The Navigation Warfare Test Bed

Sjoerd Gelsema and René Thaens
NATO Communications and Information Agency
Oude Waalsdorperweg 61, P.O. Box 174, 2501 CD The Hague, The Netherlands

{Sjoerd.Gelsema, Rene.Thaens}@ncia.nato.int

ABSTRACT

Recent NATO trials in the area of Navigation Warfare (NAVWAR) revealed the need for a flexible, but controlled environment for research on Navigation Warfare topics. This paper describes the development of the NCIA NAVWAR Test Bed, up to the stage of initial capability. It describes its purpose, its overall architecture, its principal objects, and the interaction between the objects. It illustrates the use of the Test Bed by introducing a simple, yet challenging enough scenario based on Cooperative ESM Operations. It is the aim of the paper to explain the capabilities of the Test Bed to such an extent that the reader will be able to define other additional uses of the Test Bed, and possibly contribute by writing their own objects, thus expanding the functionality, utility and benefits of the NAVWAR Test Bed.

1.0 Introduction

Modern weapon systems, military platforms and personnel engaged in air, land and maritime operations are becoming increasingly reliant on the use of a Global Navigation Satellite System (GNSS) to provide precise positioning, navigation and timing (PNT) information. In the past few years, NATO has recognized the importance of the continuous availability of PNT information.

GNSS signals are very susceptible to Radio Frequency Interference (RFI), both intentional and unintentional. Navigation Warfare (NAVWAR) is defined as *preventing the hostile use of PNT information while protecting the unimpeded use of the information by NATO forces and preserving peaceful use of this information outside the area of operations* [1]. In NATO this effort is the responsibility of the Consultation, Command and Control Board (C3B) Capability Panel on Navigation and Identification (CaP2) and its subordinate Working Group on NAVWAR . One if its activities is the development of a CONOPS [3] and related STANAGS concerning NAVWAR [2].

In 2008 the NAVWAR Working Group participated in Trial Imperial Hammer 2008 (TIH08) in Sardinia, Italy [4] to support the development of a NATO NAVWAR Concept of Operations (CONOPS). In June 2012, the NAVWAR Working Group participated in Trial Unified Vision (UV12) in Ørland, Norway [5] to test and evaluate the operational relevance of the NAVWAR CONOPS and the NAVWAR Cooperate Electronic Support Measures Operations Concept of Employment (NAVWAR-CESMO CONEMP).

TIH08 trial showed a need for a flexible, but controlled environment for further research of NAVWAR topics without the restrictions imposed by live trials or exercises. The Navigation Warfare Test Bed (NWTB) is designed to resolve this need. It was used as a backup alternative for live-flying operations in Trial UV12. This paper describes the design, implementation and the initial use of the NWTB.

2.0 Functionality

The NAVWAR Test Bed offers a dynamic and operationally relevant environment in which NAVWAR related strategies, tactics, organisational aspects and tools can be tested while reducing the dependency on costly and complex live flying exercises. In its full setup the testbed consists of a *scenario generator*, a group of simulation objects, and a group of analysis objects.

The simulation objects take information and act upon events from the scenario generator. They also interact with each other, and pass their information to the analysis object. Typical examples of simulator objects in the NAVWAR context are: a jammer platform, an ELINT platform and a viewer.

The analysis objects take information and act upon events generated by the simulator objects. Analysis object can be human-in-the-loop. A typical analysis object is a CESMO procedure, a NATO Common Elint and ESM Reporting Format (NCERF) module. However, it can also be Time Sensitive Targeting (TST) cell staff, or a NAVWAR Analysis Cell (NWAC) analyst. These modules publish the results of their analyses on a reporting network (or on an attached storage facility) for evaluation by the actual users of the testbed.

2.1 A simple example

The following example illustrates how the three groups of objects (i.e., the scenario, the simulation and the analysis) support the evaluation of a simple NAVWAR related research topic. The example is based upon the scenarios that were employed during UV12 that was held on and around the Military AirBase in Ørland, Norway in June 2012 [5].

2.1.1 Setting the scene

The analysis level contains an Operational Commander holding tactical control over missions in the area of Ørland. He needs to follow strict procedures before deciding on the response when a time sensitive target (TST) is discovered. Ultimately, the outcome of the TST procedure may be a method of attack to neutralise the target while reducing the collateral damage as much as possible. Hostile NAVWAR assets in the area may impact ingress routes to the target or reduce the effectiveness of GPS guided munition, and therefore must be included in the final plan of attack. To be able to do so, the Commander needs updated impact predictions of these NAVWAR assets during all stages of the TST planning procedure. The impact predictions are supplied by a NWAC function.

2.1.2 The goal

Hostile GPS jammers aim at denying the use of Positioning, Navigation and Timing (PNT) information for friendly units in theatre. The mere fact that GPS Jammers are present does not provide sufficient information to the Commander to decide on how to deal with these. As was shown during TIH08 and UV12, various impact prediction models can be used to estimate the influence of the jammers on the PNT performance of affected units. Also, various procedures of reporting jamming and jamming impact can be followed. A typical use of the NWTB could be to establish the preferred methods and procedures of determining jammer impact on operations and for reporting the findings to the Operational Commander.

2.1.3 The scenario

The Operational Commander has an arsenal of GPS/INS guided cruise missiles at his disposal. These weapons are ready to be used to support TST against an enemy Forward Operating Base (FOB). It is known that the area is protected by one ore more GPS jammers which could be intermittently turned on and off, and which could impact the accuracy of the weapon systems.

To employ a cruise missile, the Commander needs an estimate of its accuracy at a target. Therefore, an estimate of the jammer locations are needed. The Commander tasks the NWAC to provide an assessment. NWAC in turn provides tasking requirements to Electronic Support (ES) sensors in the area. Based on the tasking, the ES sensors provide Lines of Bearing (LOB) and signal externals to the NWAC through a CESMO network. Based on this data, the NWAC estimates jammer locations and error ellipses, that form inputs for the GPS jamming impact assessments for the Commander.

The following specifications apply to the scenario

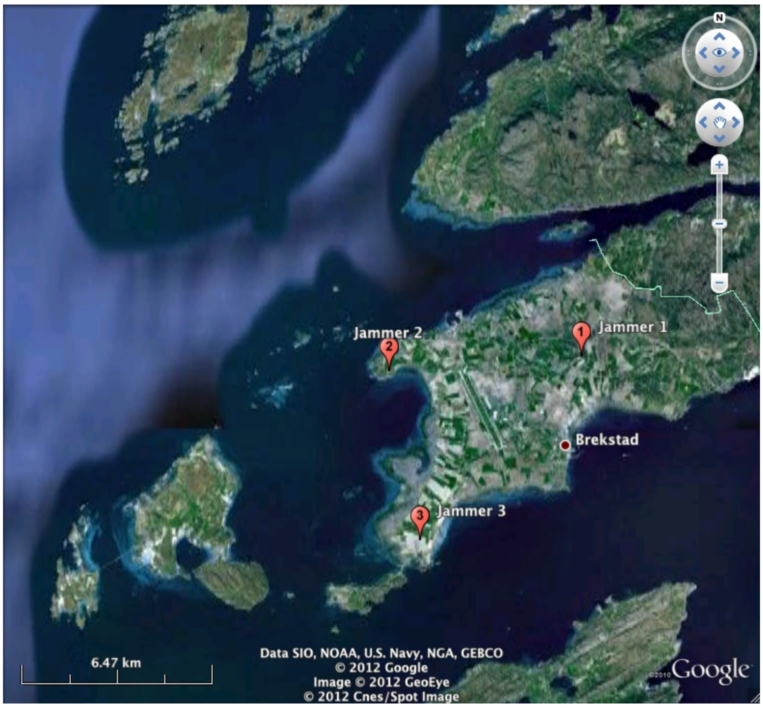
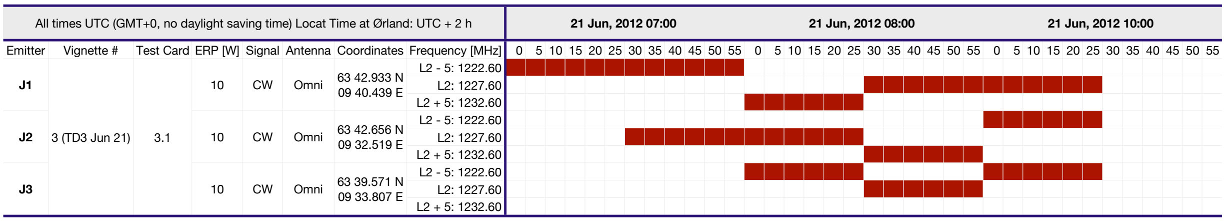


Figure 1: Jammer locations, depicted by red triangles



All times UTC (GMT+0, no daylight saving time) Locat Time at Ørland: UTC + 2 h								21 Jun, 2012 07:00												21 Jun, 2012 08:00												21 Jun, 2012 10:00											
Emitter	Vignette #	Test Card	ERP [W]	Signal	Antenna	Coordinates	Frequency [MHz]	0	5	10	15	20	25	30	35	40	45	50	55	0	5	10	15	20	25	30	35	40	45	50	55	0	5	10	15	20	25	30	35	40	45	50	55
J1	3 (TD3 Jun 21)	3.1	10	CW	Omni	63 42.933 N 09 40.439 E	L2 - 5: 1222.60	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥																								
							L2: 1227.60																			🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥						
							L2 + 5: 1232.60													🟥	🟥	🟥	🟥	🟥	🟥																		
J2			10	CW	Omni	63 42.656 N 09 32.519 E	L2 - 5: 1222.60																									🟥	🟥	🟥	🟥	🟥	🟥						
							L2: 1227.60							🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥																		
							L2 + 5: 1232.60																			🟥	🟥	🟥	🟥	🟥	🟥												
J3			10	CW	Omni	63 39.571 N 09 33.807 E	L2 - 5: 1222.60													🟥	🟥	🟥	🟥	🟥	🟥							🟥	🟥	🟥	🟥	🟥	🟥						
							L2: 1227.60																			🟥	🟥	🟥	🟥	🟥	🟥												
							L2 + 5: 1232.60																																				

Figure 2: The Jamming Schedule determines when each platform becomes active (red).

- The total duration of the scenario is three hours, starting at 09:00 and ending at 11:30 on June 12, 2012,
- The basic jammer platform behaviour is **a:** emit on a certain frequency near GPS frequency L2 for a certain amount of time, **b:** switch to another frequency, or **c:** switch jammer off,
- Jamming is continuous and omni-directional with ERP of 10 W on relevant GPS frequencies near L2,
- Jammers operate from the following locations:

	LAT	LON
J1	63°42.933 N	9°40.439 E
J2	63°42.656 N	9°32.519 E
J3	63°39.571 N	9°33.807 E

Figure 1 shows the jammer locations. The jamming schedule is according to the scheme depicted in Figure 2. During the start of the simulation, only one jammer is active per time slot. Gradually the scenario gets more complicated when after roughly two hours, all jammers start emitting simultaneously.

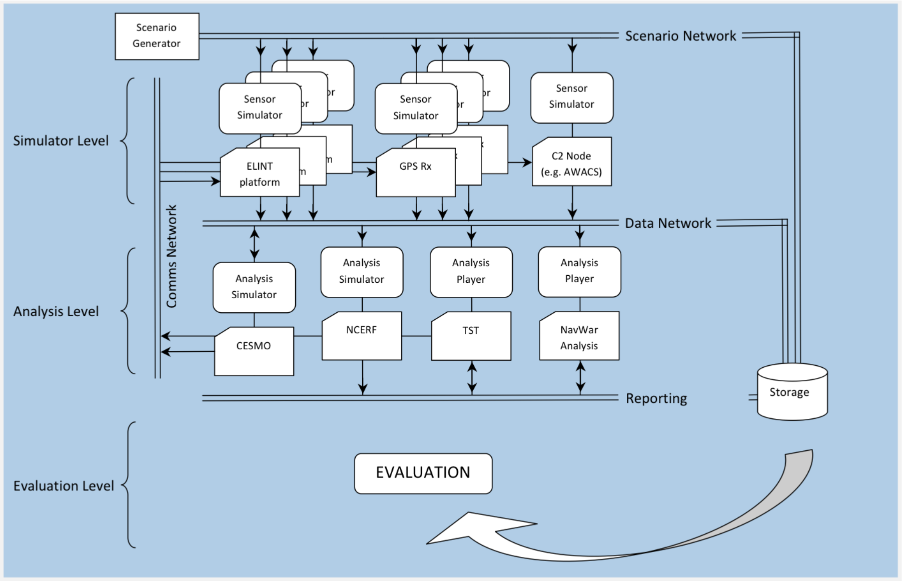


Figure 3: NWTB full architecture

Two ELINT platforms will be flying racetracks above the Norwegian Sea. They will scan the RF spectrum looking for hostile signals. Once they detect these kinds of signals, they estimate a bearing on the signals and cooperatively start a geo-location process to determine the exact position of the source of the emissions (CEMSO). Once geolocated, the emitter details are reported to C2 units. These units will compare the findings with other sources of information e.g., intelligence data or other ELINT reports and shall inform operational units about the newly discovered threat.

While processing through the TST decision cycle to determine the proper course of action for a time sensitive target, the Operational Commander maintains situational awareness on the target's whereabouts. Depending on the level of IT support available him, this information may be displayed on a geographical display, ranging from a simple Google Earth overview to the more elaborate applications.

After hostile GPS jammers are reported to the Commander, the NWAC can use the jammer's position to determine the impact of that jammer in the area. The impact prediction follows from EM coverage calculation and simulation models. These calculate the signals in space caused by the jammer and determine the expected performance for GPS receivers, given the build-in protection of these receivers.

3.0 NWTB architecture and implementation

The full NAVWAR Test Bed architecture as shown in Figure 3 was designed to support the analysis in the example outlined above (Section 2.1.3). It reflects the three groups of objects mentioned in Section 2.0), i.e. scenario, simulation and analysis. These levels are separated by three different networks, each transporting information originating from the respective groups of objects.

The use of networks as a means of transporting information between objects allows the test bed to be used in a distributed way and allows participants to connect to the test bed remotely in order to participate in tests or simulations. Furthermore, the distributed set up facilitates connectivity to deployed forces or other trials or exercises at the same time.

3.1 Networks

The origin of the system is a dynamic real-time scenario generator providing so-called Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs) [6, 7] on the scenario network. Each DIS PDU is a message

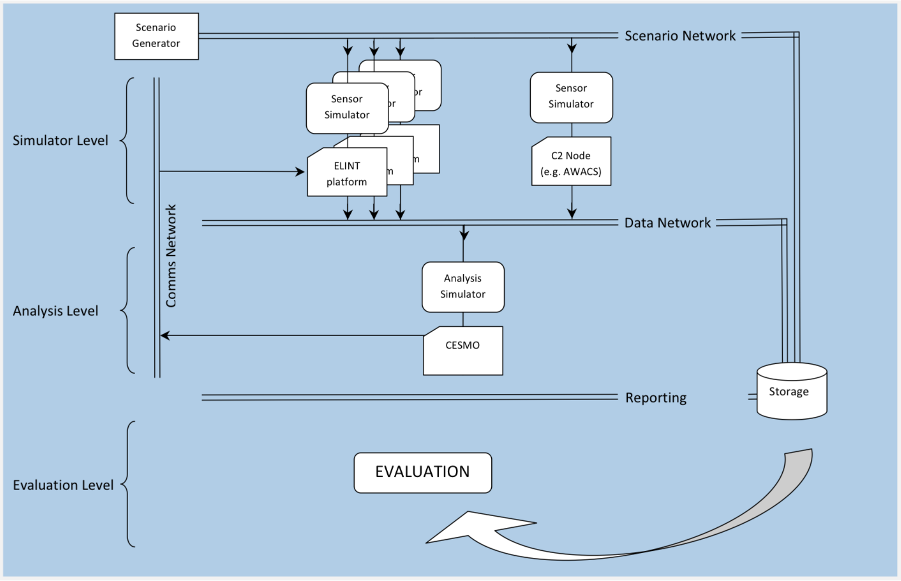


Figure 4: NWTB initial architecture

describing a particular event in the scenario i.e., the movement of jammer or ELINT platform, or activation and deactivation of jammer platforms.

The scenario network contains the information from the scenario generator and acts as the input network for the simulation objects. Simulation participants monitor the scenario network for relevant information that might affect their state. Upon finding relevant information on the scenario network, a simulation object may alter its state, resulting in the delivery of a PDU back to the scenario network. Alternatively, it may interpret events and extract information from it, which it in turn publishes on the data network. An ELINT platform, monitoring emission PDUs and generating Lines of Bearing (LOBs) on these emissions typically shows the latter behavior.

The third major network in the simulation is the reporting network to which all players at the analysis level report their ’products’ to. It serves as input for the evaluation.

3.2 Initial capability

The first major milestone in the development of the testbed that was reached at the time of this writing, is initial capability. Initial Capability was defined as being able to play the Cooperative ESM Operations (CESMO) game. It encompasses a limited scenario which includes the movement of jamming vehicles, the presence of jamming signals, the detection of these signals by ELINT platforms, the generation of LOBs, and their combination into geo-locations. The architecture needed for intial capability contains the complete network architecture of the full planned Test Bed, but only a limited number of objects interacting. Figure 4 shows the architecture for initial capability.

3.3 Principal objects

Figure 5 shows the principle objects that compose the NAVWAR Test Bed. The `NavWarTestBed` object and the `Time` are required to run in each local simulation environment for the Test Bed to function. The other objects are optional and can be seen as so called *plugins*. The sections below will describe the roles of the main objects in the NAVWAR Test Bed.

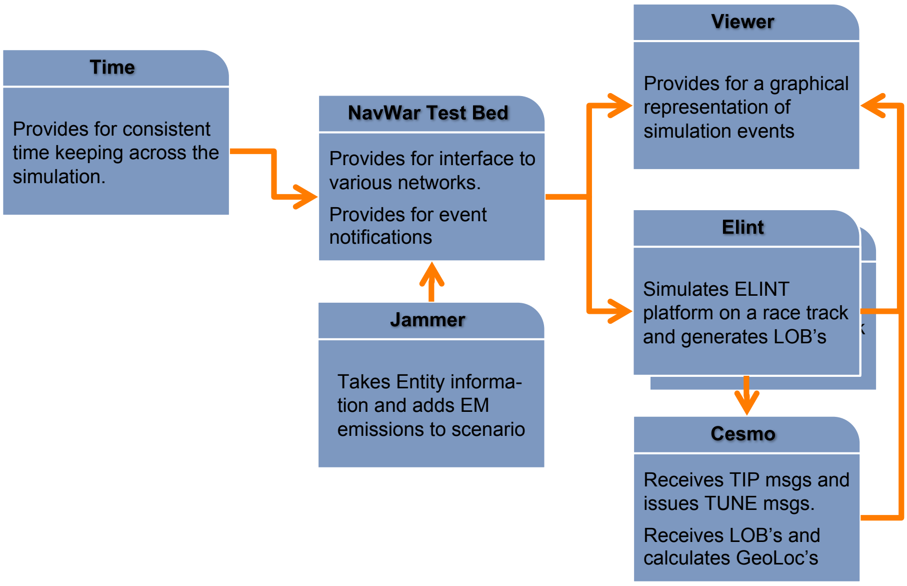

Time
Provides for consistent time keeping across the simulation.
NavWar Test Bed
Provides for interface to various networks.
Provides for event notifications
Jammer
Takes Entity information and adds EM emissions to scenario
Viewer
Provides for a graphical representation of simulation events
Elint
Simulates ELINT platform on a race track and generates LOB's
Cesmo
Receives TIP msgs and issues TUNE msgs.
Receives LOB's and calculates GeoLoc's

Figure 5: NWTB principal objects

3.3.0.1 NavWarTestBed Together with the `Time` object, the `NavWarTestBed` object forms the heart of the simulation. It forms the connection point between the networks and the other objects. It constantly monitors the networks (triggered by the fast heartbeat provided by the `(Time)` object). If it detects data on one of the networks, it analyses the data and notifies the other objects in the simulation by sending appropriate MATLAB *events*. Additionally, in case of PDU data on the scenario network, it extracts the timestamp from the PDU and updates the `Time` object with the current time.

3.3.0.2 Time The `Time` object provides for consistent time keeping across the local simulation. It is defined globally, hence every other object in the simulation can access to it to get the current time, which they typically do by calling its `now()` method. Each time the `now()` method is called, the `Time` object calculates the elapsed time since the last `update()` call using the MATLAB `tic()`-`toc()` construct.

Setting the time is done by using `Time`'s `update()` method with the current time as a parameter. `update()` Is typically called by the `NavWarTestBed` object (see above).

Additionally to the `now()` and `update()` methods, the `Time` object contains two timers that can be looked upon as the heartbeat of the simulation. Both timers send MATLAB *events* at regular intervals, the first one with a frequency of 100/s and the other one with 1/s. This is done to facilitate other objects to perform periodic tasks, for instance polling the network ports by the `NavWarTestBed` object, or sending entity PDU's by the `Elint` object.

3.3.0.3 Jammer The `Jammer` object is responsible for transmitting Electro Magnetic Emission PDUs on the scenario network. As such, it is actually part of the scenario generator, but since the current generator is not capable of issuing Emission PDUs it takes over this task. It contains a schedule (Figure 2 which determines when each of the platforms in the scenario's becomes an active jammer. If an Entity PDU appears on the scenario network, the `Jammer` object the takes the Entity information and adds EM emissions to the scenario, if the schedule dictates so.

Notice, that the `Jammer` object is reactive, i.e., it only sends Emission PDUs *in response* to an Entity PDU. This implies that it cannot send "stop-emission" messages, and thus that the Emission PDU is only valid for a short time interval. This interval typically is of the length if the time interval between two consecutive PDUs of the same entity, which is typically 10 s.

3.3.0.4 Elint The `Elint` object simulates an ELINT platform, both by defining it as an entity in the simulation and by acting as a sensor to detect Electro Magnetic Emission PDUs.

The platform entity method uses a race track, defined by two focal points, a radius and a velocity parameter. With every slow heartbeat of the simulation (see `Time` object), the method calculates the platform's actual position and transmits an Entity PDU on the scenario network.

The objects sensor method is reactive, i.e., it becomes active with every occurrence of an Electro Magnetic Emission PDU on the scenario network. It then determines if the sensor 'sees' the emission by sampling a detection probability distribution based on the calculated signal-to-noise ratio (SNR) which in turn takes into account the distance between the platform's actual position and the emitter's position.

Upon detection, the `Elint` object generates a Line-of-Bearing (LOB) message on the data network. This message consists of a number of datafields, concatenated into a single ASCII string, separated by a forward-slash character and terminated by two forward-slash characters. It includes a trigram indicating a line-of-bearing message, i.e. '`LOB/`', the time of measurement (in seconds since 01 Jan 1970), the Elint ID, the Latitude and Longitude of the measurement platform, the bearing estimate of the detection (in decimal degrees relative to North and increasing over East), the precision of the bearing estimate (in decimal degrees), a NATO Spot Number (NSN) indicating the signal type of the detection, the measured frequency of the signal (in MHz), and the estimated power (in W) A typical LOB message looks like `LOB/1340271394/20/63.471562/7.604336/75.38/3.00/17503/1232.60/10.00//`

Note that the sensor method at the moment is very basic. Apart from the distance of the platform to the emission, it does not take into account other dynamics, such as receiver instantaneous tuning characteristics, busy states, or antenna pointing direction. A goal for a future version of the NAVWAR Test Bed is the development of a more sophisticated ELINT receiver object.

3.3.0.5 Cesmo The `Cesmo` object listens to the simulation network for LOBs generated by the ELINT platforms in the simulation. Upon receipt of two LOBs that originate from two different platforms, and that arrive within a short (predefined) timeframe, the `Cesmo` object calculates the intersection of the two LOBs, including an error estimate. The result is called a *geolocation* (see Appendix A for a mathematical derivation of the geolocation calculation.)

The `Cesmo` object reports the geolocation back to the simulation network using a message similar to the LOB messages. It includes a trigram indicating a geolocation message, i.e. '`GEO/`', the time of measurement (in seconds since 01 Jan 1970), the Latitude and Longitude of the intersection, the major and minor axes of the error ellipse (in metres), the orientation of the error ellipse (in decimal degrees relative to North and increasing over East), and a NATO Signal Number (NSN), indicating the signal type of the detection. A typical geolocation message looks like `GEO/1340271432/63.755424/9.686156/9133.18/1322.82/59.08/17503//`

A more realistic `Cesmo` object will combine more than two LOBs into a single geolocation or will update an existing geolocation with a new LOB. It may possess mechanisms to respond to preliminary emission indications from ELINT platforms (TIP messages) or to task the platforms to listen for specific emissions (TUNE messages) if it needs to update or refine a geolocation. In a later version, the `Cesmo` object may be updated to send and receive these TIP and TUNE messages on the communication network, for a more dynamic simulation (see Figure 4).

3.3.0.6 Viewer The final object of interest to the user is the `Viewer` object. A `Viewer` object listens to both the scenario and the simulation network and graphically displays all the information on a geological map. This information is updated with every event and includes the location of jammer platforms and ELINT platforms, the emission of jamming signals, lines of bearing, and geolocations. The figures in section 4.0 show the output of the `Viewer` object.

3.4 Communication

There are two mechanisms by which the objects comprising the testbed communicate. Firstly, they communicate by sending User Datagram Protocol (UDP) messages over the networks. These messages are therefore not confined to the local simulation but are distributed to all systems that are part of the testbed. On the scenario network, they are commonly PDU messages, describing the state of an object (see also Section 3.1. On the simulation network, these messages are simulation specific messages. In the current implementation, they comprise of LOB messages and GEO-location messages. We've seen examples of these messages in section 3.3.

The second type of messages are MATLAB *events*. These messages are confined to a single MATLAB instance. Hence, they control the course of action only locally. Two of them are generated by the `Time` object with a regular rate of 1/s and 100/s. These events can be looked upon as the heartbeat of the simulation. The slow event triggers the `Elint` objects in the simulation to update their state information and issue a PDU to the scenario network.

The fast event triggers the `NavWarTestBed` object to check if there is UDP data available on one of the networks. If this is the case, `NavWarTestBed` generates a `dataOnPort` event, notifying other objects in the simulation. Object that are subscribed to this specific event include the `Jammer` object to append an Electro Magnetic Emission PDU if appropriate (see Section 3.3), the `Elint` object to try a detection if an Electro Magnetic Emission PDU shows up on the scenario network, and the `Cesmo` object to generate a geolocation if LOB messages show up on the simulation network.

3.5 Implementation

The current version of the NAVWAR Test Bed is implemented in MATLAB version 2010a, using MATLAB's Object Oriented features. Nearly all of the actions, function and procedures that make up the test bed are actually readily available in MATLAB, except for some specific networking actions.

All classes that implement the composing and parsing of PDU messages follow the class definitions from the Open Source implementation of the Distributed Interactive Simulation Protocol (Open-DIS) [8].

Although MATLAB contains classes that implement the handling of UDP traffic on the NWTB networks, they are part of the Instrument Control Toolbox. However, for reasons of cost and accessibility, we decided to use the Java implementation instead. Specifically, the NAVWAR Test Bed implements UDP traffic handling using the `java.net.DatagramSocket` and the `java.net.DatagramPacket` classes.

4.0 The NWTB in action

This section illustrates the functionality of the Test Bed by gradually stepping through the commands required to start a simulation.

In the example, the testbed runs on a single laptop, which is attached to the local network `localhost`. We start by instantiating a global `Time` object

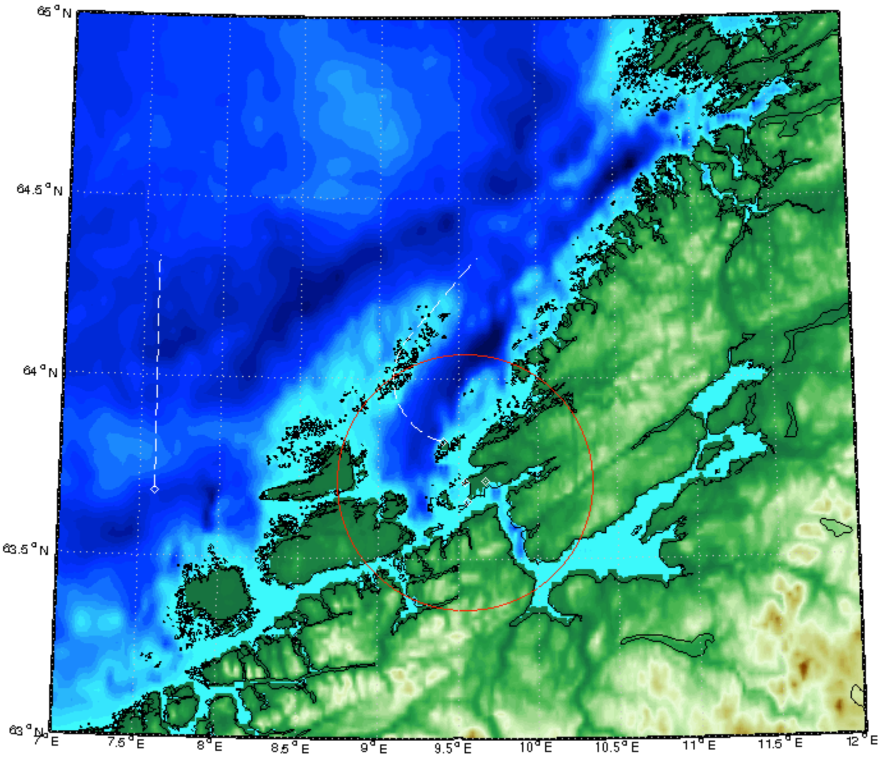
```
>> VignetteStart = '21-Jun-2012 09:00';
>> T0 = 24*3600*(datenum(VignetteStart)-datenum('01-Jan-1970'));

>> global TIME; TIME = Time(T0, 1);
```

Then, we define the scenario and the simulation networks, which are actually just ports on which the UDP packages are published and received.

```
>> scenario = jUpd(localhost:3000);
>> simulation = jUdp(localhost:3100);
```

Next, we combine the two networks together in the actual Test Bed class



```
>> VignetteStart = '21-Jun-2012 09:00';
>> T0 = 24*3600*(datenum(VignetteStart)-datenum('01-Jan-1970'));
>>
>> global TIME; TIME = Time(T0+1800, 1);
>>
>> scenario   = jUdp('localhost:3000');
>> simulation = jUdp('localhost:3100');
>> nwtb       = NavWarTestBed(scenario, simulation);
>>
>> scenario.startlistening
Entity with id =  2 added to list; Now keeping track of 1 entities.
Entity with id =  1 added to list; Now keeping track of 2 entities.
Entity with id =  3 added to list; Now keeping track of 3 entities.
>>
>> R1 = RaceTrack(T0, [63.5; 8], [64.3; 8], 20000, 5000, 250);
>> R2 = RaceTrack(T0, [64; 9.5], [64.5; 10.5], 20000, 5000, 250);
>>
>> elint1 = Elint(nwtb, DisID(42, 4, 20), R1);
Entity with id = 20 added to list; Now keeping track of 4 entities.
>>
>> elint2   = Elint(nwtb, DisID(42, 4, 21), R2);
Entity with id = 21 added to list; Now keeping track of 5 entities.
>>
>> music = Jamming(ntwb);
```

Figure 6: NWTB in action I

```
>> nwtb = NavWarTestBed(scenario, simulation);
```

and we attach a viewer object to the Test Bed

```
>> viewer = mViewer(nwtb);
```

We instantiate two `Elint` objects with ID 20 and 21, respectively, flying different race tracks (note that a racetrack consists of two half circular trajectories that are connected by straight trajectories. It is defined by a start time, two focal points [in lat-long coordinates], a radius [in metres], an altitude [in metres], and a velocity [in m/s])

```
>> R1 = RaceTrack(T0, [63.5; 8], [64.3; 8], 20000, 5000, 250);
>> R1 = RaceTrack(T0, [64; 9.5], [64.5; 10.5], 20000, 5000, 250);

>> elint1 = Elint(nwtb, DisID(42, 4, 20), R1);
>> elint2 = Elint(nwtb, DisID(42, 4, 21), R2);
```

Finally, we start the jamming, and start listening on the scenario network for activity.

```
>> music = Jamming(nwtb);

>> scenario.startlistening;
```

Figure 6 shows the `viewer` window. It shows a number of vehicles moving on the island of Sardinia, and two ELINT platforms doing their racetracks above the Tyrrhenian Sea. Additionally, the Figure shows an emission originating from one of the vehicles. The output of the NAVWAR Test Bed to the console is depicted under the `viewer` window.

The next step is to start listening on the simulation network for LOBs generated by the ELINT platforms and to start a `Cesmo` instance that combines them into a geolocation

```
>> simulation.startlistening;

>> sia = Cesmo(nwtb);
```

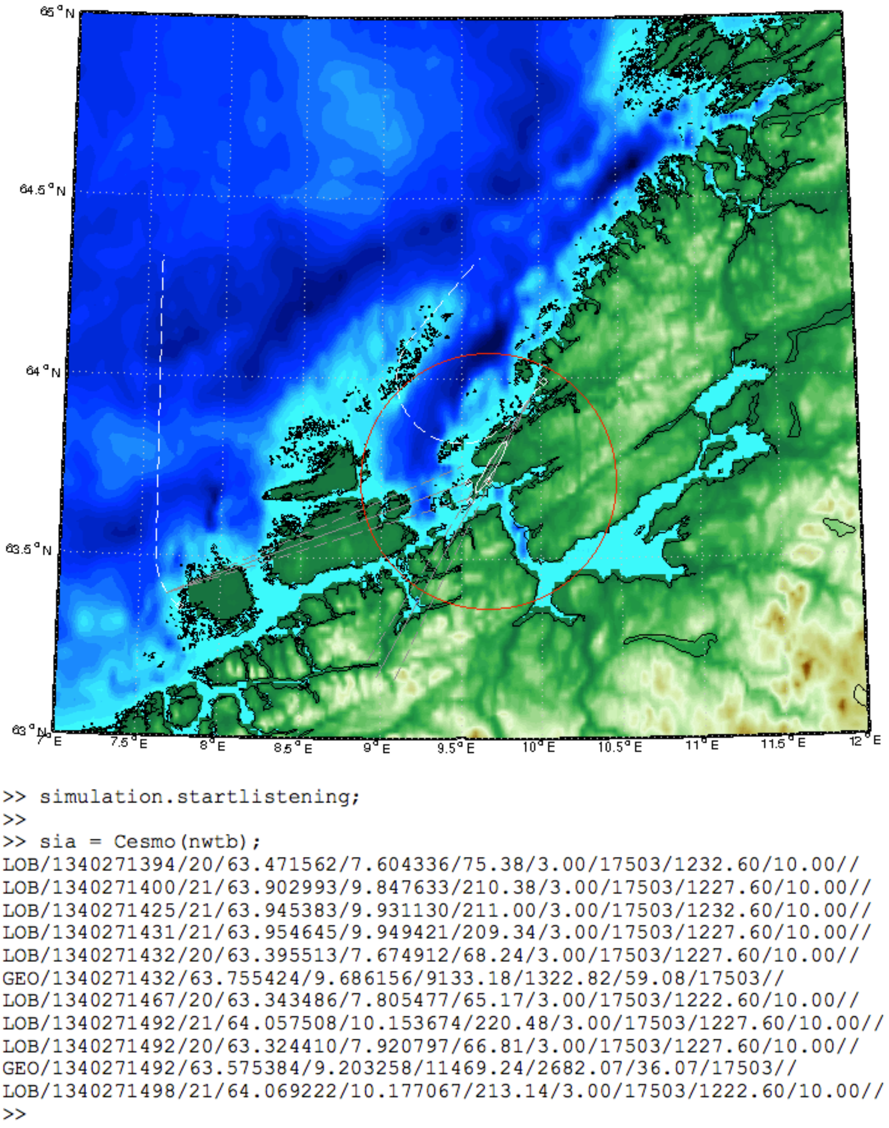
After invoking these two commands, the LOB messages generated by the ELINT platforms are displayed as literal text on the console and as dispersing lines on the graph. If the ELINT platforms respond to an emission simultaneously, the `Cesmo` instance combines them into a geolocation and publishes it on the simulation net. Again, the `nwtb` object displays it as text on the console and as an ellipse on the graph. Figure 7 shows the resulting `viewer` window.

5.0 Conclusion

This paper described the development of a NAVWAR Test Bed. It described the architecture of different networks, which the Test Bed uses as means of communication and which facilitates its distributed nature. It described the top level objects of which the Test Bed consists and how they interact to achieve a specific simulation goal.

The Test Bed has reached a maturity that allowed it to be tested and used in an environment that is wider than the NCIA laboratory. These include testing it in operational trials like UV12.

The simple setup of Section 4.0 illustrates one of many possible uses of the Test Bed. E.g., with this setup, a user can quickly assess the optimal ELINT trajectories that maximise the probability of jammer signal intercept while minimising the resulting geolocation errors. Other aspect of the CESMO concept van be easily studied as well.



```
>> simulation.startlistening;
>>
>> sia = Cesmo(nwtb);
LOB/1340271394/20/63.471562/7.604336/75.38/3.00/17503/1232.60/10.00//
LOB/1340271400/21/63.902993/9.847633/210.38/3.00/17503/1227.60/10.00//
LOB/1340271425/21/63.945383/9.931130/211.00/3.00/17503/1232.60/10.00//
LOB/1340271431/21/63.954645/9.949421/209.34/3.00/17503/1227.60/10.00//
LOB/1340271432/20/63.395513/7.674912/68.24/3.00/17503/1227.60/10.00//
GEO/1340271432/63.755424/9.686156/9133.18/1322.82/59.08/17503//
LOB/1340271467/20/63.343486/7.805477/65.17/3.00/17503/1222.60/10.00//
LOB/1340271492/21/64.057508/10.153674/220.48/3.00/17503/1227.60/10.00//
LOB/1340271492/20/63.324410/7.920797/66.81/3.00/17503/1227.60/10.00//
GEO/1340271492/63.575384/9.203258/11469.24/2682.07/36.07/17503//
LOB/1340271498/21/64.069222/10.177067/213.14/3.00/17503/1222.60/10.00//
>>
```

Figure 7: NWTB in action II

The Test Bed may also be used as input for procedural studies. By taking the information from the simulation network and feeding them into a real-life NAVWAR Analysis Cell, for instance, different aspect of a draft NAVWAR Concept of Operations (CONOPS) can be studied, thus facilitating CONOPS development.

The real meat of the Test Bed is in the functionality and in the level of realism of the objects themselves. For optimal use, third parties should contribute by writing and attaching their own simulator objects. The Test Bed facilitates all the interfacing and communication between them, thus building a large simulator with possibilities that surpass the sum of those of its components.

References

[1] NATO STANAG 4621 - *NAVWAR definition*

[2] NATO STANAG 4665 - *Navigation Warfare Operational Planning and Management*

[3] NAVWAR CONOPS - *Navigation Warfare Concept of Operations - A guide to Commanders and Staff on the integration of NAVWAR in operations and the planning process* **Final (4th) Draft**

[4] Sjoerd J Gelsema, Jennifer K Hebert, Anthony J Rocco and René Thaens, *Trial Imperial Hammer 2008: Navigation Warfare*, **Technical Note 1439**, NATO Consultation, Command and Control Agency, The Hague, Netherlands, April 2010. (NATO UNCLASSIFIED)

[5] Sjoerd J Gelsema *et.al*, *Unified Vision 2012: Navigation Warfare*, **Technical Note 0000**, NATO Consultation, Command and Control Agency, The Hague, Netherlands, *To be published*. (NATO UNCLASSIFIED)

[6] *IEEE Standard for Distributed Interactive Simulation - Application Protocol* **IEEE Std 1278.1-1995**, Institute of Electrical and Electronic Engineers, 21 September 1995.

[7] *IEEE Standard for Distributed Interactive Simulation - Application Protocol* **IEEE Std 1278.1a-1998**, Institute of Electrical and Electronic Engineers, 18 August 1998.

[8] *Open-DIS, An Open Source Implementation of the Distributed Interactive Simulation Protocol*, http://open-dis.sourceforge.net/Open-DIS.html

Appendix A

Angle of Arrival Analysis

Emitter position estimation

Let the angle of arrival (AoA) of a signal originating from a location (x_{Em}, y_{Em}) at receiver n at location (x_n, y_n), be denoted by φ_n. Then

$$\varphi_n = \tan^{-1}\left(\frac{x_{Em} - x_n}{y_{Em} - y_n}\right).$$

Let α_n be defined by

$$\alpha_n = \tan \varphi_n = \left(\frac{x_{Em} - x_n}{y_{Em} - y_n}\right). \tag{1}$$

The emitter position (x_{Em}, y_{Em}) can then be calculated from two AoA measurements (also known as Lines of Bearing (LOBs)) from aircraft positions (x_1, y_1) and (x_2, y_2) as [1]

$$x_{Em} = \frac{x_1 - x_2\frac{\alpha_1}{\alpha_2} - y_1\alpha_1 + y_2\alpha_1}{1 - \frac{\alpha_1}{\alpha_2}}, \tag{2}$$

and

$$y_{Em} = \frac{x_1 - x_2\frac{\alpha_1}{\alpha_2} - y_1\alpha_1 + y_2\alpha_1}{(\alpha_2 - \alpha_1)} - \frac{x_2}{\alpha_2} + y_2. \tag{3}$$

Precision of location estimation

Covariance matrices

The uncertainty in aircraft position is negligible. The uncertainty in measured AoA, however, is much larger and cannot be ignored. It is treated as independent random variable $\Delta\varphi_n$, having zero mean and variance $\sigma^2_{\varphi_n}$. The associated covariance matrix $C_{\varphi_{12}}$ can be written as

$$C_{\varphi_{12}} = \begin{bmatrix} \sigma^2_{\varphi_1} & 0 \\ 0 & \sigma^2_{\varphi_2} \end{bmatrix}.$$

The emitter position covariance matrix $C_{x_{Em}y_{Em}}$ due to uncertainties in emitter angles of arrival (φ_1, φ_2) can be computed as

$$C_{x_{Em}y_{Em}} = JC_{\varphi_{12}}J^T,$$

with

$$\begin{aligned} J &= \begin{bmatrix} \frac{\partial x_{Em}}{\partial \varphi_1} & \frac{\partial x_{Em}}{\partial \varphi_2} \\ \frac{\partial y_{Em}}{\partial \varphi_1} & \frac{\partial y_{Em}}{\partial \varphi_2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial x_{Em}}{\partial \alpha_1}\frac{\partial \alpha_1}{\partial \varphi_1} & \frac{\partial x_{Em}}{\partial \alpha_2}\frac{\partial \alpha_2}{\partial \varphi_2} \\ \frac{\partial y_{Em}}{\partial \alpha_1}\frac{\partial \alpha_1}{\partial \varphi_1} & \frac{\partial y_{Em}}{\partial \alpha_2}\frac{\partial \alpha_2}{\partial \varphi_2} \end{bmatrix}. \end{aligned}$$

Partial derivatives

With 1 we find that

$$\frac{\partial \alpha_n}{\partial \varphi_n} = 1 + \tan^2 \varphi_n = 1 + \alpha_2^2$$

To facilitate the derivation of the partial derivatives of x_{Em} and y_{Em} let us first rewrite 2 to

$$x_{Em} = \frac{x_1\alpha_2 - x_2\alpha_1 - y_1\alpha_1\alpha_2 + y_2\alpha_1\alpha_2}{(\alpha_2 - \alpha_1)}.$$

and 3 to

$$y_{Em} = \frac{x_1 - x_2 - y_1\alpha_1 + y_2\alpha_2}{(\alpha_2 - \alpha_1)}.$$

We can now derive the partial derivatives of x_{Em} as

$$\begin{aligned}\frac{\partial x_{Em}}{\partial \alpha_1} =& \frac{-x_2 - y_1\alpha_2 + y_2\alpha_2}{(\alpha_2 - \alpha_1)} + \\ & \frac{x_1\alpha_2 - x_2\alpha_1 - y_1\alpha_1\alpha_2 + y_2\alpha_1\alpha_2}{(\alpha_2 - \alpha_1)^2} \\ =& \frac{x_1\alpha_2 - x_2\alpha_2 - y_1\alpha_2^2 + y_2\alpha_2^2}{(\alpha_2 - \alpha_1)^2},\end{aligned}$$

and

$$\begin{aligned}\frac{\partial x_{Em}}{\partial \alpha_2} =& \frac{x_1 - y_1\alpha_1 + y_2\alpha_1}{(\alpha_2 - \alpha_1)} - \\ & \frac{x_1\alpha_2 - x_2\alpha_1 - y_1\alpha_1\alpha_2 + y_2\alpha_1\alpha_2}{(\alpha_2 - \alpha_1)^2} \\ =& - \frac{x_1\alpha_1 - x_2\alpha_1 - y_1\alpha_1^2 + y_2\alpha_1^2}{(\alpha_2 - \alpha_1)^2}.\end{aligned}$$

Similarly, we derive the partial derivatives of y_{Em} as

$$\begin{aligned}\frac{\partial y_{Em}}{\partial \alpha_1} =& -\frac{y_1}{\alpha_2 - \alpha_1} + \\ & \frac{x_1 - x_2 - y_1\alpha_1 + y_2\alpha_2}{(\alpha_2 - \alpha_1)^2} \\ =& \frac{x_1 - x_2 - y_1\alpha_2 + y_2\alpha_2}{(\alpha_2 - \alpha_1)^2}\end{aligned}$$

and

$$\begin{aligned}\frac{\partial y_{Em}}{\partial \alpha_2} =& \frac{y_2}{\alpha_2 - \alpha_1} + \\ & \frac{x_1 - x_2 - y_1\alpha_1 + y_2\alpha_2}{(\alpha_2 - \alpha_1)^2} \\ =& -\frac{x_1 - x_2 - y_1\alpha_1 + y_2\alpha_1}{(\alpha_2 - \alpha_1)^2}\end{aligned}$$

Error ellipse

The major axis, minor axis and the orientation of the ellipse can be readily derived from the covariance matrix $C_{x_{Em}y_{Em}}$ as follows. Let λ_1 and λ_2 be the eigenvalues of the covariance matrix $C_{x_{Em}y_{Em}}$ and let e_1 and e_2 be the corresponding eigen vectors. The major and minor axis of the error ellipse are given by $\sqrt{\lambda}_1$ and $\sqrt{\lambda}_2$, respectively and the direction of the ellipse is given by $\arctan \frac{e_{1_y}}{e_{1_x}}$.

References

[1] Richard G. Wiley, *ELINT - The Interception and Analysis of Radar Signals.*, **ISBN-10: 1-58053-925-4**, Artech House Inc., 685 Canton Street Norwood, MA 02062, 2006.