

AD-A282 858

ADST/WDL/TR-93-003208



ADST
Software Design Document
for the
BDS-D VIDS-equipped M1

Loral Western Development Labs
Electronic Defense Systems Software Department
Software Engineering Laboratory
3200 Zanker Road
P.O. Box 49041
San Jose, CA 95161-9041

Dec 23, 1993

Contract No. N61339-91-D0001
CDRL No. A001

Prepared for

Simulation Training and Instrumentation Command
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 328266-3275

LOREAL

DTIC
ELECTE
AUG 09 1994
S B D

5888 **94-24946**

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC QUALITY INSPECTED 1

94 8 08 015

REPORT DOCUMENTATION PAGE			Form approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 23 December 1993		3. REPORT TYPE AND DATES COVERED
4. TITLE AND SUBTITLE ADST Software Design Document for the BDS-D VIDS-equipped M1.			5. FUNDING NUMBERS Contract No. N61339-91-D-0001	
6. AUTHOR(S) Stivers, Douglas				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Loral Systems Company ADST Program Office 12443 Research Parkway, Suite 303 Orlando, FL 32826			8. PERFORMING ORGANIZATION REPORT NUMBER ADST/WDL/TR-93-003208	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Simulation, Training and Instrumentation Command STRICOM Naval Training Systems Center 12350 Research Parkway Orlando, FL 32826-3275			10. SPONSORING ORGANIZATION REPORT	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (Maximum 200 words) This document provides detailed information describing the software design of the Vehicle Integrated Defense System (VIDS) equipped M1.				
14. SUBJECT TERMS			15. NUMBER OF PAGES 54	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	17. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	17. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102

TABLE of CONTENTS

1.	Scope.....	1
1.1.	Identification.....	1
1.2.	System overview.....	1
1.3.	Document overview.....	2
2.	Referenced documents.....	2
2.1.	Government documents.....	2
2.2.	Non-Government documents.....	2
3.	Preliminary design.....	2
3.1.	CSCI overview.....	2
3.1.1.	CSCI architecture.....	4
3.1.2.	System states and modes.....	4
3.1.3.	Memory and processing time allocation.....	6
3.2.	CSCI Design Description.....	6
3.2.1.	VIDS-GT CSC.....	6
3.2.1.1.	VIDS_File_Read CSC.....	7
3.2.1.2.	VIDS_Init CSC.....	7
3.2.1.3.	VIDS_Simul CSC.....	8
3.2.1.4.	VIDS_Shutdown CSC.....	8
3.2.2.	PC-Resident VIDS CSC.....	8
4.	Detailed Design.....	10
4.1	VIDS-GT CSC Detailed Design.....	10
4.1.1.	VIDS_File_Read CSC.....	10
4.1.2.	VIDS_Init CSC.....	10
4.1.3.	VIDS_Simul CSC.....	10
4.1.3.1.	Get_CCDP_Updates CSU.....	11
4.1.3.2.	React_to_Power_State_Changes CSU.....	11
4.1.3.3.	Identify_Threats CSC.....	11
4.1.3.3.1.	EO_Sensor_Simul CSC.....	12
4.1.3.3.1.1.	Update_Delayed_Threats CSU.....	12
4.1.3.3.1.2.	Process_New_Threats CSU.....	12
4.1.3.3.1.3.	Test_Sensor_Coverage_Limits CSU.....	12
4.1.3.4.	Manage_Countermeasures CSC.....	13
4.1.3.4.1.	Prioritize_Threats CSC.....	13
4.1.3.4.1.1.	Fuse_Correlate_Threats CSU.....	14
4.1.3.4.1.2.	Get_Threat_Classification CSU.....	14
4.1.3.4.1.3.	Get_Threat_Priority CSU.....	16
4.1.3.4.1.4.	Sort_Prioritized_Threats CSU.....	16
4.1.3.4.1.5.	Update_All_Prioritized_Threats CSU.....	16
4.1.3.4.2.	Select_Countermeasures CSU.....	16
4.1.3.4.3.	Update_CPS_Coverage CSU.....	17
4.1.3.4.4.	Individual_CM_Simul CSU.....	17
4.1.3.4.4.1.	ROS_Simul CSU.....	18
4.1.3.4.4.2.	EO_CM_Simul CSU.....	19
4.1.3.4.4.3.	TCS_Simul CSU.....	20

4.1.3.4.4.4.	Decon_CM_Simul CSU.....	20
4.1.3.4.4.5.	One_Shot_CM_Simul CSU.....	21
4.1.3.4.4.6.	CFire_Simul CSU.....	21
4.1.3.4.4.7.	Turret_Scanning_Simul CSU.....	21
4.1.3.5.	Send_Updates_to_CCDP CSC.....	21
4.1.3.6.	Send_Updates_to_Network CSC.....	22
4.1.3.7.	Need_To_Release_Turret CSU.....	22
4.1.3.8.	CM_Rotation_Simul CSC.....	22
4.1.3.9.	Turret_Rotation CSC.....	23
4.1.3.10.	VIDS_Shutdown CSC.....	23
4.1.4.	XField CSC.....	24
4.1.5.	Cloud CSC.....	24
4.1.6.	Modifications to Existing Code.....	25
4.2.	VIDS-PC Detailed Design.....	26
4.2.1.	SMI_Init CSC.....	26
4.2.2.	SMI_Simul CSC.....	26
4.2.2.1.	Get_Button CSU.....	26
4.2.2.2.	Check_Alarms CSU.....	27
4.2.2.3.	Process_Rcv_PDU CSC.....	27
4.2.3.	SMI_Shutdown CSU.....	27
5.	CSCI data.....	28
6.	CSCI data files.....	38
7.	Requirements traceability.....	38
8.	Notes.....	50
Appendix A. Sensor/Countermeasure Configuration File Content.....		51

LIST of FIGURES

Figure 1. VIDS External Interfaces.....	3
---	---

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Date	
Additional Notes	
Dist	Special
A-1	

1. Scope.

1.1. Identification.

This document describes the software design for the Vehicle Integrated Defense System (VIDS) simulation and its inclusion into the existing M1 Tank Simulator software. This software design satisfies requirements contained in the Requirements traceability tables (Section 7).

1.2. System overview.

The VIDS-equipped M1 Tank Simulator exists to support a series of survivability experiments. The nature of the experiments requires that the VIDS simulation be parameter driven. The VIDS parameters not only define available sensors and countermeasures, but also define their respective sensitivities and response times. For the present, eight sensors and nine countermeasures are simulated:

Sensors

- a. Laser Warning Receiver (LWR).
- b. Missile Warning System (MWS).
- c. Future Armored System Radar (FASR).
- d. Seismic Mine Sensor.
- e. Non-Imaging System (NIS).
- f. Tank Radar Warning Receiver (TRWR).
- g. Muzzle Flash Detector (MFD).
- h. Nuclear Chemical Sensor (NCS).

Countermeasures

- a. Multi-Salvo Smoke Grenade Launcher/Rapid Obscuration System (ROS).
- b. Missile Countermeasure Device (MCD).
- c. Combat Protection System (CPS).
- d. Laser Countermeasure Device (LCMD).
- e. Vehicle Magnetic Signature Duplication (VEMASID).
- f. Nuclear Biological Chemical Overpressure (NBCOP).
- g. Advanced Threat Radar Jammer (ATRJ).
- h. Threat Countermeasure System (TCS).
- i. Chaff/Flares.

In general, the VIDS system responds to perceived threats in the following ways:

- a. by displaying visual icons on the Commander's Controls Display Panel (CCDP).

- b. by generating alert tones or playing alert messages which can be heard on the tank crew intercom.
- c. by examining user-selected countermeasure activation modes.
- d. by seizing control of the turret.
- e. by activating a selected countermeasure for each perceived threat.

Because VIDS can seize control of the turret, automatic turret rotation for counterfire is supported. Furthermore, VIDS supports automatic turret slewing for visual detection of threats.

1.3. Document overview.

This document identifies and describes new software CSCs and CSUs, as well as changes to and reuse of existing M1 Simulator CSCs and CSUs. Diagrams and narratives are used to explain how the new VIDS simulation executes within the framework of the existing M1 Simulator.

2. Referenced documents.

2.1. Government documents.

SPECIFICATIONS:

- 1. PM-Survivability: VIDS Armored Vehicle Survivability Equipment (AVSE) BDS-D Functional Specifications, 29 May 1992.

2.2. Non-Government documents.

- 1. Loral: Battlefield Distributed Simulation-Development (BDS-D) Vehicle Integrated Defense System (VIDS) Feasibility Analysis Report, 14 October 1992.
- 2. BBN: The SIMNET Network and Protocols, Report 7627, June 1991.

3. Preliminary design.

3.1. CSCI overview.

The VIDS-equipped M1 Tank Simulator (hereafter referred to as the VIDS-equipped tank) exists as one of many entities participating within a simulated battle. Each entity communicates with other entities by sending and receiving Protocol Data Units (PDUs) on an Ethernet© network. The external interfaces for the VIDS-equipped tank are illustrated in Figure 1.

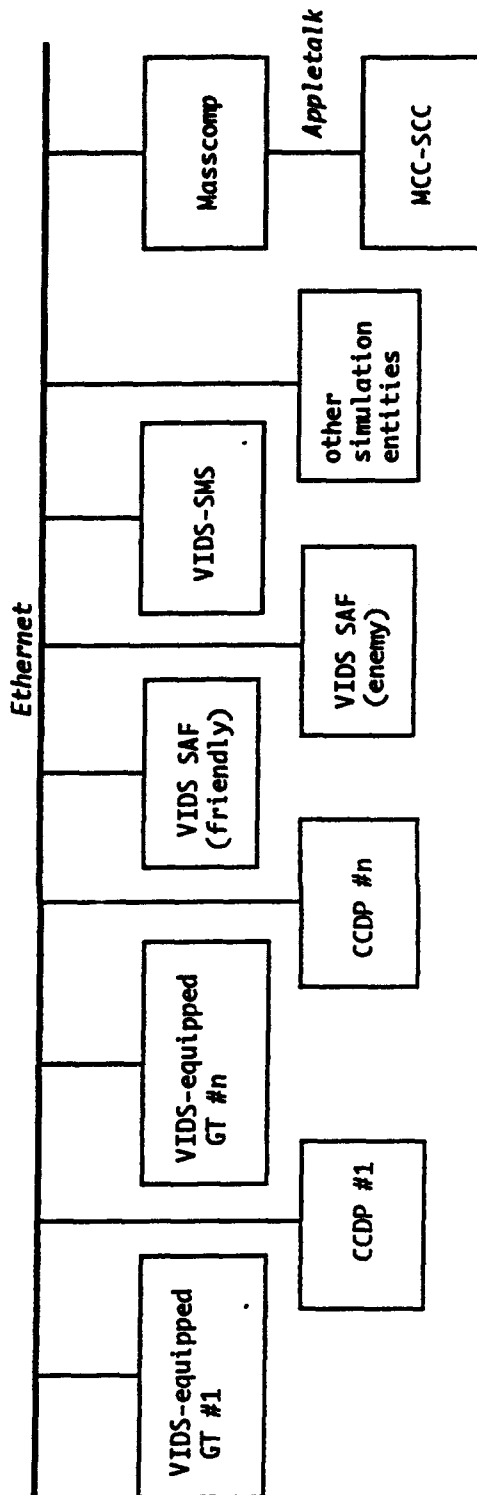


Figure 1. VIDS External Interfaces

Incoming PDUs describe the activity in one or more simulated battlefields. Because the number of incoming PDUs can be quite large, a series of high-level filters are applied to retain only those PDUs which are applicable to a

specific entity. Applicable PDUs are then filtered based upon entity-specific parameters such as exercise number, distance and available sensors. Remaining PDUs are then classified and used to influence either the entity behavior or what the entity can detect.

Outgoing PDUs describe the visual appearance or behavior of the VIDS-equipped tank. Typically, these define the current tank hull position and orientation, the turret orientation, the presence of smoke clouds, chaff/flares or the existence of electro-optical energy. For experimental purposes, a subset of the outgoing PDUs contain instrumentation information which can be used by analysts to better understand VIDS behavior and the use of the soldier-machine interface.

3.1.1. CSCI architecture.

The VIDS capability is partitioned between two host computers. One host is the current M1 tank GT hardware; the other host is a PC with an Elographics touchscreen mounted in front of a 13 inch color video monitor. The software executing on the PC supports the Soldier Machine Interface (SMI), hereafter referred to as the CCDP. This includes all the VIDS buttons, setup windows and the display windows. The software executing on the GT simulates the behavior of the sensors, countermeasures and threat resolution module (TRM).

The VIDS-PC and VIDS-GT communicate with one another just like other entities participating within a simulated battle exercise. Because there may be multiple VIDS-equipped tanks within the same exercise, the VIDS-PC and VIDS-GT are paired so only appropriate network messages are recognized and processed. In other words, the VIDS-PC knows the unique identifier (VehicleID) of its corresponding VIDS-GT, and the VIDS-GT knows the unique VehicleID of its corresponding VIDS-PC.

3.1.2. System states and modes.

The VIDS-GT CSCI operates in one of six predefined states. These states are:

- a. Startup.
- b. Idle.
- c. Initialize.
- d. Simulate.
- e. Stop.
- f. Exit.

Within the VIDS context, only the Startup, Initialize and Simulate states are significant.

VIDS SDD

During the Startup State, specific hardware devices are initialized and parameter files are read. It is during this state that the VIDS parameter file (VIDS.D) is read to establish the types and behaviors of available sensors and countermeasures. This file also contains a recommended list of countermeasures for each type of threat.

Once all the tank parameter files have been read, a communication link to the Simulation Network (SIMNET) is established. Having successfully completed these tasks, a transition is made to the Idle State.

During the Idle State, the M1 Tank Simulator waits to receive an activation request from SIMNET. (The activation request is generated by a user of the MCC-SCC console.) When an Activation Request PDU is received, a transition is made to the Initialize State.

During the Initialize State, additional hardware and internal software initialization is performed. For VIDS, sensor detection and identification probability tables are built; and default alert, safety, countermeasure coverage and turret scanning sector settings are sent to the VIDS-PC. Additionally, the list of available countermeasures and the inventories of the expendable countermeasures are sent to the VIDS-PC. Having successfully completed this initialization, a transition is made to the Simulate State.

The Simulate State represents the main processing loop for the VIDS-GT. PDUs sent by the VIDS-PC are read and used to alter the behavior of VIDS. Electro-optical PDUs from other entities are read and used to determine if a threat is present. When a threat is detected, PDUs are sent to the VIDS-PC to provide visual and audible cues. Furthermore, detected threats are prioritized; and countermeasures are selected and activated. The Simulate State continues until:

- a. An impact PDU is received which destroys the tank.
- b. A deactivation PDU is received which forces a transition to the Stop State.
- b. A reconstitute PDU is received which forces a transition to the Idle State.
- c. An exit command is received from the M1 Console keyboard which forces a transition to the Exit State.

During the Stop State, a transition is made to the Idle State, followed by a transition to the Exit State.

For the VIDS-PC, there are only three states: Initialize, Simulate and Shutdown. During the Initialize State, data files are read which define button positions, content and behavior. Furthermore, the VIDS-PC waits to receive the default alert, safety, countermeasure coverage and turret scanning sector

settings from the VIDS-GT. Once these settings have been received, a transition is made to the Simulate State.

As with the VIDS-GT, the Simulate State represents the main processing loop for the user interface. The touchscreen is continually monitored to determine if a button has been depressed or released. Specific button actions may generate brief user alert messages to appear on the display panel. Changed button values or sector coverage widths are sent back to the VIDS-GT to influence the behavior of the sensors and countermeasures. The network buffer is continually polled to determine if PDUs sent by the VIDS-GT require updates to the display or if audible alerts must be activated or terminated.

During the Shutdown State, dynamic memory is released, special interrupt handling is suspended and control is released to the normal operating system.

3.1.3. Memory and processing time allocation.

At the present time, there are no memory budgets more restrictive than those imposed by the respective host computers. However, the VIDS-GT functions which execute during the Simulate State must execute faster than 1/15th of a second. This is due to the fundamental execution cycle on the GT. In fact, the VIDS software execution speed must take only a relatively small percentage (20% or less) of the 66.67 milliseconds since the sum total of all simulated M1 behavior must execute within this time frame.

3.2. CSCI Design Description.

Because the simulated VIDS system is partitioned between two host computers, the description of the VIDS CSC is divided into two parts: the VIDS-GT CSC and the VIDS-PC CSC. Note that the VIDS-PC CSC is also referred to as the Soldier Machine Interface (SMI) and the Commander's Controls Display Panel (CCDP).

3.2.1. VIDS-GT CSC

The VIDS-GT CSC handles the job of simulating the behaviors of available sensors and countermeasures. Parameters sent by the VIDS-PC CSC are used to constrain the behavior of the VIDS-GT CSC. Parameters sent by the VIDS-GT CSC to the VIDS-PC CSC are used to inform the tank commander what is known about any hostile threats. Specific design features include:

- a. The VIDS-GT CSC satisfies all requirements presently allocated to the GT. Refer to the table in Section 7 to locate which specific requirements are satisfied.

VIDS SDD

- b. The VIDS-GT CSC is subdivided into three lower-level CSCs: VIDS_File_Read, VIDS_Init and VIDS_Simul.
- c. Each of the three lower-level CSCs are executed in sequential order. VIDS_File_Read and VIDS_Init are executed only once; VIDS_Simul is executed 15 times a second as part of the M1 code which executes in the Simulate State.

3.2.1.1. VIDS_File_Read CSC

The VIDS_File_Read CSC handles the job of reading a specific text file (VIDS.D) defining the available sensors and countermeasures and the corresponding behaviors. Specific design features include:

- a. This CSC satisfies the requirements for a parameter-driven set of sensor and countermeasure behaviors. Refer to the table in Section 7 to locate which specific requirements are satisfied.
- b. This CSC sequentially reads a specific text file. Each line is either a comment or a keyword-value(s). Comment lines are skipped. Keywords are used to discriminate which values are being read, what format must be used, and where they must be stored.
- c. The text file containing the sensor and countermeasure behaviors is read only once.

3.2.1.2. VIDS_Init CSC

The VIDS_Init CSC handles the job of preallocating dynamic memory, initializing queues, initializing countermeasure rotation CSCs and sending default parameters to the VIDS-PC. Specific design features include:

- a. This CSC does not satisfy any system-level requirements.
- b. This CSC handles the job of preallocating and initializing the dynamic memory to be used during the simulation of the VIDS behavior. Furthermore, it performs a critical initialization step by sending the VIDS-PC a set of default alert, safety, countermeasure coverage, turret scanning sector settings the list of available countermeasures and the inventories of the expendable countermeasures.
- c. This CSC satisfies the design requirements for preallocating and initializing dynamic memory and providing default parameters to the VIDS-PC.

- d. This CSC satisfies the design requirements for initializing internal tables used by the CSCs which handle countermeasure rotations.

3.2.1.3. VIDS_Simul CSC

The VIDS_Simul CSC handles the job of managing the majority of other lower-level CSCs. It is these lower-level CSCs which model the behavior of the available sensors, countermeasures and TRM. Specific design features include:

- a. This CSC and its lower-level CSCs satisfy a majority of the system-level requirements allocated to the GT. Refer to the table in Section 7 to locate which specific requirements are satisfied.
- b. This CSC handles the job of sequentially executing lower-level CSCs. These CSCs perform the following functions:
 - 1. Getting updates from the CCDP.
 - 2. Reacting to main and turret power state changes.
 - 3. Determining if there are new threats.
 - 4. Classifying threats based upon what is currently known.
 - 5. Prioritizing the current threats.
 - 6. Selecting countermeasures.
 - 7. Activating countermeasures.
 - 8. Sending updated threat status to the VIDS-PC.
 - 9. Sending countermeasure activation PDUs to other entities participating within the same simulated battle exercise.
 - 10. Managing the rotation of the independent slewing countermeasures.
- c. This CSC satisfies the design requirement for monitoring the main tank and turret power states.

3.2.1.4. VIDS_Shutdown CSC

The VIDS_Shutdown CSC handles the job of terminating the VIDS simulation and sending the final power states to the VIDS-PC. Specific design features include:

- a. This CSC does not satisfy any system-level requirements.
- b. This CSC satisfies the design requirements of formally deallocating the dynamic memory used during the simulation of the VIDS behavior. Furthermore, it prints a summary of the memory usage to the main console.

3.2.2. PC-Resident VIDS CSC

VIDS SDD

The VIDS-PC CSC handles the job of simulating the CCDP. This includes a set of multi-function buttons as well as the ability to activate audible alarms and display threat information. The display screen is used to portray the type and position of threats relative to the tank. Specific design features include:

- a. The VIDS-PC CSC satisfies all requirements presently allocated to the SMI. Refer to the table in Section 7 to locate which specific requirements are satisfied.
- b. The VIDS-PC CSC is subdivided into 3 lower-level CSCs: SMI_Init, SMI_Simul, SMI_Shutdown.
- c. Each of the three lower-level CSCs are executed in sequential order. SMI_Init and SMI_Shutdown are executed only once; SMI_Simul is executed endlessly until a keyboard Control-C or the right mouse button is depressed.

4. Detailed Design.

The detailed design is divided into two parts. The first part describes the VIDS-GT CSC and the second part describes the VIDS-PC CSC.

4.1 VIDS-GT CSC Detailed Design

4.1.1. VIDS_File_Read CSC

VIDS_File_Read reads a text file (VIDS.D) which defines the list of available sensors and countermeasures and the corresponding behaviors. This CSC is executed only once during the Startup State of the existing M1 code. Furthermore, the text file contains automatic turret rotation rates for each countermeasure, counterfire and turret scanning. The text file also contains the unique identification (VehicleID) of the PC which simulates the behavior of the corresponding CCDP.

Once all the parameters have been read, they are stored in tables. For sensors, this includes coverage angles, range limitation, detection probabilities and reaction/delay times. For countermeasures, this includes the coverage angles, reaction/delay times and in some cases (ROS and Chaff/Flares) launch distances. For the Threat Resolution Module (TRM), this includes the recommended countermeasures and priorities for each threat type.

4.1.2. VIDS_Init CSC

VIDS_Init preallocates dynamic memory structures which are used frequently during the execution of the VIDS_Simul CSC. Preallocation is done here purely for efficiency because VIDS_Init is invoked during a non-critical processing state.

VIDS_Init also sends default settings and expendable countermeasure inventories to the CCDP. This done to allow alert, safety, countermeasure coverage and turret scanning sector settings to be graphically portrayed when the CCDP is powered on.

Finally, VIDS_Init invokes functions which initialize the independently slewing countermeasure rotation tables. During this initialization, slewing countermeasure devices are aligned in the same direction as the main gun.

4.1.3. VIDS_Simul CSC

VIDS_Simul serves as the primary entry point for simulation of the sensors, TRM and countermeasures. It represents the root of a functional hierarchy which is executed once during each execution cycle of the existing, M1

simulation software. During a single execution cycle, the following high-level functions are executed:

```
Get_CCDP_Updates();
React_to_Power_State_Changes();
Identify_Threats();
Manage_Countermeasures();
Send_Updates_to_CCDP();
Send_Updates_to_Network();
Need_To_Release_Turret();
CM_Rotation_Simul();
Poison_Simul();
```

Each of these functions represent a functional sub-hierarchy which is described in the following sections.

4.1.3.1. Get_CCDP_Updates CSU

Get_CCDP_Updates retrieves the current CCDP settings. These settings are changed by user interaction with the touch screen. It is assumed that all error checking is performed by the VIDS-PC. Consequently, all individual values are assumed to be error-free and that combinations of settings are valid.

4.1.3.2. React_to_Power_State_Changes CSU

React_to_Power_State_Changes determines if the VIDS-GT system should continue to respond to sensor input and activate countermeasures. This is done by checking that both the tank Master_Power and Turret_Power are on. Only when they are both on can VIDS be on.

When VIDS is off, internal data structures used to maintain knowledge of threats and active countermeasures are discarded. Later on in Send_Updates_to_CCDP, the VIDS_Power_State is sent to the CCDP so that a similar cleanup can occur on the VIDS-PC.

4.1.3.3. Identify_Threats CSC

Identify_Threats serves as the primary entry point of sensor simulation. Only when VIDS is on is a test made to determine if there are any new threat reports. For sensors which track threat positions and proximities (NIS and FASR), periodic queries are made to retrieve current threat reports. Each report is sent to Process_Detected_Threat() to determine if the required sensor is available. When the corresponding sensor is available, the threat report is placed into a queue for sensor-specific processing.

Since a threat can be manually deleted at any time, Manual_Threat_Update is invoked to determine if the current CCDP_Control_Settings indicate a

manual threat deletion. When a deletion is indicated, the supplied threat identification is used to search and delete its record from the prioritized threat list.

Finally, each new threat report is processed by the corresponding sensor. The required sensor is sent to EO_Sensor_Simul. EO_Sensor_Simul is a generic function which is invoked to model the behavior of a specified sensor.

4.1.3.3.1. EO_Sensor_Simul CSC

EO_Sensor_Simul serves as the primary entry point for simulation of all sensors. This CSC is subdivided into three functional parts: one which simulates reaction delay, one which handles detection probability, and one which processes new threats as a function of sensor-specific coverage limits. The three functional parts are

```
Update_Delayed_Threats();  
Process_New_Threats();  
Test_Sensor_Coverage_Limits();
```

Each of these functions are described in the following sections.

4.1.3.3.1.1. Update_Delayed_Threats CSU

Each invocation of Update_Delayed_Threats decrements a delay counter associated with each threat in the wait queue. The counter symbolizes the remaining delay time for a given threat. The initial delay time for each threat type is assigned to each threat when it is initially detected in Test_Sensor_Coverage_Limits.

4.1.3.3.1.2. Process_New_Threats CSU

When the counter for a specific threat reaches zero, a detection probability is used by Process_New_Threats to decide if a threat satisfies the probability of detection. A detected threat is moved from the wait queue to the new threats queue. (The new threats queue will be examined later in Fuse_Correlate_Threats). A non detected threat is deleted from the wait queue.

Finally, if the detected threat is a mine, the vehicle brakes are applied immediately .

4.1.3.3.1.3. Test_Sensor_Coverage_Limits CSU

Test_Sensor_Coverage_Limits performs a series of tests to determine if a new threat report is detectable by a specific sensor. The logic of Test_Sensor_Coverage_Limits is constructed to be as generic as possible.

This means that tests for each sensor have been combined into a single set of tests which are applied equally to all threat reports. Differences between individual sensors are handled by a table of sensor behaviors where a given entry contains sensor-specific parameters. Instances where tests are inappropriate for a specific sensor are handled by providing compensating tolerances. For example, the MWS and MFD sensors require a test to determine if the threat is heading towards the tank. For these two sensors, only threats which are approaching the tank within a narrow approach angle are identified as threats. For sensors not requiring an approach angle test, the widest possible approach angle is used ($\pm 180^\circ$) so these reports will not be discarded prematurely.

Threat reports which pass the approach angle test are further tested to determine if the threat falls within the current alert sector, and sensor azimuth and coverage sector angles. (The alert sector is one of the CCDP settings and can be changed at any time, whereas the sensor approach, azimuth and coverage sectors remain constant during the simulation.)

To simplify calculations, the threat position is mathematically transformed into the coordinate system of the tank hull. At this point, the relative threat azimuth and elevation angles are computed. If the threat is heading towards the tank and falls within the alert, sensor azimuth and sensor elevation coverage sectors, it is added to the wait queue with the sensor-specific reaction/delay time. Otherwise, the threat report is discarded.

4.1.3.4. Manage_Countermeasures CSC

Manage_Countermeasures serves as the primary entry point for countermeasure simulation. Countermeasure simulation satisfies the requirement to prioritize threats, select appropriate countermeasures and to activate individual countermeasures for each threat. These activities are accomplished by invoking the following functions:

```
Prioritize_Threats();  
Select_Countermeasures();  
Update_CPS_Coverage();  
Individual_CM_Simul();
```

Each of these functions are described in the following sections. Note that Prioritize_Threats, Select_Countermeasures and Update_CPS_Coverage are invoked only when the VIDS power is on.

4.1.3.4.1. Prioritize_Threats CSC

Prioritize_Threats simulates the behavior of the VIDS Threat Resolution Module (TRM). Its primary purpose is to classify threats based upon the

current sensor reports. Furthermore, `Prioritize_Threats` sorts the threats so countermeasures for the most lethal threats will be activated first. Finally, threats are automatically deleted if no new sensor reports are received within a predefined threat lifetime.

These activities are accomplished by invoking the following functions:

```
Fuse_Correlate_Threats();  
Sort_Prioritized_Threats();  
Update_All_Prioritized_Threats();
```

Each of these functions are described in the following sections. Note that `Sort_Prioritized_Threats` is invoked only when the queue of active threats has changed through an addition, update or deletion.

4.1.3.4.1.1. `Fuse_Correlate_Threats` CSU

The new threats queue built by `Process_New_Threats` is examined by `Fuse_Correlate_Threats` to determine if a new threat report supplies additional information for a known threat. When a new report correlates with previous reports (specifically, the SIMNET vehicle identification values match), the new sensor information is consolidated with the previous information. Sensor detection, energy type categories (some sensors like LWR can detect different types of laser energies) and threat platforms are combined into sets; azimuth, elevation and range values are replaced by the current threat report data. Only when a new report better identifies the threat are guise and vehicle type information updated.

When a new report does not correlate with a known threat, the report is added to the prioritized threat queue as a new threat. Furthermore, an alarm warning is queued. The type of alarm warning corresponds to the detecting sensor.

`Get_Threat_Classification` is invoked for both updated and new threats to return a threat classification. The threat classification is then used by `Get_Threat_Priority` to return a threat priority.

4.1.3.4.1.2. `Get_Threat_Classification` CSU

`Get_Threat_Classification` uses what is currently known about a threat to assign a classification. Threats detected by the NCS or Mine sensor have straightforward classification assignments. However, for the remaining sensors, classifications are based upon moderately complex combinations of sensor reports. Consequently, the following PDL succinctly summarizes the classification strategy:

VIDS SDD

```

if (Sensor_Reports.MINE)
    return Class_Mine;
end if
if (Sensor_Reports.NCS)
    return Class_Chemical;
end if

if (Sensor_Reports.MFD)
    Class = Class_Muzzle;
    if (Sensor_Reports.LWR && Category.LRF)
        return Class_Muzzle_w_LRF;
    end if

    if (Available_CM.TCS)
        return Class_Muzzle_w_TCS;
    end if
end if

if (Sensor_Reports.FASR)
    if (Platform.Infantry_Support)
        Class = Class_Infantry_Support;
    else if (Platform.Tank)
        Class = Class_Tank;
    else if (Platform.Helicopter)
        Class = Class_Helicopter;
    end if

    if (Sensor_Reports.NIS)
        Class = Class_Helicopter;

    if (Sensor_Reports.TRWR)
        if (Category.RADAR_Uplink)
            if (Platform.Helicopter)
                Class = Class_3; /* AT-6 */
            else
                Class = Class_1; /* AT-2C */
            end if
            if (Platform.Unknown)
                Class = Class_6; /* AT-2C or AT-6 */
            end if
            if (Platform.Infantry_Support)
                Class = Class_Infantry_Support;
            end if
        end if
    end if

    if (Sensor_Reports.MWS)
        Class = Class_9; /* Any ATGM */

        if (!Available_Sensors.LWR & !Sensor_Reports.TRWR)
            Class = Class_7; /* AT-4, AT-9, or AT-11 */
        end if
        if (!Sensor_Reports.LWR)
            if (!Available_Sensors.TRWR)
                Class = Class_8; /* AT-2C, AT-4, or AT-6 */
            else if (!Sensor_Reports.TRWR)
                Class = Class_2; /* AT-4 */
            end if
        end if
    end if

    if (Sensor_Reports.LWR)
        if (Category.LDES)
            Class = Class_4; /* AT-9 */
        else if (Category.LBR)
            Class = Class_5; /* AT-11 */
        else if (Category.LRF)
            Class = Class_10; /* general threat */
        end if
    end if

    return Class;

```

4.1.3.4.1.3. Get_Threat_Priority CSU

Get_Threat_Priority is a simple function which uses the supplied threat class as index into a table of priorities. The priority for a given threat class is contained in VIDS.D. A copy of VIDS.D is included as Appendix A.

4.1.3.4.1.4. Sort_Prioritized_Threats CSU

Sort_Prioritized_Threats visits each threat in the prioritized threat queue to verify each threat is positioned correctly within the queue. Threats which have activated a countermeasure are placed lower in the queue than threats which have not. A threat which is inside the safety sector is lower in priority than one which is outside. When two threats have equal priority, the one which is closest to the main gun has higher priority. When two threats are equal in angular proximity from the main gun, the one which will be reached first with a clockwise turret rotation has higher priority.

4.1.3.4.1.5. Update_All_Prioritized_Threats CSU

Update_All_Prioritized_Threats visits each threat in the prioritized threat queue to decrement its lifetime. When the lifetime for a threat reaches zero, it is removed. Since this changes the status of the prioritized threat queue, a new prioritized threat message is sent to the CCDP so that the corresponding threat icon will be removed.

4.1.3.4.2. Select_Countermeasures CSU

Select_Countermeasures assigns countermeasures to new threats and reconfirms that the current countermeasure for an existing threat is correct. This is done by visiting each threat in the prioritized threat list and determining if it has been assigned a countermeasure. When a countermeasure has not been assigned, a table lookup is used to find the first available countermeasure. When a countermeasure has been assigned, a table lookup is still performed to confirm that the same countermeasure is still recommended. This is done because the type of threat may have changed due to sensor fusion or because an expendable countermeasure is no longer available. If the recommended countermeasure has changed, the new recommended countermeasure will be activated even if the previous countermeasure has been activated.

Once each threat has been assigned a countermeasure, a check is made to determine if there has been a manual change in the order of countermeasure activation. If the CCDP settings indicate a change, the corresponding countermeasure will be activated first in Individual_CM_Simul.

4.1.3.4.3. Update_CPS_Coverage CSU

Update_CPS_Coverage converts the CPS azimuth coverage sector from the current CCDP control settings and replaces the current CPS azimuth coverage values. The changed azimuth coverage values will be retrieved the next time CPS is activated as a countermeasure.

4.1.3.4.4. Individual_CM_Simul CSU

Individual_CM_Simul controls the activation and deactivation of countermeasures. In general, individual countermeasures are activated and deactivated simultaneously until all threats have been handled. Only when an individual countermeasure is needed to defeat multiple threats is the countermeasure activated sequentially. Furthermore, Individual_CM_Simul supports automatic modes for counterfire rotation and turret slewing.

Countermeasure activation, counterfire rotation and turret slewing can all be activated automatically or semi-automatically. (Semi-automatic activation is equivalent to automatic activation when the commanders thumb switch is engaged.) Countermeasures can be activated manually using buttons on the CCDP, but manual counterfire rotation and turret slewing is still controlled by either the tank commander or gunner controls. Note, however, that all countermeasure activations require arming. A button on the CCDP arms all countermeasures.

Manual countermeasure activation occurs when countermeasures are armed and a countermeasure button is depressed (back lighted) on the CCDP. Electro-optical countermeasure energy is transmitted endlessly until either the corresponding button is released or countermeasures are made safe (disarmed). Manual ROS activation launches a salvo of grenades within the CM Coverage sector. Manual chaff or flares activation launches all expendables at once. Furthermore, manual jamming or a salvo of smoke grenades, chaff or flares can occur concurrently with any mode of turret slewing.

Automatic countermeasure activation occurs when all of the following conditions exist:

- a. the CM mode is automatic or semi-automatic.
- b. the recommended countermeasure for a threat is available.
- c. countermeasures are armed.
- d. the commanders thumb switch is engaged (necessary only for semi-automatic activation).

Furthermore, if a countermeasure requires turret rotation prior to activation, automatic modes for both counterfire rotation or turret slewing are temporarily suspended.

VIDS SDD

Automatic rotation for counterfire will occur when the following conditions exist:

- a. the CFIRE mode is automatic or semi-automatic.
- b. automatic countermeasure activation has not seized control of the turret.
- c. the commanders thumb switch is engaged (necessary only for semi-automatic activation).

Automatic turret rotation temporarily suspends automatic turret slewing.

Automatic turret slewing will occur when the following conditions exist:

- a. the SCAN mode is automatic or semi-automatic.
- b. automatic countermeasure or counterfire rotation has not seized control of the turret.
- c. the commanders thumb switch is engaged (necessary only for semi-automatic activation).

The following countermeasures are simulated: ROS, MCD, CPS, ATRJ, LCMD, VEMASID, TCS, NBCOP, Chaff/Flares. Furthermore, the radar search energy of the FASR sensor and the forward-looking infrared (FLIR) energy of the MINE sensor are simulated here. However, to simplify their design, countermeasures have been simulated by generic CSUs. Those countermeasures which have similar operating characteristics have been grouped together. The modules and the countermeasures they simulate follow:

- a. ROS_Simul: ROS.
- b. EO_CM_Simul: MCD, CPS, ATRJ, LCMD, VEMASID, the radar search energy of the FASR sensor, and the FLIR energy of the MINE sensor.
- c. TCS_Simul: TCS.
- d. Decon_CM_Simul: NBCOP.
- e. One_Shot_CM_Simul: Chaff and Flares.

Each of these modules are described in the following sections.

4.1.3.4.4.1. ROS_Simul CSU

ROS_Simul serves as the primary entry point for simulation of the Rapid Obscuration System (ROS). This system launches smoke grenades to temporarily hide the tank position from electro-optically guided or terminal homing missile threats.

For simulation, the turret is conceptually divided into 24 equal sectors each with 15 degrees of coverage. Each sector may contain zero or more grenades; and there may be more than one smoke grenade type for individual sectors.

For manual activation, the number and sectors are specified by the CCDP countermeasure coverage sector. Smoke grenades are launched a predefined distance from the tank hull.

For automatic activation, launch sectors are selected dynamically. Launch sectors are selected which require the minimum turret rotations to get the recommended smoke grenades between the threat and the tank hull. Once the turret has rotated a launch sector into position, one or more grenades are launched from adjacent sector positions. Note that if turret rotation is required by ROS, gunner and commander turret controls are disabled; and automatic rotation for counterfire or turret slewing is temporarily suspended.

As grenades are launched, the inventory of available smoke grenades is decremented. Once all the recommended grenades have been launched, the prioritized threat record is updated so that additional smoke grenades will not be launched towards the same threat; gunner and commander turret controls are reenabled; and automatic rotation for counterfire or turret slewing is resumed.

4.1.3.4.4.2. EO_CM_Simul CSU

EO_CM_Simul serves as the primary entry point for simulation of the MCD, CPS, ATRJ, LCMD, VEMASID, the radar search energy of the FASR sensor and the FLIR energy of the MINE sensor. This module simulates systems which direct electro-optical energy towards a threat.

Because of independent slewing, the center of the electro-optical energy is in one of two directions. In manual mode, the countermeasure slews to be coincident with the direction of the main gun; in an automatic mode, the countermeasure slews to angle of the detected threat.

Electro-optical begins when the countermeasure device arrives at the required azimuth angle and when the reaction/delay time has expired. For manual activation, the jamming energy continues until it is manually deactivated. For automatic activation, the electro-optical continues until the predefined jamming time expires. Furthermore, the prioritized threat record is updated so that jamming energy will not be automatically directed against the same threat.

The radar search energy from the FASR sensor begins only when the corresponding CCDP button is depressed and after the reaction/delay time has expired. The center of radar search energy is always aligned with the main gun. Releasing the FASR button immediately terminates the radar search energy.

The FLIR energy from the MINE sensor begins only when this sensor is available and when the reaction/delay time has expired. Since the MINE sensor is not manually activated/deactivated, the FLIR energy continues as long as the tank is operational.

4.1.3.4.4.3. TCS_Simul CSU

The function TCS_is_Not_Effective serves as the main entry point for the TCS simulation. It is invoked by failure_check_cat_kill (found in the m1_failure module) when it is necessary to compute damages from the impact of a missile or main gun round. If TCS is not available as a countermeasure, damage calculations are computed normally. However, when TCS is available a series of tests are performed to determine if the impact should be ignored.

The first set of tests determine if TCS is in a ready mode. TCS is in a ready mode if the following conditions are satisfied:

- a. countermeasures are armed, and
- b. the CM mode is automatic, or
the CM mode is semi-automatic and the commanders thumb switch is engaged, or
- c. the CM mode is manual the TCS button is depressed.

If the TCS is in the ready mode, the final set of tests are performed. Conceptually, the hull is divided into quadrants: front right, front left, rear left, rear right. The direction of the velocity vector of the projectile with respect to the hull orientation is used to derive which quadrant's inventory must be examined. If the quadrant's inventory is not depleted, the inventory is reduced by one and TCS has saved the tank crew from a catastrophic kill.

4.1.3.4.4.4. Decon_CM_Simul CSU

Decon_CM_Simul and Poison_Simul serve as the primary entry points for simulating the NBCOP. Identify_Threats registers when a poison is present regardless of the availability of NCS by invoking Contaminant_Exists. When poison is present, Poison_Simul decrements a life-expectancy timer. A catastrophic kill is initiated when the timer expires.

Decontamination begins when the following conditions are satisfied:

- a. countermeasures are armed and,
- b. the CM mode is automatic, or
the CM mode is semi-automatic and the commanders thumb switch is engaged, or
- c. the CM mode is manual and the NBCOP button is depressed.

For manual activation, the decontamination continues until it is manually deactivated. For automatic activation, the jamming continues until the predefined jamming time expires.

4.1.3.4.4.5. One_Shot_CM_Simul CSU

One_Shot_CM_Simul serves as the primary entry point for Chaff and Flares countermeasure simulation. This module launches chaff or flares to decoy radar directed or infra-red guided/directed missiles and munitions.

Because of independent slewing, the final azimuth launch angle is in one of two directions. In manual mode, the countermeasure slews to be coincident with the direction of the main gun; in an automatic mode, the countermeasure slews to azimuth angle of the detected threat.

Chaff or flares are launched as soon as the countermeasure device arrives at its prescribed angle and when the reaction delay time has expired. For automatic activation, the prioritized threat record is updated so that other countermeasures will not be automatically directed against the same threat. Furthermore, once chaff or flares are launched, the entire chaff or flares inventory is considered to be depleted.

4.1.3.4.4.6. CFire_Simul CSU

CFire_Simul serves as the main entry point of automatic turret slewing for counterfire. This module rotates the turret to aim the main gun in the direction of a threat.

As automatic turret rotation begins the gunner and commander turret controls are disabled. Once the main gun is positioned to the threat azimuth angle, gunner and commander turret controls are reenabled.

As a final note, the main gun is never fired automatically.

4.1.3.4.4.7. Turret_Scanning_Simul CSU

Turret_Scanning_Simul serves as the main entry point of automatic turret slewing. This module rotates the turret within the turret scanning sector.

As automatic turret rotation begins the gunner and commander turret controls are disabled. Only when automatic turret scanning is disabled are the gunner and commander turret controls reenabled.

4.1.3.5. Send_Updates_to_CCDP CSC

Send_Updates_to_CCDP serves as the primary communication channel for sending information updates from the VIDS-GT to the VIDS-PC. The following types of information are sent:

- a. changes to the top ten threats.
- b. changes to hull or turret orientations.
- c. changes to master or turret power states.
- d. audible alerts for new or changed threats.
- e. changes in expendable countermeasure (ROS, TCS, Chaff/Flares) inventory.

Services provided by existing code are used to package and transmit the messages to the corresponding VIDS-PC.

4.1.3.6. Send_Updates_to_Network CSC

Send_Updates_to_Network serves as the primary communication channel for sending the state of the VIDS-equipped tank to other entities participating within the same simulated battle exercise. The following types of informational messages are sent:

- a. the presence of smoke, chaff or flares.
- b. the activation/deactivation of electro-optical energy.
- c. instrumentation (used only for data collection and analysis).

Only instrumentation messages are sent at regular intervals. However, an instrumentation message will be sent sooner if one of the following conditions exist:

- a. the state of the Master or Turret Power changes.
- b. the state of the commander's thumb switch changes.
- c. the number of threats exceeds the maximum number of displayable icons on the CCDP.
- d. turret control is seized or released by VIDS.

Services provided by existing code are used to package and transmit the information to other entities.

4.1.3.7. Need_To_Release_Turret CSU

Need_To_Release_Turret provides a safeguard to release turret control when no threats are present and CM, CFIRE and SCAN modes are in manual.

4.1.3.8. CM_Rotation_Simul CSC

CM_Rotation_Simul supports the need for independently slewing countermeasures. Furthermore, it manages conflicting rotation requests using priorities.

Each independently slewing countermeasure corresponds to an entry in the countermeasure device table. For each invocation of CM_Rotation_Simul, individual countermeasure devices are rotated to the next position by adding the device rotation rate to the current position. The rotation rate can be positive or negative to move a device in a specific angular direction. Because all simulated devices (except VEMASID) are mounted to the turret, hull and turret rotations can accelerate or retard the time required to move a device to a specified angle. However, once the device has arrived, it remains perfectly positioned until a higher priority request is received.

Conflicting requests to rotate a specific countermeasure are handled using priorities. The priority of each countermeasure rotation request is compared to the previous rotation request. Lower priority requests are ignored. A request with equal or higher priority than the current priority takes control of the countermeasure device. The new priority replaces the current priority.

4.1.3.9. Turret_Rotation CSC

Start_Turret_Rotation and Stop_Turret_Rotation serve as the main entry points for automatic turret rotations required by ROS_Simul, CFire_Simul and Turret_Scanning_Simul. Like CM_Rotation_Simul, rotation requests are based upon a priority.

Conflicting requests to rotate the turret or stop the current rotation are handled using priorities. The priority of each turret request is compared to the previous request. Lower priority requests are ignored. A request with greater than or equal to the current priority takes control of the turret. The new priority replaces the current priority.

Start_Turret_Rotation is invoked to initiate automatic turret rotation. The minimum rotation direction is computed using the current and final turret positions when the request satisfies the priority test. The direction is used to compute the sign of the turret rotation increment. It is this increment which is used to modify the current turret position during each execution cycle.

Stop_Turret_Rotation is invoked to terminate automatic turret rotation. Only when the request satisfies the priority test is the automatic turret rotation terminated.

4.1.3.10. VIDS_Shutdown CSC

VIDS SDD

The VIDS_Shutdown CSC handles the job of shutting down the VIDS simulation. The local Master and Turret Power states are cleared and the final power state message is sent to the CCDP. This has the net effect of forcing the CCDP to redisplay the initial (power off) menu.

Each internal queue is examined and each member is formally deallocated. Once this is complete, memory statistics are written to the main console. The statistics can indicate if there is a memory management problem or if additional memory should be preallocated during the earlier execution of VIDS_Init.

4.1.4. XField CSC

XField handles the low-level simulation of electro-optical energy. XField PDUs retrieved from the network are examined to determine the kind and spatial extent of electro-optical energy. If the VIDS-equipped tank falls within the energy field, the information describing the field is added to an internal list of other fields. Additionally, the presence of clouds (smoke) is used to determine if the field energy is absorbed. If field energy is absorbed, the field is not made available to the higher-level Identify_Threats CSC.

Fields are removed from the list when either an explicit XField PDU defines that the field no longer exists or the specified field lifetime expires.

An XField PDU sent by the VIDS-equipped tank (refer to the EO_Sensor_Simul and One_Shot_CM_Simul CSUs) is tagged appropriately to distinguish it from fields sent by other vehicles. Furthermore this type of field is periodically retransmitted to the network as long as the field is present.

4.1.5. Cloud CSC

Cloud handles the low-level simulation of electro-optic energy absorbing smoke clouds. Smoke Cloud PDUs are initially transmitted to the network by VIDS-equipped tanks (refer to the ROS_Simul CSU) to inform other vehicles that new smoke clouds exist.

Each smoke grenade is simulated as a single cloud. Parameters are supplied which define the smoke type and corresponding spatial dynamics. This allows other vehicles to model the smoke growth, dissipation and interference with electro-optical energy.

Like XFields PDUs, Cloud PDUs are periodically retransmitted to the network as long as the smoke is potentially effective as an obscurant. When the smoke from a grenade is no longer effective, a Cloud PDU is transmitted to the network so that other vehicles can drop it from their internal lists.

4.1.6. Modifications to Existing Code

Modifications to existing code were made to support the VIDS-GT capability. The files and changes follow:

- a. `m1_main.c`
 - Added invocation of `VIDS_Init` in the `veh_spec_init` function.
 - Added invocation of `VIDS_File_Read` in the `veh_spec_startup`.
 - Added invocation of `VIDS_Simul` in the `veh_spec_simulate`.
 - Added `VIDS_Shutdown` to the `veh_spec_exit` to send a `TankPowerStateVariantmsg` to the CCDP to turn it off.
 - Added a test to determine if the engine was running before invoking `VIDS_Simul()` to support the NIS-unique detection and identification probabilities.
 - Made changes to access new `idc_value` array element which holds the state of the new commander's thumb switch.
 - Inserted `field_tick` and `cloud_tick` before `VIDS_Simul`.
- b. `m1_turret.c`
 - Added the `set_vids_az` function to support automatic turret rotations to specific azimuth angles.
 - Added the `set_vids_relative` function to support final turret angles relative to the tank hull.
 - Added the `set_vids_north` function to support final turret angles relative to true north.
 - Added the `set_vids_auto_on` function to disable the gunner and commander turret rotation controls.
 - Added the `set_vids_auto_off` function to enable the gunner and commander turret rotation controls.
 - Added the `set_vids_slew_rate` function to support the specification of a rotation rate.
 - Added the `get_vids_rate` function to retrieve the current, VIDS-specific turret rotation parameters.
- c. `proc_a_pkt.c`
 - Added code to recognize and reformat VIDS (CCDP Control Settings) PDUs.
 - Added code to recognize ROS, TCS and Chaff/Flares munition resupply PDUs so the initial supply and placement of expendable countermeasures could be instantly restored.
- d. `m1_ctl_tpc.c`
 - Made changes to support the use of a new commander's thumb switch which will be used instead to activate semi-automatic CMs.
- e. `m1_failure.c`
 - Added the `TCS_is_Not_Effective` function to intercept a fatal impact from a missile or projectile.
- f. `m1_resupp.c`

Added code to recognize the ROS, TCS and Chaff/Flares variants of the munition resupply PDU and invoke the corresponding CSC resupply functions.

g. `read_pars.c`:

Added code to recognize and store the name of the VIDS parameter file.

Added `get_vids_data_file` to return the name of the VIDS parameter file.

4.2. VIDS-PC Detailed Design

4.2.1. SMI_Init CSC

`SMI_Init` preallocates dynamic memory structures associated with drawing menus and icons which will be displayed during the execution of the `SMI_Simul` CSC. Data files are read which define the placement and appearance of buttons and icons as well as the unique identifier (VehicleID) of the corresponding GT. Parameters are read which define active buttons and how long they must be pressed for a corresponding action to be activated. VIDS (Default Sector Setups and Expendable CM Inventory) PDUs sent by the corresponding GT supplies initial values for the alert, countermeasure, safety, scanning and CPS coverage sectors, the list of available sensors and countermeasures and the inventory of the expendable countermeasures. The list of available countermeasures is used assign spare buttons to individual countermeasures. Finally, links are established between buttons and function invocations.

4.2.2. SMI_Simul CSC

`SMI_Simul` serves as the primary entry point for simulation of the real CCDP. It represents the root of a functional hierarchy which is executed endlessly until a keyboard Control-C or right mouse button event is received. During a single execution cycle, the following high-level functions are executed:

```
Get_Button();  
Check_Alarms();  
Process_Rcv_PDU();
```

Each of these functions represent a functional sub-hierarchy which is described in the following sections.

4.2.2.1. Get_Button CSU

`Get_Button` serves the need to monitor button, mouse, and keyboard activity. A right mouse button or keyboard Control-C signals a request to terminate

the SMI_Simul CSC by transitioning to SMI_Shutdown. Otherwise, a test is made to determine if a displayed button has been held down. If a button has been held down long enough and it corresponds to a predefined action, the action is initiated through a corresponding function call. The corresponding function may change the current menu, the content of the display, the operating state or a combination of these changes. When a button changes one of the VIDS operating states, a network message is sent to the VIDS-GT to update its corresponding CCDP settings. Additionally, when any button state is changed or when a user alert message is displayed, an instrumentation message is sent to the network for data collection and analysis.

4.2.2.2. Check_Alarms CSU

Check_Alarms manages the VIDS alarm tones heard on the tank intercom. The status of each alarm type is checked to determine if it should be activated or terminated. When an alarm is activated, the alarm is heard for a predefined duration. An alarm is terminated when the duration has expired, a termination message was received from the GT or VIDS is powered off.

4.2.2.3. Process_Rcv_PDU CSC

Process_Rcv_PDU manages received network messages. Only messages sent by the corresponding VIDS-GT are processed. All other messages are discarded.

Depending upon the type of message, the display or alarm tones are changed. The message types which change the display include the following:

- a. Tank Power State updates.
- b. Tank Orientation updates.
- c. Prioritized Threat updates.
- d. Automatic CM Activation/Deactivation updates.
- e. Default Setups.
- f. Changes to the inventory of expendable countermeasures.

Only the Alarm Control message type affects what is heard on the tank intercom. Refer to the IDD in section 5 for the exact content of each message type.

4.2.3. SMI_Shutdown CSU

SMI_Shutdown releases memory allocated during SMI_Init and restores the mouse and display behaviors before terminating.

5. CSCI data.

Within the VIDS-GT CSC, there is only one global data element: `vids_debug`. It is a Boolean object which is toggled between two states to either activate or deactivate diagnostic messages. Under normal conditions, `vids_debug` is false.

Within the VIDS-PC CSC, the following arrays represent global elements:

- a. Icon.
- b. Threat.
- c. Frame.
- d. Display.
- e. Vary.
- f. Buttons.
- g. `fcnptrs`.

These arrays used to support low-level drawing operations. Refer to the following header files for more details:

`global.h`
`alarm.h`
`buttons.h`

The following table lists the type and content of the messages exchanged between the PC, GT and SAF. Additionally, the content of the GT and PC instrumentation messages are included.

CCDP_Control_Settings (sent from CCDP to M1)	Alert_Sector	Start_Angle	0 to 360 degrees.	Defines the start angle of the alert sector. The start angle is defined relative to the front of the hull. Positive angles are counterclockwise. Threats which fall outside alert sector will be ignored.
		Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divisible by 15 degrees. If Delta equals zero, threats will not be reported.
	Safety_Sector	Start_Angle	0 to 360 degrees.	Defines the start angle of the safety sector. The start angle is defined relative to the front of the hull. Positive angles are counterclockwise. Neither the turret nor any CM will be activated within the Safety Sector.
				Furthermore, the turret will not be automatically slewed into this sector for either Semi or Auto CFire.
		Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divisible by 15 degrees.
				If Delta equals 0, then no safety region is defined. In other words, countermeasures can be activated/deployed in any direction.
	Turret_Scanning_Sector	Start_Angle	0 to 360 degrees.	Defines the start angle of the turret scanning sector. The start angle is defined relative to the front of the hull. Positive angles are counterclockwise.
		Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divisible by 15 degrees.
				If Delta equals 0, then turret scanning is disabled.
	CM_Coverage_Sector	Start_Angle	0 to 360 degrees.	Defines the start angle of the CM coverage sector. The start angle is defined relative to the front of the hull. Positive angles are counterclockwise.
		Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divisible by 15 degrees.

Field	Symbol	Range	Units	Soft
CPS_Coverage_Sector	Start_Angle	0 to 360 degrees.		If Delta equals 0, then no smoke grenades will be fired by a Manual Salvo. A sector defined by Delta which is greater than 0 defines the coverage of smoke grenades when the Salvo button is depressed in Manual mode. Defines the start angle of the CPS coverage sector. The angle is defined to be relative to the main gun. Positive angles are counterclockwise.
Manual_Grenade_Salvo	Delta	5 to 120 degrees.		Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divisible by 5 degrees.
VIDS_Power_State		Off, On		Defines an array which stores the number of smoke grenades to launch for each type of grenade. Smoke grenade type (LBA1, M76, XM81) is used to index into the array. Each array element contains the number of grenades to launch.
Turret_Mode		Manual, Semi, Auto		Defines whether VIDS is on or off. When VIDS is off, only the Salvo and Arm_Safe Buttons operate.
CFire_Mode		Manual, Semi, Auto		Defines the turret scanning mode. The turret mode will return to Manual if manually deactivated, or if a threat is detected and CFire_Mode or CM_Mode is Semi or Auto.
CM_Mode		Manual, Semi, Auto		Defines the counterfire mode. Semi or Auto provide for automatic turret slewing and positioning.
Arm_Safe_State		Safe, Armed		Defines the countermeasure mode. Semi and Auto provide for automatic turret slewing and automatic CM activation.
ROS_Button_State		Deactivated, Activated		Defines the Arm/Safe button state. The state must be Armed before any countermeasure is activated. This state is ignored for all CFire and Turret Scanning modes.
				Defines the ROS countermeasure state. The ROS countermeasure may be activated manually or automatically. Manually deactivating the ROS countermeasure immediately terminates either type of activation. Otherwise, deactivation occurs after the recommended number of smoke grenades have been launched.
MCD_Button_State		Deactivated, Activated		Defines the MCD countermeasure state. The MCD countermeasure may be activated manually or automatically. Manually deactivating the MCD countermeasure immediately terminates either type of activation. Manual activation always requires manual deactivation; automatic deactivation follows automatic activation after a predefined delay.

Structure	Fields	Sub-Fields	Range of Values	Descriptions
	.Flares_Button_State		Deactivated, Activated	Defines the Flares countermeasure state. The Flares countermeasure may be activated manually or automatically. Manually deactivating the Flares countermeasure immediately terminates either type of activation. Otherwise, deactivation occurs after the entire inventory of flares has been launched.
	.LCMD_Button_State		Deactivated, Activated	Defines the LCMD countermeasure state. The LCMD countermeasure may be activated manually or automatically. Manually deactivating the LCMD countermeasure immediately terminates either type of activation. Manual activation always requires manual deactivation; automatic deactivation follows automatic activation after a predefined delay.
	.Delete_Threat		SIMNET Vehicle Id	Defines which target to be deleted. A value of 0 means that no manual deletions are required.
	.Selected_Top_Threat		SIMNET Vehicle Id	Defines which target is the top priority. A value of 0 means that no change is required.
	.Threat_Sensor_Filter		LBR, LDES, LRF, ATGM, HELO, TANK	Defines an array of bits which is indexed by LBR, LDES, LRM, ATGM, HELO, TANK. When a bit is set to 1, sensed threats of the specified type will be ignored.
	.Gulise_Filter		UNKNOWN, FRIEND, FOE	Defines an array of bits which is indexed UNKNOWN, FRIEND, FOE. When a bit is set to 1, identified threats of the specified type will be ignored.
	.Auto_Activated_CMIs		ROS, MCD, CPS, VEMASID, ATRJ, TCS, MFD, NBC, Chaff_Flares, LCMD	The bit map is indexed by CM type. Each array element defines which CM can be automatically activated.
Tank_Power_State (sent from M1 to CCDP)				
	.Master_Power_State		Off, On	Defines the Master Power State. When this is Off, no CCDP button can be "depressed", all lights go out, the display is blanked, and any audible warning tone is terminated.
	.Turret_Power_State		Off, On	Defines the Turret Power State. When this is Off, no CCDP button can be "depressed", all lights go out, the display is blanked, and any audible warning tone is terminated.

Da... Structure Tank_Orientation (sent from M1 to CCDP)	Fields	Subfields	Range of Values	Descriptions
	.Hull_Orientation		0 to 360 degrees.	Defines the angle of the hull with respect to true North. Positive angles are counterclockwise.
	.Turret_Orientation		0 to 360 degrees	Defines the angle of the turret with respect to the hull. Positive angles are counterclockwise.
Expendable_CM_Inventory (sent from M1 to CCDP)				
	.ROS_Launch_Tubes		[3,24]	Defines an array of smoke grenade totals. The first array element is indexed by LBA1, M76, XM81. Each array element contains the remaining grenades of the specified type in the 24 launch tubes. The second array element corresponds to the launch tubes. For simulation, each launch tube is positioned at 15 degree increments with an initial offset of 7.5 degrees from the main gun.
				Note, each tube can launch one of three types of grenades: LBA1, M76, XM81. Grenades can be launched in any order from a simulated tube.
	.Remaining_ROS_Grenades		[3]	Provides a subtotal of remaining grenade types.
	.ROS_Grenade_Angles		[24]	Defines the grenade launch angles. These are used to display smoke grenade icons on the CCDP.
	.Fired_ROS_Grenade_Count		0,24	Defines the number of grenade angles and ultimately the number of smoke grenade icons to display on the CCDP.
	.TCS_Launch_Tubes		[4]	Defines an array of 4 launch tubes where each tube holds 2 expendable TCS CMs.
	.Remaining_TCS			Provides the remaining inventory of TCS.
	.Remaining_Chaff		0,30	Defines the remaining inventory of chaff.
	.Remaining_Flares		0,30	Defines the remaining inventory of decoy flares.
Alarm_Control (sent from M1 to CCDP)				
	.Alarm_Index		1-255	Defines the index into a prerecorded table of alarm tones.
	.Alarm_Activation		Off, On	Defines when the tone is played or terminated.

Structure	Fields	SubFields	Range of Values	Descriptions
	.Alarm_Duration		0 - 60	Defines the number of seconds that a warning tone must be heard.
Prioritized_Threats (sent from M1 to CCDP)				
	.Total_Threat_Count		0-65535	Defines the number of recognized threats. Note that only the first 10 threats will be sent to the CCDP, and that the threats will be sent in priority order.
	.Threat_List	.Threat_Type	LRF, LBR, LDES, ATGM, MINE, HCPTR, TANK, SUPPVEH, NBC, UPLINK, SEARCH, TRACK, AH64, OH590, HAVOC, HIND, HIP	Identifies the type of threat.
		.Vehicle_Id	SIMNET Vehicle Id	Uniquely defines the threat.
		.Azimuth_Angle	0 to 360 degrees	Defines the angle of the threat with respect to true North. Positive angles are counterclockwise.
		.Elevation_Angle	90 to -90 degrees	Defines the angle of the threat with respect to the horizon. Positive angles are above the horizon.
		.Range	0 - 10km	A subset of the sensors report a threat distance.
		.Guise	Friendly, Foe, Unknown	Defines if the threat is friendly. For the present all threats are foes.
		.Recommended_CM	NULL, MCD, ROS, CPS, CMINE, ATRJ, TCS, NBCOP, CHAFF, F_ARES, LOMD	Defines the recommended CM for automatic activation.
		.ROS (variant)	.Grenade_Type_Array	Defines an array of smoke grenade types. For ROS, the three types are L8A1, M76, XM81. Each array element defines the number of smoke grenades to launch for a given threat.
Auto_CM_State_Change (sent from M1 to CCDP)	.CM_States_Array		ROS, MCD, CPS, VEMASID, ATRJ, TCS, MFD, NBC, Chaff_Flares, LOMD	The array is indexed by CM type. Each array element defines which button/display to light/extinguish when a countermeasure is automatically activated/deactivated.
		.State	Deactivated, Standby, Activated	Defines the automatic activation state of a specific CM. Standby indicates the reaction/delay time period prior to actual activation.
Default_CCDP_Setups (sent from M1 to CCDP)				
	.Alert_Sector	.Start_Angle	0 to 360 degrees.	Defines the start angle of the alert sector. The start angle is defined relative to the front of the hull. Positive angles are counterclockwise. Threats which fall outside alert sector will be ignored.
		.Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divisible by 15 degrees.
	.Safety_Sector	.Start_Angle	0 to 360 degrees.	Defines the start angle of the safety sector. The start angle is defined relative to the front of the hull. Positive angles are counterclockwise. Neither the turret nor any CM will be activated within the Safety Sector.

Structure	Fields	Sub-fields	Range of Values	Description
		.Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divided. If Delta equals 0, then no safety region is defined. In other words, countermeasures can be activated/deployed in any direction.
	.Turret_Scanning_Sector	.Start_Angle	0 to 360 degrees.	Defines the start angle of the turret scanning sector. The start angle is defined to be relative to the front of the hull. Positive angles are counterclockwise.
		.Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divided by 15 degrees.
				If Delta equals 0, then turret scanning is disabled.
	.CM_Coverage_Sector	.Start_Angle	0 to 360 degrees.	Defines the start angle of the CM coverage sector. The start angle is defined relative to the front of the hull. Positive angles are counterclockwise.
		.Delta	0 to 360 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divided by 15 degrees.
				If Delta equals 0, then no smoke grenades will be fired by a Manual Salvo. A sector defined by Delta which is greater than 0 defines the coverage of smoke grenades.
	.CPS_Coverage_Sector	.Start_Angle	0 to 360 degrees.	Defines the start angle of the CPS coverage sector. The angle is defined to be relative to the main gun. Positive angles are counterclockwise.
		.Delta	5 to 120 degrees.	Defines the angular offset from the start angle. The sector defined by the delta from the start angle is evenly divided by 5 degrees.
	.Manual_Grenade_Salvo		[3]	Defines an array which stores the number of smoke grenades to launch for each type of grenade. Smoke grenade type (LBA1, M76, XM81) is used to index into the array. Each array element contains the number of grenades to launch.
	.Available_CMs		ROS, MCD, CPS, VEMASID, ATRJ, TCS, MFD, NBC, Chaff_Flares, LCMD	Defines a bit map indexed by CM type. Each array element defines which CM is available for either manual or automatic activation.
	.Auto_Activated_CMs		ROS, MCD, CPS, VEMASID, ATRJ, TCS, MFD, NBC, Chaff_Flares, LCMD	Defines a bit map indexed by CM type. Each array element defines which CM can be automatically activated.
	.Available_Sensors		LWTJ, MWS, MFD, SEESAW, NIS, NCS, ITWRI, FASR	Defines a bit map indexed by Sensor type. Each array element defines which Sensor is available for manual activation. Presently, this is used only for FASR.

	Max_Threats			Defines the maximum number of threats to be displayed on the CCUP.
XField (exchanged between SAF and VIDS-equipped vehicles)				
.field_ID			0 to 65535	Uniquely identifies the field.
.target_ID			SIMNET Vehicle Id	Uniquely defines the target.
.based_on			Position_Based_On_Vehicle_Hull, Rotation_Based_On_Vehicle_Hull, Rotation_Based_On_Vehicle_Turret, Rotation_Based_On_Vehicle_Gun, Rotation_Based_On_VIDS, Range_Finder_Laser, Beam_Rider_Laser, Designating_Laser, Radar_Jam, Acoustic, IR, MF, ATGM, RADAR_Search, RADAR_Track, RADAR_Uplink, CPS_Laser, Laser_Target_Decoy, Mine_FLIR, Mine_Magnetic, Real_Mine_Illumed, False_Mine_Illumed, FASR_Search, FASR_Illumed, BCIS, Transponder, Nuclear, Biological, Chemical, Flares, Chaff.	Defines the type of XField.
.subtype				
.type			Unknown, Sector, Omnidirectional, Beam	Defines the XField directionality.
.location			SIMNET WorldCoordinates	Defines the location of the XField.
.rotation				Defines the rotation of the XField. When Rotation_Based_On_VIDS, this matrix defines the vehicle-independent (stabilized) azimuth angle of the XField.
.velocity			9 element rotation matrix	Defines the XField velocity.
.age			SIMNET VelocityVector	Defines the XField age.
.exp_duration			0 to the largest unsigned integer.	Expected duration (lifetime) of the field.
.theta_1			0 to the largest unsigned integer.	One of two azimuthal angles which define the field.
.theta_2			0 to 360 degrees.	One of two azimuthal angles which define the field.
.phi_1			0 to 180 degrees.	One of two altitudinal angles which define the field.
.phi_2			0 to 180 degrees.	One of two altitudinal angles which define the field.
.power			0.0 to maximum floating point number.	Defines the total power emitted by the field.
.frequency			0.0 to maximum floating point number.	Defines the base frequency of the field.
.theta_sweep_frequency			0.0 to maximum floating point number.	Defines the frequency in Hertz for the theta dynamics to complete a cycle.

Ux .structure	Fields	SubFields	Range of Values	Descriptions
	.theta_sweep_amplitude		0.0 to maximum floating point number.	Defines the frequency in Hertz for the theta dynamics to complete a cycle.
	.phi_sweep_frequency		0.0 to maximum floating point number.	Defines the frequency in Hertz for the theta dynamics to complete a cycle.
	.phi_sweep_amplitude		0.0 to maximum floating point number.	Defines the frequency in Hertz for the theta dynamics to complete a cycle.
	.radius		0.0 to maximum floating point number.	Defines radius of the field at the source.
	.k			For future simulations.
Cloud (sent by VIDS-equipped vehicles)				
	.location		SIMNET WorldCoordinates An array of 3 floating point rates.	Defines the location of the smoke cloud in the Defines the rate (meters/second) at which a cloud moves from its origin.
	.drift		0 to maximum floating point number.	Defines the age of the smoke cloud
	.age		Visual_Obscured, Visual_IR_Obscured, Millimeter_Wave_Obscured	Defines the energy absorbing properties of the cloud.
	.type			
VIDS-GT Instrumentation (sent to the Data Logger)				
	.VehicleId		SIMNET Vehicle Id 0 to 10	Defines which vehicle issued the Instrumentation PDU
	.Sensor_Count			Defines the number of available threats in the
	.Sensors		[10] [10]	Avail_Sensors array. Each element uniquely defines the available sensor.
	.Sensor_Accuracy			Each element defines the accuracy of the corresponding sensor.
	CCH_Thumb_Switch		released, depressed	Defines the current state of the Commanders Control Handle thumb switch.
	.Turret_Controlled_by_VIDS		false, true	Defines if VIDS has seized control of the turret.
	.Excess_Threats		0 to 255	Defines the number of detected threats but not currently displayed on the CCDDP.
VIDS-PC Instrumentation (sent to the Data Logger)				
	.VehicleId		SIMNET Vehicle Id	Defines which vehicle issued the Instrumentation PDU.
	.EventId		SIMNET Event Id	Defines the unique identification number of the PDU.
	.Button		0 to 255	Defines which button is changing state.
	.Menu		0 to 255	Defines which menu the button is on.
	.New_Menu		0 to 255	Defines which menu will be displayed next.
	.Button_State		released, depressed	Defines the state of the button.
	.Alert_Msg		[21]	Holds a copy of the alert message displayed on the CCDDP. The string is null terminated.

6. CSCI data files.

Files are not shared between VIDS CSCs or CSUs.

7. Requirements traceability.

The following table depicts the requirements traceability. The requirements are grouped into sections representing incremental refinements. For example, requirements beginning with the number 1 signify Phase 1 requirements; requirements beginning with the number 2 signify Phase 2 requirements.

[illegible]

[illegible]

VIDS SDD

8. Notes.

Acronyms

ATRJ
AVSE
CCDP
CFIRE
CM
CPS
CSC
CSCI
CSU
FASR
FLIR
GT
IDD
LCMD
LWR
MCC
MCD
MFD
MWS
NBCOP

NCS
NIS
PC
ROS
SAF
SCC
SIMNET
SMI
TCS
TRWR
VehicleID

VEMASID
VIDS

Definitions

Advanced Threat Radar Jammer
Armored Vehicle Survivability Equipment
Commander's Controls Display Panel
Counterfire
Countermeasure
Combat Protection System
Computer Software Component
Computer Software Configuration Item
Computer Software Unit
Future Armored System Radar
Forward Looking Infrared
Graphics Technologies
Interface Description Document
Laser Countermeasure Device
Laser Warning Receiver
Management Command and Control
Missile Countermeasure Device
Muzzle Flash Detector
Missile Warning System
Nuclear Biological Chemical Overpressure System
Nuclear Chemical Sensor
Non-Imaging Sensor
Personal Computer
Rapid Obscuration System
Semi-Automated Forces
Simulation Control Console
Simulation Network
Soldier Machine Interface
Threat Countermeasure System
Tank Radar Warning Receiver
An integer triplet consisting of site, host and vehicle numbers. Used to uniquely identify an entity within a battle exercise.
Vehicle Magnetic Signature Duplication
Vehicle Integrated Defense System

Appendix A. Sensor/Countermeasure Configuration File Content

```

# This is the VIDS data file. It is read at startup.
# All data elements are preceded by a label. After the label
# there will be one or more data values which will be delimited
# by tabs. Any characters inserted between tabs will be
# considered to be a data value (even a blank space).
#
# All time, angular measurements, probabilities and distances
# are defined as floating point values.
#
VIDS_Console_Site          1
VIDS_Console_Host         10
VIDS_Console_Vehicle      1
#
Available_Sensors          LWR    MWS    FASR    MFD    NCS    NIS    TRWR    MINE
Available_CMs              MCD    ROS    CMINE   TCS    NBCOP  NCS    TRWR    MINE
Auto_Activated_CMs        MCD    ROS    CMINE   TCS    NBCOP  NCS    TRWR    MINE
#
Max_Threats                10
#
VIDS_Processing_Delay      0.2
#
# Sensor Parameters
#
MWS.Response_Time_in_sec   1.2
MWS.Alarm_Duration_in_sec  3.0
MWS.Max_Detection_Distance_in_meter 6000.0
MWS.Max_Approach_Angle_in_Deg 22.5
MWS.Azimuth_Coverage_Central_Angle_in_Deg 0.0
MWS.Azimuth_Coverage_Delta_in_Deg 180.0
MWS.Elevation_Coverage_Central_Angle_in_Deg 15.0
MWS.Elevation_Coverage_Delta_in_Deg 25.0
MWS.Detection_Probability  0.98
MWS.Detection_Accuracy_in_Deg 2.0
MWS.Life_Countdown_in_sec  30.0
#
LWR.Response_Time_in_sec   0.5
LWR.Alarm_Duration_in_sec  3.0
LWR.Azimuth_Coverage_Central_Angle_in_Deg 0.0
LWR.Azimuth_Coverage_Delta_in_Deg 180.0
LWR.Elevation_Coverage_Central_Angle_in_Deg 15.0
LWR.Elevation_Coverage_Delta_in_Deg 25.0
LWR.Detection_Probability_LRF 0.92
LWR.Detection_Probability_LBR 0.97
LWR.Detection_Probability_LDES 0.97
LWR.Detection_Accuracy_in_Deg 3.0
LWR.Life_Countdown_in_sec  30.0
#
FASR.Response_Time_in_sec  0.5
FASR.Alarm_Duration_in_sec 3.0
FASR.Max_Detection_Distance_in_meter 5000.0
FASR.Azimuth_Coverage_Central_Angle_in_Deg 0.0
FASR.Azimuth_Coverage_Delta_in_Deg 10.0
FASR.Elevation_Coverage_Central_Angle_in_Deg 0.0
FASR.Elevation_Coverage_Delta_in_Deg 5.0
FASR.Detection_Probability 0.95
FASR.Identification_Probability 0.90
FASR.Detection_Accuracy_in_Deg 1.0
FASR.Update_Frequency_in_sec 5.0
FASR.Life_Countdown_in_sec 60.0
#
MINE.Response_Time_in_sec  0.2
MINE.Alarm_Duration_in_sec 2.0
MINE.Max_Detection_Distance_in_meter 36.5
MINE.Azimuth_Coverage_Central_Angle_in_Deg 0.0
MINE.Azimuth_Coverage_Delta_in_Deg 11.8
MINE.Elevation_Coverage_Central_Angle_in_Deg -43.51
MINE.Elevation_Coverage_Delta_in_Deg 1.49
MINE.Detection_Probability_Real 0.90
MINE.Detection_Probability_False 0.10
MINE.Detection_Accuracy_in_Deg 0.0
MINE.Life_Countdown_in_sec 30.0
#
NIS.Response_Time_in_sec  4.0
NIS.Alarm_Duration_in_sec 3.0

```

VIDS SDD

NIS.Azimuth_Coverage_Central_Angle_in_Deg	0.0					
NIS.Azimuth_Coverage_Delta_in_Deg	180.0					
NIS.Elevation_Coverage_Central_Angle_in_Deg	0.0					
NIS.Elevation_Coverage_Delta_in_Deg	60.0					
NIS.Detection_Distances_Engine_Off	5000.0	7000.0	10000.0	12000.0	15000.0	18000.0
NIS.Detection_Probabilities_Engine_Off	1.0	0.93	0.90	0.85	0.73	0.50
NIS.Identification_Distances_Engine_Off	5000.0	7000.0	10000.0			
NIS.Identification_Probabilities_Engine_Off	1.0	0.88	0.66			
NIS.Detection_Distances_Engine_On	2000.0	4000.0	7000.0	10000.0	12000.0	15000.0
NIS.Detection_Probabilities_Engine_On	1.0	0.90	0.79	0.71	0.64	0.57
NIS.Identification_Distances_Engine_On	2000.0	4000.0	7000.0			
NIS.Identification_Probabilities_Engine_On	1.0	0.84	0.71			
NIS.Detection_Accuracy_in_Deg	6.0					
NIS.Update_Frequency_in_sec	5.0					
NIS.Life_Countdown_in_sec	60.0					
#						
TRWR.Response_Time_in_sec	1.0					
TRWR.Alarm_Duration_in_sec	3.0					
TRWR.Azimuth_Coverage_Central_Angle_in_Deg	0.0					
TRWR.Azimuth_Coverage_Delta_in_Deg	120.0					
TRWR.Elevation_Coverage_Central_Angle_in_Deg	37.5					
TRWR.Elevation_Coverage_Delta_in_Deg	42.5					
TRWR.Detection_Probability	0.99					
TRWR.Detection_Accuracy_in_Deg	10.0					
TRWR.Life_Countdown_in_sec	30.0					
#						
MFD.Response_Time_in_sec	0.5					
MFD.Alarm_Duration_in_sec	0.5					
MFD.Azimuth_Coverage_Central_Angle_in_Deg	0.0					
MFD.Azimuth_Coverage_Delta_in_Deg	180.0					
MFD.Elevation_Coverage_Central_Angle_in_Deg	15.0					
MFD.Elevation_Coverage_Delta_in_Deg	25.0					
MFD.Detection_Probability	0.95					
MFD.Detection_Accuracy_in_Deg	2.0					
MFD.Life_Countdown_in_sec	30.0					
#						
NCS.Response_Time_in_sec	0.5					
NCS.Alarm_Duration_in_sec	3.0					
NCS.Azimuth_Coverage_Central_Angle_in_Deg	0.0					
NCS.Azimuth_Coverage_Delta_in_Deg	180.0					
NCS.Elevation_Coverage_Central_Angle_in_Deg	0.0					
NCS.Elevation_Coverage_Delta_in_Deg	90.0					
NCS.Detection_Probability	0.95					
NCS.Detection_Accuracy_in_Deg	0.0					
NCS.Life_Countdown_in_hour	8.0					
#						
# Countermeasures						
#						
ROS.Coverage_Angle_in_Deg	15.0					
ROS.Max_Turret_Rotation_Rate	45.0					
ROS.Launch_Distance_in_meter	30.0					
ROS.Response_Time_in_sec	0.0					
#						
MCD.Response_Time_in_sec	0.2					
MCD.Jam_Time_in_sec	3.0					
MCD.Azimuth_Coverage_Central_Angle_in_Deg	0.0					
MCD.Azimuth_Coverage_Delta_in_Deg	11.0					
MCD.Elevation_Coverage_Central_Angle_in_Deg	0.0					
MCD.Elevation_Coverage_Delta_in_Deg	5.0					
MCD.Max_Turret_Rate	128.57					
#						
CPS.Response_Time_in_sec	0.5					
CPS.Jam_Time_in_sec	3.0					
CPS.Azimuth_Coverage_Central_Angle_in_Deg	0.0					
CPS.Azimuth_Coverage_Delta_in_Deg	60.0					
CPS.Elevation_Coverage_Central_Angle_in_Deg	15.0					
CPS.Elevation_Coverage_Delta_in_Deg	15.0					
CPS.Max_Turret_Rate	128.57					
#						
LCMD.Response_Time_in_sec	0.5					
LCMD.Jam_Time_in_sec	3.0					
LCMD.Decoy_Distance_in_meter	20.0					
LCMD.Max_Turret_Rate	360.0					
#						
CMINE.Response_Time_in_sec	0.5					
CMINE.Jam_Time_in_sec	3.0					
#						
NBCOP.Response_Time_in_sec	30.0					
NBCOP.Crew_Life_Countdown_in_sec	60.0					
NBCOP.Activation_Time_in_hour	8.0					
NBCOP.Decontamination_Probability	0.95					

VIDS SDD

```
#
ATRI.Response_Time_in_sec      0.5
ATRI.Jam_Time_in_sec          3.0
ATRI.Azimuth_Coverage_Central_Angle_in_Deg 0.0
ATRI.Azimuth_Coverage_Delta_in_Deg 180.0
ATRI.Elevation_Coverage_Central_Angle_in_Deg 40.0
ATRI.Elevation_Coverage_Delta_in_Deg 50.0
ATRI.Max_Turret_Rate          128.57
#
TCS.Coverage_Angle_in_Deg      90.0
TCS.Inventory                  2      2      2      2
#
Chaff.Response_Time_in_sec      0.5
Chaff.Inventory                30
Chaff.Duration_in_sec          8.0
Chaff.Launch_Distance_in_meter 75.0
Chaff.Launch_Angle_in_Deg      45.0
Chaff.Radius_in_meter          21.0
Chaff.Max_Turret_Rate          128.57
#
Flares.Response_Time_in_sec     0.5
Flares.Inventory                30
Flares.Duration_in_sec          4.0
Flares.Launch_Distance_in_meter 150.0
Flares.Launch_Angle_in_Deg     45.0
Flares.Radius_in_meter         22.0
Flares.Max_Turret_Rate         128.57
#
CFIRE.Max_Turret_Rate           45.0
#
TSCAN.Max_Turret_Rate           3.0
#
Data_Collection_PDU_Period_in_sec 30.0
#
```

The following defines the grenade load for the turret sectors
starting from the gun (top) and going clockwise in 15 degree increments.

Label tab L8A1 tab M76 tab XM81 , top and clockwise

Sector	L8A1	M76	XM81
Sector_0	2	2	0
Sector_15	2	2	0
Sector_30	2	2	0
Sector_45	1	1	0
Sector_60	1	1	0
Sector_75	1	1	0
Sector_90	1	1	0
Sector_105	2	2	0
Sector_120	2	2	0
Sector_135	1	1	0
Sector_150	1	1	0
Sector_165	1	1	0
Sector_180	1	1	0
Sector_195	1	1	0
Sector_210	1	1	0
Sector_225	1	1	0
Sector_240	1	1	0
Sector_255	2	2	0
Sector_270	2	2	0
Sector_285	1	1	0
Sector_300	1	1	0
Sector_315	1	1	0
Sector_330	1	1	0
Sector_345	2	2	0

The following is the CM Threat Mapping to be used by VIDS.

Valid Order values: 0 - 8. The 0 order is the first CM to be used.

#threat class	priority	list of CMs in priority selection order							
Class_1	14	CPS	MCD	VIS	4	IR	4	Flares	
Class_2	13	CPS	MCD	VIS	4	IR	4	Flares	
Class_3	10	CPS	MCD	VIS	4	IR	4		
Class_4	6	CPS	LCMD	IR	4	VIS	4		
Class_5	8	CPS	IR	4	VIS	4	4		
Class_6	11	CPS	MCD	VIS	4	IR	4		
Class_7	7	CPS	IR	4	VIS	4	4		
Class_8	12	CPS	MCD	VIS	4	IR	4		
Class_9	9	CPS	MCD	IR	4	VIS	4		
Class_10	5	CPS	VIS	4	IR	4	4		
Class_Mine	1	CMINE							
Class_Chemical	2	NBCOP							
Class_Muzzle	18	CPS	VIS	4	IR	4	4		
Class_Muzzle_w_LRF	4	CPS	VIS	4	IR	4	4		
Class_Muzzle_w_TCS	3	CPS	VIS	4	IR	4	4		

VIDS SDD

Class_Helicopter	15
Class_Tank	16
Class_Infantry_Support	17