1998

REAL-TIME SYSTEM-LEVEL INTEGRATED TESTING

by
Dr. Tim Wells
NMD Test and Evaluation

Mr. Keith Maples
Teledyne Brown Engineering

The Integrated System Test Capability (ISTC) is being developed as the primary Ballistic Missile Defense Organization (BMDO) ground test asset for the National Missile Defense (NMD) system. It provides the opportunity to test all NMD processors and operational software in an integrated systems context prior to flight test, and then provide, in evolutionary fashion, performance extrapolation to a representative NMD C1 capability system. This paper discusses the representation of the Ground–Based Interceptor (GBI) in ISTC; tools and techniques used to meet ISTC test requirements, including real–time target signature and image generation; and current status and results.

Introduction (ISTC Context)

This paper addresses the requirements, design, and lessons learned from development of real–time processing hardware and software used to conduct high–fidelity flight processor–in–the–loop 'egrated systems tests for the NMD 'gram.

ɜ Integrated System Test Capability is a nputer-based system for testing actual listic missile defense system data ·cessing hardware and software in an ːgrated configuration through the use of ulated environments. The ISTC provides capability to :

- evaluate the performance of the NMD system
- determine the interoperability of the deployed software in the NMD system
- determine the operational suitability of the human interfaces in the NMD system.

To achieve its purpose, the ISTC must:

- provide for the physical incorporation of the actual mission and communications processors of the NMD system
- utilize the actual software that will be installed in the mission and communications processors
- operate in real time
- drive the NMD system processors with realistic scenarios
- subject the system assets to realistic threat and environmental effects in demanding scenarios
- collect data to support post-test system-level performance analyses.

The ISTC has a Hardware-in-the-Loop (HWIL) capability with the capacity to integrate all the computers in an NMD system. Each element segment of the NMD system is represented in the ISTC on individual, standalone multiprocessing computers called nodes. These computers incorporate actual system element mission and communications processors, running actual element software. The individual element nodes are interconnected by an NMD system communications network driven by real–time system interfaces and threat and environment input data. The test configuration is truly representative of individual elements operating in concert as part of an overall architecture and provides an accurate exercise of the NMD system under the realistic stresses that will be encountered in operation.

The ISTC test architecture is composed of segments, with each segment associated with a specific system function, as shown in Figure 1.

Distribution Statement A: Approved For Public Release; Distribution is Unlimited

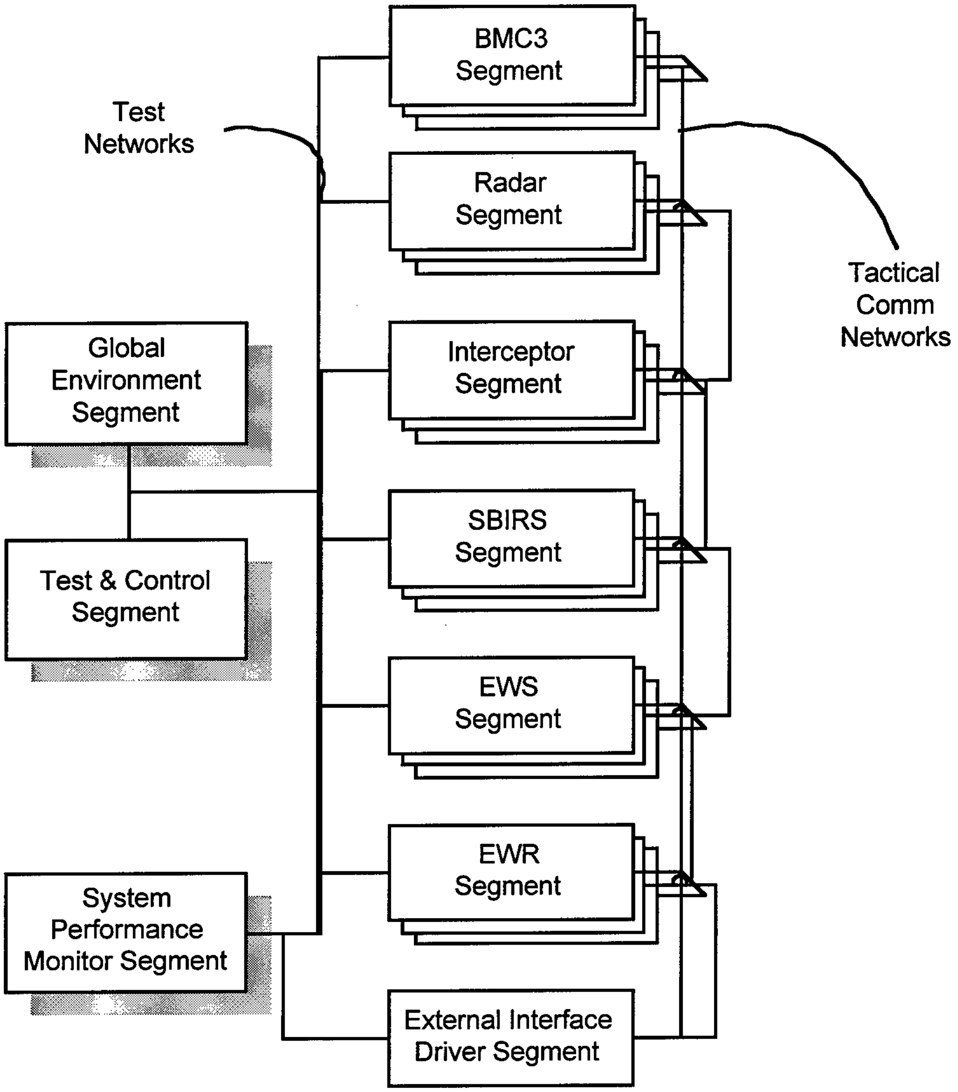


Figure 1. ISTC System Diagram

The Ground-Based Interceptor Exo-atmospheric Kill Vehicle (EKV) is the GBI component in the most complete state of development in ISTC. Complete processor test environments for each GBI EKV competitor have been integrated and tested in ISTC. The EKV segment has been a pathfinder for integrating external software and hardware. Figure 2 illustrates the constituents and immediate interfaces of the GBI component of ISTC.

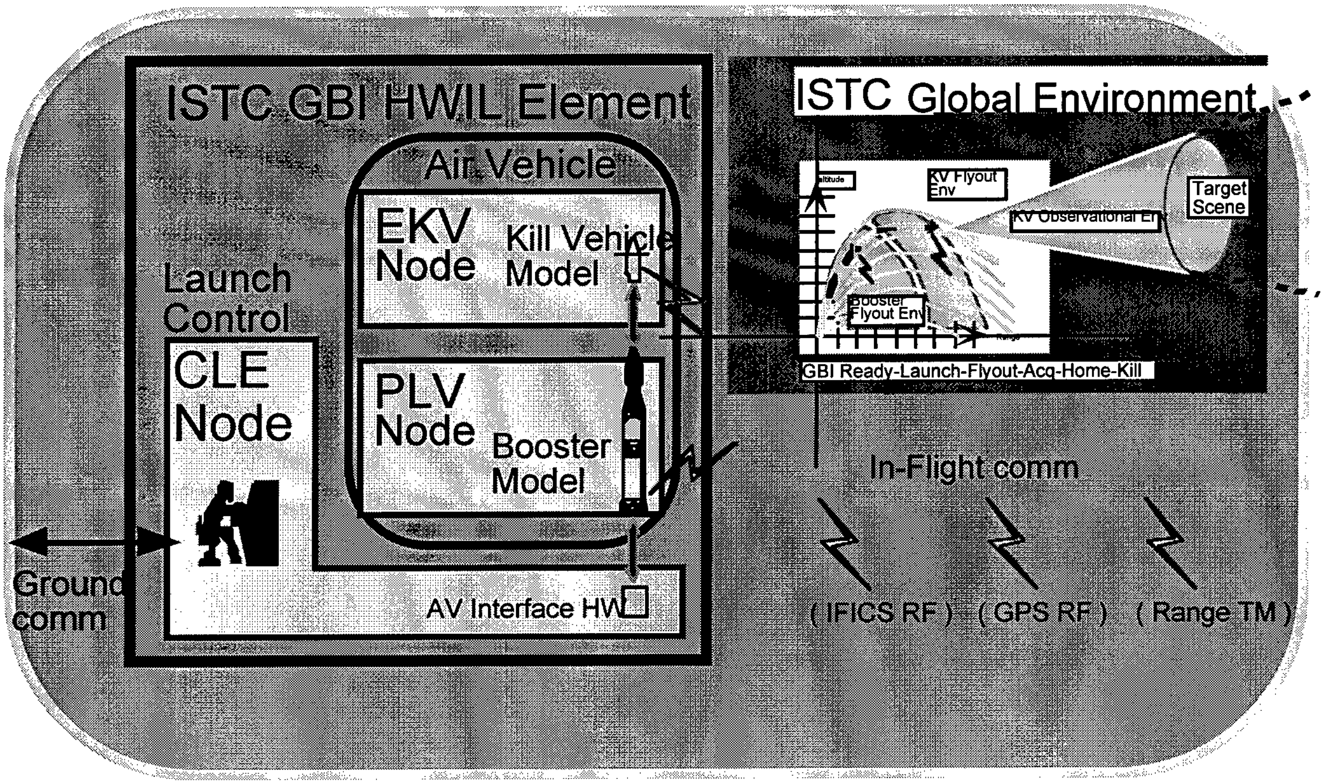


Figure 2. Embedded GBI Components and Their Interfaces

The GBI segment consists of a booster, the EKV, and a launch controller. To date, contractor-supplied representations of the EKV have been integrated into ISTC and test drivers (low-fidelity, ISTC-developed models) have been utilized for representation of the launch controller and booster. Integration of a high-fidelity simulation of the Payload Launch Vehicle (PLV) into ISTC is currently under way. The PLV is the flight test booster which has been utilized in interceptor flight tests to date at Kwajalein Missile Range (KMR). The PLV will remain the test vehicle of choice for additional flight tests occurring in CY 1999. Lockheed Martin Missiles and Space Corporation (LMMS) builds the PLV and delivers the PLV simulation to ISTC. In addition, LMMS builds the current Command Launch Equipment (CLE) used at KMR. The CLE serves as launch controller and as such is currently being integrated into the ISTC. The software and hardware delivered by LMMS will be equivalent to the actual hardware and software functioning at KMR as the Launch Controller and Missile Interface Unit segments of the CLE.

The EKV is the most mature segment of the GBI in ISTC; the remainder of this paper will focus on that segment.

Hardware and Operating System

The GBI application software achieves its real-time performance through the use of the GBI platform infrastructure which consists of the underlying hardware, operating system, programming language, and the Run Time Infrastructure (RTI). The correct combination and use of these elements is crucial to enabling the GBI application software to achieve its stated objectives.

The underlying hardware is a Silicon Graphics Inc. (SGI) Origin 2000 running the IRIX 6.4 (UNIX-based) operating system. This high-performance server is configured with 16 MIPS R10000 Central Processing Units (CPUs) running at 190 megahertz, each containing 2 megabytes (MB) of level 2 cache. The CPUs are integrated into SGI's Scaleable Shared-memory Multiprocessing (S2MP) architecture which scales efficiently to 128 processors. The S2MP is a hardware-based cache coherent distributed shared memory system utilizing the CrayLink Interconnect. All CPU, memory, and input/output (I/O) interconnects sustain gigabyte (GB) bandwidths and are crossbar switched to support multiple simultaneous connections at these rates.

The IRIX operating system features shared memory process threading, cache affinity scheduling, and application memory migration to ensure efficient use of the hardware's resources and minimize thread context switch latencies. Because of the unique requirements of real time, the operating system provides functions which allow the user to configure and/or override default system behavior. Some examples include the ability to lock threads and their memory to specific compute resources and to raise their priority above that of other threads and the operating system itself to reduce latencies associated with context switching and servicing interrupts. Although the operating system default behavior works well in most cases, deliberate use of these capabilities is effective in achieving real time, particularly when applied to those threads concerned with processing I/O.

The Origin 2000 supports many high performance I/O interfaces directly through its 1.28 GB per second XIO bus. One of these, the High Performance Parallel Interface (HiPPI), provides 80+ MB per second throughput. The Origin 2000 also contains the industry standard Peripheral Component Interconnect (PCI) bus. This bus provides the interconnect to VMIC's reflective memory product. The reflective memory provides 29.9 MB per second sustained bandwidth and single digit microsecond latencies.

The programming language is Ada95 and the compiler is GNAT [Gnu's Not Unix (GNU) Ada Translator], supplied by SGI and integrated with its suite of development tools and compilers (multiprocessor debugger, etc.). Ada95 was selected because its ingrained tasking (threads) model and synchronization primitives lend themselves to the highly parallel nature of the GBI application software. SGI has integrated the Ada tasking model of GNAT into its threads model, providing a seamless and efficient parallel implementation on which the scheduler can operate.

Run Time Infrastructure

The final element of GBI's platform infrastructure is the RTI called Distributed Modeling Support Environment (DMSE). DMSE evolved from the earliest days of ISTC to address the problem of application code portability (reuse) caused by the relentless pace of technological advance and application performance requirements. DMSE provides the application developer a layer of immunity from the Commercial-off-the-Shelf (COTS) hardware and operating system, thus providing transparent operation over multiple hardware platform and operating system combinations (currently VME, SUN, and SGI supported). Virtually all application software interface to the underlying architecture, as well as to external entities, is handled through DMSE. Almost immediately, because of its position in the software hierarchy, DMSE began serving additional purposes, including performance enhancement, functionality augmentation, data collection, and incorporation of external interfaces.

The DMSE Application Programmer's Interface consists primarily of an operating system interface (and, in some cases, an operating system adjunct) and a programmer's model. DMSE's real-time programmer model is composed of message passing and shared memory, both with user-specified, nonintrusive run time data collection capability. Wherever possible, DMSE seeks to enhance performance by customizing the behavior of the underlying operating system and/or bypassing the operating system and going directly to the hardware.

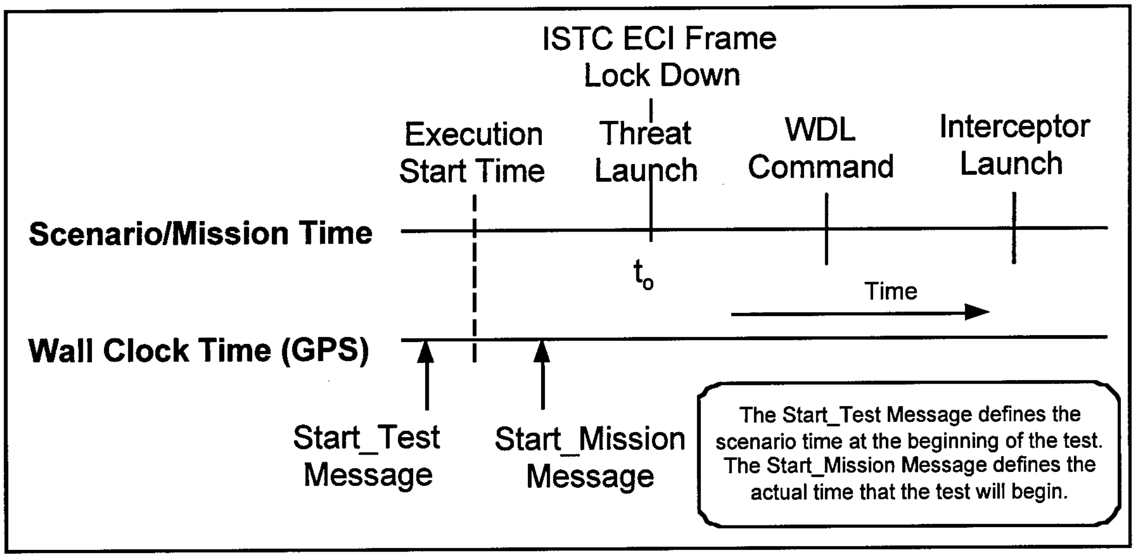
External interfaces are provided by DMSE for both the message passing and shared memory communication mechanisms using a variety of standard and custom protocols including TCP/IP, HiPPI Physical/Framing, and Reflective Memory polling and/or interrupts. Multiple message layer protocols are supported, including Distributed Interactive Simulation (DIS), Strategic Defense System (SDS), Capability Increment 3 (CI3) (X), DMSE, and generic (unknown); others are easily added. Some examples of the supplemental functionality DMSE provides include a real-time Global Positioning System (GPS) synchronized clock, startup and shutdown synchronization, periodic events, Asynchronous Signal Handlers, and file I/O.

Pre–computation Versus Real–Time Computation

One of the keys to achieving high fidelity simulation in real time is a timely phasing of tasks so that, generally, computations are performed as early as possible. The ISTC timeline definition is illustrated in Figure 3. The Test and Control (T&C) activation message contains the wall clock (actual) time that test execution will begin (test initiation). Each node begins execution independently at test initiation time, as determined by their own clocks synchronized with GPS timing. The T&C activation message contains the scenario (simulation) time at the moment of test initiation and is provided before test initiation time to allow each node ample time for synchronization. Currently the scenario start time (t_o) is concurrent with first threat launch.

Knowledge of the threat structure and interceptor locations permits pre–computation of many scenario characteristics before test execution. For example, backgrounds can be defined for a range of launch times and azimuths. These are referred to as off–line computations. After the Start_Test message is sent, messages will be sent to each node instructing it to perform its individual node initialization. At this time, and before commencement of real–time operations, various other computations can be performed which take advantage of knowledge of the selected scenario and scenario time. For example, a priori threat information is known once scenario, engagement time, and conditions are fixed, enabling pre–storage of target temperatures, which are used later, in real time, to compute engagement dependent signatures.

Additional computations, which may depend on early mission events such as GBI launch messages, can be performed during booster flyout. This allows computation of additional background scenes or pre-computation of a few extended images, such as those obtained just before impact. The



The Start_Test Message defines the scenario time at the beginning of the test. The Start_Mission Message defines the actual time that the test will begin.

Figure 3. ISTC Timeline Definition

latter computation would have to be deferred until the engagement conditions are well known — typically at booster burnout — so that there will be minimal error due to inaccuracies in the range to target and aspect angle. Although this option has been implemented, in practice it is preferable to restrict scene granularity so that all scenes can be computed in real time.

NTE/PTE Interface

The EKV is composed of two parts — the Node Test Environment (NTE) and the Processor Test Environment (PTE). A version of the PTE is supplied by each of the two GBI EKV contractors, and strict controls are maintained on access to these versions to protect their competition-sensitive nature. For any single test, only one of these PTE versions is executed in the ISTC simulation; the other PTE is inaccessible. Special versions of the NTE are maintained to supply appropriate interfaces for each PTE version. To provide an impartial assessment of the two PTE units, commonality of the NTE units is maximized. Differences between the units encompass both hardware interfaces, message protocols, message rates, and message contents. However, the nature of the information that is passed is the same in both cases. All information provided here is common to the two PTE units and is not competition sensitive.

The NTE/PTE interface provides true point-source, background, and extended-object scene information to the PTE sensor model. These three scene components are generated separately by the NTE. Background and extended objects are presented separately to the PTE as irradiances at the aperture of the telescope. The point-source data, background data, and extended-object data provided are then processed by the PTE's sensor model, which adds optical blurring and includes the effects of seeker motion and other noise sources on the composite scene signals.

Point-source components include unresolved threat complex objects, Resident Space Objects, celestial objects, and other objects smaller than a single pixel (a detector sub-pixel). Point sources are provided to the PTE in real time, at a pre-defined interface rate. Point sources which are obscured by extended targets are identified by the NTE and not passed to the PTE.

Background and extended targets are provided to the PTE as arrays of irradiances, along with information describing the location of these arrays in a reference frame which depends on nominal viewing geometry but not on instantaneous line-of-sight (LOS) or field-of-view (FOV). Backgrounds and extended targets are combined in the PTE; no shadowing or superimposition is performed within the NTE.

Frame rate requirements are dictated by interfacing contractor sensor scene composition and processing software and hardware. To meet these frame rates, inherently high data bandwidth, low latency physical interfaces have been implemented which have been successfully demonstrated with both all-digital and early brassboard contractor configurations.

Target Signatures

It is axiomatic in HWIL testing that the more calculations that can be done in real time, the higher the fidelity of the simulation will be. Therefore, the test philosophy adopted for the ISTC is to accomplish as much of the signature-related calculations in real time as is technically feasible. Through the years of ISTC development, beginning with the Proof-of-Principle prototype in 1989, it has been proven that for an important class of interceptor scenarios — the exo-intercept of NMD strategic systems — it is possible to calculate target signatures in real time consistent with interceptor seeker framing rates. ISTC's experience in NMD has also shown that the requirements for updating the benign scene background are not as stringent as for near-field target/threat objects because the background dynamics change more slowly. Thus, background scenes need to be updated typically at only a few frames per second, as opposed to targets which require signature updates at the sensor framing rates.

Traditionally, the elements of the radiometric scene for HWIL testing have been computed off-line and composited into series of time–sequenced frames that are stored for future playback during test runs. There are distinct disadvantages to this approach, as well as some cogent reasons why it has been used. The main impediment to the use of pre-computed, or "canned" scenes, is that an interceptor simulation cannot be run in a "closed loop," that is, the dynamics of the engagement cannot be fed back into the signature generation process. Rather, the interceptor fly-out must be planned ahead of time and based on known target trajectories to ensure intercept. Analysis has shown that the largest error source in the pre-computation of composited scenes is the inability to predict the correct orientation of resolved objects and background components relative to the seeker. This orientation can only be predicted accurately in real time. A truly stressing test of the interceptor system's components cannot be conducted when the answer is known a priori.

Over the last 3 years, ISTC has developed a real-time HWIL infrared (IR) target signature generation capability. In the ISTC calculation scheme, the object surface temperature map is calculated during the interceptor fly-out prior to sensor activation in non-real time, stored in arrays, and temporally interpolated for use in the real-time signature determination. In addition to closing the simulation loop by feeding state vector information to the signature calculation in real time, substantial productivity improvement can be realized in both test pre-preparation, since target signatures do not have to be pre-calculated and stored, and test run turnaround time. A similar approach for backgrounds, or at least less stressing background components, is also employed in ISTC.

Ideally, a test should be conducted with target parameters unknown to the interceptor and with signatures calculated using the most current interceptor and target state vectors, that is, in real time. The main reason this has not been done previously is the large computational load imposed by generating signatures in real time. However, recent exponential growth of available computational power at ever-decreasing cost, coupled with use of a fast signature generation code [Optical In-line Signature Generation (OPTISIG)] has changed this paradigm, at least for one major class of interceptor simulations, that is, exo-atmospheric intercepts of strategic systems. The ISTC real-time point-source signature capability is derived from the Teledyne Brown Engineering (TBE) developed OPTISIG code and uses the GBI developed Target and Threat Model Libraries (TMLs), also built by TBE. Signature calculations consistent with the typical sensor framing rates can even be accomplished on Pentium-class machines. Sustained real-time calculations and throughputs have been demonstrated on SGI R10000 machines running the ISTC. Table 1 shows early code benchmarks for point-source images which were produced on a PC and used to predict real–time throughput.

Target Type	Seeker Frame Rates (fps)
Medium RV	1350
Rigid Ultra Light Replica	2030
Medium Balloon	1580
Canisterized Traffic Balloon	1590
Large Balloon	2070
Computer Used: Harris Night Hawk with 100 MHz Power PC	

Table 1. Target Signature Performance, EKV NTE Point-Source Signatures

As illustrated in Table 1, the implementation of OPTISIG contained in the ISTC allows for generation of point-source signatures at high rates (well above nominal seeker frame rates). Although the table is for single object computations, in practice, more often than not, there are multiple threat objects contained in the seeker FOV. The R10000 processor used in ISTC is capable of producing point-source signatures at the seeker frame rate for multiple threat objects, nominally between 5 and 10. There are two options built into the ISTC software for handling cases where the number of objects in the seeker FOV exceeds processor capability. First, because the software is flexible, it allows for the addition of processors to run OPTISIG. Second, if the computation requirements exceed the processing power available, a scheme of interpolation has been implemented. The available processors compute point-source signatures for all objects in the seeker FOV at the fastest rate possible. When OPTISIG cannot run for every object at every frame, an interpolation scheme becomes active. Signatures are computed via OPTISIG for as many frames as possible and interpolated values are computed for missed frames. Errors introduced are a function of the number of missed frames and in practice have been very low (less than 1% difference in interpolated values vs. OPTISIG generated value).

An important element of the ISTC real-time signature concept is to use government standardized and accredited target model libraries. To ensure signature commonality among all community participants, the TML concept, pioneered by Synthetic Scene Generation Model and adopted for the OPTISIG program by the Ground-Based Surveillance and Tracking System Program Office, has been carried forward by GBI and is the accepted government standard method of producing fast OPTISIG signatures. The GBI/EKV element in the ISTC adopted the approach utilized by the GBI Program Office to use TMLs developed by TBE under GBI sponsorship. These TMLs are distributed to the EKV contractors and other interested parties, such as ISTC, thus ensuring commonality among all GBI players. Having a level playing field with regard to the signatures used in testing is an important consideration in a competitive environment. When threat/target models are built, considerable efforts are made to ensure that the most up-to-date Intel threat description information is used and that this information is provided through accredited BMDO sources.

The ISTC approach to generating real-time IR signatures provides the operational testers with the required flexibility and the appropriate stressing environment to test the interceptor components of the GBI EKV system for the entire spectrum of flight target objects and threat systems through the complete exo-atmospheric intercept window.

Extended Target Signatures

In the typical intercept geometry scenario, the focal plane image of near-field objects grows as a squared power of the decreasing range. Although these objects are initially point sources, at some point in time they resolve onto more than one detector. This results in the most stressing situation for real-time calculations, that is, the resolved object image during the interceptor end-game when the aim-point selection must be exercised to kill the target. The ISTC's unique, real-time resolved signature capability traces its origins to concepts first demonstrated in the late 1980's for the Target Signature Analysis program for what is now Eglin Air Force Base Wright Laboratories.

The extended image of resolved objects is provided as an array of irradiance values. These arrays represent integrated irradiances at the aperture, where the integration is performed over a small pixel. The coordinate system used for background images is utilized here. An attempt is made to fit the pixel map to the size of the 2–dimensional projected target, and, except possibly for the final image before impact, the pixel map will contain the entire target. Note that the irradiance array represents a proper subset of the FOV and is rotated about its three–dimensional centroid so that it corresponds to the predicted EKV sensor orientation at the effective time of the image.

The resolution of the target array — that is, the number of pixels in the array — is determined by the target size and the resolving ability of the sensor. A requirements analysis was performed that evaluated shape accuracy of the reported image for several resolution levels. As a result of this analysis, the initial image array was set so that the target always subtends at least 4 pixels in at least one direction.

An array of irradiance values is computed for each waveband, for each frame from target extension until impact. Processing of extended targets was designed to be very flexible and responsive to differing NTE and PTE computation rates. The processing logic was designed so that if it were determined that the largest images could not be computed in real time, these larger images would be queued and computed just–in–time and in parallel with the smaller, real–time extended images. In this approach, for all but the last second or two of viewing, extended image computation would be done in real time. In any case, computation is done and transferred to the PTE to ensure that the PTE will have sufficient time to process the images. The number of leading frames is determined off–line and is based on benchmarks of total system throughput. In practice, it has been found that PTE speed is sufficient to maintain real–time processing right up until impact.

Table 2 presents code benchmarks for resolved images, showing the framing rates that ISTC has achieved.

Beginning at time of extension, target images are provided to the PTE at the standard PTE frame rate. As the table illustrates, ISTC supports real-time resolved image signatures at a framing rate and with fidelity sufficient to meet the GBI requirements for the EKV terminal phase with a just-in-time target image generation .

Background

ISTC is required to supply local environments for the EKV sensors, which

Variety of Targets at 90 degree Aspect Angle				
128 x 128 Resolution				
No. Facets	Secondary Pixels	Facet Project (ms)	Pixelization (ms)	Framing Rate (fps)
432	166	11.5	8.5	49.7
800	216	18.3	9.3	36.3
144	196	6.6	8.8	64.9
152	204	6.2	10.5	59.5
96	256	4.4	12.5	59.1
192	186	6.7	7.8	68.8
128	166	4.5	7.7	81.6
96 Facets				
256 x 256	512	11.8	44.2	17.9
128 x 128	256	4.3	11.6	63.2
64 x 64	128	1.7	2.5	235.9
432 Facets				
64 x 64	84	5.7	1.8	133.8
800 Facets				
64 x 64	108	9.1	1.7	92.0

Table 2. Target Signature Performance, Resolved Image Performance

add to the target signature and represent the measurable ambient intensities in the appropriate bandwidths. The EKV representation must be an integral part of a consistent “truth” model of the physical environment across the entire system of multiple element nodes and element types. In particular, because of the closed–loop interactions between radar reports and EKV reports and results, the scene presented to the radar element must be consistent with that presented to the EKV.

The following scene aspects must be reproduced:

- signal–to–noise ratios of the point-source targets
- background absolute levels, especially those arising from significant sources such as Earth limb
- background gradients
- dynamic characteristics of the background due to rapidly changing altitude of the GBI
- dynamic characteristics of the background due to linear motion of the EKV sensor boresight or rotation of the sensor about the boresight
- point sources — artificial satellites, stellar objects, and distant solar bodies
- extended sources — nearby solar bodies, especially the Sun and Moon.

In addition to these benign sources, the following results of hostile actions must be accounted for:

- nuclear events — specifically, gamma rays
- debris of various shapes, spectral characteristics, and temporal dynamics.

The ISTC EKV celestial background includes both solar system (Sun, Moon, and planets) and stellar components. The Celestial Background Scene Descriptor code from Phillips Laboratory is used to model the Sun, Moon and planets. The ISTC star background is derived from astronomical databases, the Catalog of Positions of Infrared Stellar Sources (CPIRSS) from the U. S. Naval Observatory, the MSX Infrared Astronomic Catalog (MSX IRAC) from Phillips Laboratory, and a database of over 400 IR calibration star spectral irradiances, also provided by Phillips Laboratory. Both the CPIRSS and MSX IRAC catalogs were partially derived from the IRAS measurements of the mid 1980's. The three source catalogs are used by ISTC to generate star catalogs for EKV with stellar intensities computed for the EKV wavebands.

The viewing time covered by the background scenes is from commencement of seeker operations to nominal impact with the target. This viewing window is enlarged on either side by a few seconds to account for uncertainty in seeker uncap time and intercept time. Within the viewing window, nonstructured extended backgrounds (currently only Earth limb) are computed at intervals determined by GBI altitude and LOS elevation.

The real–time stipulation for extended EKV backgrounds is a stressing requirement, due to the large number of pixel intensities that must be quantified. The pixel requirement arises from both pixel granularity — the need to resolve the background down to a maximum pixel size — and the pixel update frequency. The latter is determined by the rate at which the background changes, due primarily to rate of change of altitude of the EKV and to the elevation scanning rate of the sensor. Because computational resources create a hard pixels/second limit, there is a tradeoff between pixel resolution and pixel update rate. Fortunately, NTE scenarios are generally associated with higher EKV viewing altitudes and elevation angles. As a result, ISTC can focus on providing higher resolution background images, all the way down to the detector resolution level.

A second error source lies in alignment of the provided background irradiance array.

Real–time indexing into the previously computed array results in an array that may be misaligned by as much as .5 pixel. From frame to frame, the change in alignment will result in an apparent dithering of the background of the same order of magnitude. Although the errors resulting from this dithering are two orders of magnitude below the errors due to gain in altitude (at low operating altitudes) it could still be significant for large pixels. By using small background pixels, ISTC has made this error negligible.

Background data is provided as an array covering a viewing region larger than the FOV. Each array represents a matrix of integrated irradiances at the aperture, where integration is performed over a small pixel. An array of background irradiance values is computed for each waveband. The size of the pixel is determined at run time and depends on the spatial frequencies of the background features represented.

ISTC uses the PLEXUS code (Phillips Laboratory Expert User Simulation) for the Earth limb environment. PLEXUS Atmospheric Integrated Model, including Strategic High-Altitude Atmospheric Radiance Code and Moderate Transmission, is used to generate spectral databases for each candidate launch site. Currently, one database has been generated for daytime and nighttime at both the Kwajalein and Grand Forks GBI sites. In practice, for each database several overlapping "maps" of the logarithm of irradiance are generated off–line, one for each various GBI altitude and elevation boresight. In real time, the background generating node receives a message requesting a background with a specified boresight and rotation angle with respect to the local horizon. The node then computes an intensity at several points distributed uniformly across the focal plane via interpolation in overlapping maps. Pixels across the entire focal plane are then calculated by means of biquadratic interpolation in the logarithm of the intensity function.

In a manner similar to that used for extended target calculations, a provision was made for pre-computation of backgrounds. If it were determined that background update frequencies exceed real-time computational capabilities, pre-computation would be performed. These background scenes would be pre–computed and provided to the PTE during the time interval after Weapon Data Load and before commencement of seeker operations. To the greatest extent possible, backgrounds would be computed in real time and provided to the PTE with adequate time for PTE optical and detector processing. Required image update frequencies would be computed during the initialization phase. Note that in all cases of pre–computation (even the nominal real–time case), there would be uncertainty in actual GBI position and LOS at the time the scene data is computed. Hence, the pre–computed scenes would need to be slightly larger than the FOV. In practice, it has been found that this feature of the design is unnecessary. ISTC has been able to maintain full real–time calculation of the scenes, with a constant refresh rate computed before the mission.

Although not yet implemented, ISTC will ultimately account for natural clutter sources (aurora, Earth backgrounds) and nuclear effects. It is anticipated that inclusion of nuclear effects will require significant modifications to this interface.

Summary

This paper addressed the requirements, design, and lessons learned with development of real–time processing hardware and software used to conduct high–fidelity flight processor–in–the–loop integrated systems tests for the NMD program. Significant advances were necessary in the areas of background and target signature generation capability to meet the simultaneous requirements of high fidelity and real–time operation.

Our hardware/software design, based on multiple SGI Origin 2000s, is extensible and takes maximum advantage of parallelism so that our object processing capability can grow without software breakage through the addition of processor boards. Special interface hardware and communications protocols support completely embedded hardware/software components of competing EKV contractors. This will be phased into HWIL components in the coming year.

In the signatures area, signature models for real–time implementation of point sources and resolved image scenes have been adapted, providing ISTC test articles with the capability to interact dynamically with their environment without the need to script engagement–dependent parameters.

In backgrounds, the industry standard PLEXUS database has been utilized using multiple off–line computed tables for various scenarios, operating altitudes, and boresight elevations. Real–time logarithmic interpolation to a fine pixel level has provided a high-fidelity benign background for the EKV sensor.

Currently the Kinetic Impact Debris Distribution model is being implemented in real time to represent debris and chaff from other intercepts.

In short, it is crucial to carefully divide resources, first in allocating tasks across an expandable array of processors and second in time–managing the required calculations so that real–time algorithms are inherently simpler and can rely on judiciously pre–computed databases and models.