Physical modelling of naval infrared decoys in TESS and SE-WORKBENCH-EO for ship self-protection

C.R. Viau[a], I. D'Agostino[a], T. Cathala[b]
[a]Tactical Technologies Inc., 356 Woodroffe Ave., Ottawa, ON, CAN, K2A3V6
[b]OKTAL Synthetic Environment, 11 avenue du Lac, Vigoulet-Auzil, FRA, 31320

ABSTRACT

Electronic countermeasures (ECM) are an essential part of platform protection and survivability. Developing effective countermeasure starts with an in-depth understanding of the physical interactions between the threat, the target, the countermeasures and their environment. A significant amount of this knowledge can be acquired using physics-based modelling and simulation tools. To address the rapidly evolving requirements for physics-based simulation tools, a collaboration was established between Tactical Technologies Inc. (TTI) and OKTAL-SE with the purpose of coupling OKTAL-SE's simulation suite SE-WORKBENCH-EO and TTI's Tactical Engagement Simulation Software (TESS). TESS simulations are physics-based tools developed in the MATLAB/Simulink framework that enable users to analyze, evaluate, understand and optimize the effectiveness of ECM against threat weapon systems. The primary motivation for coupling the two simulation packages was to create a complete set of powerful research tools capable of generating complex and realistic electro-optical (EO) and infrared (IR) synthetic environments (air, sea and land) in support of IR countermeasure systems development, tactics and effectiveness assessment against the latest generation of imaging IR (IIR) guided weapons.

The first phase of this project was to develop an interface between TESS and SE-WORKBENCH-EO, which was successfully accomplished. The next phase addressed the specific coupling with TESS ASM(IR) simulator and the development of naval IR decoys in SE-WORKBENCH-EO. This paper first focuses on the physical modelling of the naval IR decoys in TESS (ejection, ignition, bloom, sustain, and decay phases, deceleration due to drag, components that make up the IR signature) and how these parameters are interfaced to SE-WORKBENCH-EO. This paper also discusses user input parameters as well as output parameters used to assess the effectiveness of the countermeasure. Typical deployment tactics for ship self-protection are used to visually demonstrate the naval IR decoys and their effectiveness against IIR guided anti-ship missiles.

Keywords: naval countermeasures, imaging infrared seeker, decoys, TESS, SE-WORKBENCH, modelling, simulation

1. INTRODUCTION

The anti-ship missile (ASM) has been a serious threat to naval platforms since the Second World War and has been reportedly operational in many international conflicts from the Cold War to the Syrian Civil War[1]. ASMs can be launched from land, air, naval and subsurface platforms and have effective ranges in the order of hundreds of nautical miles. Modern ASMs can travel at supersonic speeds (Mach 1-3), leaving very little time for the target platform to react. Typically, ASMs use a combination of inertial navigation, GPS, or terrain contour matching (while traveling over land) during the mid-course phase. In the terminal phase, ASMs may use a single or multi-mode seeker (active radar, laser-guided, IIR) to autonomously guide the threat to the desired target. Naval platforms, in comparison to land and air platforms, are particularly vulnerable to guided missiles because of the large radar cross sections, electromagnetic and thermal signatures as well as their relatively slow speeds. The design of modern naval platform has been greatly driven by the emergence and the advancements of the ASMs. Stealth technologies integrated into the platform design aimed at reducing signature and detectability have significantly improved platform survivability. Understanding and suppressing sources of radiation is a preventive and essential step in maximizing survivability. Reducing a platform's signature decreases the chances of acquisition as the threat transition from mid-course to terminal phase. However, reducing detectability and susceptibility of a platform may not be sufficient to ensure survivability. Other concealment and deceptive techniques (countermeasures) are required to defeat the modern ASM threat.

Naval countermeasures can be grouped into two categories: hard-kill and soft-kill. Hard-kill countermeasures attempt to physically damage the approaching threat and include missiles, close-in weapon system (CIWS) and currently in development laser-based directed energy weapon (DEW) and high-energy lasers (HEL). Although naval platforms are fitted with hard-kill countermeasures, they are not as commonly used as soft-kill and are considered a last line of defense. Soft-kill countermeasures attempt to conceal the exact location of the platform and/or deceive the approaching threat by introducing errors in its tracking loops. The choice of soft-kill countermeasures is dependent on the type of guidance system used by the threat. Chaff and radio frequency (RF) decoys (both passive and active) can be effective against radar-guided systems while multispectral obscurant smoke, flares and IR decoys can be effective against laser-guided and IR-guided threats (see Figure 1).

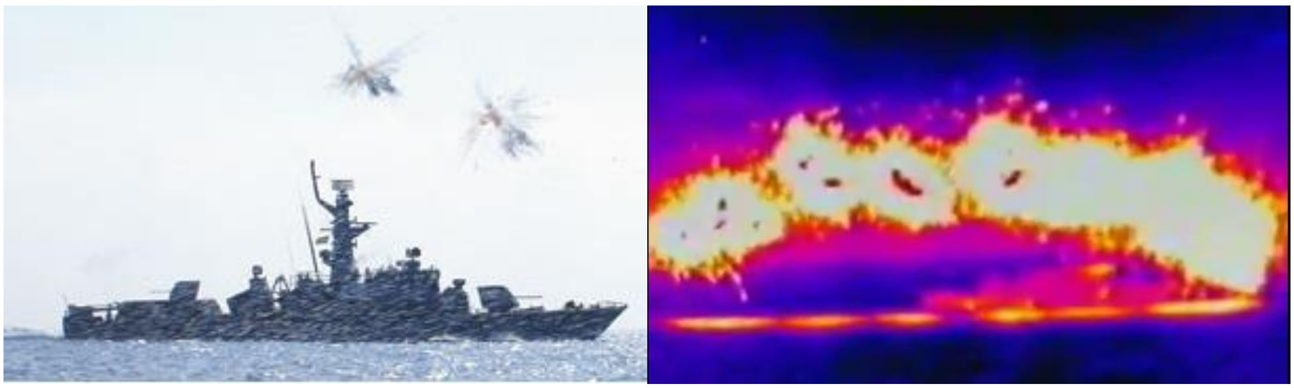

Figure 1 - Chaff[2] burst (left) and IR decoy[3] soft-kill countermeasures

In order to develop countermeasure systems and tactics, scientists and engineers require an in-depth understanding of the physical interactions between the systems, their environment and their overall effectiveness. This knowledge is typically acquired using physics-based simulation tools, hardware-in-the-loop simulators and reverse engineering when available. The system modelling and simulation (M&S) phase is a critical step in the development process where the developer can gain a broad understanding of the algorithm, tactic or system's capabilities and vulnerabilities. The M&S process allows the developer to conduct trade-off studies, performance analyses and virtually an unlimited number of scenarios that cannot be conducted in field trials due to the cost and time restrictions. A physics-based M&S tool allows developers to optimize and assess their algorithms, tactics and systems effectiveness prior to hardware development and field trials.

A physics-based simulation tool called Tactical Engagement Simulation Software (TESS™) was originally developed in mid 1990s to support radar and IR countermeasure analysis work for the international defense community. TESS simulations are physics-based, developed in the MATLAB® and Simulink® framework and enable users to analyze, evaluate, understand and optimize the effectiveness of countermeasures against threat weapon systems. TESS distinguishes itself from other simulation products with its available source code which allows users to inspect, verify and modify any of the underlying mathematical equations and algorithms of the simulation. In 2006, TESS was extended to simulate IIR threat systems and countermeasures and in 2013 a research and development collaboration was established to couple TESS with OKTAL-SE's SE-WORKBENCH-EO. The primary motivation for coupling the two simulation packages was to create a complete set of powerful research tools capable of generating complex and realistic electro-optical (EO) and IR synthetic environments (air, sea and land) in support of countermeasure systems development, tactics and effectiveness assessment against the latest generation of IIR guided threat systems.

This paper focuses on the physical modelling of naval IR decoys in TESS and SE-WORKBENCH-EO. This paper provides an overview of the simulation tools, and discusses user input parameters and simulation outputs used to assess the effectiveness of the countermeasures. Typical deployment tactics for ship self-protection are used to visually demonstrate the naval IR decoys and their effectiveness against IIR-guided ASM.

2. OVERVIEW OF TESS ASM(IR)+

TESS ASM(IR)+ is a member of the TESS Sea IR family of physics-based simulation tools. ASM(IR)+ models closed-loop engagements and interactions between a ship platform and a sea or air launched IR-guided (imaging) anti-ship missile. To defend itself from the incoming threat, the maneuvering ship can deploy distraction or seduction IR decoys with optional sub-munitions to achieve a "walk-off" effect. In long range engagements, a mid-course estimation option optimizes the missile's flight path to achieve maximum range. ASM(IR)+ models most phases of the engagement but the dynamic part of the simulation starts in the terminal phase with the missile seeker turned-on in either Search or Track mode and operating autonomously until end-game.   Measures of effectiveness such as miss distance, probability of kill and probability of survival are computed at the end of each simulation run. Like other TESS simulations, ASM(IR)+ is built in the MATLAB/Simulink environment and, with its available source code, users can review, inspect and modify any of the underlying models and algorithms. A front-end database allows the user to define and store data libraries of Targets, Countermeasures and Threats. A programmable batch runner is included for executing batch runs (Monte Carlo) of simulated tactical engagements.

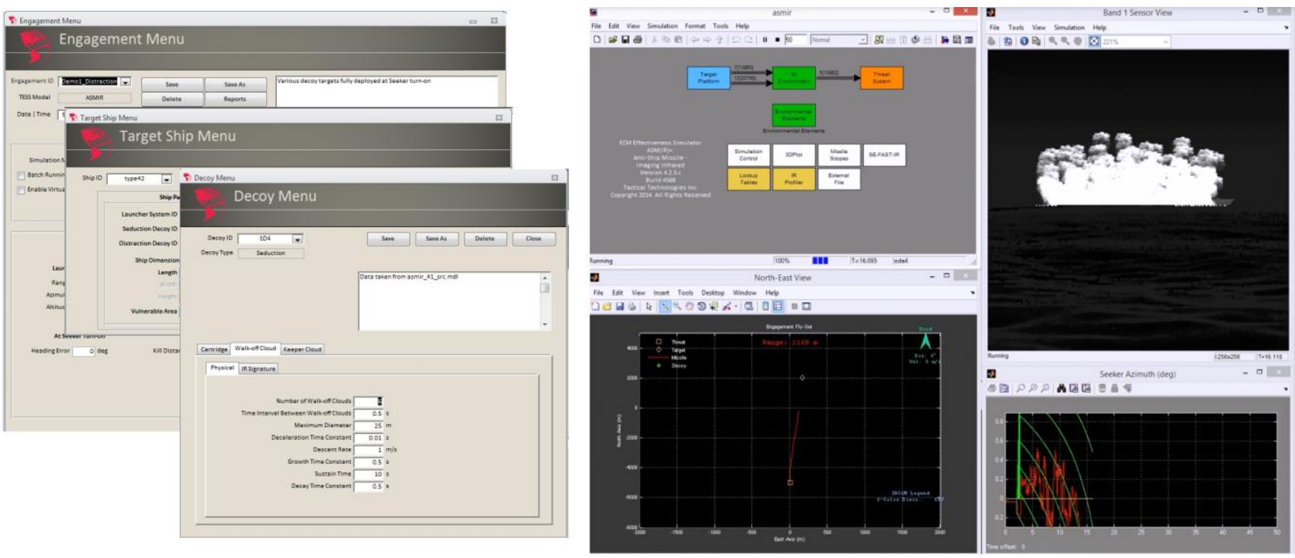

Figure 2 - TESS ASM(IR)+ Master Interface (left) and Simulink model with typical displays (right)

The coupling[4] to OKTAL-SE's software suite allows TESS to leverage the power of SE-WORKBENCH in order to generate complex and realistic IR synthetic environments (land, air, naval) with SE-FAST-IR or SE-RAY-IR. Jointly, the tools provide engineers and researchers the ability to model closed-loop engagements and physical interactions between a target platform deploying countermeasures and an IIR guided missile system.

SE-WORKBENCH-EO is a complete suite of tools used by the international research community to perform multi-sensor simulations. It enables the user to create virtual and realistic multispectral 3D scenes that may contain several types of targets (and now countermeasures), and then generate the physical signal received by a sensor, typically an IR sensor. In the SE-WORKBENCH-EO workshop, SE-FAST-IR is a set of physics-based software and libraries that allows preparing and visualizing 3D databases in real-time for the EO domain. SE-FAST-IR makes an intensive use of OpenGL Shader state-of-the art technology. Most of the physics computations are processed on the graphics board, resulting in a minimal CPU load that thus remains available for other simulation tasks. SE-RAY-IR is a ray-tracing software, which enables the computation of highly realistic images in both the visible and the IR spectrum resulting from complex scenario runs that include 3D terrain, 3D targets, atmospheric conditions, sensors models and associated trajectories.

3. DECOY MODELLING IN TESS AND SE-WORKBENCH-EO

3.1 Distraction Decoys

A distraction decoy is intended to delay or prevent the acquisition of the target ship by the IIR seeker as it transitions to autonomous guidance in its terminal phase. When distraction decoys are selected, the simulation runs start with the decoys fully bloomed. The decoy deployment and bloom phases are not modeled in this type of engagement. Some of the user input parameters associated with distraction decoys include deployment tactic parameters (number of decoys, deployment altitude range and angle) and decoy physical parameters (maximum size, descent rate and sustain time).

The horizontal motion of the distraction decoy is dictated by the wind while the vertical motion is governed by the gravitational force. The decoy position [*D*] is first computed in the target ship's local frame of reference $(]^S)$ as follows:

$$[D]^S = \begin{bmatrix} LR\cos\varphi_L \\ LR\sin\varphi_L \\ DA \end{bmatrix} \qquad (1)$$

where *LR* is the launch range (m), *DA* is the deployment altitude (m) and φ_L (rad) is the horizontal deployment angle with respect to the ship's bow. The decoy position is then transformed into the North-East-Down or local frame $(]^L)$ using the ship's orientation to construct the appropriate Euler transformation matrix[5(p75)] $([TM]^{SL})$. The final decoy position in the local frame $([D]^L)$ is computed by integrating and adding the wind vector $[S_W]^L$ as follows:

$$[S_W]^L = \int \begin{bmatrix} V_W^X \cos\varphi_W \\ V_W^Y \sin\varphi_W \\ 0 \end{bmatrix} dt \qquad (2)$$

$$[D]^L = [TM]^{LS}[D]^S + [S_W]^L \qquad (3)$$

The decoy's vertical motion is computed by subtracting the integrated user-defined descent rate from the initial decoy deployment altitude (DA).

The user-defined sustain and decay characteristics of the clouds are passed from ASM(IR)+ to SE-WORKBENCH-EO where they are used to drive the dynamic dimensions of decoys.

3.2 Seduction Decoys

A seduction decoy is used to lure the IIR seeker's tracking point away from the targeted ship. A group of seduction decoys can also be used to create a temporary curtain of obscurant smoke to conceal the exact position of the target from the approaching threat. In TESS ASM(IR)+, the seduction decoy can also be used in a series of coordinated smaller bursts followed by a final larger burst effectively creating a "walkoff and keeper" effect. The modelling of the seduction decoys consists of three primary subsystems, a launcher, a cartridge and the cloud of pyrophoric material. Each of these subsystems is described in the following sections.

3.3 Launchers

In ASM(IR)+, any number of fixed or trainable launchers can be created and positioned on the ship target platform. For fixed launchers, the azimuth pointing direction and number of barrels per launcher can be defined. For the trainable launchers, the azimuth and elevation gimbal limits and number of rounds per launcher can be characterized. The programmable batch runner permits the user to create, customize and optimize decoy deployment algorithms and tactics using system and engagement parameters such as wind speed and direction, launcher positions and coverage, number of rounds, deployment range and firing interval.

3.4 Cartridges

The motion of the cartridges is modeled from the time of the initial ejection up until the final sub-munition burst using a series of differential equations. Following the ejection, the motion of the undetonated cartridge follows a ballistic trajectory affected by drag and gravitational forces. In the case of a walkoff/keeper decoy round, the cartridge releases sub-munitions that ignite at user-defined intervals to create multiple decoy clouds along the flight path of the cartridge. Some of the input parameters available to characterize the motion the cartridges include ejection velocity, caliber, mass, average drag coefficient and interval burst times.

The forces experienced by the cartridge in flight are the drag force and gravitational force. The forces $[F_D]^C$ are modelled in the cartridge's frame of reference $(]^C)$ using the conventional drag equation[6] and standard gravity at sea level:

$$[F_D]^C = \begin{bmatrix} -\frac{1}{2}\rho v_x^2 C_D A_C - 9.81\sin\theta_{CL} \\ 0 \\ 9.81\cos\theta_{CL} \end{bmatrix} \quad (4)$$

where ρ is the air density at sea level, C_D is the drag coefficient, A_C is the cartridge cross sectional area and θ_{CL} is the elevation angle of the cartridge with respect to the horizontal plane. From equation (4), the acceleration in the cartridge frame of reference $[A_D]^C$ is determined by dividing with the payload mass. The velocity vector $[V_D]^L$ is obtained by integrating the acceleration and performing a coordinate transformation $[TM]^{LC}$ from the cartridge frame $(]^C)$ to the local NED frame of reference $(]^L)$. The velocity of the cartridge is integrated once more to obtain the actual position $[S_D]^L$ in the local frame of reference.

$$[V_D]^L = [TM]^{LC}[V_D]^C \quad (5)$$

$$[S_D]^L = \int_{IC} [V_D]^L dt \quad (6)$$

The subscript *IC* indicates the *Initial Conditions* of the integration which are the physical location of the launcher in the local frame.

The cartridge orientation throughout the ballistic flight is derived from the acceleration vector[5(p299)]:

$$\begin{bmatrix} A_X \\ A_Y \\ A_Z \end{bmatrix} = \begin{bmatrix} \dot{V} \\ \dot{\varphi}_{CL} V \cos\theta_{CL} \\ -\dot{\theta}_{CL} V \end{bmatrix} \quad (7)$$

where V is the magnitude of the velocity vector and θ_{CL} is the elevation angle from the previous time step. It is assumed that the roll of the cartridge has little impact on the trajectory and as a result the A_X is set to 0. The azimuth (φ_{CL}) and elevation (θ_{CL}) angles are obtained by isolating the angular rotation rates ($\dot{\varphi}_{CL}, \dot{\theta}_{CL}$) and integrating over time:

$$\begin{bmatrix} \varphi_{CL} \\ \theta_{CL} \end{bmatrix} = \begin{bmatrix} \int \frac{A_Y}{\overline{V}\cos\theta_{CL}} dt \\ \int \frac{-A_Z}{\overline{V}} dt \end{bmatrix} \quad (8)$$

Alternatively, the motion of the cartridges can be extracted from a user-defined lookup table using altitude and range as a function of time after deployment.

3.5 Decoy Cloud Motion

The motion of the seduction decoy cloud after a burst is modeled similarly to the distraction decoy (see equations 2 and 3) with two exceptions. The first is that the initial position of the decoy cloud is based on the location of the cartridge at the time of the burst. The second difference is that the decoy has an initial velocity vector pointing in the direction of the cartridge. The deceleration of the growing cloud is modeled using an exponential decay curve with a user-defined time constant. In addition, a user-defined descent rate is applied to control the vertical motion of the decoy.

3.6 Decoy Cloud Dimension

The cloud dimensions are approximated using a linear model with user-defined growth, sustain and decay times. These values as well as a maximum cloud diameter are passed from TESS to SE-WORKBENCH-EO to control the dimensions of the growth, sustain and decay phases of each decoy.

3.7 Building Decoys with SE-WORKBENCH-EO

The challenge to building a decoy in SE-WORKBENCH-EO is a challenge typical of physical system modelling. That is, how to model a complex real-world system with the limited resources of a computer model. The first step in the process is to identify and understand what needs to be modeled. This requires that the system first be deconstructed down to its most basic components. As illustrated in the left-hand side of Figure 3, a decoy in bloom is composed of thousands of small, very hot particles, which as they burn up, leave an opaque trail of hot smoke. Both particle types exhibit a non-constant thermal profile over the particle lifetime. Particle density is greater toward the center of the decoy bloom, with all particles drifting away from the center of the burst at a rate determined by the initial charge detonation. It becomes immediately apparent that the decoy must be designed as a composite of several different particle types: hot burning spark-like particles, and larger, less hot, opaque smoke particles. Ideally, each decoy is modeled as a composite of thousands of burning particles, with each particle then emitting its own smoke trail as it burns up. This however is not practical from either a processing or programming standpoint. From a processing perspective, updating and redrawing thousands of particles each time step is too processor-intensive. Similarly, programming thousands of individual smoke particle systems would be overly complex. To this end it was decided to add a third element to the decoy composite, a single larger burning particle to model the particle-dense center of the decoy bloom. The larger and hotter core is well depicted in the right-hand side of Figure 3. Thus, the goal is to model the bloom core as a single dynamic large burning particle, model the less dense bloom extremity with a more manageable number of small burning particles, and model the smoke trail as a continuous emission of mid-sized smoke particles emanating from the decoy bloom core.

Figure 3 - Naval IR decoys in the visual[7] (left) and the IR[8] (right) spectrums

The physical modelling starts by choosing an object and material files to serve as the base particle designs. To model the burning decoy core and extremity particles, a fire material file (fire.mat) was selected from the SE-WORKBENCH-EO library of resources. The fire material object is modeled as a textured sphere, transitioning from orange to yellow to white from outer radius to center in the visible spectrum. Additionally, the material contains a modifiable thermal association which provides the flexibility to customize the material's heat signature. To model the smoke particles, a likewise smoke material file (smoke.mat) was selected from the SE-WORKBENCH-EO library of resources. This object is similar in construction to the fire object, but contains an RGB texture derived from a 2D image of a smoke cloud. Some thermal and physical modifications are performed to both particles using the SE-PHYSICAL-EDITOR, and by directly modifying the thermal influence files themselves.

The next step in achieving the design goal is to model the composite decoy directly in SE-SCENARIO, the scene generation tool of the SE-WORKBENCH-EO. Particle system design is enabled by using the SE-SPECIAL-EFFECTS product, an extension of SE-SCENARIO. This tool allows specifying and visualizing design constraints for the various decoy components. These parameters include specifying a particle model (billboard material files); establishing growth, transparency, and radiance profiles; specifying update rates, emission quantities, and emission rates; and a wide variety of other parameters. Instant design feedback is possible by animating the decoy in the various visible and IR contexts available in SE-SCENARIO.

3.8 Integrating Decoys into Simulink via the SE-TOOLKIT/SE-TK-PARTICLES

With a composite decoy design now constructed, the next step is to integrate the design into the TESS model. During typical usage of SE-SCENARIO, scene entities are designed and saved directly to the program scenario, as a scenario (*.scnx) file. This approach is well-suited to scenarios having a fixed number of decoys, but in order to make a dynamic range of decoys available to the user, it was opted to create and parameterize the decoys at runtime. This is made possible using the SE-TOOLKIT API and SE-TK-PARTICLES API. A complete listing of all SE-TOOLKIT and SE-TK-PARTICLES API functions is available from the SE-TOOLKIT SDK[9] and SE-TK-PARTICLES SDK[10] documentation. The SE-TK-PARTICLES API provides a special calling syntax to allow a source mex-file to create custom decoys at runtime using a specification provided via the TESS model. During the Initialization phase of model execution, a call is made from the mdlStart function of the source mex-file to a subroutine setting up each particle system belonging to each decoy. In total, a composite of three particle systems is created for each decoy: a particle system creating single core particle, a particle system creating a burst of small fringe particles, and a smoke trail particle system. Each particle system subroutine contains all of the necessary SE-TK-PARTICLES API commands for creating and setting the specifications particular to each particle system. This parameterization is a combination of user-specified inputs and predefined attributes.

Particle emission and maintenance form the two principle decoy management functions required at each iteration of the simulation loop. Single emission particle systems (core and extremity particles) are only enabled once according to the decoy bloom flag assigned to each decoy from the TESS Simulink model. Slightly more complex, particle systems having more than one particle emission (eg/ smoke trail) must continue to release particles throughout the lifetime of the decoy. Using the SE-TK-PARTICLES API, commands have been added to instruct each smoke particle system to release 2 to 3 new smoke clouds every 0.5 seconds. With these parameters set, the release of subsequent smoke clouds is automated by the scenario. To ensure that each new smoke cluster is released from the correct location in 3D space, the smoke particle system position must match the decoy position from the Simulink model. To achieve this, at each sample time, decoy XYZ-positions are retrieved from the S-function input port, and assigned to the corresponding decoy using an SE-TOOLKIT API command (seTkEntitySetLocalTransformation).

Particle maintenance is required for all existing particles in the scenario, and defines the manner in which particle position is updated. While handles to parent particle systems, like other scene entities (ship, sensor, etc.) are easily accessed and updated via the SE-TOOLKIT API, particles themselves are numerous (hundreds of spark particles and dozens of cloud particles per decoy), and not within the scope of the SE-TOOLKIT API. In order to access the handle to an existing particle, a scheduled interrupt program is created and defined in the SE-TK-PARTICLES as a Particle System User Program. This user program allows controlling particle updates via a callback interrupt routine, instead of automating particle motion according to fluid dynamics principles. Since wind effects and decoy position are driven by the TESS Simulink model, automated particle motion is not suitable for this application. To add a user program to a particle system, a user program is named and created via the SE-TK-PARTICLES API function seTkParticleSystemSetUserProgram. The corresponding user program function is then created in the source mex-file, and the particle system update rate is set to match the sensor sampling rate defined in the TESS model. This instructs each particle to access its associated user program at an update rate interval defined by the parent particle system.

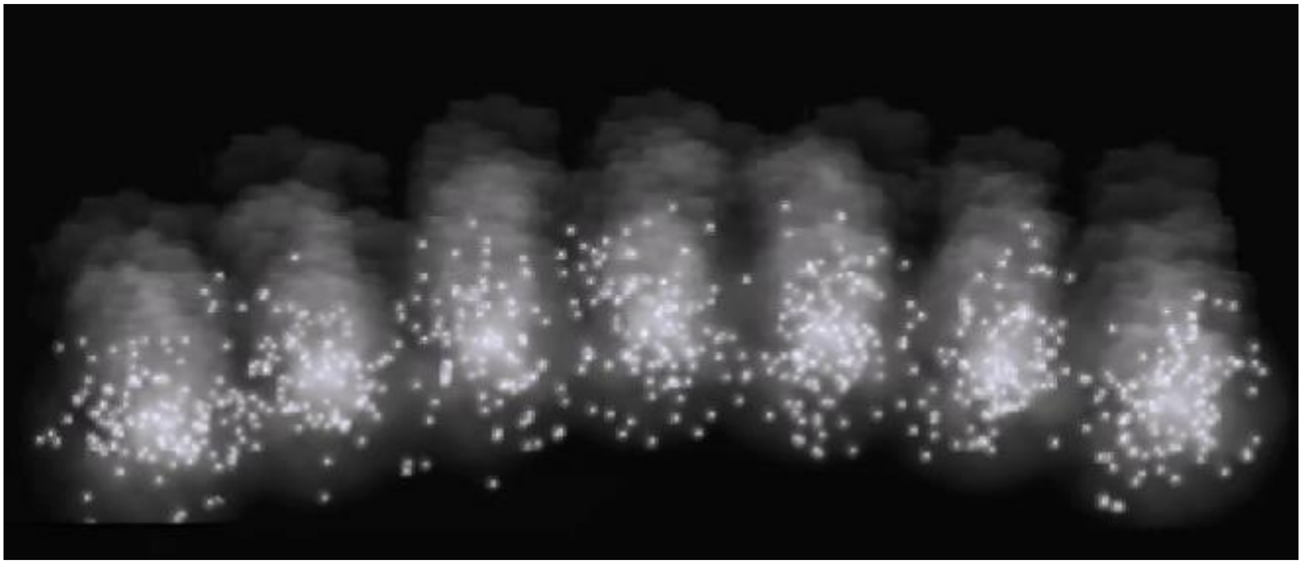

Figure 4 - Naval IR decoys in TESS ASM(IR)+

For the large core particle of each decoy, a system-specific user program is created to reposition the particle at each sample time. This position is defined by the decoy xyz-coordinates provided by the TESS model. The particular challenge when relying on a User Program is that there is no way to identify from the interrupt callback which decoy the particle originated from. Compounding this is the fact that the User Program callback does not have access to the mex-file SimStruct, meaning the program does not have access to block inputs or parameters. To overcome this issue, emphasis is placed on SE-TOOLKIT API resources that the user program does have access to. First, the current position of the particle is retrieved (seTkUserProgramGetParticleAbsoluteTransformation). Next, the positions of all other decoy particle systems in the scenario are determined (seTkScenarioGetEntity). Since it is known that each smoke particle system is kept aligned with its corresponding Simulink model decoy position, a comparison is made between the position of the core particle and all decoy smoke particle systems in the scenario. By determining which smoke particle system is most proximal, the required coordinates for the core particle position can be calculated.

For smoke and spark particles, the User Programs for these particles are much more straight-forward. Here, each User Program must add the effect of wind and ejection velocity to the current particle position, and for spark particles only, calculate the displacement due to descent rate. First, the particle position is retrieved via the SE-TOOLKIT API as was described for the core particle User Program. Then, the fixed descent rate (spark particles only), wind rate, wind direction and the deceleration speed of the decoys are all retrieved directly from the TESS model (using the mxGetPr and mexPutVariable commands from the MEX Library). Since the sample rate is known, the displacement of the particle is easily calculated and its position updated.

3.9 TESS Connections to Scenario Decoys

In order to define and control the physical characteristics and behavior of the decoy in the SE-WORKBENCH-EO scenario, user inputs are required via the TESS model. This data is transmitted in several ways: via direct input to the Simulink S-function block itself; via the S-function block mask (S-function parameters); and via access through the MATLAB Base workspace. Figure 5 highlights the decoy input connections to a fully integrated model S-Function in TESS ASM(IR)+. The following parameter list details all TESS user input used to construct the decoy in SE-WORKBENCH-EO, and the method by which this data is transmitted.

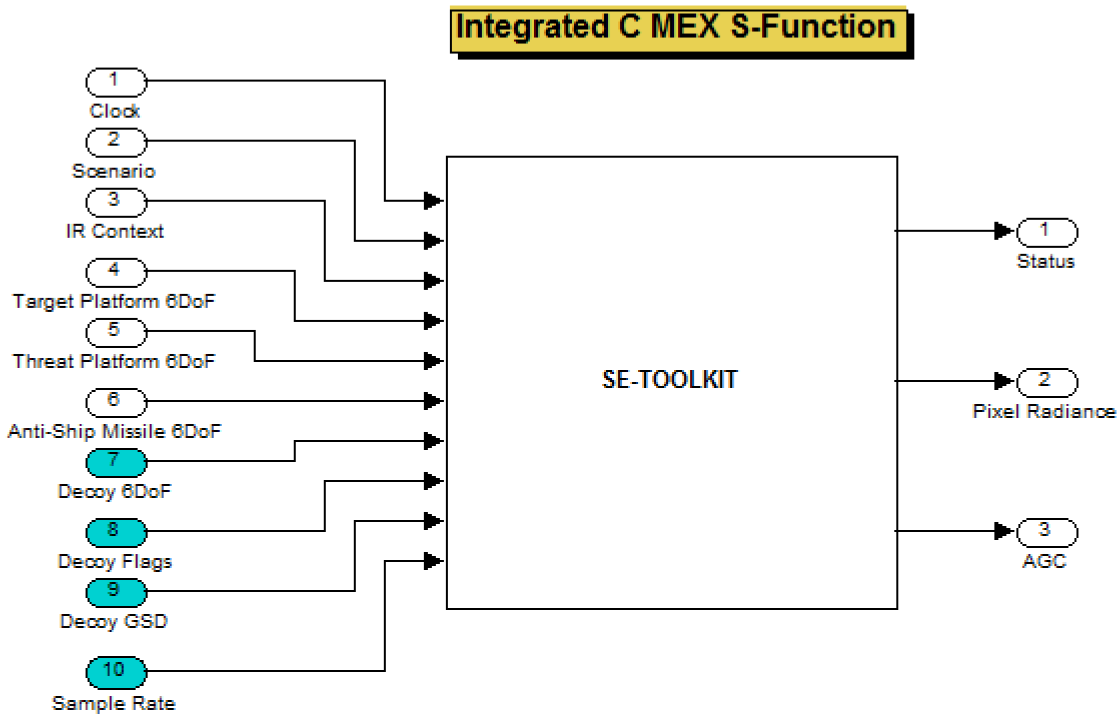

Figure 5 – Decoy inputs to an integrated C MEX S-Function

Decoy 6DoF – This data is transmitted via direct feedthrough to an S-function input port. This dynamic data is updated every major model time step to control the decoy position in 3D space.

X-coordinate (m)	The x-position (East-West) of the decoy, relative to the origin (0,0,0)
Y-coordinate (m)	The y-position (North-South) of the decoy, relative to the origin (0,0,0)
Z-coordinate (m)	The altitude/z-position (Up-Down) of the decoy, relative to the origin (0,0,0)
Azimuth (rad)	The heading of the decoy, relative to North
Elevation (rad)	The elevation of the decoy, relative to horizontal
Roll (rad)	The decoy roll, relative to level flight

Decoy Flags – This data is transmitted via direct feedthrough to an S-function input port. This dynamic data is updated every major model time step to track the various launch and bloom states of each decoy.

Launch Flag	1 if the decoy cartridge has launched, 0 if not
Bloom Flag	1 if the decoy has burst from the cartridge, 0 if not
Size Flag (m)	Current decoy size (not used)

Decoy Size – This data is transmitted via direct feedthrough to an S-function input port. This user-entered data remains static throughout the simulation, establishing decoy growth and motion constants.

Growth Time (s)	The time period where the decoy is growing, beginning at decoy burst and ending when the decoy maximum size is reached

Sustain Time (s)	The time period where the decoy sustains its maximum size
Decay Time (s)	The time period where the decoy decreases in size, until the decoy lifetime has elapsed
Maximum Diameter (m)	The decoy maximum diameter
Descent Rate (m/s)	The rate at which the decoy descends

Sample Time – This data is transmitted via direct feedthrough to an S-function input port. This user-entered data remains static throughout the simulation, establishing the sample rate for the S-function.

Sensor Sample Rate (Hz)	The sensor sample rate, which is used as the update rate of the decoy animation.

Wind Parameters – This data is transmitted via a mexMatlab function call from the binary mex-file to the MATLAB Base workspace. Wind data is directly accessed from a model Environment block. This data is used by decoy interrupts where access to the SimStruct is not available.

Wind Rate (m/s)	The magnitude of the wind vector
Wind Direction (rad)	The heading of the wind vector, relative to North

4. ENGAGEMENT ANALYSIS

4.1 Distraction Decoys

As discussed in sections 3.1and 3.2, naval IR decoys can be used pre-emptively (distraction) or reactively (seduction) when the seeker's track point is locked onto the target ship. Figure 6 illustrates two distraction decoys fully bloomed at the start of a simulation run. The automatic gain control (AGC) of the seeker is set relatively high and as a result, the presence of the target ship and the background radiance (between the two decoys) are not visible in this image. The use of distraction decoys has several effects on the seeker's detection capability. In this simulated engagement, only two distraction decoys were used but in a real scenario, likely several more decoys would be deployed forcing the seeker to consider and assess each of them as the potential true target. A greater number of targets to evaluate can delay the acquisition and lock-on process and increase the chances of an incorrect decision. If the target discrimination algorithm uses some form of centroid tracking, the presence of the decoys may introduce enough error in the tracking loop to prevent a direct hit. If the threat system uses a more sophisticated tracking algorithm based on image and pattern recognition, the distraction decoys may only be effective if they are positioned in the line-of-sight to the approaching threat where they obscure some parts of the ship. If deployed correctly, the distraction decoys may occlude enough of the ship to prevent the seeker from recognizing distinct features and establishing a positive track.

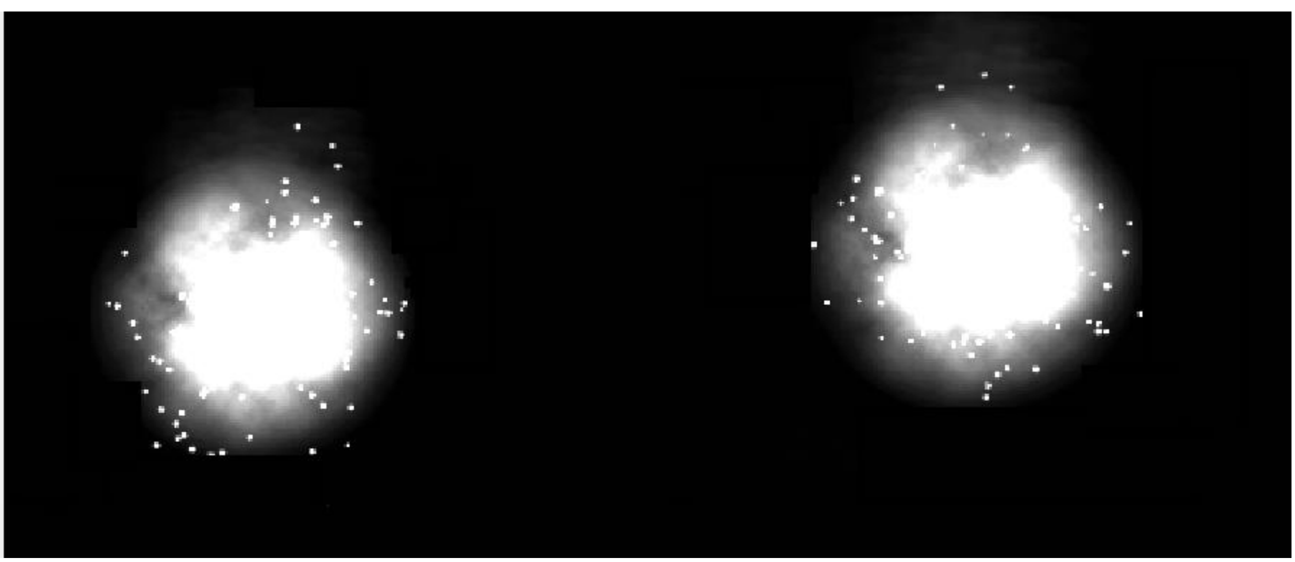

Figure 6 - Two distraction decoys fully bloomed at the start of a simulation run

An engagement run was carried out in TESS ASM(IR)+ to illustrate the effects of distraction decoys on the seeker's ability to lock-on to the target ship. In this engagement, two distraction decoys were deployed approximately 60 meters away from the ship at +/- 45° with respect to the bow of the ship. The threat system was approaching from 10° angle of arrival (nearly head on) with respect to the ship heading and transitioned to IIR terminal guidance 6 km away. The simulated engagement started with distraction decoys fully bloomed and the seeker in search mode. At the start of the engagement, both decoys and the target ship were within the field of view of the threat. Some of the more relevant engagement scopes (Seeker Azimuth Angle, Seeker Az/El Rates, Missile Body Accelerations) were captured and are illustrated in Figure 7.

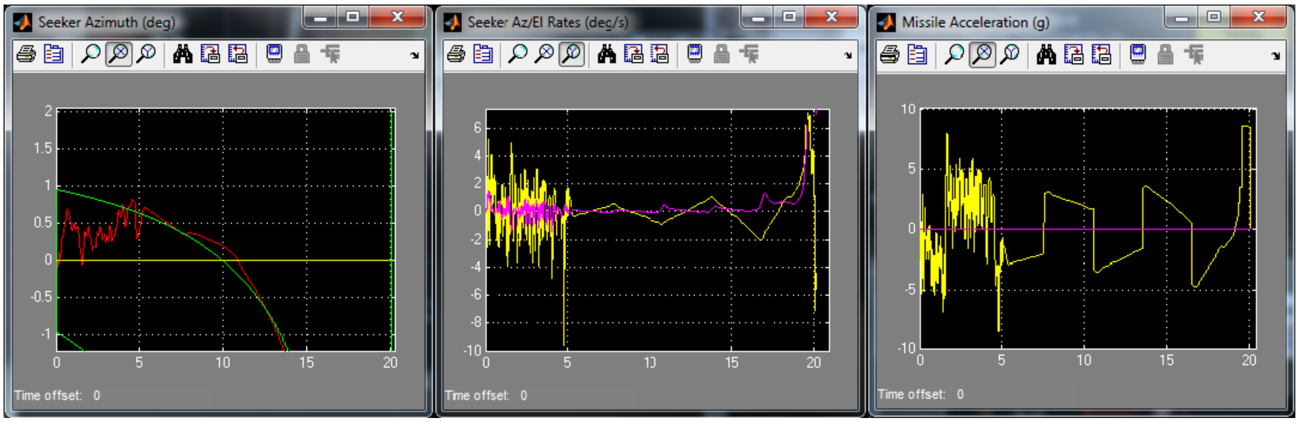

Figure 7 - Distraction decoy engagement results. Seeker Az angle (left), Az/El Rates (centre), Missile Body Accelerations (right)

The Seeker Azimuth scope (left) illustrates the seeker head (red trace) and decoy (green) azimuth angles relative to the target (yellow). From this scope, it can be observed that the presence of the decoy prevents the seeker from directly locking onto the target ship. For the first five seconds of the engagement, the seeker's tracking cell moves between the target and one of the decoys. This change in pointing direction results in an oscillation of the seeker azimuth gimbal (yellow trace in centre image) and consequently missile body acceleration commands (yellow trace in the right image). After the 5-second mark of the engagement, the seeker locked on to one of the decoys and tracked it until end game (this is observed in the Seeker Azimuth scope where the red trace settles and follows the green trace). Also visible in the Seeker Az Rate and Missile Acceleration scopes is the user-defined 2.5 g (6 sec period) terminal weave used to evade the potential hard-kill CIWS. The results of this engagement can vary significantly by changing the decoy deployment range and angles, wind speed and direction, or the type of the target recognition and tracking algorithm used by the threat system.

4.2 Seduction Decoys

In Section 4.1, distraction decoys were described as a means to prevent or delay the target ship acquisition from the threat system. However, in certain engagements, such as littoral environments where the threat may be launched from land and have an operator-in-the-loop, or when faced against a more advanced anti-ship threat (with multi-mode seeker or advanced IR counter-countermeasures), distraction decoys may not be effective. In such cases, other types of decoys or other deployment tactics are required. As the name describes, seduction decoys attempt to seduce and lure the threat system's tracking cell away from the target ship with the hopes of creating a break-lock.

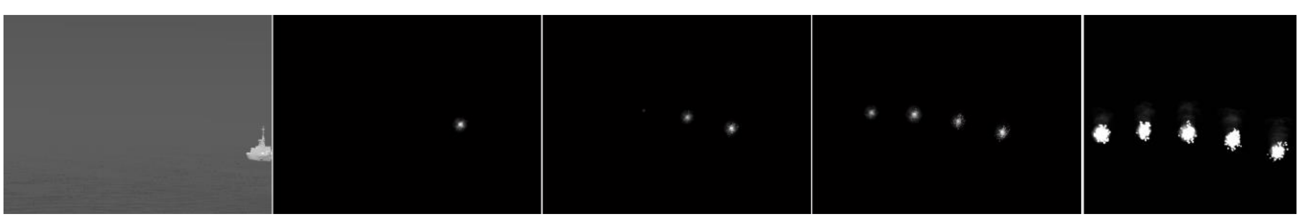

Figure 8 – Walk-off seduction decoy sequence in TESS ASM(IR)+

One common type of seduction decoy is that using a single round with sub-munitions to create a walk-off effect. In some cases, the walk-off clouds may be of shorter duration than the last (keeper) burst. The final keeper burst may also be of a much larger size and intensity in order to keep the seeker's track point locked on. A simulated deployment sequence of walk-off/keeper decoys in TESS ASM(IR)+ is illustrated in Figure 8. The AGC shading effect temporarily hides the ship and details of the sea and background following the first decoy burst. Figure 9 illustrates a walk-off deployment sequence at two different launch angles (20° and 45°).

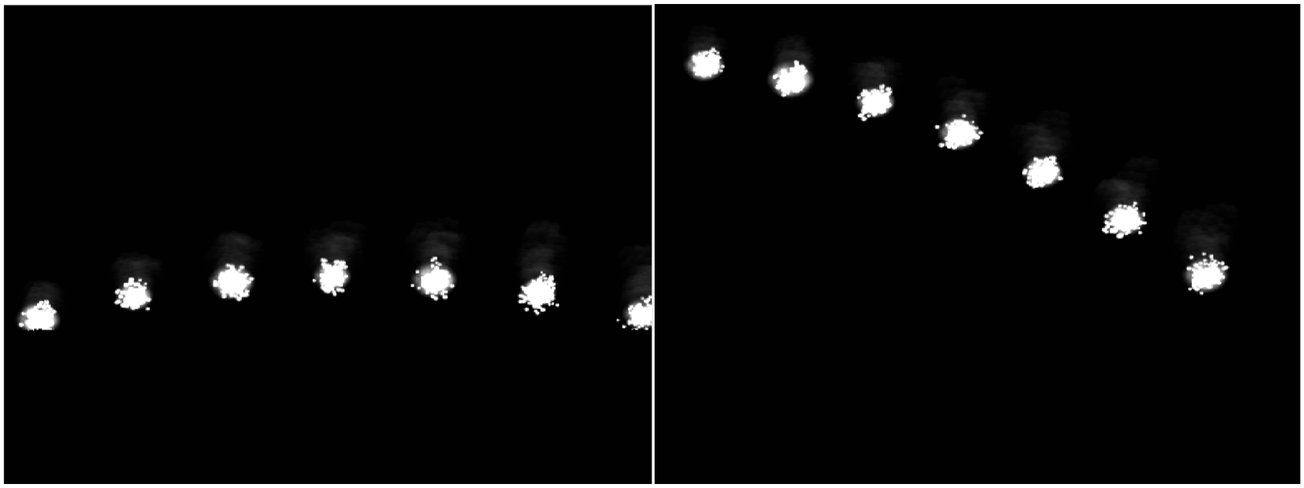

Figure 9 - Seduction decoy deployment at elevation angles of 20° (left) and 45° (right)

Another commonly used decoy deployment technique is illustrated in Figure 10 and is referred to as screening. A series of decoys are rapidly deployed in the line-of-sight between the target ship and the approaching threat system. The screen is used temporarily to hide the exact position of the ship and may allow it to perform an undetected change of direction. In the illustrated sequence of Figure 10, the ship speed and direction, the deployment range and angles as well as the position of the IR camera observing the scene are all coordinated to provide enough time for the ship to move out of the field of view undetected. The effectiveness of this deployment technique was assessed in TESS ASM(IR)+.

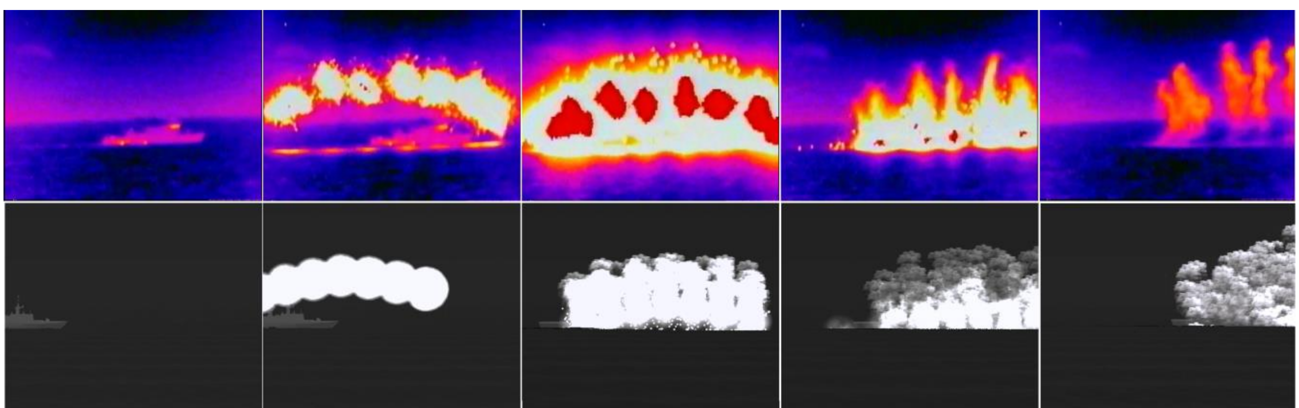

Figure 10 - Real[3] (top row) and simulated (bottom row) decoy deployment sequence to create a screening effect

In this experiment, the seeker's terminal phase was activated at 4.5 km away from the target at an angle of arrival of 90°. In the top sequence illustrated in Figure 11, the seeker has no difficulty detecting and tracking the target ship from the start of the terminal phase until end game. In the bottom sequence, the ship deployed seven decoys to create a temporary screen between itself and the approaching threat. In this example, the decoy deployment prevents the seeker from extracting the exact position and specific characteristics of the ship that may be used for image and feature recognition. The decoy screen in this scenario fills up the seeker's field of view at approximately 20 seconds into the engagement preventing the seeker from reacquiring the ship until end game. However, this deployment strategy nearly fails at approximately 17 seconds into the engagement (see 3rd and 4th image of the bottom sequence) as the ship starts to appear from behind the decoy screen. A seeker using feature detection such as edges or corners may have been able to reacquire the target at that point.

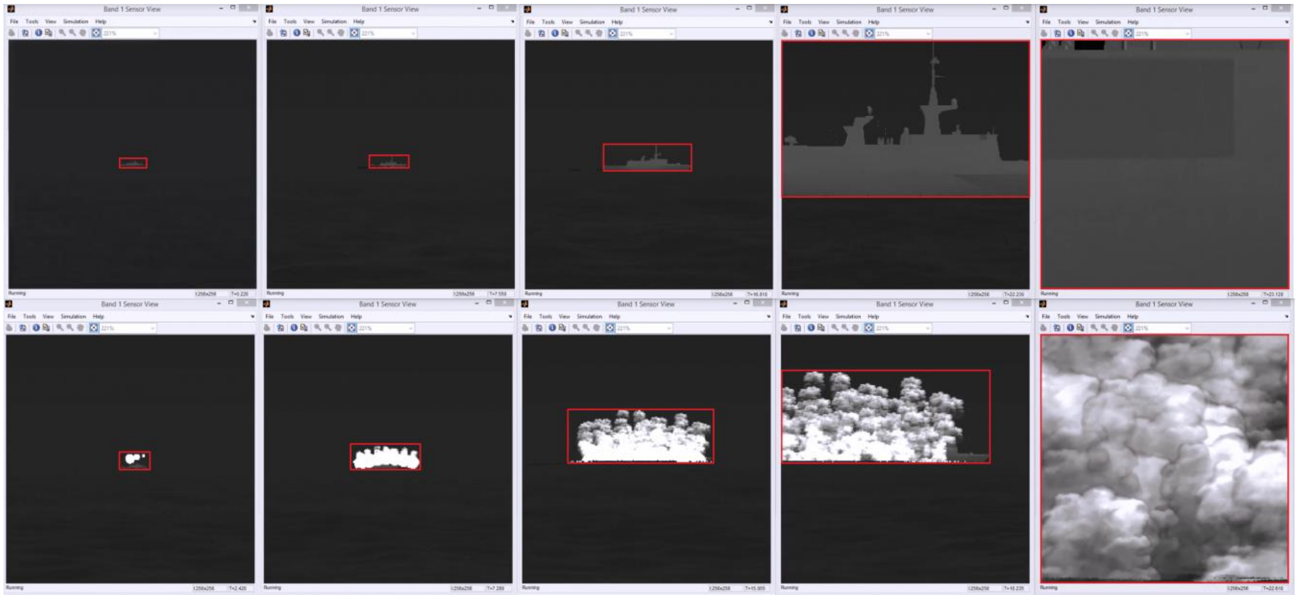

Figure 11 - Target-Threat engagement without (top row) and with naval IR countermeasures (bottow row) in TESS ASM(IR)+

This scenario depicts only one possible engagement geometry. The development of deployment tactics requires thousands of runs to be performed and analyzed to understand effective and vulnerable regions. TESS ASM(IR)+ was used to conduct Monte Carlo batch runs of 3 different deployment tactics and the results were used to generate the effectiveness plots displayed in Figure 12. These plots illustrate the probability of kill of the target platform as a function the threat direction of arrival and decoy deployment tactic. The blue regions indicate high probability of survival (successful countermeasure deployment tactic) while the red regions represents high probability of kill (unsuccessful countermeasure deployment tactic). The analysis tool provides an automated calculation of the kill zone (red) area and percentage. From these three deployment tactics, it is observed that the centre tactic shows a slightly better performance than the tactic in the left plot while the tactic used in the right plot shows a significant drop in effectiveness in comparison to the other two.

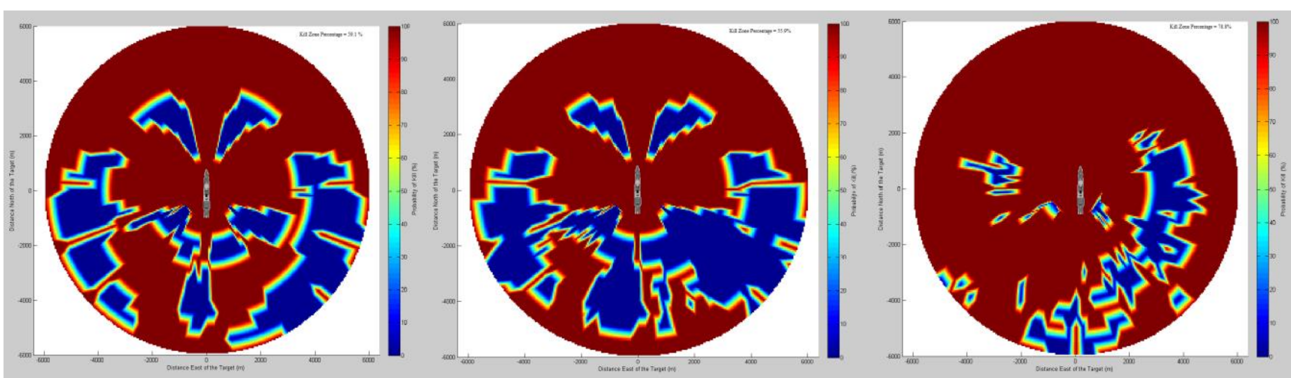

Figure 12 - Countermeasures effectiveness plots

5. CONCLUSION AND FUTURE WORK

The ASM has threatened both military and merchant ships for many years. With the technological advancements in focal plane arrays, detectors and electronic imaging, ASMs equipped with IIR terminal guidance capabilities will only become more lethal. Platform survivability is built on two fundamental concepts; signature management and effective countermeasures. Developing effective countermeasures starts with an in-depth understanding of the physical interactions between the threat, the target, the countermeasures and their environment. A significant amount of this knowledge can be acquired using physics-based modelling and simulation tools such as TESS. The main focus of this report was on the physical modelling of the decoy launchers, the cartridge trajectory and the decoy motion after a burst. This report also presented the approach used to create realistic representation of IR decoys in SE-WORKBENCH-EO.

The collaboration between TTI and OKTAL-SE first focused on the coupling of TESS ASM(IR)+ and SE-WORKBENCH-EO for the development and effectiveness assessment of naval countermeasures. The next part of this ongoing collaboration will be to build on this effort and potentially integrate SE-WORKBENCH-EO with other TESS IIR simulators such as TESS ILAPS (Integrated Land Active Protection System) and TESS SAAM(IIR) (Surface-to-Air and Air-to-Air Missiles). The advanced IR synthetic environment capabilities of SE-WORKBENCH-EO now provide the possibility to integrate other systems in TESS such as imaging DIRCM, missile approach warning systems (MAWS) and airborne/seaborne IR search and track (IRST) systems. The coupling with SE-FAST-IR now opens the possibility to use TESS with SE-FAST-HWIL in a hardware-in-the-loop configuration in support of DIRCM, MAWS, IRST and other EO/IR sensor development.

REFERENCES

[1] "Russia Provides Syria With Advanced Missiles - NYTimes.com.", <http://www.nytimes.com/2013/05/17/world/middleeast/russia-provides-syria-with-advanced-missiles.html?hp&_r=1&> (28 May 2014).

[2] "11/4_PassiveCountermeasures.", <http://www.frontline-defence.com/index_archives.php?page=1745> (28 May 2014).

[3] "Rheinmetall MASS - IIR / Laser -Screening - YouTube.", <https://www.youtube.com/watch?v=WPqSZh3QGac&list=PLFE88E1F348AB011E> (15 May 2014).

[4] Viau, C., D'Agostino, I.., Cathala, T., "Coupling of TESS with SE-WORKBENCH for EO/IR countermeasure development and effectiveness assessment," presented at 10th International IR Target and Background Modelling & Simulation Workshop, 23 June 2014, Ettlingen, Germany.

[5] Zipfel, P. H., Modeling and simulation of aerospace vehicle dynamics, American Institute of Aeronautics and Astronautics, Reston, VA (2000).

[6] "Drag (physics) - Wikipedia, the free encyclopedia.", <http://en.wikipedia.org/wiki/Drag_(physics)> (22 May 2014).

[7] "Test Firing Mark 36 SRBOC Chaff & Infrared Decoy System - YouTube.", <https://www.youtube.com/watch?v=CoduG_AkQeQ&index=7&list=PLFE88E1F348AB011E> (21 May 2014).

[8] "Rheinmetall MASS - AGC shading - YouTube.", <https://www.youtube.com/watch?v=vq7qTssx0Kw&list=PLFE88E1F348AB011E&index=10> (21 May 2014).

[9] OKTAL-SE., "SE-TOOLKIT SDK: Programming Manual."

[10] OKTAL-SE., "SE-TK-PARTICLES SDK: Programming Manual."