. To do that, we provide a derivation of optimal intercept time and course in three dimensions for an interceptor with a non zero kill radius, engaging a missile attack.

Future plans

As mentioned earlier, this simulation is part of a larger objective to assess the impact on vulnerability of a CPF as a function of signature type and strength. This simulation requires inputs such as the probability of detection, the probability of track initiation, the probability of seduction and the probability of a hit/miss etc. In the future, we will investigate each signature, research the associated threats and model defensive systems to determine each probability, and combine them through this engagement simulation to evaluate the integrated effectiveness of a CPF.

Mury, B and Nguyen, B. U. 2007. A recursive engagement simulation tree for use in maritime defence. DRDC Atlantic TM 2006-096. Defence R&D Canada - Atlantic.

Sommaire

Introduction

Le groupe de la gestion intégrée des signatures (*Integrated signature management* ou *ISM*) est une récente addition à RDDC Atlantique. Il cherche des moyens de minimiser la susceptibilité des signatures d'une plate-forme navale. Quatre types de signatures existent généralement : la signature radar, la signature infrarouge (IR), la signature acoustique et la signature électromagnétique (EM). Chaque signature présente des caractéristiques distinctes et est associée à des menaces qui l'exploitent. Les missiles antinavires sont habituellement dotés d'autodirecteurs soit radar soit IR. Les torpilles exploitent les signatures acoustiques. Les signatures EM ou acoustiques peuvent déclencher les mines. En général, la vulnérabilité d'un navire est proportionnelle à la force de ses signatures.

Résultats et portée

Appuyant le programme de gestion intégrée des signatures du groupe ISM, le groupe de recherche opérationnelle à RDDC Atlantique est à mettre au point un modèle pour évaluer les avantages que présente la réduction des signatures d'une frégate canadienne de patrouille (FCP). Nous avons réalisé dans le cadre de ce mandat une simulation stochastique visant à déterminer les résultats d'un scénario de guerre. Cette simulation est conçue de manière qu'elle fournit un moyen naturel de comparer entre eux les critères d'efficacité relatifs aux signatures. Ces critères comprennent la probabilité de neutraliser toutes les menaces et le nombre prévu d'armes défensives utilisées en fonction du type et de la force d'une signature. La simulation enregistre également le temps d'engagement et modélise les tactiques. Par exemple, une simple tactique est celle où la défense engage une menace avec des armes de neutralisation par déroutement ou des armes de destruction jusqu'à ce que la menace soit neutralisée ou que le temps et les ressources de la défense s'épuisent. Nous démontrons les possibilités d'application de la simulation en examinant des scénarios où un navire se défend contre des missiles multiples avec des intercepteurs multiples. À cet effet, nous fournissons en trois dimensions une dérivation du temps et de la trajectoire d'interception optimaux pour un intercepteur ayant un rayon de destruction non nul, engageant une attaque au missile.

Recherches futures

Comme mentionné plus haut, cette simulation fait partie d'un objectif plus grand, qui consiste à évaluer l'effet sur la vulnérabilité d'une FPC en fonction du type et de la force d'une signature. Dans l'avenir, nous examinerons chaque signature, étudierons les menaces connexes, modéliserons des systèmes de défense et les rassembleront à l'aide de la présente simulation d'engagement, afin de doter une FPC d'une efficacité intégrée.

Mury, B. and Nguyen, B. U.. 2007. A recursive engagement simulation tree for use in maritime defence. DRDC Atlantic TM 2006-096. Defence R&D Canada - Atlantic.

Table of contents

List of figures

List of tables

Acknowledgements

The authors would like to thank Dr. Dan Hutt and Dr. Zahir Daya for reviewing the manuscript.

1. Introduction

The ISM (Integrated Signature Management) group is a recent addition to DRDC Atlantic that looks for ways to reduce the signatures of a naval platform to minimize its susceptibility. Typically, there are four types of signatures: the radar signature, the infrared (IR) signature, the acoustic signature and the electromagnetic (EM) signature. Each of these has distinct frequency bands and characteristics. Associated with each signature are threats that the enemy can employ to exploit them. Anti-ship missiles usually carry radar or IR seekers. Torpedoes exploit acoustic signatures. EM or acoustic signatures can trigger mines. In general, the higher the signature strength is, the more vulnerable the ship is.

In support of ISM, the Operational Research group at DRDC Atlantic is developing a model to assess the benefits of signature reduction of a Canadian Patrol Frigate (CPF), Ref [1]. Within this mandate, we have built a stochastic simulation to determine the outcomes of a defending ship engaging a multiple missile attack. This simulation provides a natural means to compare the Measures of Effectiveness (MOEs) as a function of signature type and strength. The MOEs include the probability of neutralizing all threats and the closest point of approach of the threats. Our simulation models both the engagement timeline and tactics. An engagement timeline typically records detection, classification, identification, engagement and kill assessment events. A simple tactic can be the Shoot-Look-Shoot tactic. Once a threat is identified, the defence will engage that threat possibly with a surface-to-air missile. If the interceptor defeats the threat then the engagement is over. However, if the defence misses the threat then it will re-engage provided there is enough time and inventory. The defensive weapons can be hard kill, soft kill or a combination of both.

To the best of our knowledge, there are at least three types of engagement model. The first type of engagement model is a Monte Carlo simulation. The second type of engagement model is based on the use of a generating function such as the one in Ref [4]. This may be the most efficient in terms of computational time. The third type of engagement model is the decision tree, Refs [5-6]. This is definitely the most generic model.

A Monte Carlo simulation is perhaps the most commonly used technique. It makes use of random numbers to determine the outcomes of a battle. For example, a possible outcome when the defence engages two threat missiles is a hit and a miss i.e. threat number 1 is neutralized while threat number 2 is not neutralized. The simulation is executed several times for a same scenario and collects the statistics e.g. those favourable to the defence and those favourable to the threats, Refs [2-3]. At the end of the day, the statistics collected can be converted into approximate MOEs.

A generating function generally makes use of polynomials, e.g. a binomial series, where each term of a polynomial corresponds to a possible outcome. Additionally, the generating polynomial groups equivalent outcomes together e.g. the outcome where threat number 1 is hit and threat number 2 is miss may be equivalent to that where threat number 1 is miss and threat number 2 is hit. By doing so, the generating function only needs to perform calculations in terms of groups instead of individual outcomes which is the case in a Monte Carlo simulation. Moreover, since a group collects equivalent outcomes, this implies that the number of groups is less than the number of outcomes. Therefore, the use of a generating function can be, and often is, substantially more efficient than other techniques including Monte Carlo. Note, however, that this efficiency is due to the assumption that there are equivalent outcomes. There are times when this assumption is not valid. For instance, if threat number 1 is a different type from threat number 2 then the two events in the example above are not equivalent because the probability of a hit against threat number 1 is not necessarily the same as the one against threat number 2.

As described above, the Monte Carlo technique makes use of statistics to provide approximate MOEs while the generating function technique relies on simplifying assumptions. We have found that the decision tree algorithm is a natural remedy to these drawbacks for two reasons. First, a decision tree always yields the exact MOEs as it generates all possible outcomes during a battle even if it could be costly in computational time. However, in this study, only simple scenarios are considered. Hence, computational time is not a concern. Second, it is generic since it makes no assumptions about inputs and outputs. For one thing, a decision tree does not assume that threat number 1 is identical to threat number 2. Hence, it is naturally a good candidate to study the impacts of distinct signatures.

Essential to engagement models are the probabilities of possible outcomes. They may include the probability of detection, the probability of track initiation, the probability of seduction and the probability of a hit/miss etc. Each of these may depend on the type and strength of signature. Their determination is currently investigated case by case and must be completed in order to demonstrate the utility of integrated signature management.

The general scenario that we consider is that of a ship or multiple ships defending themselves against a number of threats with a limited inventory of interceptors or weapons as shown in Figure 1.

Section 2 describes the concept of a decision tree. Section 3 portrays a model of missiles including results for an intercept of a missile threat by a missile interceptor at a nonzero kill radius in the shortest time. Section 4 illustrates the decision tree as an engagement model. Section 5 is a brief description of a hard kill model while Section 6 provides examples of inputs and outputs of the Recursive Engagement Simulation Tree (REST). Section 7 provides further examples based on soft kill weapons and mine threats. Finally, Section 8 presents a discussion and conclusions.

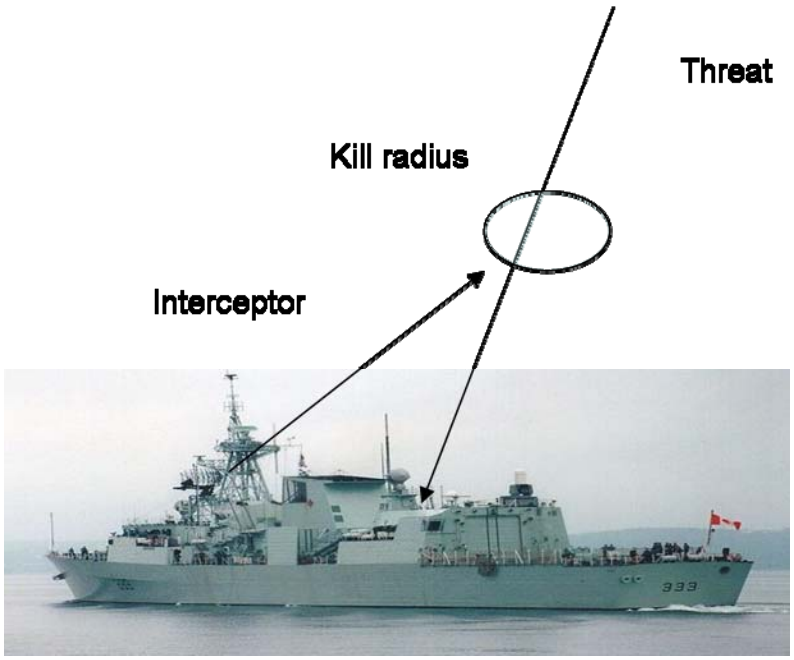

Figure 1. General scenario.

2. Decision Tree

The main idea of a decision tree is to generate all the possible outcomes of a scenario. We do this by time stepping the simulation from the beginning of an attack until the end of that attack. Each time a significant event occurs such as the detection or the non-detection of a threat, it is recorded in the tree as well as the corresponding probability of that occurrence. If the tree is correctly modelled then each possible outcome of the attack corresponds to a branch of the tree.

In terms of computer programming languages, a decision tree is a data structure. Each event is a node and each outcome is represented by a branch. The probability of an outcome is thus the product of all the probabilities associated with the nodes forming that branch. This makes it possible to determine the probability of each possible outcome which in turn is useful in determining the MOEs. For example, the probability of neutralizing all threats is the sum of the probabilities of all branches whose outcomes indicate that each threat was successfully defeated. For testing purposes, note that the sum of the probabilities of all the branches must be one since a tree generates all possibilities.

The decision tree used in this report is a binary tree. That is, each node or event has exactly two possible outcomes. One favours the defence, while the other favours the offence. The algorithm for building a binary tree is well known and can be found in many textbooks such as Ref [5]. Essentially, when a new event occurs, the algorithm traverses the tree and inserts a corresponding node at the appropriate locations in the tree. We implement the decision tree by defining two classes: node type and tree type.

The node type class contains only data. It encapsulates the data that is specific to an event and has no knowledge of the tree. The data includes the time of the occurrence of an event such as a detection event or an engagement event, labels for threats and defensive weapons, defensive weapon's salvo size, and the probabilities of a hit and miss.

The tree type class encapsulates the structure of the tree. It is designed as a recursive data structure. That is, each sub-tree is a tree. Due to the recursivity of the tree, we have implemented the algorithm in C++ as it is an object oriented computer language and hence a suitable choice. The data of a tree includes the nodes as defined above, the left sub-tree (favouring the defence) and the right sub-tree (favouring the offence), the inventory of defensive weapons, the list of the threats and whether they are alive or killed.

3. Missile Model

3.1 Data Structure

For programming purposes, we define three classes: missile, threat and interceptor classes.

The missile class is the base class for the threat and interceptor classes. Given an initial position and a velocity vector, it tracks the current position of the missile as a function of time.

The threat class is derived from the missile class. It allows the threats to be defined with an initial position, a velocity vector, and a launch time. The threat class also records the launch and impact locations of the threat missile as well as their corresponding times.

The interceptor class is also derived from the missile class. The interceptor class will determine the velocity vector required to achieve the minimal intercept time based on these parameters: launch position of the interceptor, its speed, its weapon range; the threat's trajectory and its velocity. The interceptor class also records the launch and impact parameters of the interceptor.

3.2 Optimal Intercept at a Nonzero Kill Radius

In this high level study, we assume that missiles travel in straight lines. To model interceptors and threats, we make use of the formulas for minimal intercept time and corresponding velocity vector derived from Annex A and B respectively. These formulas are obtained based on a scenario where an interceptor is launched toward a threat missile with the aim to explode at a nonzero kill radius from the threat missile. The results obtained in Annex A and B are useful because they have close form solutions yet they embed a non trivial kill radius. In addition, the weapon range of the threat missile will affect the miss distance in a critical way. That is, the threat missile may not explode right at the defending ship but it still may damage the ship if its weapon range is non zero.

Annex A provides the following formula for minimal intercept time at a non zero kill radius:

$$T_I = \frac{r}{v_T}\left(\frac{-b+\sqrt{b^2-4\cdot a\cdot c}}{2\cdot a}\right) \tag{1}$$

where

$$a = \frac{v_M^2}{v_T^2} - 1$$
$$b = 2 \cdot \hat{r} \cdot \widehat{v_T} + 2 \cdot \frac{v_M}{v_T} \cdot \frac{r_W}{r}$$
$$c = \left(\frac{r_W}{r}\right)^2 - 1 \tag{2}$$
$$\vec{r} = \overrightarrow{r_M^0} - \overrightarrow{r_T^0} - \overrightarrow{v_T} \cdot t'$$
$$\hat{r} = \vec{r} / r$$
$$\widehat{v_T} = \overrightarrow{v_T} / v_T$$

v_M is the missile's speed; $\overrightarrow{r_M^0}$ is its position at time $t = t'$; v_T is the target's speed; $\overrightarrow{v_T}$ is its velocity vector; $\overrightarrow{r_T^0}$ is its position at time $t = 0$ and r_W is the kill radius.

Annex B provides the following formula for the velocity vector of the missile that is associated with the minimal intercept time above:

$$\overrightarrow{v_M} = \frac{\left(\overrightarrow{r_T^0} - \overrightarrow{r_M^0} + \overrightarrow{v_T} \cdot T_I\right) \cdot v_M}{\left(r_W + v_M \cdot \left(T_I - t'\right)\right)} \tag{3}$$

To illustrate the effect of a kill radius on intercept time, Table 1 shows intercept time $\left(T_0\right)$ based on a zero kill radius, intercept time $\left(T_I\right)$ as a function of kill radius, the difference of the two previous intercept times in magnitude $\left(\Delta T\right)$ and in percentage $\left(\Delta T\ \%\right)$. Note that time is measured in seconds, and kill radius ranges from zero m to 50 m in increments of 10 m. The interceptor travels at $140\ m/s$ while the threat travels at $200\ m/s$. We perform this calculation for three scenarios. In the first scenario the threat is located at $\left(100\ m, 0\ m, 0\ m\right)$, $\left(200\ m, 0\ m, 0\ m\right)$ in the second scenario and $(1000\ m, 0\ m, 0\ m)$ in the third scenario. In each scenario, the defending ship is located at the origin while the interceptor launch site is located at $(0\ m, 100\ m, 0\ m)$.

Table 1. Intercept time as a function of kill radius

RANGE	100 M				200 M				1000 M			
Kill radius	T_0	T_I	ΔT	ΔT %	T_0	T_I	ΔT	ΔT %	T_0	T_I	ΔT	ΔT %
0 m	∞	∞	∞	100	0.78	0.78	0.00	0.00	2.98	2.98	0.00	0.00
10 m	∞	∞	∞	100	0.78	0.74	0.04	5.65	2.98	2.95	0.03	1.02
20 m	∞	∞	∞	100	0.78	0.69	0.09	11.01	2.98	2.92	0.06	2.05
30 m	∞	0.50	∞	100	0.78	0.65	0.13	16.15	2.98	2.89	0.09	3.11
40 m	∞	0.43	∞	100	0.78	0.62	0.16	21.12	2.98	2.86	0.12	4.19
50 m	∞	0.38	∞	100	0.78	0.58	0.20	25.95	2.98	2.83	0.15	5.29

In scenario 1, the interceptor with a kill radius less than or equal to 20 m can not intercept the threat because the speed of the interceptor is less than that of the threat. For convenience, the corresponding intercept times are set to an infinite value meaning that the interceptor will never intercept the threat in these cases. Hence, the corresponding ΔT and $\Delta T\ \%$ are not well defined. The infinite intercept times are shown in red and the associated ΔT and $\Delta T\ \%$ are shown in yellow. The ones shown in green imply that intercept is possible. In general, Table 1 shows that the improvement in intercept time is relatively small. This improvement decreases as a function of initial separation between a threat and a defending ship.

As an illustration, here is an example detailing on how to obtain the results in scenario 1, shown in Table 1, when the initial threat location is $\overline{r_T^0} = (200\ m, 0\ m, 0\ m)$, and the kill radius of the interceptor is $20\ m$.

Recall that the interceptor's speed is $v_M = 140\ m/s$, and the threat's speed is $v_T = 200\ m/s$. As the threat aims at the ship (located at the origin), the threat's velocity vector is $\overrightarrow{v_T} = (-200, 0, 0)\ m/s$. In addition, the initial location of the interceptor is $\overrightarrow{r_I^0} = (0, 100, 0)\ m$, and so the initial distance between the threat and the interceptor is $r = |(200, 0, 0) - (0, 100, 0)|\ m = 223.607\ m$. Assuming a time delay $t' = 0\ s$, we obtain the following results from Equation 1 and 2:

$$a = -0.51$$
$$b = 1.914$$
$$c = -0.992$$
$$T_I = 0.694\ s$$

Repeating the same calculation with $r_w = 0\ m$ gives:

$$a = -0.51$$
$$b = 1.789$$
$$c = -1$$
$$T_0 = 0.78\ s$$

The values for T_I and T_0 are displayed in the third row (kill radius = $20\ m$) of scenario 2 $(range = 200\ m)$.

3.3 Limiting Angles of Approach

The formula for minimal intercept time in Section 3.2 contains a square root expression in the numerator. If intercept is possible then the intercept time must be real. Hence, the argument of the square root must be non negative. However, Refs [7-8] reveal that this argument can be negative if the speed of the interceptor is lower than that of the threat. As a result, to intercept a threat with a lower speed interceptor, the interceptor's launch location must be in such a way that the argument of the square root is non negative. This condition turns out to be the criteria for limiting angles of approach.

Generally, a lower speed interceptor's initial location has to be in the forward zone with respect to the threat. If the interceptor's kill radius is accounted for then this forward zone will be expanded. Figure 2 displays two sets of limiting angles of approach. The first is the inner triangle with light grey shades corresponding to an interceptor with a zero kill radius. The second is the same triangle augmented by the two blue shaded triangles one on each side of the inner triangle. This naturally leads to improvement in the capabilities of an interceptor. Observe that in three dimensions, these triangles are in reality cones.

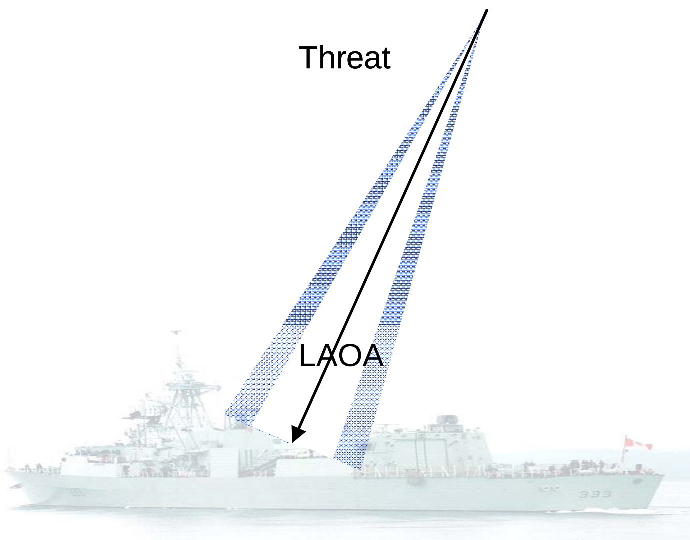

Figure 2. Limiting angles of approach (geometry).

The criteria of limiting angles of approach may guide the defence in terms of relative geometry within a task force. This is significant in scenarios where a ship is defended against missile threats by other surrounding ships. That is, if these surrounding ships lie outside of the limiting angles of approach then they will not be able to intercept a threat that has a higher speed than the one of the interceptors.

Figure 3 displays limiting angles of approach, defined in Figure 2, as a function of α and β. Here $\alpha = \frac{r_W}{\left| r_M^0 - r_T^0 \right|}$ is the ratio of the kill radius to the initial separation between the threat and the interceptor while $\beta = \frac{v_T}{v_M}$ is the ratio of the threat's speed to that of the interceptor. Figure 3 shows that limiting angles of approach increase as a function of α and decreases as a function of β. The first observation must be true since the kill radius increases as α increases and so the interceptor capability must also increase as a function of α. The second observation must also be true since the interceptor speed decreases as β increases and so the interceptor capability must also decrease as a function of β.

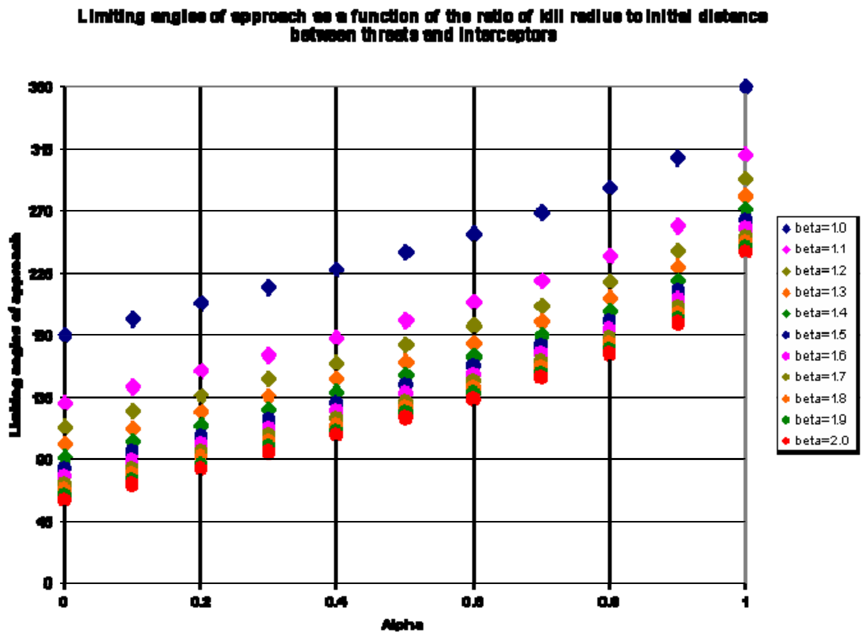

Figure 3. Limiting angles of approach.

4. Engagement Model

The engagement model called REST replicates engagement scenarios. It encodes the timeline and the firing tactics that are used. It draws on the missile model to simulate missiles, and the decision tree to determine the statistics at the end of the simulation. The engagement model is implemented by the engagement class.

The engagement class makes use of the threat, interceptor, and tree classes to execute the simulation and calculate the statistics. Currently this class implements a very general scenario: it allows multiple threats to be engaged by multiple interceptors. However, we assume perfect classifications and identifications. The data structure of a node makes it only a matter of time to implement these events. Note that the eventual implementations of these events will provide more accuracy to the MOEs but at the same time they will increase computational time.

For example, if only detection and engagement events are modeled then each engagement against a threat yields three possible outcomes as shown below. If we enumerate the outcomes from left to right then the first outcome is a detection event designated by the symbol D followed by a hit event designated by the symbol H. The second outcome is a detection event D followed by a miss event M. The third outcome is a non detection event D'. If a threat is not detected then the defence can not engage it. Hence, the third outcome consists only of D'.

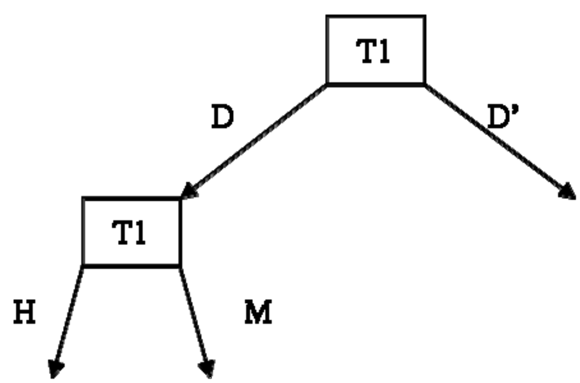

Figure 4. Detection followed by engagement.

However, if classification and identification events are also modeled then each engagement against a threat yields five possible outcomes as shown in Figure 5. Let C denote the event of classification and I denote the event of identification of a threat. The first outcome is a detection event D followed by a classification event C, an identification event I and ended with a hit event H. The second outcome is the same as the first outcome with the exception that it is ended with a miss event M. The third outcome represents a threat that is detected (D) and classified (C) but not identified (I') and so the defence can not engage it. Similarly, the fourth outcome represents a threat that is detected (D) but not classified (C'). Finally, the fifth outcome represents a threat that is not detected D'.

Observe that the difference in the number of outcomes between the two scenarios above is equal to two and hence a small number. However, if there is more than one threat, say 16 threats each engaged by a Sea Sparrow from a Halifax Class Frigate, Ref [10], then the difference in the number of outcomes between the two scenarios is equal to $5^{16} - 3^{16} \approx 10^{11} - 10^{7} \approx 10^{11}$ and would constitute a significant difference in computational time. This exposes the limitation of an engagement model based on a decision tree. That is, the total number of outcomes grows exponentially as a function of number of threats. However, as mentioned earlier, we consider only simple scenarios in this study. As a result, REST is a suitable model for analysis. Nevertheless, the user is cautioned to use common sense to simplify the scenario as much as possible. Based on experience, if the complexity is in the order of 10^{4} then execution takes less than one minute.

For example, if there is an incoming missile threat aiming at the defending ship then the probability of classification and identification are equal to 100 percent since classification and identification are obvious in such a scenario. Thus, classification and identification events can be eliminated to minimize computational time.

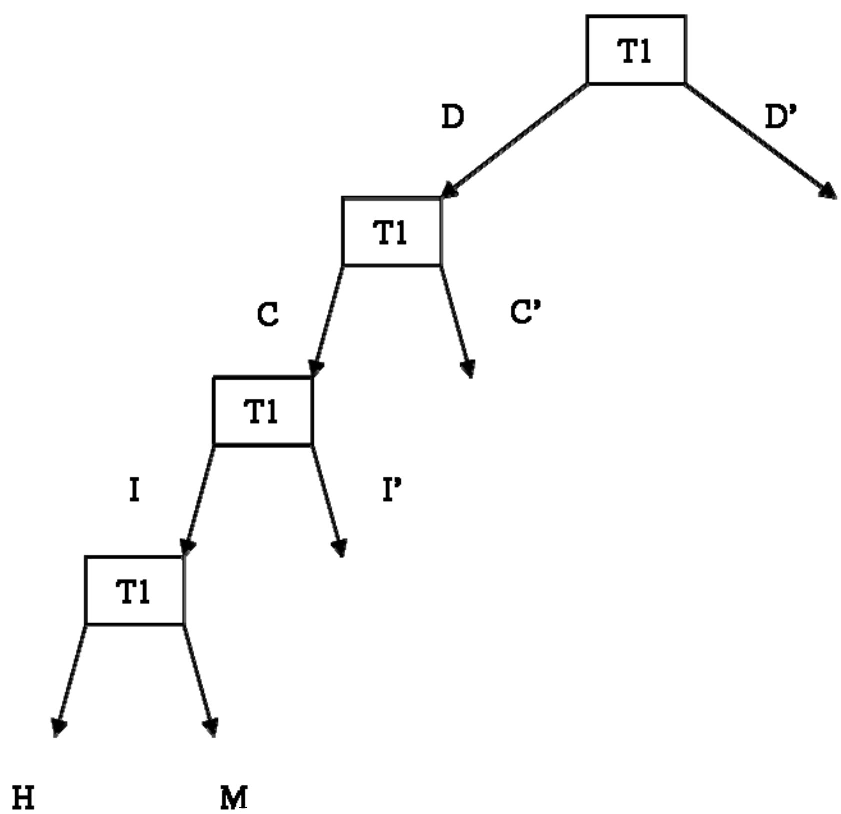

Figure 5. Detection, classification, identification followed by engagement.

5. Interceptor Model

While the software for REST should eventually have a graphical user interface, it currently executes from a command line interface and reads its input from text files. The input file contains two types of information: one for threats, and the other for interceptors. It can also contain comments, which are lines starting with a semi-colon. The input file is specified by giving its name as a command line argument to the executable.

5.1 Threats

Each threat is defined by a line in the input file. The line must begin with the keyword "threat". The remaining of this line contains information about the threat: a label which is a unique identifier for the threat, its initial position in Cartesian coordinates (x, y, z), its velocity vector also in Cartesian coordinates (v_x, v_y, v_z), and the time when it was launched with time zero being the start time of the scenario. The format for such an input is shown below:

“threat label x0 y0 z0 vx vy vz t0”

For example, the input line below defines a threat labelled as t1, launched at time zero from location $(1000,1000,1000)$, with velocity vector $(-25,-25,-25)$.

“threat t1 1000 1000 1000 -25 -25 -25 0”

5.2 Interceptors

Similarly, each type of interceptor is defined by a line in the input file. Unlike inputs for threats, where each individual threat is defined by a separate line, there is only one line for each type of interceptor. The line must begin with the keyword "interceptor". The remaining of the line contains information about the interceptor: a label identifying the type of interceptor, the number of interceptors of that type, its kill radius, speed, range, salvo size, and SSPK. The format for interceptors is shown below:

“interceptor label count killradius speed range salvo SSPK”

For example, the input line below defines an interceptor of type i1 of which there are three, each interceptor of type i1 has a kill radius equal to 1, a speed equal to 175, a detection range equal to 1500, a salvo size of 1, and a SSPK equal to 70 percent.

`“interceptor i1 3 1 175 1500 1 0.7”`

There are three observations to be made. First, REST determines the velocity vector for the interceptor in a way such that the intercept occurs in a minimal time as described earlier. Hence, only the speed of the interceptor is input. Second, the user needs to use consistent units. For example, if the location of a threat is in metres and the launch time is in seconds then the speed must be in metres per second. Third, a threat is detected when its distance from the defending ship is equal to the detection range. This implies that the defence can not launch its interceptors before the threats reach within the detection range.

6. Hard Kill Examples

6.1 Example 1

Example 1 consists of two threats launched at a defending ship. The defence carries three interceptors of the same type. The Single Shot Probability of Kill (SSPK) of the interceptors is equal to 70 percent. We assume that detections, classifications, identifications and kill assessments are all equal to 100 percent.

Each engagement of a threat by an interceptor has two outcomes: a hit or a miss denoted as H or M respectively. Threat number one is engaged by the first interceptor as shown in Figure 6. The left branch implies that threat one was hit while the right branch implies that threat one was missed.

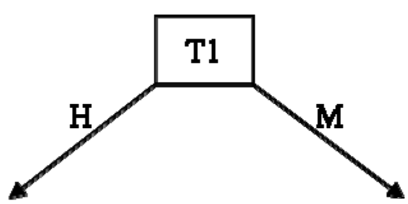

Figure 6. First engagement against threat 1 for Example 1.

After the first engagement, the defence engages threat number two. The outcomes of this second engagement are then inserted in the tree above. Note that the second engagement is independent of the first engagement. Thus, the new node is inserted to both branches of the tree as shown in Figure 7.

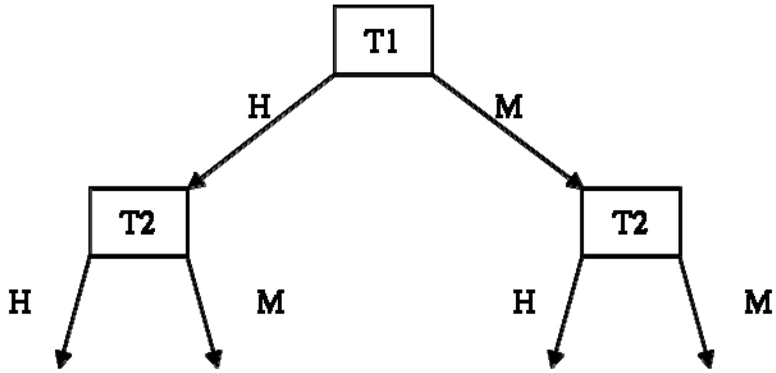

Figure 7. First engagement against threat 1 followed by first engagement against threat 2 for Example 1.

The third interceptor engages the threat that was missed in the two previous engagements with preference for threat number one if both threats were missed. Note that preference for threat number one is artificial since both threats are assumed identical and their initial ranges from the ship are also identical. This means that we could have chosen to reengage threat number two and the outcomes would have been unchanged. This artificiality allows the algorithm to pick one of the two identical threats. Note also that the new node associated with the third interceptor was not added to the leftmost branch. This is simply because this branch represents an outcome where both threats have already been hit. Once all threats are hit, the defence stops engaging. The final decision tree is shown in Figure 8.

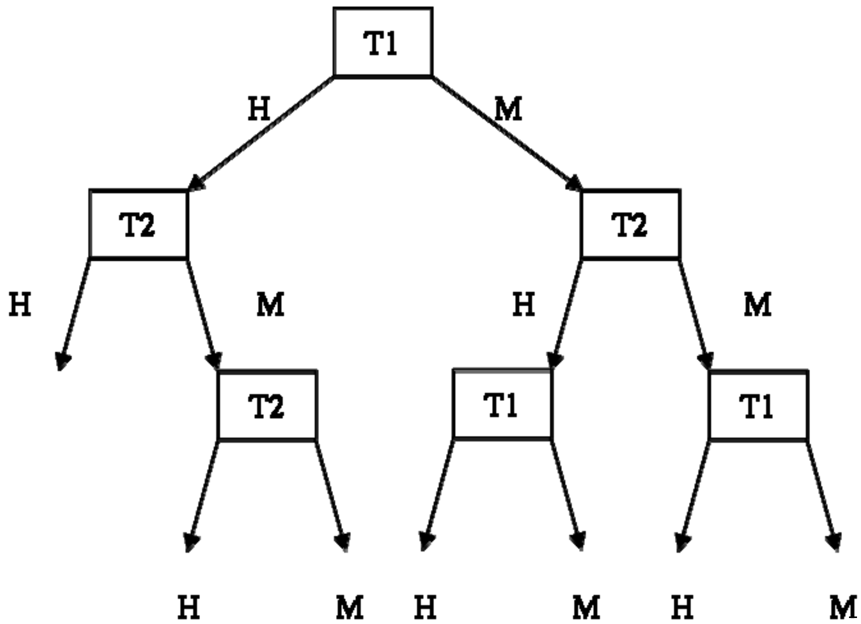

Figure 8. Final decision tree for Example 1.

The tree above has seven branches. Table 2 displays all the possible outcomes of the scenario in Example 1. Branches are enumerated from left to right. For example, branch number one corresponds to two hits: threat number one is hit by one interceptor and then threat number two is hit by another interceptor. Similarly, branch number seven corresponds to three misses: threat number one is missed twice and threat number two is missed once.

In Table 2, branch number one, two and four each yields two hits. This implies that these three branches correspond to outcomes where all threats are neutralized. Hence the probability of raid negation $\left(P_{raid}\right)$ is equal to the sum of the three associated probabilities. That is:

$$P_{raid} = 0.490 + 0.147 + 0.147 = 0.784$$

We can also determine the expected number of threats hit $\left(E_{raid}\right)$ by summing all probabilities each weighted by the number of hits:

$$E_{raid} = 2 \cdot 0.490 + 2 \cdot 0.147 + 1 \cdot 0.063 + 2 \cdot 0.147 + 1 \cdot 0.063 + 1 \cdot 0.063 + 0 \cdot 0.027 = 1.757$$

Similarly, the expected number of interceptors expended (E_{int}) is equal to the sum of all probabilities, each weighted by the number of interceptors expended:

$$E_{int} = 2 \cdot 0.490 + 3 \cdot 0.147 + 3 \cdot 0.063 + 3 \cdot 0.147 + 3 \cdot 0.063 + 3 \cdot 0.063 + 3 \cdot 0.027 = 2.510$$

Table 2. *Outcomes for Example 1*

BRANCH	BRANCH PROBABILITY	HITS	INTERCEPTORS EXPENDED	PRAID	ERAID	EINT
1	0.490	2	2	0.490	0.980	0.980
2	0.147	2	3	0.147	0.294	0.441
3	0.063	1	3	0.000	0.063	0.189
4	0.147	2	3	0.147	0.294	0.441
5	0.063	1	3	0.000	0.063	0.189
6	0.063	1	3	0.000	0.063	0.189
7	0.027	0	3	0.000	0.000	0.081
Total	1.000			0.784	1.757	2.510

The input file for Example 1 is shown below. It indicates that there are two threats, three interceptors and each engagement is made with a salvo size equal to one. It also provides information on the interceptor and those on the threat. The interceptor has a kill radius equal to one, a speed equal to 175 m/sec, a detection range equal to 1500 m, and a SSPK equal to 70 percent. Threat one's initial position is equal to $(1000\ m, 1000\ m, 1000\ m)$, a velocity vector equal $(-25\ m/\sec, -25\ m/\sec, -25\ m/\sec)$, and a launch time equal to 0. The information for threat two can be understood in the same way as those for threat number one.

```
; 2 threats
; 3 interceptors
; salvo 1
```

```
;interceptor  label  count  killradius  velocity  range  salvo  SSPK
interceptor   i1     3      1           175       1500   1      0.7

;threat  label  x0     y0     z0     vx   vy   vz   t0
threat   t1     1000   1000   1000   -25  -25  -25  0
threat   t2     -1000  -1000  -1000  25   25   25   0
```

The output tree for Example 1 is shown below as well as the corresponding MOEs. Note that the tree from the simulation is displayed horizontally. The bottom branch of TreeS is the same as the leftmost branch of TreeD while the top branch of TreeS is the same as the rightmost branch of TreeD.

```
                   ,-M
          ,-t1(0.7)
          |        `-H
  ,-t2(0.7)
  |       |        ,-M
  |       `-t1(0.7)
  |                `-H
--t1(0.7)
  |                ,-M
  |       ,-t2(0.7)
  |       |        `-H
  `-t2(0.7)
          `-H
```

Figure 9. Final decision tree for Example 1 from simulation.

$Pq[0]$ is the probability of zero hit, $Pq[1]$ is the probability of one hit and $Pq[2]$ is the probability of two hits which in this scenario is the same as the probability of raid negation. Comparing these results to those derived in the previous sub section confirms that the two methodologies are consistent with one another.

```
Praid      0.784
Eraid      1.757
Eint       2.510
Pq[0]      0.027
Pq[1]      0.189
Pq[2]      0.784
sum of Pq  1.000
```

6.2 Example 2

The input file for Example 2 is shown below. It indicates that there are three threats, fourteen interceptors and each engagement is made with a salvo size equal to two. The information on interceptors and threats can be understood in the same way as those from Example 1.

```
; 3 threats
; 14 interceptors
; salvo 2

;interceptor   label  count  killradius  speed  range  salvo  SSPK
interceptor    i1     14     1           175    1500   2      0.7

;threat  label  x0     y0     z0     vx   vy   vz   t0
threat   t1     1000   1000   1000   -25  -25  -25  0
threat   t2     -1000  -1000  -1000  25   25   25   0
threat   T3     1000   1000   1000   -25  -25  -25  0
```

The output for Example 2 is shown below. We have chosen not to display the tree since it is too big to be included.

```
Praid      1.000
Eraid      3.000
Eint       6.593
Pq[0]      4.783e-08
Pq[1]      3.385e-06
Pq[2]      0.000
Pq[3]      1.000
sum of Pq  1.000
```

6.3 Example 3

The input file for Example 3 is shown below. It indicates that there are five threats, twenty interceptors and each engagement is made with a salvo size equal to two. The information on interceptors and threats can be understood in the same way as those from Example 1.

```
; 5 threats
; 20 interceptors
; salvo 2

;interceptor   label  count  killradius  speed  range  salvo  SSPK
interceptor    i1     20     1           175    1500   2      0.7

;threat  label  x0     y0     z0     vx   vy   vz   t0
```

```
threat   t1     1000   1000   1000   -25  -25  -25  0
threat   t2     -1000  -1000  -1000  25   25   25   0
threat   t3     1000   -1000  1000   -25  25   -25  0
threat   t4     1000   -1000  1000   -25  25   -25  0
threat   t5     -1000  1000   1000   25   -25  -25  0
```

The output for Example 3 is shown below. Again, we have chosen not to display the tree since it is too big to be included.

```
Praid      1.000
Eraid      4.000
Eint       10.989
Pq[0]      3.487e-11
Pq[1]      3.526e-09
Pq[2]      1.604e-07
Pq[3]      4.325e-06
Pq[4]      7.653e-05
Pq[5]      1.000
sum of Pq  1.000
```

7. Other Examples

While the first three examples above (Section 6) deal with hard kill weapons, there is no reason why a tree can not accommodate soft kill (Refs [10-11]) or any kinds of defence mechanisms for that matter. What is more, the recursive algorithm can always be applied. In this section, we provide two more examples demonstrating the versatility of the binary tree. The first example considers soft kill weapons and the second example considers underwater mines. However, note that the timelines corresponding to these examples are not provided by the interceptor module in Section 6 simply because these examples do not entertain interceptors. Their timelines form the subject of future work. They need to be examined case by case. To give the reader an idea of what to expect, the timeline for chaff against radar missiles can be obtained for example from Ref [12]. This may include the jamming time of the defence against the seeker of the threat missile and the launch, and bloom times of the seduction chaff. Associated with the timeline are probabilities such as the probability that the seeker captures a chaff cloud after jamming ends or the probability that the seeker recaptures the defending ship once it has locked on to the chaff cloud.

7.1 Example 4 – Soft Kill

Below is an example of a soft kill scenario obtained from Ref [11]. There are two incoming missiles $(T1 \text{ and } T2)$ launched at a defending ship (DS). Each missile is engaged by a seduction chaff. $S\,(U)$ means that the threat is (not) seduced by the chaff. $H'\,(M')$ implies that the defending ship is hit (missed) by the missile(s).

Given the decision tree in Figure 10, we get the following MOEs:

$$\begin{aligned} P_{raid} &= S^2 \\ P_{survival} &= S^2 + 2 \cdot S \cdot U \cdot M' + U^2 \cdot M'^2 \\ &= \left(S + U \cdot M'\right)^2 \end{aligned}$$

where $P_{survival}$ is the probability of survival of the defending ship and P_{raid} is the same probability of raid negation as in Section 6 but this time using soft kill weapons. Note that, in addition to the treatment in Ref [11], in this example, we have added the possibility that the threat may hit (H') or may not hit (M') the ship even when the soft kill weapons were not successful. This additional feature aims to reveal the flexibility of the binary tree.

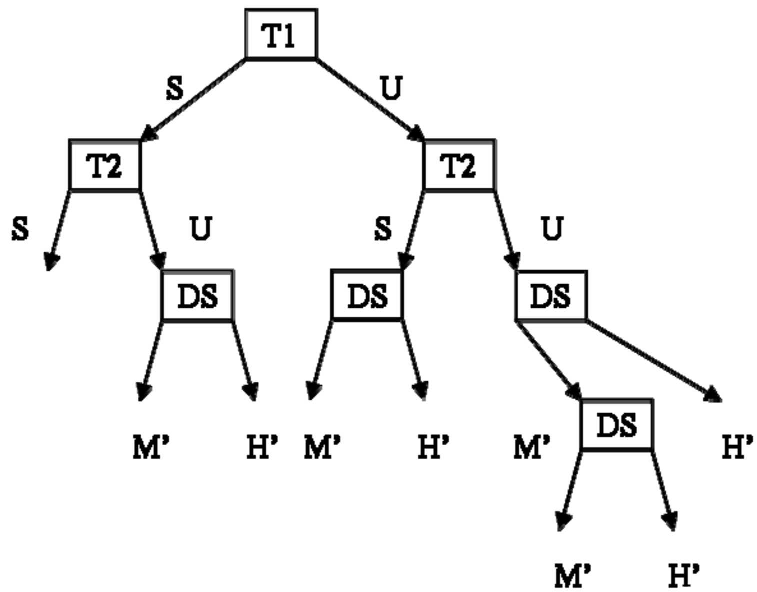

Figure 10. Final decision tree for Example 4 (soft kill).

7.2 Example 5 – Mines

Example 5 consists of a ship transiting through an area horizontally where there are three mines as shown in Figure 11. Each mine has an effective range (r_e) and a corresponding probability of hit (A) i.e. if the defending ship is within the effective range of a mine then there is a probability equal to A that this mine would actuate. This is called a cookie cutter model. In Figure 12, we show the final decision tree for Example 5. We mean by $A(A')$ that the mine has (has not) actuated, and by $M'(H')$ that the defending ship (DS) survives (is destroyed) by the actuation of the mines.

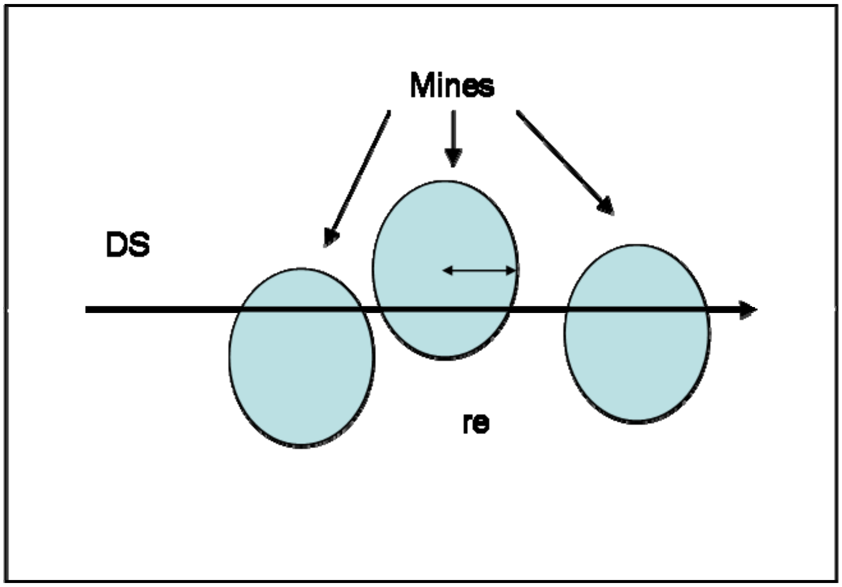

Figure 11. A defending ship (DS) transiting through a mined area.

The MOEs corresponding to the binary tree in Figure 12 can be written as:

$$
\begin{aligned}
P_{raid} &= (A')^3 \\
P_{survival} &= (A')^3 + 3\cdot(A')^2\cdot(A\cdot M') + 3\cdot A'\cdot(A\cdot M')^2 + (A\cdot M')^3 \\
&= (A' + A\cdot M')^3
\end{aligned}
$$

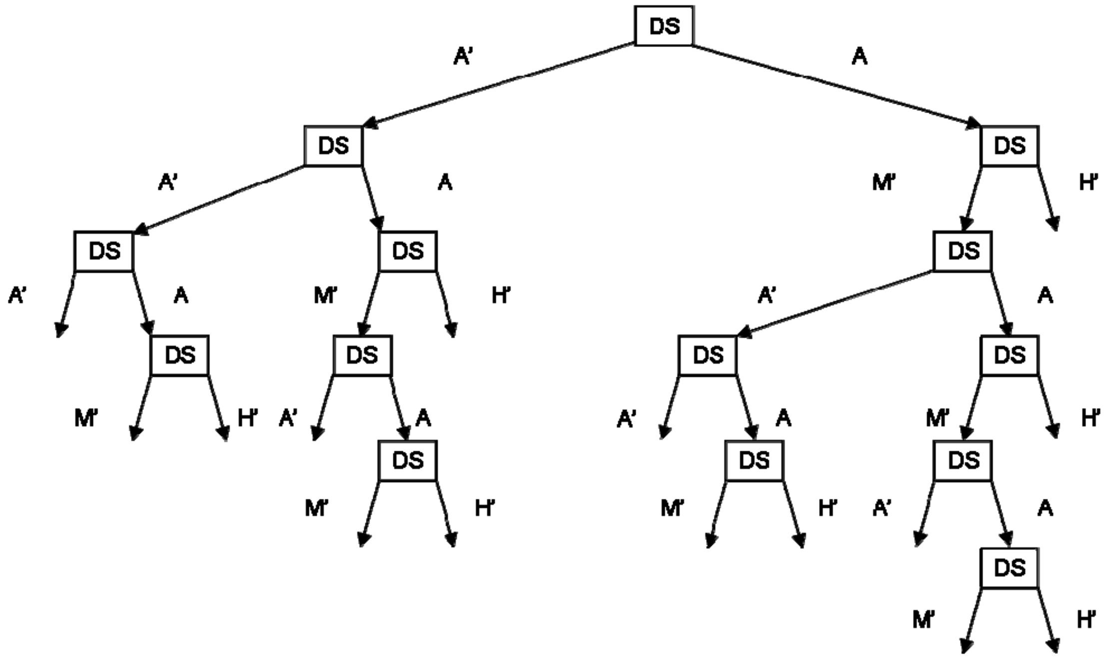

Figure 12. Final decision tree for Example 5 (mines).

8. Discussion

The engagement model described in this report is a first step in modelling the impact of ISM. It provides a common and consistent framework to compare the impact of each signature. In essence, it is a methodology to keep track of all possible outcomes in a battle. It also records the timeline of significant events and their corresponding probabilities. So far, REST has modelled only the intercept event i.e. the hit and miss outcomes. In reality, detections, classifications, identifications, launches, intercepts and kill assessments also need to be modelled as mentioned in Section 4.

Note that the examples provided in this report assume that the hit and miss probabilities are identical for all engagements. Nevertheless, this assumption is not essential in REST. It was only made to simplify the illustrations. For example, two detection events of a threat occurring at different times might have different probabilities as they may be associated to different signature strengths of the defending ship and different orientations of that ship. Intrinsically, the model is only as good as its input data. That is, the value of the probability of a node is an input to REST. They can be obtained, derived or further developed from more accurate models or real experiments. This task is by no means trivial. It is part of an intensive and ongoing research, and will be reported in the future before the ISM project is completed.

Even though, the aim is to determine the vulnerability of a CPF, this engagement model can be applied to all scenarios irrespective of the threats and signatures e.g. torpedoes and ship acoustic signature, or radar missiles and ship radar cross section. In addition, it can model multiple threats of multiple types as well as multiple defensive systems. For example, we can examine a scenario where a torpedo is launched at the defending ship and some time during this attack an infrared missile is fired at the same ship. Such a complex scenario may or may not arise in real life. However, it would identify the weaknesses and strengths of the defence as well as illustrate the impact of lowering the strength of one type of signature while raising that of another.

REST can also analyze tactics and manoeuvres of the defence. For one thing, it is known in torpedo defence that there are established manoeuvres conducted by the defending ship to evade the torpedo threats. Such an analysis may shed light on how these tactics need to be modified to accommodate an additional infrared missile threat during a torpedo attack.

9. References

1. B. U. Nguyen, "Assessing the Vulnerability Reduction by Integrating Signatures in Generic Maritime Operations", draft DRDC Technical Memorandum, Jul 05.

2. Averill M. Law and W. David Kelton, *Simulation modeling and analysis*, McGraw–Hill, 2000, chapter 1, pp. 90 – 91.

3. William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, *Numerical Recipes*, Cambridge university press, 1992, chapter 7, p. 290.

4. B. U. Nguyen, P. A. Smith and D. Nguyen, "An engagement model to optimize defense against a multiple attack assuming perfect kill assessment", Naval Research Logistics, volume 44, 97, pp. 687-697.

5. Thomas A. Standish, *Data structures, algorithms and software principles*, Addison – Wesley Publishing Company, 1994, chapter 9, pp. 337–401.

6. B. U. Nguyen and L. Briggs, "Computing measures of effectiveness in EADSIM (Extended Air Defense Simulation) using decision tree", NORAD/AN Technical Memorandum, Jun 04.

7. B. U. Nguyen, "Determining the Optimal Intercept Angle for Attack on a Surface Ship by a Submarine with Non-Zero Weapon Range", DMOR Research Note, Feb 94.

8. B. U. Nguyen and D. Nguyen, *Optimal intercept course of vessels to a non-zero range*, SIAM Review, Dec 96.

9. http://en.wikipedia.org/wiki/Halifax_class_frigate

10. A. Jesion, "A methodology for the inclusion of seduction chaff in DMOR's Ship Area Air Defence Simulation (SAADS)", ORAE Project Report, PR579, Mar 92.

11. B. U. Nguyen and M. A. Gammon, "Seduction chaff: a new defence capability for the Ship Area Air Defence Simulation (SAADS)", ORA Research Note, RN 94/7, Nov 94.

12. Guy Ferland, "A combined jamming and chaff solution algorithm for SISWS", DRDC Ottawa Contractor Report, CR 2005-085, Jun 05.

Annex A – Optimal Intercept Time at a Nonzero Kill Radius

In this Annex, we derive the shortest intercept time for an interceptor with a nonzero kill radius aiming at a missile threat. The threat assumes a given constant speed and a predetermined straight line trajectory. The interceptor is also a straight line. However, the interceptor's direction is to be determined by the optimal intercept time.

The equation of motion of a threat can be written as:

$$\overrightarrow{r_T} = \overrightarrow{r_T^0} + \overrightarrow{v_T} \cdot t$$

where $\overrightarrow{r_T}$ is the position of a threat at time t; $\overrightarrow{r_T^0}$ its position at time $t = 0$ and $\overrightarrow{v_T}$ is its velocity vector. Similarly, the equation of motion of an interceptor can be written as:

$$\overrightarrow{r_M} = \overrightarrow{r_M^0} + \overrightarrow{r_W} + \overrightarrow{v_M} \cdot (t - t')$$

where $\overrightarrow{r_M}$ is the position of an interceptor at time t; $\overrightarrow{r_M^0}$ its position at time $t = t'$; $\overrightarrow{r_W}$ represents the kill radius, that is $\left|\overrightarrow{r_W}\right| = r_W$ and $\overrightarrow{v_M}$ is the velocity vector to be determined from the optimal intercept time.

At the time of intercept, the position of a threat is the same as that of an interceptor, i.e. $\overrightarrow{r_T} = \overrightarrow{r_M}$. Refs [7-8] dictate the following formula for intercept time:

$$T_I = \frac{r}{v_T}\left(\frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}\right)$$

where

$$a = \frac{v_M^2}{v_T^2} - 1$$

$$b = 2 \cdot \hat{r} \cdot \widehat{v_T} + 2 \cdot \frac{v_M}{v_T} \cdot \frac{r_W}{r}$$

$$c = \left(\frac{r_W}{r}\right)^2 - 1$$

$$\bar{r} = \overrightarrow{r_M^0} - \overrightarrow{r_T^0} - \overline{v_T} \cdot t'$$

$$\hat{r} = \bar{r} / r$$

$$\widehat{v_T} = \overline{v_T} / v_T$$

T_I is the shortest time to intercept a target at a nonzero kill radius r_W with a constant speed v_M .

Annex B – Optimal Intercept Direction at a Nonzero Kill Radius

In Annex A, we have derived the optimal intercept time at a nonzero kill radius. In this Annex, we derive the corresponding direction for an interceptor to intercept a threat with the optimal intercept time or equivalently at the kill radius. The geometry of this problem is shown in Figure 13 below:

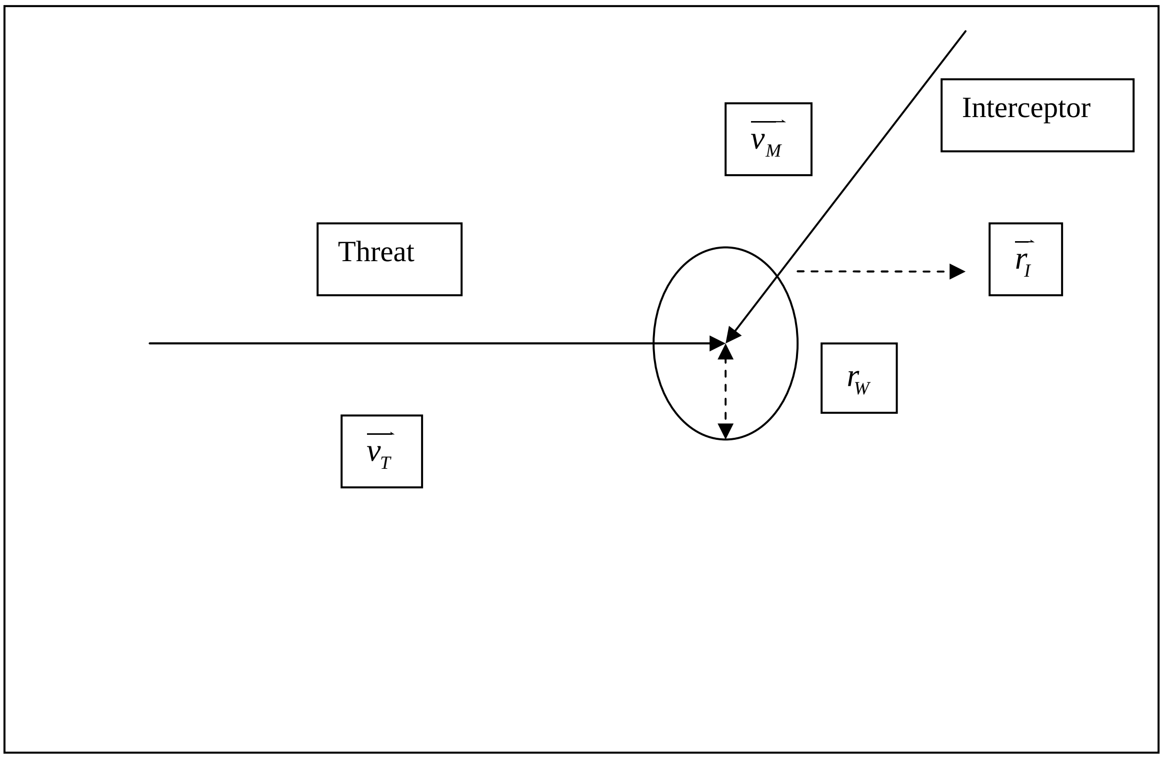

Figure 13. Geometry of optimal intercept.

A threat is moving along the velocity vector $\vec{v}_T$ while an interceptor is moving along the velocity vector $\vec{v}_M$. The intercept occurs at the surface of a sphere having a radius equal to r_W at location $\vec{r}_I$.

Based on the geometry of Figure 13, we can infer that:

$$\overline{r_M^0} + \overline{v_M} \cdot \left(T_I - t'\right) + r_W \cdot \widehat{v_M} = \overline{r_T^0} + \overline{v_T} \cdot T_I$$

where $\widehat{v_M}$ is the unit vector along $\overline{v_M}$. Thus, the velocity vector of the missile can be written as:

$$\widehat{v_M} = \frac{\overline{r_T^0} - \overline{r_M^0} + \overline{v_T} \cdot T_I}{\left(r_W + v_M \cdot \left(T_I - t'\right)\right)}$$

$$\overline{v_M} = v_M \cdot \widehat{v_M}$$

Annex C – Classes

REST is programmed in C++, an object oriented language, which allows the use of class. There are five classes in REST. To build the binary tree, we define the Node and Tree class. To model threats and interceptors, we define the Missile, Threat and Interceptor class.

```
string      m_label;
float       m_time;
string      m_threat;
string      m_interceptor;
int         m_salvoSize;
float       m_probHit;
float       m_probMiss;
enum        EventType { Detect, Engage } m_eventType;
```

`m_label` is a label representing an event such as a detection or an engagement event. In REST, an event is called a Node. The label of a Node is displayed in the decision tree.

`m_time` is the time when the event occurs.

`m_threat` and `m_interceptor` are text strings which refer to the threat and interceptor involved in an event. Each threat is identified by a unique label. The label of an interceptor must match one of those defined in the Tree class.

`m_salvoSize` indicates the salvo size that is fired against a threat. For example, if `m_salvoSize` is equal to two then the defence engages a threat with two interceptors at each engagement opportunity.

`m_probHit` is the SSPK. `m_probHit` represents the probability of hitting a threat with one interceptor. `m_probMiss` represents the probability of missing a threat with one interceptor. `m_probMiss` is the complement of `m_probHit`: $m\_probMiss = 1 - m\_probHit$. Hence, only the value of `m_probHit` needs to be input.

`m_eventType` indicates whether this is a detection or an engagement event. Classification, identification and kill assessment events can easily be added to `m_eventType` in the future. Such additions will improve the functionality of REST.

Tree Class

The Tree class encapsulates the structure of a tree. It is designed as a recursive data structure. That is, each sub-tree of a tree is a tree. The data structure of a Tree is listed below:

```
Node                      m_node;
Tree                      *m_pHit;
Tree                      *m_pMiss;
map<string, int>          m_arsenal;
set<string>               m_threats;
struct  MonteCarloStruct  float          cumulativePb;
                          float          Pb;
                          int            killed;
                          int            expended;
static                    vector<MonteCarloStruct>
                          m_monteCarloVector;
```

`m_node` is the Node object associated with an event.

`m_pHit` and `m_pMiss` are pointers to two sub-trees: one corresponding to the left sub-tree favouring the defence and the other to the right sub-tree favouring the threat.

`m_arsenal` contains a list of the interceptor types and their inventories.

`m_threats` contains the list of all threats.

`m_monteCarloVector` is an option allowing the user to run REST as a Monte Carlo simulation. This option can be used to compare the MOEs obtained by the decision tree to those obtained by Monte Carlo's methodology.

The functionality of a tree requires a number of public functions listed below:

```
bool  IsEmpty();
void  AddArsenal(string weapon, int count);
void  AddEvent(const Node &node,
      const set<string> &dependsOn,
      const set<string> &dependsOnNot);

void  PrintTree();
void  PrintStats();
void  PrintMonteCarlo(int samplesize);
```

`IsEmpty()` confirms whether a tree is empty or not.

`AddArsenal` adds weapons to the arsenal.

`AddEvent()` adds events to a tree.

`PrintTree()` displays a graphical representation of a decision tree.

`PrintStats()` and `PrintMonteCarlo()` print out the MOEs. The former function's MOEs are obtained by traversing the tree while those of the latter function are obtained by Monte Carlo's methodology.

The Tree class is designed to handle simple dependencies between events. The dependencies are supplied through the `AddEvent()` function. Two sets of strings can be supplied: `dependsOn` and `dependsOnNot`. An event will only be added to a tree when one of the events in `dependsOn` has occurred, or when one of the events in `dependsOnNot` has not occurred.

Missile Class

The Missile class is the base class for the Threat and Interceptor classes. Given an initial location and a velocity vector, it tracks the current position of the missile as a function of time.

The data structure a Missile class is listed below. The notation should be obvious. For example, the initial position of a missile is represented by (m_x0, m_y0, m_z0) while the velocity vector is represented by (m_vx, m_vy, m_vz).

```
string                m_label;
// Starting position  double m_x0;
                      double m_y0;
                      double m_z0;
// Starting time      double m_t0;
// Velocity           double m_vx;
                      double m_vy;
                      double m_vz;
                      double m_v;
// Current position   double m_x;
                      double m_y;
                      double m_z;
// Current time       double m_t;
enum                  State { Unfired, Running, Expended } m_state;
```

The public functions made available to the Missile class are listed below:

```
string  GetLabel();
State   GetState();
void    GetPosition
        (double &x,
        double &y,
        double &z);
void    GetDistance
        (double x,
        double y,
        double z,
        double &distance);
double  GetTime();
```

`GetLabel()` provides the label of a missile.

`GetState()` provides the state of a missile which can be: `Unfired`, `Running`, or `Expended`.

`GetPosition()` provides the position (x, y, z) of a missile in three dimensions.

`GetDistance()` calculates the distance of a missile from the defending ship.

`GetTime()` provides the time when the missile was last updated.

Threat Class

The Threat class is derived from the Missile class. A threat in the Threat class is defined with a starting position and a velocity vector. The Threat class also records the launch and impact time. The public functions made available to the Threat class are listed below:

```
void  Init
      (string label,
      double x0,
      double y0,
      double z0,
      double vx,
      double vy,
      double vz,
      double t0);
bool  Update(double t);
```

`Init()` is used to provide the initial position (x_0, y_0, z_0), the velocity vector (v_x, v_y, v_z), and the launch time t_0 of a threat.

`Update()` is a function of the current time t. It will "fire" the missile at a specified launch time, updates the position of the missile while it is flying, and stops the time increment when the missile reaches the aimed point.

Interceptor Class

The Interceptor class is also derived from the Missile class. It launches an interceptor from an initial location, with a given speed, and a given kill radius, against a threat. Note that the speed is a scalar not a vector. The Interceptor class will determine the velocity vector required to achieve the minimal intercept time based on the interceptor's characteristics and the nature of the threat as described in Annex B. Similarly to the Threat class, the Interceptor class will stop the time increment once the interceptor reaches the kill radius from the threat.

The data structure of the Interceptor class requires additional parameters listed below:

```
double  m_r;
double  m_interceptTime;
Threat  *m_threat;
```

`m_r` is the kill radius of the interceptor. That is, the range between an interceptor and a threat necessary to kill that threat.

`m_interceptTime` is the time when the range between an interceptor and a threat is equal to the kill radius.

`m_threat` is a pointer to the threat being engaged.

The public functions made available to the Interceptor class are listed below:

```
double  Fire
        (string label,
        double x0,
        double y0,
        double z0,
        double v,
        double r,
        double t,
        Threat &threat);
Threat  *GetThreat();
```

`Fire()` launches an interceptor against a specified threat from an initial position (x_0, y_0, z_0) with a speed v, a kill radius r at time t. The velocity vector is computed as explained above.

Annex D – A Detailed Output

This is a sample detailed output based on the scenario defined in Example 1. It shows the position of each missile at each time increment, as well as indicating when the missiles are fired and when they complete their flights.

The input file for Example 1 is shown below. It indicates that there are two threats, three interceptors and each engagement is made with a salvo size equal to one. It also provides information on the interceptor and those on the threat. The interceptor has a kill radius equal to $1\ m$, a speed equal to $175\ m/\sec$, a maximal detection range equal to $1500\ m$, and a SSPK equal to 70 percent. Threat one's initial position is equal to $(1000\ m, 1000\ m, 1000\ m)$, its velocity vector equal $(-25\ m/\sec, -25\ m/\sec, -25\ m/\sec)$, and its launch time equal to $0\ \sec$. Threat two's initial position is equal to $(-1000\ m, -1000\ m, -1000\ m)$, its velocity vector equal $(25\ m/\sec, 25\ m/\sec, 25\ m/\sec)$, and its launch time equal to $0\ \sec$.

```
; 2 threats
; 3 interceptors
; salvo 1

;interceptor  label  count  killradius  velocity  range  salvo  SSPK
interceptor   i1     3      1           175       1500   1      0.7

;threat  label  x0     y0     z0     vx   vy   vz   t0
threat   t1     1000   1000   1000   -25  -25  -25  0
threat   t2     -1000  -1000  -1000  25   25   25   0
```

The details of the output are shown below. The two threats are launched at time $0\ \sec$. At time equal to $6\ \sec$, both threats enter the maximal detection range of the defending ship. At this time, one interceptor is launched against threat one and another interceptor is launched against threat two. Based on the kill radius, the intercept time for both engagements occurs at time equal to $12.7395\ \sec$. The simulation time steps until time equal to $12\ \sec$. Before time equal to $13\ \sec$, a node representing the intercept of threat one is added to the tree as well as a second node representing the intercept of threat two.

At time equal to $13\ \sec$, an interceptor is launched against each threat if it was missed by the previous engagements. The simulation repeats the same calculation as that of the first two engagements until both threats reach their impact point. Observe that, in this scenario, we assume that there are as many interceptors available as needed. In reality, if there are only three interceptors then the defence will stop engaging at the second engagement opportunity which occurs at time equal to 13 sec .

```
time: 0
t1: 1000 1000 1000
t2: -1000 -1000 -1000

time: 1
t1: 975 975 975
t2: -975 -975 -975

time: 2
t1: 950 950 950
t2: -950 -950 -950

time: 3
t1: 925 925 925
t2: -925 -925 -925

time: 4
t1: 900 900 900
t2: -900 -900 -900

time: 5
t1: 875 875 875
t2: -875 -875 -875

time: 6
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  680.935 680.935 680.935
Intercept time:   12.7395

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -680.935 -680.935 -680.935
Intercept time:   12.7395

t1: 850 850 850
t2: -850 -850 -850
i1: 0 0 0
i1: 0 0 0

time: 7
t1: 825 825 825
t2: -825 -825 -825
i1: 101.036 101.036 101.036
i1: -101.036 -101.036 -101.036

time: 8
t1: 800 800 800
```

```
t2: -800 -800 -800
i1: 202.073 202.073 202.073
i1: -202.073 -202.073 -202.073

time: 9
t1: 775 775 775
t2: -775 -775 -775
i1: 303.109 303.109 303.109
i1: -303.109 -303.109 -303.109

time: 10
t1: 750 750 750
t2: -750 -750 -750
i1: 404.145 404.145 404.145
i1: -404.145 -404.145 -404.145

time: 11
t1: 725 725 725
t2: -725 -725 -725
i1: 505.181 505.181 505.181
i1: -505.181 -505.181 -505.181

time: 12
t1: 700 700 700
t2: -700 -700 -700
i1: 606.218 606.218 606.218
i1: -606.218 -606.218 -606.218

time: 12.7395
i1: end of run
i1: end of run

time: 13
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  540.647 540.647 540.647
Intercept time:   18.351

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -540.647 -540.647 -540.647
Intercept time:   18.351

t1: 675 675 675
t2: -675 -675 -675
i1: 0 0 0
i1: 0 0 0

time: 14
t1: 650 650 650
t2: -650 -650 -650
i1: 101.036 101.036 101.036
i1: -101.036 -101.036 -101.036
```

```
time: 15
t1: 625 625 625
t2: -625 -625 -625
i1: 202.073 202.073 202.073
i1: -202.073 -202.073 -202.073

time: 16
t1: 600 600 600
t2: -600 -600 -600
i1: 303.109 303.109 303.109
i1: -303.109 -303.109 -303.109

time: 17
t1: 575 575 575
t2: -575 -575 -575
i1: 404.145 404.145 404.145
i1: -404.145 -404.145 -404.145

time: 18
t1: 550 550 550
t2: -550 -550 -550
i1: 505.181 505.181 505.181
i1: -505.181 -505.181 -505.181

time: 18.351
i1: end of run
i1: end of run

time: 19
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  420.401 420.401 420.401
Intercept time:   23.1609

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -420.401 -420.401 -420.401
Intercept time:   23.1609

t1: 525 525 525
t2: -525 -525 -525
i1: 0 0 0
i1: 0 0 0

time: 20
t1: 500 500 500
t2: -500 -500 -500
i1: 101.036 101.036 101.036
i1: -101.036 -101.036 -101.036

time: 21
t1: 475 475 475
```

```
t2: -475 -475 -475
i1: 202.073 202.073 202.073
i1: -202.073 -202.073 -202.073

time: 22
t1: 450 450 450
t2: -450 -450 -450
i1: 303.109 303.109 303.109
i1: -303.109 -303.109 -303.109

time: 23
t1: 425 425 425
t2: -425 -425 -425
i1: 404.145 404.145 404.145
i1: -404.145 -404.145 -404.145

time: 23.1609
i1: end of run
i1: end of run

time: 24
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  320.195 320.195 320.195
Intercept time:   27.1691

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -320.195 -320.195 -320.195
Intercept time:   27.1691

t1: 400 400 400
t2: -400 -400 -400
i1: 0 0 0
i1: 0 0 0

time: 25
t1: 375 375 375
t2: -375 -375 -375
i1: 101.036 101.036 101.036
i1: -101.036 -101.036 -101.036

time: 26
t1: 350 350 350
t2: -350 -350 -350
i1: 202.073 202.073 202.073
i1: -202.073 -202.073 -202.073

time: 27
t1: 325 325 325
t2: -325 -325 -325
i1: 303.109 303.109 303.109
i1: -303.109 -303.109 -303.109
```

```
time: 27.1691
i1: end of run
i1: end of run

time: 28
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  240.031 240.031 240.031
Intercept time:   30.3757

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -240.031 -240.031 -240.031
Intercept time:   30.3757

t1: 300 300 300
t2: -300 -300 -300
i1: 0 0 0
i1: 0 0 0

time: 29
t1: 275 275 275
t2: -275 -275 -275
i1: 101.036 101.036 101.036
i1: -101.036 -101.036 -101.036

time: 30
t1: 250 250 250
t2: -250 -250 -250
i1: 202.073 202.073 202.073
i1: -202.073 -202.073 -202.073

time: 30.3757
i1: end of run
i1: end of run

time: 31
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  179.907 179.907 179.907
Intercept time:   32.7806

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -179.907 -179.907 -179.907
Intercept time:   32.7806

t1: 225 225 225
t2: -225 -225 -225
i1: 0 0 0
i1: 0 0 0
```

```
time: 32
t1: 200 200 200
t2: -200 -200 -200
i1: 101.036 101.036 101.036
i1: -101.036 -101.036 -101.036

time: 32.7806
i1: end of run
i1: end of run

time: 33
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  139.825 139.825 139.825
Intercept time:   34.3839

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -139.825 -139.825 -139.825
Intercept time:   34.3839

t1: 175 175 175
t2: -175 -175 -175
i1: 0 0 0
i1: 0 0 0

time: 34
t1: 150 150 150
t2: -150 -150 -150
i1: 101.036 101.036 101.036
i1: -101.036 -101.036 -101.036

time: 34.3839
i1: end of run
i1: end of run

time: 35
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  99.7427 99.7427 99.7427
Intercept time:   35.9872

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -99.7427 -99.7427 -99.7427
Intercept time:   35.9872

t1: 125 125 125
t2: -125 -125 -125
i1: 0 0 0
i1: 0 0 0

time: 35.9872
```

```
i1: end of run
i1: end of run

time: 36
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  79.7016 79.7016 79.7016
Intercept time:   36.7888

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -79.7016 -79.7016 -79.7016
Intercept time:   36.7888

t1: 100 100 100
t2: -100 -100 -100
i1: 0 0 0
i1: 0 0 0

time: 36.7888
i1: end of run
i1: end of run

time: 37
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  59.6605 59.6605 59.6605
Intercept time:   37.5905

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -59.6605 -59.6605 -59.6605
Intercept time:   37.5905

t1: 75 75 75
t2: -75 -75 -75
i1: 0 0 0
i1: 0 0 0

time: 37.5905
i1: end of run
i1: end of run

time: 38
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  39.6194 39.6194 39.6194
Intercept time:   38.3921

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -39.6194 -39.6194 -39.6194
Intercept time:   38.3921
```

```
t1: 50 50 50
t2: -50 -50 -50
i1: 0 0 0
i1: 0 0 0

time: 38.3921
i1: end of run
i1: end of run

time: 39
i1 fired against: t1
Intercept vector: 101.036 101.036 101.036
Intercept point:  19.5783 19.5783 19.5783
Intercept time:   39.1938

i1 fired against: t2
Intercept vector: -101.036 -101.036 -101.036
Intercept point:  -19.5783 -19.5783 -19.5783
Intercept time:   39.1938

t1: 25 25 25
t2: -25 -25 -25
i1: 0 0 0
i1: 0 0 0

time: 39.1938
i1: end of run
i1: end of run

time: 40
t1: end of run
t2: end of run
```

List of symbols/abbreviations/acronyms/initialisms

DND	Department of National Defence
DRDC	Defence Research Development Canada
ISM	Integrated Signature Management
SSPK	Single Shot Probability of Kill
IR	Infrared
EM	Electromagnetic
MOEs	Measures Of Effectiveness
REST	Recursive Engagement Simulation Tree

Distribution list

Internal Distribution

5 Dr. Bao Nguyen

5 DRDC Atlantic Library

1 Dr. Dan Hutt

1 Dr. Zahir Daya

1 Mr. Boris Vasiliev

1 Dr. Jim L. Kennedy

1 Dr. Jim S. Kennedy

1 Dr. Dan Peters

1 Mr. David Hopkin

1 Dr. John Fawcett

1 Dr. Sean Pecknold

1 Mr. Dave Hazen

1 Dr. Mark Gammon

External Distribution

1 NDHQ/DRDKIM

1 Dr. Abe Jesion,
NDHQ/DRDC CORA/MORT
101 Colonel By Drive, Pearkes Building
Ottawa, ON
Canada K1A 0K2

1 Mr. Paul Saunders,
MARLANT/DCOS OR, N02OR
PO Box 99000 Stn Forces
Halifax, NS
Canada B3K 5X5

1 Dr. Michael Crawford
CFMWC
PO Box 99000 Stn Forces
Halifax, NS
Canada B3K 5X5

1 Dr. Roy Mitchell,
NDHQ/DRDC CORA
101 Colonel By Drive, Pearkes Building
Ottawa, ON
Canada K1A 0K2

1 Mr. R.W. Funk
DCOS Operational Research (J02 OR)
Maritime Forces Pacific
Joint Task Force (Pacific)
PO Box 17000 Stn Forces
Victoria, BC
Canada V9A 7N2

1 Dr. Murray Dixson
Canadian Forces Experimentation Centre Bldg. 4, Shirley's Bay
National Defence Headquarters
Ottawa, ON
Canada K1A 0K2

1 Mr. Ivan Taylor/Mr. Ed Emond
NDHQ/DRDC CORA/CORT
101 Colonel By Drive, Pearkes Building
Ottawa, ON
Canada K1A 0K2

1 Mr. Paul Massel
NDHQ/DRDC CORA/CORT
101 Colonel By Drive, Pearkes Building
Ottawa, ON
Canada K1A 0K2

1 Mr. Mike Ormrod
NDHQ/DRDC CORA
101 Colonel By Drive, Pearkes Building
Ottawa, ON
Canada K1A 0K2

1 Mr. Bob Burton
NDHQ/DRDC CORA
101 Colonel By Drive, Pearkes Building
Ottawa, ON
Canada K1A 0K2

1 Dr. Dave Allen
Canadian Forces Experimentation Centre Bldg. 4, Shirley's Bay
National Defence Headquarters
Ottawa, ON
Canada K1A 0K2

1 Dr. Lise Arseneau
NDHQ/CANSOFCOM COS FD
101 Colonel By Drive, Pearkes Building
Ottawa, ON
Canada K1A 0K2

1 Dr. Andrew Billyard
Building 14, Shirley's Bay Campus
3701 Carling Avenue
Ottawa, ON
Canada K2H 8S2

1 Dr. Rene Seguin/Mr. Pieter de Jong
1 CAD ORT
Box 17000 Station Forces
Winnipeg, MB
Canada R3J 3Y5

1 Mr. Jean-Denis Caron/Dr. Sean Bourdon
NORAD and USNORTHCOM J84 Analysis
250 Vandenberg Street, Suite B016
Peterson AFB, CO
USA 80914-3812

1 Dr. Handson Yip
NURC
Viale San Bartolomeo 400
19126 La Spezia
Italy

1 Mr. Dale Reding/Ms. Sylvie Martel/Dr. Yaw Asiedu
NC3A
Oude Waalsdorperweg 61, 2597 AK The Hague, Netherlands
P.O. Box 174, 2501 CD The Hague, Netherlands

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. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) DRDC Atlantic	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).

A recursive engagement simulation tree (REST) for use in maritime defence

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

Brian Mury and Bao Nguyen

5. DATE OF PUBLICATION (month and year of publication of document) October 07	6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc). 63	6b. NO. OF REFS (total cited in document) 12

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).

Technical Memorandum

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

Defence R&D Canada – Atlantic
PO Box 1012
Dartmouth, NS, Canada 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). 1GO	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).
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.) DRDC Atlantic TM 2006-096	10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

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

(x) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) 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).

In support of the Integrated Signature Management (ISM) program at DRDC Atlantic, the Operational Research group is developing a model to assess the benefits of signature reduction of a defending ship. Within this mandate, we have built a stochastic simulation to determine the Measures Of Effectiveness (MOEs) of a warfare scenario where a ship defends against multiple threats with multiple defensive systems. The simulation is designed with the aim to compare the MOEs of a Canadian Patrol Frigate (CPF) as a function of distinct signatures. It can be used to analyze torpedo defence, mine defence, maritime air defence as well as soft kill and hard kill capabilities. In addition, it can assess the trade off in terms of vulnerability of the defence between the reduction of one of its signatures and the rise of another. This report describes the implementation of the underlying algorithms which make use of the concept of recursivity and the notion of a decision tree. To demonstrate the applicability of this simulation, we provide examples where a ship defends against multiple threats with multiple interceptors.

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).

Integrated signature management
Decision tree
Recursion
Engagement
Detection
Radar signature
Acoustic signature
Electromagnetic signature
Infrared signature
Missile
Torpedo
Canadian patrol frigate
Limiting angles of approach

This page intentionally left blank.