A MATLAB /Simulink methodology for simulating dynamic imaging IR missile scenarios for use in countermeasure development and evaluation

J. P. Tremblay, C. R. Viau

Tactical Technologies Inc, 356 Woodroffe Ave., Ottawa, On, K2A 3V6, Canada

ABSTRACT

The paper describes a methodology for characterizing the signatures of targets for Imaging Infrared (IIR) missiles and generating dynamic missile engagement scenarios using MathWorks tools (primarily MATLAB and Simulink). The over-all objective of this work was to develop high fidelity physics-based simulations of the attack of IIR missiles on targets that are using various types of countermeasures for survivability. While the methodology has been implemented in products used for analyses of both ship and main battle tank protection this paper focuses on the ship application.

The methodology involves a multi-step process. First the infrared signatures of the objects are characterized using a graphical tool that enables the user to select individual or groups of surfaces on the objects (targets and countermeasures) and specify their surface temperatures and spectral emissivities. Second, a dynamic IR scene generator creates the scene as viewed by the missile's seeker. Then an imaging IR seeker, using the option of several tracking algorithms, discriminates the target. Finally, the inclusion of dynamic models for missile guidance, aerodynamics and propulsion together with signal propagation enable the closing of the loop in the missile's fly-out. The simulation dynamically computes the distance between each surface and the missile seeker and uses the specified atmospheric attenuation profile to produce a simulated IR image at the seeker. This is processed using several optional tracking algorithms to generate steering signals. This process is repeated every time-step of the simulation and determines the trajectory of the missile and the hit or miss of the missile at engagement completion.

The paper includes the following topics: characterizing IR signatures, generating dynamic IR scenes, simulating representative close-loop missile fly-out engagements, evaluating performance and running simulation batches.

Keywords: IR scene generation, MATLAB, Simulink, Countermeasure, IR Seeker, Missile

1. INTRODUCTION

A project was initiated to complement an existing suite of electronic warfare (EW) countermeasure effectiveness simulations. The project consisted of developing a low-cost, physics-based software simulation of an infrared (IR) scene generator using The MathWorks product suite. The IR scene generator would be used for the purpose of developing and evaluating IR countermeasures against imaging-based threat systems. The immediate need for this application was in a naval anti-ship self-protection environment. Since then, the tools has been revised, improved and used in an Active Protection System (APS) simulation for land vehicles. For the purpose of this report, the examples and demonstrations of the IR scene generator focuses on the naval applications.

Other major constraints that were imposed on this project were that the software-based IR scene generator had to be generic, customizable and expandable. The generic constraint meant that the simulation had to be based on unclassified physics, use verifiable and open-literature algorithms, and did not represent or model any specific system. The customizable constraint implied that a third-party user could turn the generic system into specific system by entering classified parameters such as objects' IR signature or the sensor's operating wavelength. Finally, the expandable constraint meant that with very little effort new objects could be added, improved physical and optical effects could later be implemented and that the IR scene generator could be reused in other applications.

The remainder of this report provides a description of the naval application in which the IR scene generator is used, an overview of the process and the tools used to generate the IR scenes. The report concludes with a performance evaluation of the close-loop simulation and a short description of the ongoing development.

2. DEFINING THE IR ENVIRONMENT

The first step in simulating an IR engagement is to define the IR environment consisting of objects that require emissive properties and a medium in which radiation is allowed to propagate.

A tool was required to import 3D CAD objects into the MATLAB/Simulink environment, overlay an infrared signature and animate the object within a reasonably realistic environment. A review of existing CAD software led to the selection of AC3D product as the preferred CAD application for three reasons. First, it was an open file format, which meant that the structure format could be easily viewed and understood. Second, it has the ability to import more common files such as 3DStudio and AutocadDXF. Third, the AC3D product has a low license cost.

An import function was created that parsed the .ac file and set up a structure which could then be used by MATLAB's *patch* function to reproduce the 3D object in a MATLAB figure. Figure 1 is an example of an object in AC3D (left) and in TTI's IRProfiler (right):
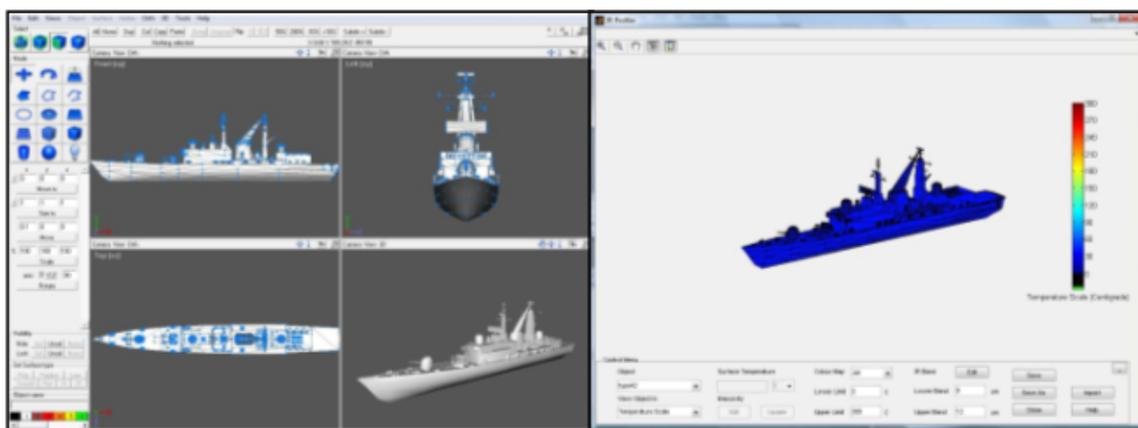

Figure 1: IR object editing

After an object is imported into the MATLAB environment, a data structure is created and used throughout the entire process. Table 1 lists the properties of the data structure. The information stored in the structure such as Vert, Face, and Center are used by the *patch* function to define the location of the object, while Emissivity, SurfaceTemperature, and Radiant Emittance are used to define the IR signature of the object within the image.

Name	Object name
SurfaceId	Vector uniquely identifying each object surface
RadiantEmittance	Radiant emittance associated with each surface (band 1 and band 2)
SurfaceTemperature	Temperature of each individual surface
Degrees	Temperature units currently selected
Fvcd	FaceVertexColorData vector for the MATLAB *patch* function (band 1 and band
Emissivity	Emissivity profile (as a function of wavelength) for each surface
Dimension	Object length, width height and size ratio
CMAP	MATLAB color map currently selected
Wavelength	Wavelength associated with the radiant emittance (band 1 and band 2)

REUpdate	Radiant Emittance update required
Vert	XYZ coordinate position of each vertices of each surface
Face	Vertice ID to create each surface
Center	XYZ coordinate position of the center of each surface
Area	Area of each surface
Normal	Normal vector of each surface

Table 1: IR object data structure properties

In the IR profiler, a user selects the desired temperature, *T*, and emissivity, *E(λ)*, of individual surfaces of the object to match the conditions in the desired engagement environment condition. The Radiant emittance, M_λ, is calculated by[1]

$$M_\lambda = \frac{2\pi hc}{\lambda^5[e^{\frac{hc}{\lambda kT}} - 1]} E(\lambda)$$

where h, c, and k are Plank's constant, the speed of light, and Boltzmann's constant respectively.

When describing the signature of a body, a user must take into account several sources that affect the body's radiance which include the body's thermal emission, the reflected skyshine, the reflected earthshine/seashine and the inter-reflections of the body's surfaces. The engagements are short and the IR signatures of the objects vary only with separation distance between the source and sensor through the duration of the simulation.

Other functions of the IR Profiler include the ability to import and duplicate 3D objects, view objects in radiant emittance or temperature scales, and compute the radiant emittance (W/m^2) at the source. Below is an example of a fictitious signature profile viewed in the temperature view:
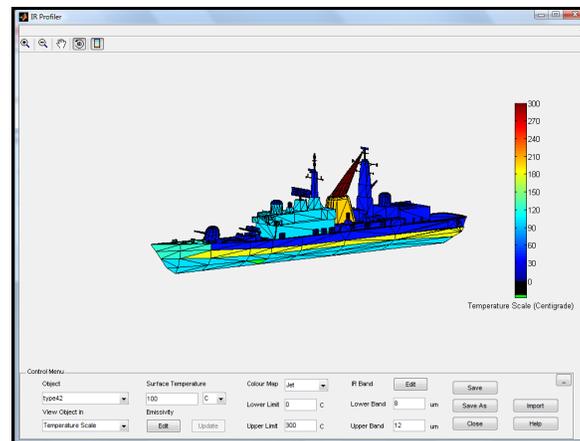

Figure 2: IR signature of a ship

3. SIMULINK MODEL

Once the desired IR signatures have been assigned, they are imported into the Simulink environment. The Simulink model is organized into subsystem blocks, grouped according to common functionality. The functional blocks visible in the top-level block diagram are color coded to indicate their associations. The orange block (on the right) representing the threat system contains the imaging IR seeker, the airframe and the threat platform. The light blue target platform block (on the left) contains the targeted platform (ship) and the infrared decoys. The lower green block models the environmental elements present in the engagement while the other green block models the atmospheric attenuation of the infrared signals between the target and the sensor.
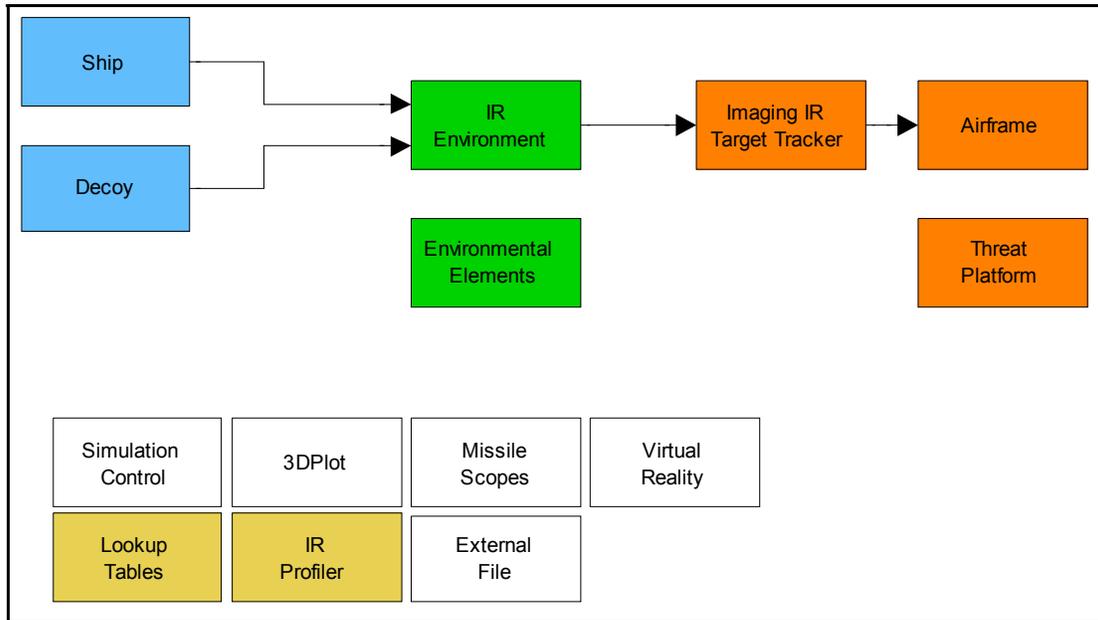

Figure 3: ASM(IR) countermeasure effectiveness simulation

Each of the major blocks shown in Figure 3 has an organization underneath as illustrated by Figure 4. User defined variables are entered via the Mask Values block. Signals from other sections of the simulation enter through the From Signal block and signals that will be used by other blocks are organized in the Goto Signals block. The signals exiting the block through the output ports are signals that reflect the information characterizing the IR radiation. The radiant emittance stored within the object's structure in MATLAB's workspace, are entered into their respective object blocks through constant Simulink blocks located in the Mask Values block. The motion of the objects and any other actions occurring by that object are modeled in the main block of the subsystem.
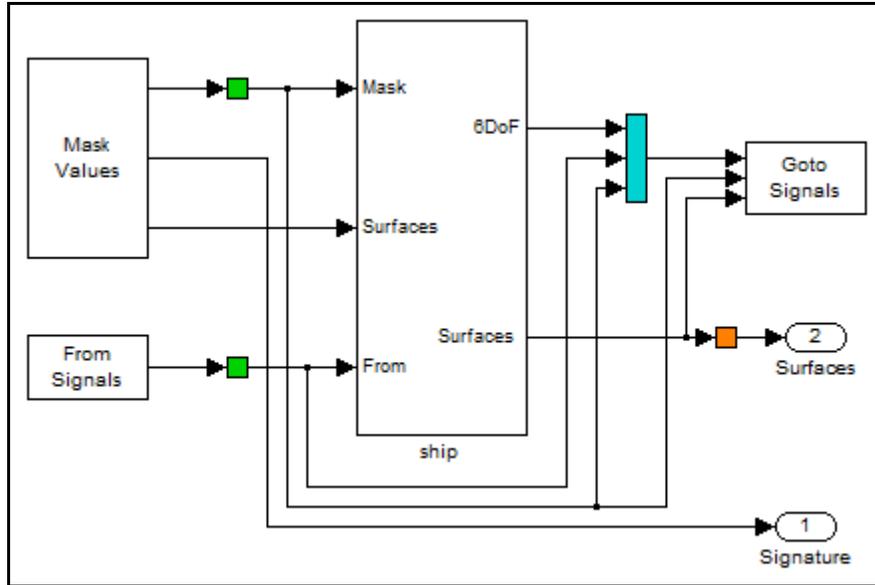
Figure 4: Upper level subsystem block

The coordinates that represent the object's position are relative to a body axis with its x-axis inline with forward longitudinal direction of the object and its z-axis pointed downwards. To simplify future calculations, coordinates are transformed to a local system with their coordinates referenced to the north, east, and downwards directions respectively. The transformation is calculated by

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix}^B = \begin{bmatrix} \cos\psi_{BL}\cos\theta_{BL} & \sin\psi_{BL}\cos\theta_{BL} & -\sin\theta_{BL} \\ \cos\psi_{BL}\sin\theta_{BL}\sin\phi_{BL} - \sin\psi_{BL}\cos\phi_{BL} & \sin\psi_{BL}\sin\theta_{BL}\sin\phi_{BL} + \cos\psi_{BL}\cos\phi_{BL} & \cos\theta_{BL}\sin\phi_{BL} \\ \cos\psi_{BL}\sin\theta_{BL}\sin\phi_{BL} + \sin\psi_{BL}\cos\phi_{BL} & \sin\psi_{BL}\sin\theta_{BL}\sin\phi_{BL} - \cos\psi_{BL}\cos\phi_{BL} & \cos\theta_{BL}\cos\phi_{BL} \end{bmatrix}^{BL} \begin{bmatrix} x \\ y \\ z \end{bmatrix}^L
$$

where ψ_{BL}, θ_{BL}, and ϕ_{BL} are the heading, elevation, and pitch angles of the body relative to the local frame respectively. All objects within the simulation are characterized by their position and orientation relative to the local frame of reference. With all objects coordinates relative to the same frame of reference, line of sight calculations and distances relative to one another are calculated with ease.

4. ATMOSPHERIC ATTENUATION

The distance separating the IR sensor from the object will affect the magnitude of the radiation signals being received[2]. Attenuation occurs due to absorption and scattering of the radiation within the transport medium in which it travels. During the simulation, the motion of the various objects in the simulation (ship, decoy, missile, etc.) is continuously updated and passed to several other subsystems. One of these subsystems is the IR Environment which has the primary purpose of attenuating the object's IR signature.

The separation distance is used in conjunction with the atmospheric attenuation profile generated by third party software such as MODTRAN to compute a transmittance factor for each individual surface. The atmospheric attenuation profile is generated offline and imported into the simulation in a data table format (in units of dB/km as a function of wavelength). Based on the sensor's operating bands, an average attenuation factor is computed and used to compute the apparent radiant emittance of each surface.

5. RADIANT FLUX CALCULATION AT AN IR IMAGE SENSOR

With all objects profiled and positioned within the simulation, attention can be focused on modeling the perception of the IR imaging sensor. The focal plane flux equation is used to determine the power of the radiant flux present at the sensor. The derivation of the equation begins with the assumption that the irradiating objects are all Lambertian surfaces and follow Lambert's cosine law. The radiant flux present at the sensor is determined from the definition of radiance[1,2]. Radiance is the total sum of exitance present within a solid angle emitted from the source. The mathematical definition of Radiance is

$$L_\lambda = \frac{\partial^2 \Phi}{\partial A \cos(\theta) \partial \Omega}$$

where Φ is the radiant flux, A is the discrete surface of the source, and Ω is the solid angle of interest. The λ subscript symbolizes the spectral radiance at a specific band of wavelength. Since the surfaces are assumed to have isotropic radiance, the radiance of an object will equal the radiant emittance divided by π. The equation for radiance can be simplified and rearranged as an expression of radiant flux as

$$\Phi_\lambda = L_\lambda A \cos(\theta) \cdot \Omega.$$

For small angles, the solid angle can be estimated as

$$\Omega = \frac{A_o}{R^2}$$

where A_0 is the aperture of the imaging system and R is the distance from the aperture to the emitting source. The sensor area projects onto the surface of the source as

$$A = (\alpha R)^2$$

where α is the projected angle of a square sensor area onto a surface separated by a distance of R. The equation is completed by accounting for atmospheric attenuation. The radiance will be attenuated by a factor of $\tau_\alpha(\lambda)$. The final equation for radiant flux is

$$\Phi_\lambda = L_\lambda (\alpha R)^2 \cos(\theta) \cdot \frac{A_o}{R^2} \cdot \tau_\alpha(\lambda)$$

6. IMAGE SENSOR RENDERING USING MATLAB

The process required to render the IR image of the sensor begins with the generation of an image scene. Objects within an image scene are defined in MATLAB using the *patch* function. The *patch* function requires a set of vertices and face mapping matrices that will describe the geometry of the object. The vertices and face data are created through the use of AC3D and stored within the object's structure as Vert and Face. The data will form a unit representation of the object that is scaled by the dimension structure. The dimension structure can be edited by a user to scale the object to the desired dimensions without the need of a new object schema. The properly sized object can then be moved to its proper position within the image by shifting the coordinates by the appropriate displacement from the origin. A value between 0 and 1 is assigned to each face of the object representing the object's IR intensity. The value of the intensity is used to assign the RGB colour of the patch within the image. Pixels that are completely white in the image represent a completely saturated detector element, while pixels that are completely black indicate the lack of detection of radiant emittance.

The next step in the imaging rendering process is to define the size of the sensor image and its frame of view. The number of sensor elements, defined by a user, determines the size of the image figure. The figures position tag is assigned a screen location in x,y coordinates followed by the image width and height. The numbers of detector elements are used as the width and the height of the image.

The MATLAB camera commands are used to assign the figures frame of view. First, the camera's position is set to equal the sensor's location on the missile through the use of the *campos* function. The line of sight of the camera is assigned by designating a coordinate representing the viewing target of the sensor through the use of the *camtarget* function. Finally, the image is scaled by setting the sensors field of view with the *camva* function. The resulting figure will illustrate the recording of the IR imaging seeker intensity reading. This image can be transformed into a sensor intensity matrix through the use of the *hardcopy* function. Rounding the values within the intensity matrix to ones or zeros relative to a threshold will create a binary image required by the target selecting process.

7. TARGET DISCRIMINATION

In order to keep the simulation generic, customizable and based on open-literature algorithms, several target discrimination criteria were developed. To match the performance of a specific threat platform, specific customized code would have to be entered to properly simulate the specific target selection and discrimination algorithms. The simulation can discriminate targets based on size, peak power, total power, or average centroid.

The simulation utilizes MATLAB's Video and Image Processing Blockset add-on tool set to perform pixel analyses. The tool block reads a binary image stored in the form of a matrix as input. The block will discriminate the separate blob entities and provide information on the blobs including the area, the centroid location, the bounding block location and dimensions, the designation label, and the number of blobs present in the image.
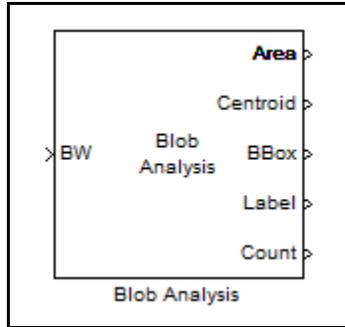
Figure 5: Simulink's blob analysis block

This information is passed to a MATLAB s-function that will select the appropriate blob based on the target discrimination criteria. The s-function will output the pixel location of the centroid of the targeted blob. The pixel locations are used to calculate the error angles used as command signals to the IR gimbal servo system. The rotational rate of the line of sight angles from the IR imaging missile to the target is calculated and input to the proportional navigation guidance system.

8. MISSILE AIRFRAME AND AUTOPILOT SIMULATION

The dynamics of the missile airframe were modeled based on the approaches of Blakelock[3] in which longitudinal and lateral behaviors were linearized and decoupled. This approach is consistent with missile body symmetry such as that for a cruiform missile and enables simplifying the dynamic response to two vibration models known as the phugoid and short-period oscillation modes. The phugoid mode has a long period and its response characteristics are considered to be negligible compared to the short-period mode response. These response characteristics when separated between the pitch response versus control surface defection and the acceleration response versus pitch enable the use of both pitch and acceleration feedback stability control as shown in the autopilot model of Figure 6.
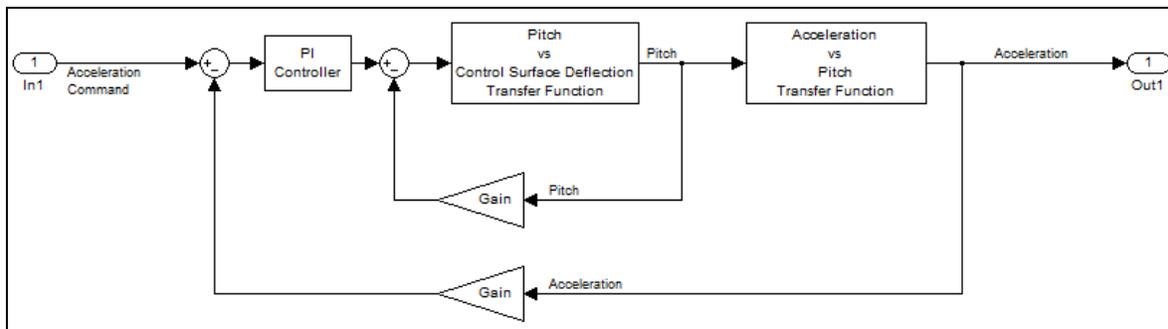

Figure 6: Typical missile acceleration autopilot feedback control loop

9. THE CLOSED LOOP ENGAGMENT

The purpose of the simulation is to evaluate platform countermeasure effectiveness against an imaging IR threat in a wide variety of scenarios. The target platform may defend itself through the combination of maneuvers and deployment of infrared decoys. The infrared decoys are categorized as distraction or seduction. Distraction decoys normally have means of propulsion that place the decoy in a stationary position away from the ship. Their purpose is to substitute the platform as the target for acquisition by of the IR seeker. Figure 7 illustrates the seeker image view of a ship with distraction decoys deployed. The image on the left shows radiant intensity and the image on the right shows the binary image used in the target selection blob analysis.

Figure 7: Ship with distraction decoys deployed

The seduction decoys are fired from the ship and are released at the end of a projectile trajectory. Their goal is to attract the attention of the IR seeker, stealing the lock of the missile on the ship. Figure 8 shows several seduction decoys launched in a pattern that will screen the ship from view, allowing the platform to maneuver away from the threat's line of sight.
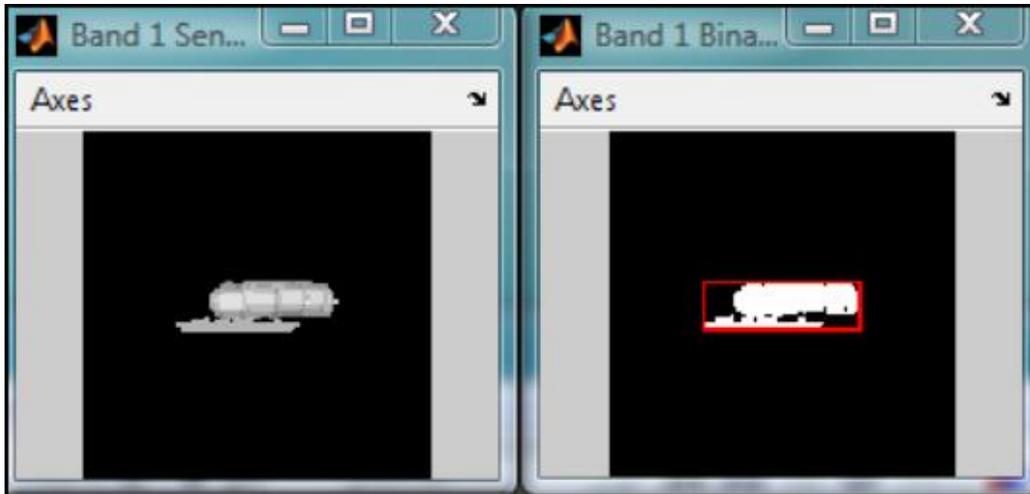
Figure 8: Seduction decoys deployed forming a screen

A special type if IR decoy called a walk-off decoy was incorporated into the ASM(IR) simulation. The walk-off decoy consists of a canister containing several IR decoys that are sequentially deployed along the trajectory of the canister's flight. The first cloud will have the effect of "walking" the IR seeker away from the ship as the earlier deployed clouds fade and the later deployed clouds grow. The final cloud, known as the keeper cloud, will have a greater size and intensity then the walk-off clouds with the intention of attracting and maintaining the lock of the IR seeker.   Figure 9 illustrates the deployment of a walk-off decoy in two radiation bands.
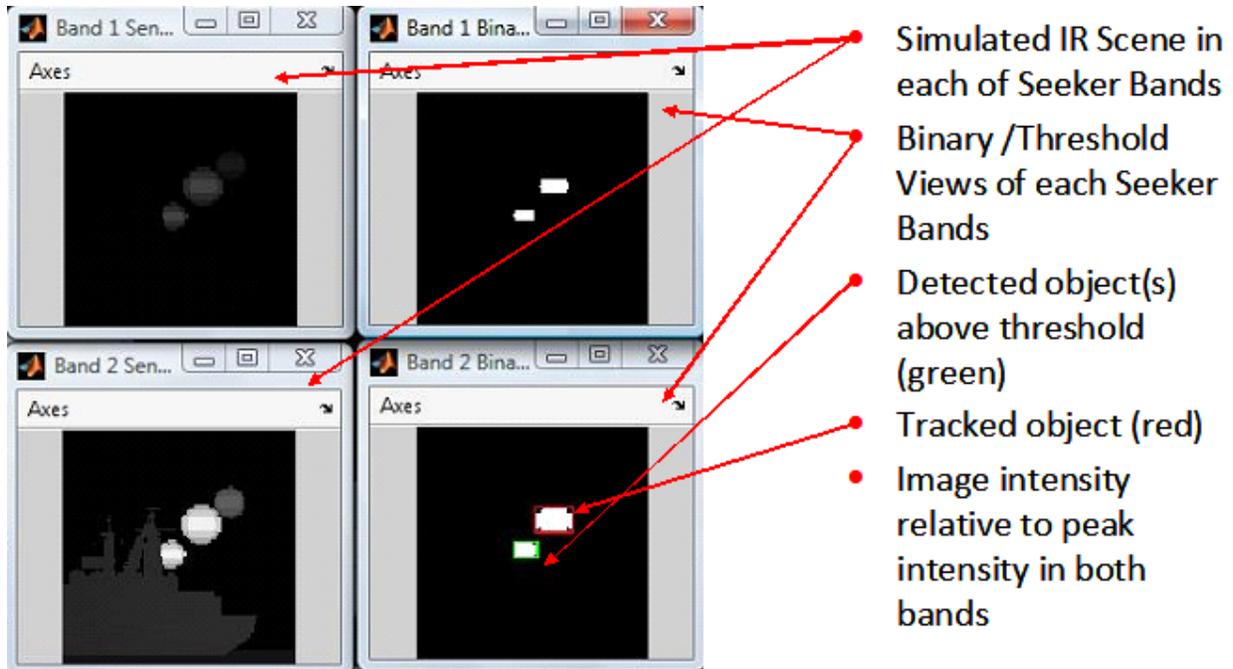

Figure 9: Deployment of a walk-off decoy

The IR threat may also potentially use counter countermeasure techniques and tactics which include terminal weaving, seeker stiff neck, and dual band discrimination.

The ASM(IR) simulation becomes effective at evaluating countermeasure protection by running many varying scenarios in a batch process. Measures of effectiveness include the missile miss distance and a probability of kill. As an example, figure 10 illustrates the batch runner results of a short scenario involving an IR missile threat attacking a ship, defending itself with the deployment of a walk-off decoy, from various headings. The simulations were performed with an Intel Core2 Extreme CPU at 3.00GHz with 2.0GB of RAM. The simulation was initialized with the missile located 1500m from the defending ship, which released its walk-off decoy 0.3s from the start of the simulation at an angle of 90° from the ship's longitudinal axis. The ship was traveling at a 330° heading, moving 10m/s. Each simulation run was completed in less then 52s. The countermeasure tactic successfully protected the ship between the headings of 320° to 20° and 140° to 200°.
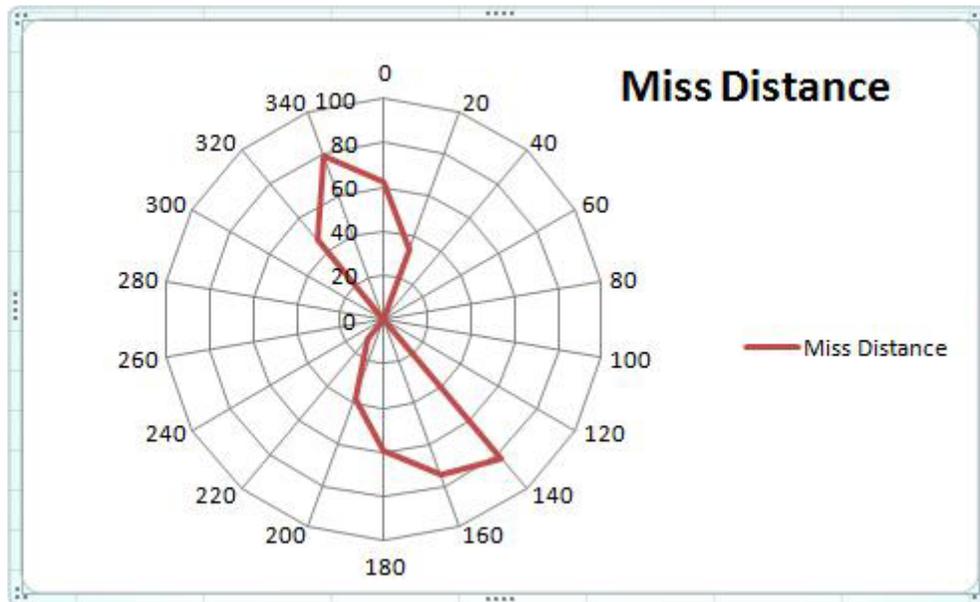

Figure 10: Miss distance results from a batch run scenario

10. Conclusion

The imaging technique described in this paper resolves the IR seeker image quickly and efficiently, resulting in a simulator capable of conduction many batch-runs in a reasonable amount of time. This advantage makes the ASM(IR) simulation a valuable tool for determining countermeasure effectiveness versus IR imaging threats.

References

[1]   Dudzik, M. C., [Electro-Optical Systems Design Analysis and Testing Volume 4], Infrared Information Analysis Center and SPIE Optical Engineering Press, Bellingham, 14-18 (1993).
[2]   Holst, G. C., [Electro-Optical Imaging System Performance], JDC Park, Winter Park, 39-46 and 265 – 272 (2006).
[3]   Blakelock, J. H., [Automatic Control of Aircraft and Missiles Second Edition], John Wiley and Sons Inc., Toronto, 7 - 111 (1991).