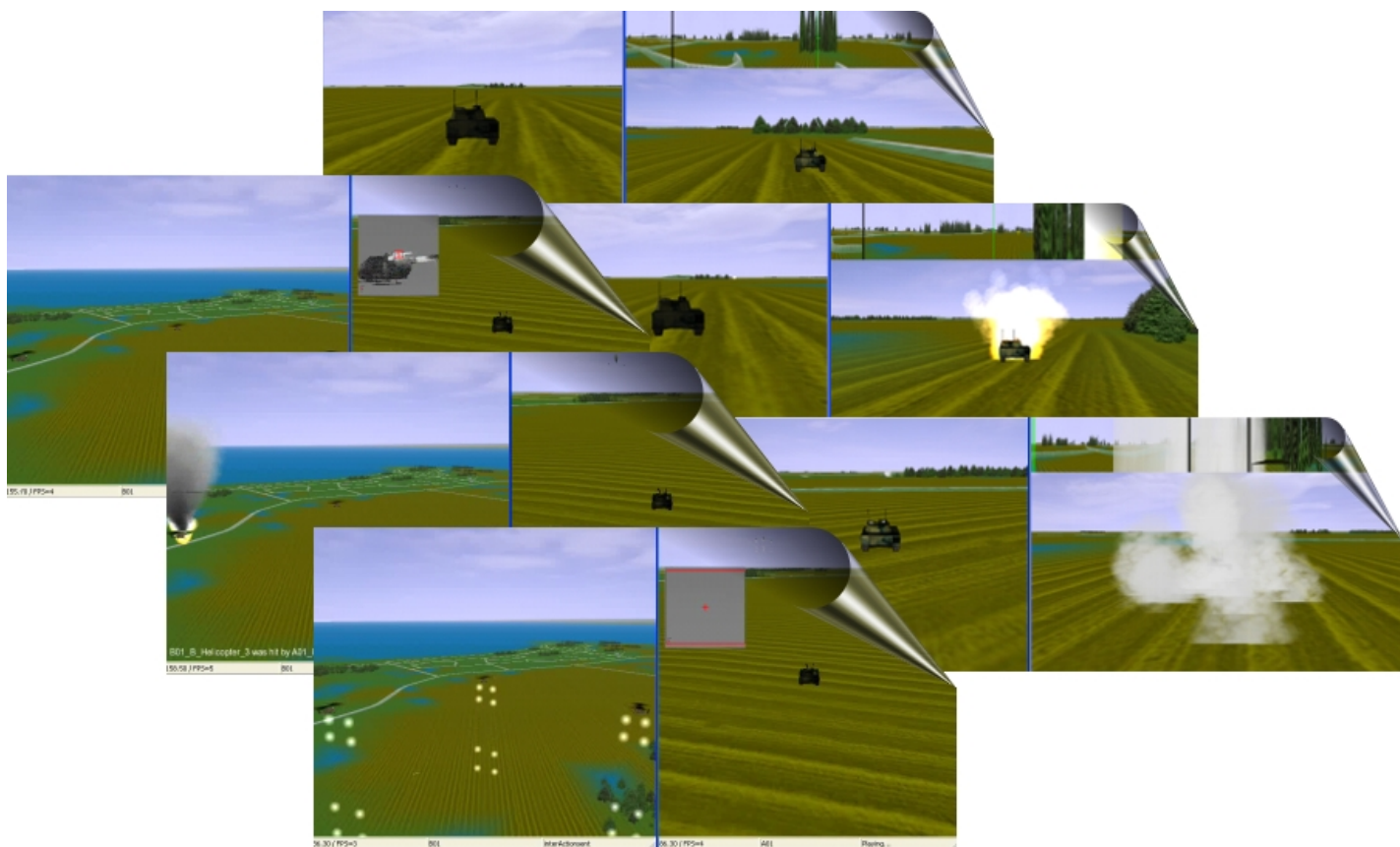


Hanna Andersson, Carl Hedberg,  
Mikael Petersson, Lars Tydén,  
Christer Wigren



FOI är en huvudsakligen uppdragsfinansierad myndighet under Förvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1350 anställda varav ungefär 950 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömningen av olika typer av hot, system för ledning och hantering av kriser, skydd mot hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.



FOI  
Totalförsvarets forskningsinstitut  
Ledningssystem  
Box 1165  
581 11 Linköping

Tel: 013-37 80 00  
Fax: 013-37 81 00

[www.foi.se](http://www.foi.se)

# Integrerad EO duellmodell



<b>Utgivare</b> FOI - Totalförsvarets forskningsinstitut Ledningssystem Box 1165 581 11 Linköping	<b>Rapportnummer, ISRN</b> FOI-R--1724--SE	<b>Klassificering</b> Metodrapport
	<b>Forskningsområde</b> 6. Telekrig och vilseledning	
	<b>Månad, år</b> September 2005	<b>Projektnummer</b> E7015
	<b>Delområde</b> 61 Telekrigföring med EM-vapen och skydd	
	<b>Delområde 2</b>	
<b>Författare/redaktör</b> Hanna Andersson Carl Hedberg Mikael Petersson Lars Tydén Christer Wigren	<b>Projektledare</b> Lars Tydén	
	<b>Godkänd av</b> Mikael Sjöman	
	<b>Uppdragsgivare/kundbeteckning</b> FM	
	<b>Tekniskt och/eller vetenskapligt ansvarig</b> Lars Tydén	
<b>Rapportens titel</b> Integrerad EO duellmodell		
<b>Sammanfattning (högst 200 ord)</b> <p>I många militära operationer är EO-system (elektrooptiska) och missiler en realitet. Det är därför viktigt att ha motmedelssystem mot dessa och veta hur man kan försvara sig. En vanlig metod att utvärdera motmedelssystem och deras användning är simuleringar.</p> <p>EWSim är ett multispektralt (radio optronik och radar) ramverk som använder sig av HLA. Detta dokument beskriver de EO-simuleringsmoduler (IRST, optikspanare, UV-varnare, laservarnare, missiler, facklor, rök, DIRCM, optik och logikenheten till VMS) som ingår i EWSim, vilka kan kombineras för simulering och värdering av bl.a. elektrooptiska telekrigdueller som därmed kan stödja utveckling, kravsättning och förståelse för samverkande telekrigssystem. I rapporten visas exempel på scenarier som kan simuleras. Scenarierna kan tyckas vara ganska enkla men tittar man närmare på dessa så har de mellan fem och tjugo modeller av system som samverkar eller motverkar varandra. Modulerna kan kombineras och sättas på alla typer av plattformar såsom stridsfordon, helikoptrar, UAV:er, flygplan, båtar och fartyg. Detta gör att sammansatta förbands telekrigföringsförmåga samt sårbarhet för telekrig kan simuleras.</p> <p>Med EWSim finns nu möjligheten att sätta upp scenarier som kan värdera några mot några i enskilt användande eller som värderingsduell där man kan bilda lag som kan duellera mot ett motståndarlag.</p>		
<b>Nyckelord</b> Simulering, telekrig, elektrooptik, EWSim, HLA		
<b>Övriga bibliografiska uppgifter</b>	<b>Språk</b> Svenska	
<b>ISSN</b> 1650-1942	<b>Antal sidor:</b> 84 s.	
<b>Distribution enligt missiv</b>	<b>Pris:</b> Enligt prislista	

<b>Issuing organization</b> FOI – Swedish Defence Research Agency Command and Control Systems Box 1165 581 11 Linköping	<b>Report number, ISRN</b> FOI-R--1724--SE	<b>Report type</b> Methodology report
	<b>Programme Areas</b> 6. Electronic Warfare and deceptive measures	
	<b>Month year</b> September 2005	<b>Project no.</b> E7015
	<b>Subcategories</b> 61 Electronic Warfare including Electromagnetic Weapons and Protection	
	<b>Subcategories 2</b>	
<b>Author/s (editor/s)</b> Hanna Andersson Carl Hedberg Mikael Petersson Lars Tydén Christer Wigren	<b>Project manager</b> Lars Tydén	
	<b>Approved by</b> Mikael Sjöman	
	<b>Sponsoring agency</b> FM	
	<b>Scientifically and technically responsible</b> Lars Tydén	
<b>Report title (In translation)</b> Integrated model of EO duel		
<b>Abstract (not more than 200 words)</b> <p>Electro-Optical (EO) systems and missiles are used in many military operations. It is therefore important to have countermeasure systems and know how to use them for defense against such threats. A common method to assess countermeasure systems and their use is to use simulations.</p> <p>EWSim is a multispectral framework (radio, radar and EO) that uses HLA. This document describes the modules for EO simulation (IRST, optic surveillance, UV missile approach warner, laser warner, missiles, flares, smoke, DIRCM, optics and DAS control unit) that is part of EWSim and can be combined for simulations and assessments of e.g. EO EW duels and can therefore be used to support development, specification of needed requirements and understanding of cooperating EW units. Examples of scenarios that can be simulated are shown in the report. The first impression might be that these scenarios are rather simple, but a closer inspection reveals between five and twenty models of systems that cooperate or act against each other in every scenario. The modules can be combined and placed on many different types of platforms such as fighting vehicles, helicopters, UAVs, aircrafts and ships. Therefore it is possible to simulate and study the EW-abilities of joint forces and to pinpoint their possible vulnerabilities to various EW-systems.</p> <p>It is now possible, using EWSim, to setup scenarios that can assess the situation few-against-few. The simulation can be controlled by one user, or teams of users can be created where one team competes against the other.</p>		
<b>Keywords</b> Simulation, electronic warfare (EW), electro-optical (EO), EWSim , HLA		
<b>Further bibliographic information</b>	<b>Language</b> Swedish	
<b>ISSN</b> 1650-1942	<b>Pages</b> 84 p.	
	<b>Price acc. to pricelist</b>	

# Innehåll

<b>1</b>	<b>INLEDNING</b>	<b>7</b>
<b>2</b>	<b>MODELLBESKRIVNING</b>	<b>9</b>
<b>2.1</b>	<b>Basklasser</b>	<b>13</b>
2.1.1	Egenskaper för BaseEntity och NetScene	13
2.1.2	PhysicalEntity	15
2.1.3	Plattform	17
<b>2.2</b>	<b>IRST</b>	<b>18</b>
2.2.1	Allmänt om IRST	18
2.2.2	IRST i EWSim	19
2.2.3	Modellbeskrivning	21
2.2.4	Beskrivning av parametrar	25
<b>2.3</b>	<b>Optikspanare</b>	<b>27</b>
2.3.1	Allmänt om optikspanare	27
2.3.2	Optikspanare i EWSim	27
2.3.3	Modellbeskrivning	29
2.3.4	Beskrivning av parametrar	30
<b>2.4</b>	<b>UV-varnare</b>	<b>31</b>
2.4.1	Allmänt om UV-varnare	31
2.4.2	UV-varnare i EWSim	31
2.4.3	Modellbeskrivning	32
<b>2.5</b>	<b>Laservarnare</b>	<b>33</b>
2.5.1	Allmänt om laservarnare	33
2.5.2	Laservarnare i EWSim	33
2.5.3	Modellbeskrivning	33
<b>2.6</b>	<b>Missil</b>	<b>34</b>
2.6.1	IR-missil i EWSim	34
2.6.2	Beskrivning av parametrar	35
<b>2.7</b>	<b>Facklor</b>	<b>43</b>
2.7.1	Allmänt om facklor	43
2.7.2	Fackelfällning i EWSim	43
2.7.3	Modellbeskrivning	43
2.7.4	Beskrivning av parametrar	45
<b>2.8</b>	<b>Rök</b>	<b>49</b>
2.8.1	Allmänt om rök	49
2.8.2	Rök i EWSim	49
2.8.3	Modellbeskrivning	49
2.8.4	Beskrivning av parametrar	50
<b>2.9</b>	<b>DIRCM</b>	<b>51</b>
2.9.1	Allmänt om DIRCM	51
2.9.2	DIRCM i EWSim	51
2.9.3	Modellbeskrivning	52
2.9.4	Beskrivning av parametrar	53
<b>2.10</b>	<b>Optik</b>	<b>57</b>
2.10.1	Optik i EWSim	57
2.10.2	Beskrivning av parametrar	57

<b>2.11</b>	<b>VMS-logik</b>	<b>60</b>
2.11.1	VMS i EWSim	60
<b>3</b>	<b>EXEMPEL PÅ SIMULERADE SCENARIER</b>	<b>62</b>
<b>3.1</b>	<b>Scenario 1: Bildalstrande missil mot ett stridsfordon</b>	<b>63</b>
3.1.1	Valda parametrar	63
3.1.2	Simulering	63
<b>3.2</b>	<b>Scenario 2: Stridsfordon mot stridsfordon</b>	<b>67</b>
3.2.1	Valda parametrar	67
3.2.2	Simulering	68
<b>3.3</b>	<b>Scenario 3: En bildalstrande IR missil mot tre Hkp</b>	<b>73</b>
3.3.1	Valda parametrar	73
3.3.2	Simulering	74
<b>3.4</b>	<b>Scenario 4: En retikelmålsökare mot IRST och DIRCM</b>	<b>78</b>
3.4.1	Valda parametrar	78
3.4.2	Simulering	78
<b>4</b>	<b>DISKUSSION</b>	<b>82</b>
<b>5</b>	<b>FORTSATT ARBETE</b>	<b>83</b>
<b>6</b>	<b>REFERENSER</b>	<b>84</b>



# 1 Inledning

I många militära operationer är elektrooptiska (EO) system och missiler en realitet. Det är därför viktigt att ha motmedelssystem mot dessa och veta hur man kan försvara sig. En vanlig metod att utvärdera motmedelssystem och deras användning är simuleringar. Att utveckla ett ramverk för simuleringar är ett omfattande arbete, och det är därför bra om ett sådant ramverk är flexibelt och fungerar för flera olika typer av plattformar både i mark-, luft- och sjöfallet. Dessutom bör ramverket använda den vanligast förekommande simuleringsstandarden HLA<sup>1</sup> som gör det lättare att bli interoperabel med modeller utvecklade av andra.

När man simulerar EO-scenarier i klottriga miljöer krävs detaljerade tredimensionella modeller av mål, bakgrund och motmedel. Simulerar man scenarier där bakgrundsklottret är försumbart, t ex en missil som närmar sig ett flygplan nerifrån en molnfri dag, räcker det däremot kanske att modellera mål och motmedel som punktojekt. Fördelen med att använda fullständiga 3D-modeller även i ett sådant scenario är att man i varje tidssteg kan kontrollera om geometrin mellan mål och missil gör att varma delar på mål och motmedel skymms av kallare delar. Av denna anledning har en 3D-scengraf (en graf som beskriver 3D-omgivningen) integrerats med modellerna så att alla dessa har möjlighet att använda bilder skapade ur en gemensam omvärldsbeskrivning.

Resultat från simuleringar av telekrigdueller kan fylla olika funktioner. En simulering med detaljerade modeller av bakgrund, mål och motmedelssystem ökar förståelsen för de tekniska detaljerna i scenariet. Ett exempel på system som innehåller mycket detaljerade modeller för att simulera telekrigdueller är OPTSIM [OPTSIM]. Nackdelen med att använda alltför detaljerade modeller är att simuleringarna blir beräkningskrävande och långsamma. Vill man ha ett statistiskt underlag från en mängd simuleringar, t ex för att kunna ta fram taktiska rekommendationer, är simuleringar med snabbare och enklare modeller att föredra framför detaljrikedom. När simuleringssmodeller tas fram vägs alltid snabbhet mot noggrannhet. Kan simuleringarna utföras i realtid kan de också användas för utbildning av personal som handhar militära plattformar med motmedel. Traditionellt sett har realtidskravet inte varit förenligt med användandet av bilder från 3D-modeller annat än för att visa upp resultatet, speciellt då det har rört sig om IR-modeller. De senaste åren har detta ändrats, och det är nu möjligt att konstruera den typen av simuleringssramverk. Möjligheten finns även att och köpa färdiga produkter för att generera IR-bilder. Dessa saknar dock helt telekrig och man måste dessutom skriva på omfattande avtal om hur dessa ska användas. Detta projekt har valt att utveckla denna del själv, med hjälp av öppen källkod. Den öppna källkoden som finns tillgänglig idag erbjuder så pass mycket funktionalitet att detta är möjligt. Dessutom har ett forskningsprojekt som detta behov av att kunna modifiera funktionalitet och sprida resultat utan förbehåll från någon leverantör.

Det här dokumentet innehåller en beskrivning av EO-delen (visuella och IR-våglängder) av det utvecklade ramverket EWSim (Electronic Warfare Simulation interface modell) för distribuerade telekrigssimuleringar. Detta ramverk har förutom EO dessutom radar och kommunikation vilket sammantaget medger multispektral systemvärdering. EWSim innehåller modeller för plattformar, motmedel, varnare, sensorer samt missiler med målsökare och rörelsemodeller. EWSim använder HLA och OpenSceneGraph (OSG<sup>2</sup>) som är ett öppet källkodsprojekt med OpenGL i botten. Inom FoT-projektet Duellsimulering Telekrig har EWSim-moduler för att kunna simulera telekrig vidareutvecklats och anpassats från tidigare

<sup>1</sup> High Level Architecture, <https://www.dmsomil/public/transition/hla/>

<sup>2</sup> <http://www.openscenegraph.org/>

modeller [EWSIM]. Dessutom har nyutvecklade moduler såsom DIRCM, IRST och optikspaning tillkommit. EWSim kan köras som en instans, ett s.k. federat, som kör ett flertal olika plattformar i ett scenario, eller som flera federater distribuerade på flera datorer i ett nätverk vilka tillsammans kör ett scenario. Detta möjliggör både enskilt användande och värderingspel med flera användare.

Syftet med EWSim är att tillhandahålla ett ramverk för multispektral systemvärdering av telekrig med möjlighet att köra scenarier med några mot några.

## 2 Modellbeskrivning

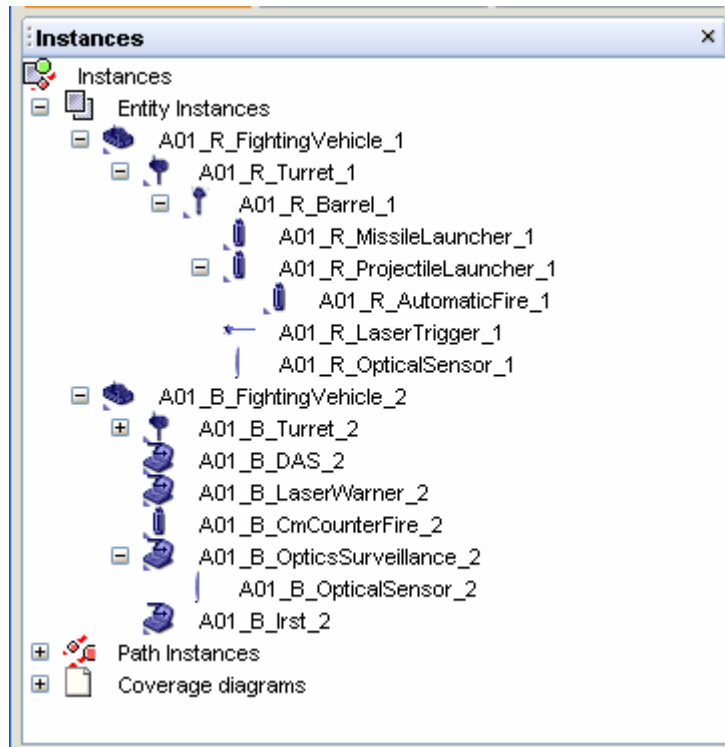
I en konfliktsituation, där telekrig kan komma till användning, kan flera objekt/plattformar vara inblandade på vardera sidan. Dessa plattformar kan vara utrustade med olika sensor-, verkans- och försvarssystem. Sensorsystem kan användas för att hitta motståndare att anfalla eller för att upptäcka ett hot (t ex missilsystem eller plattformar som skjuter projektiler) innan det hunnit verka.

Sensorer kan vara enkla kameror inom det visuella eller IR-området men de kan också vara mer avancerade system som automatiskt hittar tänkbara mål/hot. Exempel på avancerade spaningssystem inom det elektrooptiska våglängdsområdet kan vara IRST (InfraRed Search and Track) eller optikspanare. Den förra letar efter objekt med förhöjd signatur över bakgrunden och den senare letar efter optiska system riktade så att den egna plattformen finns inom dess synfält. Dessa system kan innehålla logik för att bara varna för mål som närmar sig den egna plattformen. Sensorer i form av rena varnare, t ex laservarnare eller UV-varnare kan också förekomma.

Varning (från laservarnare, UV-varnare, IRST, optikspanare m fl) tas om hand och resulterar i en motåtgärd som t ex kan vara att skjuta moteld, lägga ut rök, kasta facklor eller använda laserstörning (DIRCM). Varning kan också tas emot från en annan plattform via en radiolänk.

När en simulering sätts upp, i NetScene, konfigureras ett antal plattformar med önskade komponenter i en trädstruktur. I Figur 1 är detta gjort för två stridsfordon, A01\_R\_FightingVehicle\_1 och A01\_B\_FightingVehicle\_2. Vardera av dessa fordon har ett kanontorn (*Turret*) och varje kanontorn har ett eldrör (*Barrel*). Till det första stridsfordonets eldrör har en missil- och en projektilavfyrare (*MissileLauncher* och *ProjectileLauncher*) kopplats vilket gör att man kan avfyr en missil eller en projektil. Till projektilavfyraren har kopplats en automatisk avfyrning (*AutomaticFire*) som riktar in eldröret och avfyrar projektiler när ett fordon som inte tillhör den egna sidan finns inom synhåll. För att kunna mäta avstånd innan avfyrning finns en lasertrigger kopplad till eldröret. Dessutom finns en optisk sensor kopplad till eldröret (det kan t ex vara skyttens sikte). Förutom den automatiska avfyrningen av projektiler kan den som styr det första stridsfordonet även skjuta projektiler och missiler manuellt.

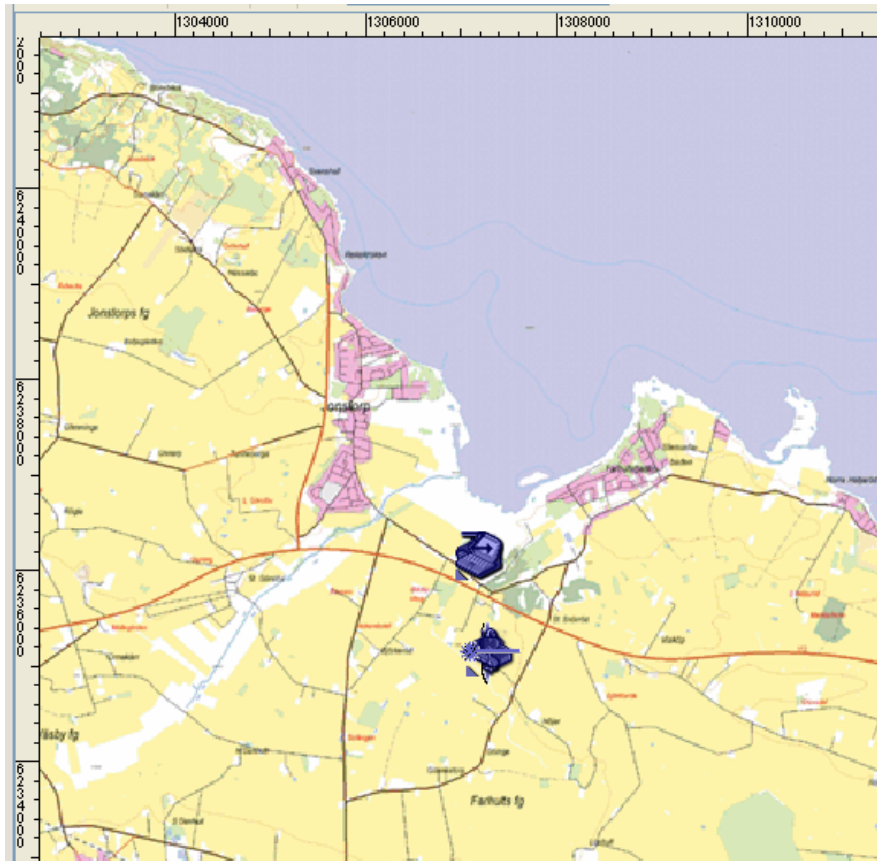
Det andra stridsfordonet är, förutom med ovan nämnda kanontorn och eldrör, utrustat med ett varnare- och motverkanssystem (VMS eller *DAS* som modulen kallas här). Detta VMS kommer att ha tillgång till alla varnare och motmedelssystem som ligger i trädstrukturen under dess förälder (ägaren till objektet, ett stridsfordon är t ex förälder till sitt torn och tornet till sitt eldrör). Föräldern är i detta fall stridsfordonet självt och dess underobjekt blir i detta fall allt under stridsfordonet. Det motmedelssystem som finns konfigurerat här är moteld (*CounterFire*) och varnare som finns är laservarnare, optikspanare (*OpticsSurveillance*) och IRST. VMS:et kan konfigureras så att den vidtar åtgärder beroende på vilken varning som kommer in.



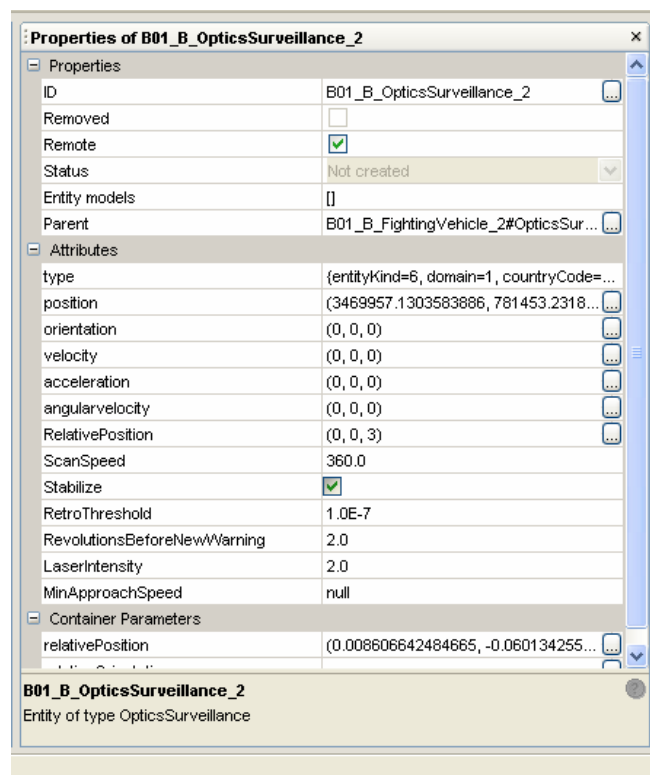
**Figur 1** Konfigurerat scenario i NetScene.

Innan simuleringen placeras stridsfordonen som beskrivits ovan ut i en karta, Figur 2. Till denna karta hör en 3D-terräng där de två fordonen kan röra sig oberoende av varandra under en simulering. Händelseförloppet under simulering skulle kunna vara följande. När stridsfordonen befinner sig på ett sådant sätt att de har möjligt att upptäcka varandra kommer den automatiska avfyrningsmodulen att vrida in kanontorn och eldrör mot det andra fordonet. Det tar lite tid för det avfyrande fordonet att bestämma sig för om fordonet är fientligt eller inte och därför avfyras en projektil inte direkt (denna tidsfördröjning kan ställas in). Det sikte som finns inriktat mot det andra fordonet kan upptäckas av dess optikspanare och eventuellt resultera i varning (beroende på inställningar). Fordonet har även en IR-signatur som även den kan resultera i varning beroende på inställningar för IRST:n. Om det första fordonet får tillräckligt mycket tid på sig att bestämma sig för att avfyra, mäter den först in avståndet till fordonet med en laseravståndsmätare vilket även det kan resultera i en varning i det andra stridsfordonets laservarnare. Hade det första fordonet skjutit iväg en missil istället för en projektil hade robotflamman kunnat ge varning till en UV-varnare ifall en sådan funnits tillgänglig. Roboten som går iväg kan också ge varning i IRST:n som nu lättare kan diskriminera det som ett verkligt hot eftersom det är ett mål som rör sig snabbt och är på väg mot det egna fordonet. Ifall roboten har en optik kan den upptäckas av optikspanaren. Beroende på inställningar i VMS:et kan varningar resultera i en motåtgärd. I detta fall finns bara moteld att tillgå men det går också att konfigurera upp t ex rök, facklor och DIRCM som motmedel. Förutom att ha automatiska hot och funktioner kan fordonen konfigureras så att de styrs av en användare som då får hitta fordon t ex med hjälp av någon sensor, ta beslut om avfyrning och eventuellt vidta motåtgärder.

Alla modellkomponenter kan konfigureras genom att, i NetScene, sätta parametervärden i modulens egenskapsfönster (ett exempel från optikspanaremodulen kan ses i Figur 3).



**Figur 2** Kartbild med två utplacerade stridsfordon, som är konfigurerade på olika sätt.



**Figur 3** Egenskaper för optikspanaren.

Detta kapitel beskriver olika modellkomponenter (moduler) som kan ingå i en telekrigssimulering inom det elektrooptiska området. Varje modul (IRST, optikspanare, UV-varnare, missiler, facklor, rök, DIRCM, optik och logikenheten till VMS) beskrivs i ett separat avsnitt som börjar med en kort allmän beskrivning och sedan går in på detaljer hur komponenten används och ställs in för användning med EWSim och NetScene. För läsare som inte är intresserade av att göra egna simuleringar kan detaljavsnitten hoppas över med gott samvete. För exempel på resultat se kapitel 3.

Alla moduler har vissa gemensamma egenskaper vilket beror på att nya typer av objekt kan skapas i NetScene genom en klasstruktur där en uppsättning parametrar ärvs från basklasser. Nya parametrar läggs till som är specifika för den nya klassen. Sedan kan andra typer av objekt ära från den klassen osv. I Figur 4 visas delar av denna klasstruktur i NetScene. Nästan alla objekt bygger direkt eller indirekt på typen *BaseEntity*. Bland dessa finns *PhysicalEntity*. Objekt som direkt bygger på *PhysicalEntity* är t ex kanontorn, eldrör och plattform. Exempel på klasser som ärver från *Platform*-klassen är *GroundVehicle*, *Aircraft* och *SurfaceVessel*, osv. Klasserna *BaseEntity*, *PhysicalEntity* och *Platform* är ganska centrala och beskrivs därför separat i det inledande avsnittet, 2.1.



Figur 4 Klasstruktur i NetScene.

## 2.1 Basklasser

I detta avsnitt beskrivs egenskaper för en del basklasser som många av de moduler som beskrivs i senare avsnitt bygger på.

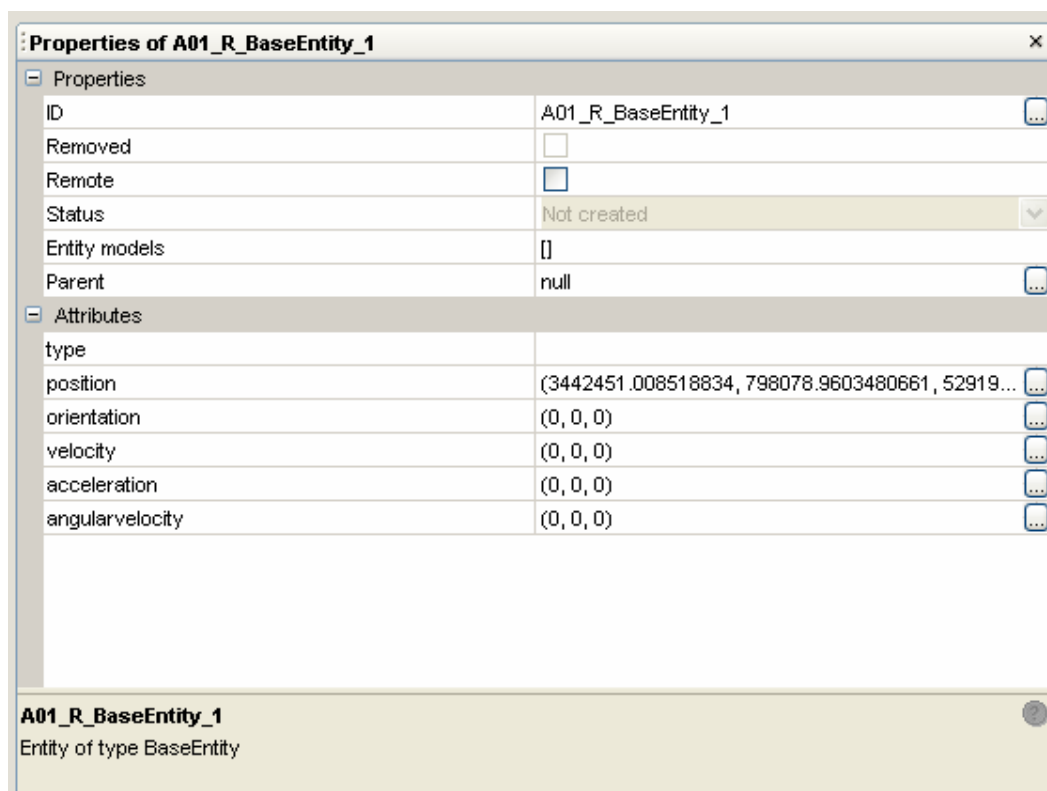
### 2.1.1 Egenskaper för BaseEntity och NetScene

NetScene-specifika egenskaper ges av parametrarna *ID*, *Removed*, *Remote*, *Status*, *Entity models* och *Parent* och bestämmer framförallt hur simuleringen skall kopplas upp och vem som skall kontrollera objektet.

*BaseEntity*-egenskaper ges av parametrar för typ, position, orientering, hastighet, acceleration och vinkelhastighet. Typen bestämmer vilket typ av objekt som skall skapas och kommer inte att ändras under simuleringens gång. De andra kan ges initiala värden i NetScene genom att skriva in siffror i dessa fält eller genom att flytta objektet i kartan.

### I Parametrar

Nedan visas de parameterinställningar som kan göras i NetScene för *BaseEntity*-objekt (och alla objekt som har *BaseEntity* som en basklass).



**Figur 5** Egenskapsparametrar för NetScene och *BaseEntity*.

**Tabell 1** Beskrivning av parametrar för NetScene och *BaseEntity*.

Parameternamn	Beskrivning
ID	Unikt namn som identifierar objektet. Bör vara av formen <b>***_X_ModulTyp_#</b> , där * är ett valfritt tecken, X en bokstav och # en siffra (***) används för att bestämma vilken federat som skall styra objektet, X används för att ange vilken sida som objektet tillhör t ex R för röd eller B för blå, # måste väljas så att namnet blir unikt).
Removed	Parameter som visar om objektet har tagits bort.
Remote	Anger om objektet skall styras från ett annat federat eller om NetScene skall styra objektet.
Status	Visar objektets status, t ex om objektet är skapat.
Entity models	Kan användas för att låta objektet gå längs en fördefinierad bana. Skall vara på formen (allt inklusive hakparenteser bör finnas med):  <pre>[SimpleMotionModel: {StartTime=0, Speed=200, PathID="Path1", OnGround=FALSE}]</pre> <i>SimpleMotionModel</i> anger vilken rörelsemodell som skall användas när objektets position beräknas. <i>StartTime</i> bestämmer när objektet skall börja röra på sig. <i>PathID</i> är ett id som svarar mot en fördefinierad bana som skall finnas med i scenariet. <i>OnGround</i> bestämmer om objektet måste röra sig på marken (lämpligt för markfordon och fartyg men kanske inte för helikoptrar och flygplan).
Parent	”Namnet på förälderobjektet” + ”#XXX”. Där XXX är ett specifikt namn på en containerplats, behållare för underobjekt. Godtagbara namn för rena <i>BaseEntity</i> objekt är:  RCSAntennaContainer AntennaContainer WarnerAntennaContainer DircmContainer LaserSourceContainer OpticalSensorContainer  Av namnet framgår vad som skall kopplas till vilken container. Om ett DIRCM-system skall kopplas till en helikopter (som i botten bygger på ett <i>BaseEntity</i> ) skall värdet på <i>parent</i> vara ”ParentID#DircmContainer”.
type	I många fall måste värdet på denna parameter bli rätt för att rätt typ av objekt skall skapas i EWSim när simuleringen startas. Värdet på denna parameter skall vara av formen (allt skall finnas med inklusive klamrar, # skall bytas mot siffror som svarar mot rätt objekt):  <pre>{entityKind=#, domain=#, countryCode=#, category=#, subcategory=#, specific=#, extra=#}</pre>

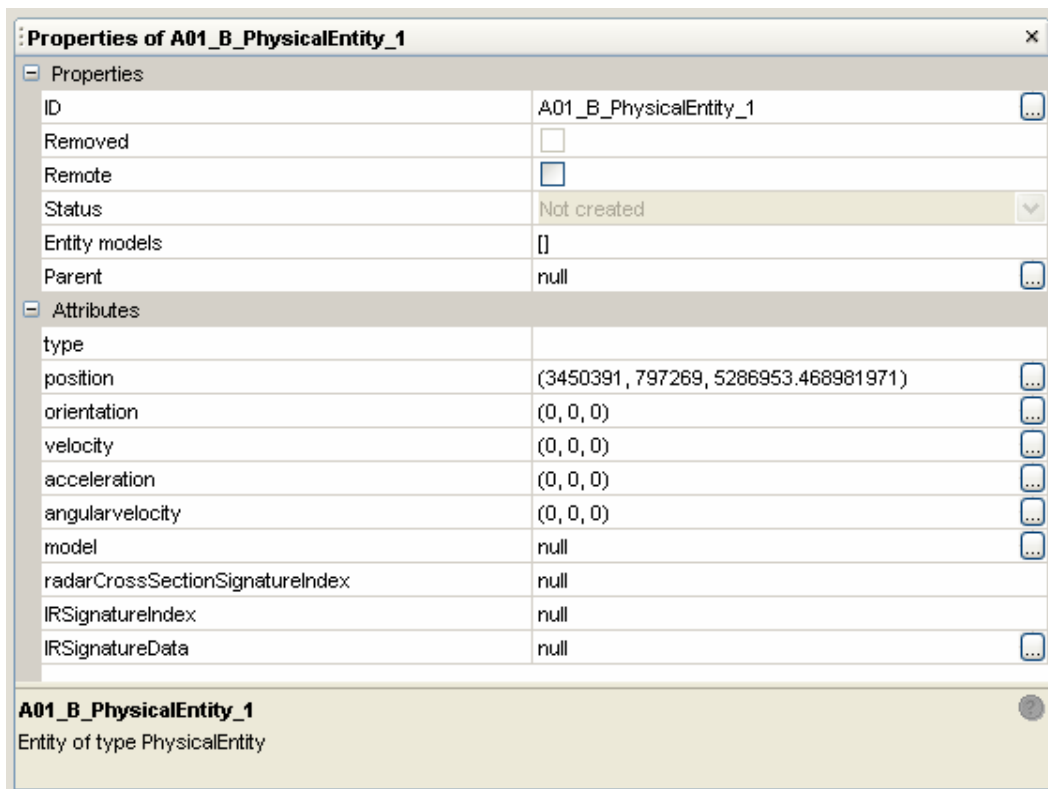


	För rätt värden se aktuell lista med typer för olika system [ENUM].
position orientation velocity acceleration angularvelocity	Dessa värden kommer att uppdateras när simuleringen pågår (NetScenes koordinatsystem används). I EWSim kommer dessa parametrar att sättas till samma som moderobjektet, men för användning i NetScene är det rekommenderat att positionen sätts till samma som för moderobjektet. (Vinklar anges i [rad]).

### 2.1.2 PhysicalEntity

En *PhysicalEntity* bygger i grunden på parametrar från *BaseEntity* men har en del tillägg för att sätta 3D-modell, radarmålteindex, IR-signaturindex och data för att beräkna IR-signaturen som funktion av våglängdsområde, motorpådrag och aspektvinkel. Det finns också containrar som kan användas till att koppla underobjekt som inte finns på rena *BaseEntity*-objekt.

#### I Parametrar



**Figur 6** Egenskapsparametrar för *PhysicalEntity*.

**Tabell 2** Beskrivning av parametrar för *PhysicalEntity*.

Parameternamn	Beskrivning
Parent	<p>”Namnet på förälderobjektet” + ”#XXX”. Där XXX är ett specifikt namn på en containerplats. Godtagbara namn för <i>PhysicalEntity</i> objekt är (förutom de som finns under <i>BaseEntity</i>):</p> <p>LaserTriggerContainer  LaserWarnerContainer  WarningFromRadioContainer  WarningToRadioTransmitterContainer  CmFlareContainer  CmFLSCFlareContainer  CmCounterFireContainer</p> <p>Av namnet framgår vad som skall kopplas till vilken container. Om en laservarnare skall kopplas till en <i>PhysicalEntity</i> (eller till något som på något sätt bygger på denna) skall värdet på parent vara ”ParentID# LaserWarnerContainer”.</p>
model	Filnamn som pekar på en 3D-modell (används inte ännu).
radarCrossSectionSignatureIndex	Index som ger radarmålarea.
IRSignatureIndex	Index som används för att hämta rätt data från <i>IRSignatureData</i> (se nedan) när IR-signaturen beräknas. Detta index kan t ex kopplas till motorpådrag på ett flygplan (som kan ändra sig dynamiskt under ett simuleringsförlopp).
IRSignatureData	<p>Skall vara på formen (kursiv text skall ersättas, icke kursiv text skall stå som det är):</p> <p>TABLE <i>namn</i>, 4, <i>antal våglängdsband</i>, <i>antal olika IR-signaturer</i>, <i>antal azimutvinklar</i>, <i>antal elevationsvinklar</i> /  <i>våglängdsindex1</i>, <i>våglängdsindex2</i>, ...,  <i>irsignaturindex1</i>, <i>irsignaturindex2</i>, ...,  <i>azimutvinkel1</i>, <i>azimutvinkel2</i>, ...,  <i>elevationsvinkel1</i>, <i>elevationsvinkel2</i>, ...,  <i>signature(våglängdsindex1, irsignaturindex1, azimutvinkel1, elevationsvinkel1)</i>,  <i>signature(våglängdsindex2, irsignaturindex1, azimutvinkel1, elevationsvinkel1)</i>, ...,  <i>signature(våglängdsindex1, irsignaturindex2, azimutvinkel1, elevationsvinkel1)</i>,  <i>signature(våglängdsindex2, irsignaturindex2, azimutvinkel1, elevationsvinkel1)</i>, ...</p> <p>Tal separeras med , eller /.</p>

### 2.1.3 Plattform

En *Plattform* bygger i grunden på parametrar från *PhysicalEntity* och har inga tillägg när det gäller parametrar men däremot kan andra typer av containrar användas, se Tabell 3.

**Tabell 3** Beskrivning av parametrar för *Plattform*.

Parameternamn	Beskrivning
Parent	<p>”Namnet på förälderobjektet” + ”#XXX”. Där XXX är ett specifikt namn på en containerplats. Godtagbara namn för Plattform objekt är (förutom de som finns under BaseEntity och PhysicalEntity):</p> <ul style="list-style-type: none"> <li>IrstContainer</li> <li>UVWarnerContainer</li> <li>OpticsSurveillanceContainer</li> <li>PlattformContainer</li> </ul> <p>Av namnet framgår vad som skall kopplas till vilken container. Om en IRST skall kopplas till en Plattform (eller till något som på något sätt bygger på denna) skall värdet på parent vara ”ParentID# IrstContainer”. PlattformContainer kan användas för att låta en plattform styra positionen på en annan (automatiska offsets läggs i så fall till).</p>

## 2.2 IRST

### 2.2.1 Allmänt om IRST

En IRST är en sensor som ofta klarar att svepa över ett stort vinkelområde av scenen och kan användas för att upptäcka och följa objekt med ett, inom IR-området, lågt signal-till-brus-förhållande (inkluderat klotter), t ex inkommande missiler. IRST förekommer både i mark-, fartygs- och flygmonterade applikationer, se exempel på användning på svenska system nedan.

#### I Markfallet

IRST-Mark (se Figur 7), en demonstrator från Saab Bofors Dynamics, är ett exempel på hur en IRST kan användas i stridsfordonsfallet för att hitta, presentera och följa mål, t ex andra stridsfordon, helikoptrar och missiler.



**Figur 7** Sensormoduler till IRST-Mark (källa [IRSTMARK]).

Lemuren (se Figur 8), som är ett stabiliserat modulärt uppbyggt sikte för stridsfordon, kan lätt konfigureras för olika behov och skulle även den kunna konfigureras för att användas som IRST. När det används som vagnchefssikte är den utrustad med en andra generationens IR-kamera och en TV-kamera. IR-kameran i Lemuren skulle kunna användas som en IRST, för att ge varning om ankommande robot.



**Figur 8** Lemursiktet (källa [LEMUR]).

## II Sjöfallet

En IRST kan användas på Visbykorvetten för att förbättra spaningskapacitet (se referens [IRSTSJÖ]).

## III Flygfallet

IR-OTIS är en IRST som utvecklas av Saab Bofors Dynamics för att användas på flygplanet JAS Gripen [IROTISI].



**Figur 9** IR-OTIS (källa [IROTISI]).

### 2.2.2 IRST i EWSim

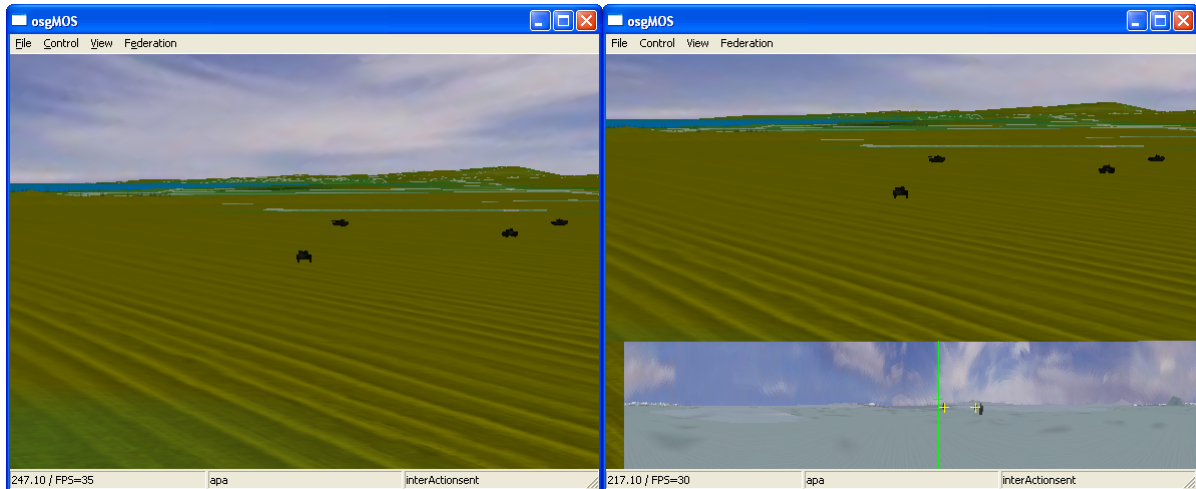
En IRST kan konfigureras i NetScene och kopplas till ett markfordon, en flygande plattform eller till ett fartyg, vilka kan styras i EWSim. IRST:n kan i modellen användas både som varnare och för att upptäcka objekt. För upptäckt krävs att objektet ligger inom IRST:ns synfält, inte är skymt och har en signatur (som ger en IR-signal) som överstiger bakgrundens (eller en given brusnivå) med en given faktor. I varnarsammanhang måste dessutom mottagen strålningseffekt öka och objektet får inte röra sig i vinkelled, vilket är karakteristiskt för ett närmande hot. Figur 10 visar de parameterinställningar som kan göras i NetScene. Dessa behandlas i avsnitt 2.2.4.

Properties	
ID	A01_B_lrst_2
Removed	<input type="checkbox"/>
Remote	<input checked="" type="checkbox"/>
Status	Not created
Entity models	[]
Parent	A01_B_FightingVehicle_2#lrstCo...
Attributes	
type	{entityKind=6, domain=1, countryCod...
position	(3468114.410395955, 781413.71 ...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
WavelengthRange	IR
VFOV	10.0
IFOVH	0.0010
IFOVV	0.0010
ScanSpeed	360.0
Stabilize	<input checked="" type="checkbox"/>
RelativePosition	(0, 0, 2)
NEI	1.0E-7
ExtinctionCoefficient	0.4
Threshold	8.0
TerrainSignatureData	!Signature index must be sorted i...
SkySignature	0.0
RevolutionsBeforeNewWarning	null
StoNIncreaseThreshold	null
MaxDetectionDeltaAngle	null
Container Parameters	
relativePosition	(0, 0, 0)
relativeOrientation	(0, 0, 0)

**Figur 10** Inmatning av parametrar för ett IRST-system i NetScene.

## I Användargränssnitt i EWSim

När ett objekt har ett IRST-system kan detta aktiveras i EWSim genom att den som styr moderobjektet trycker på tangenten 'y'. I detta fall aktiveras ett fält längst ner i bilden och visar vad IRST:n ser, se Figur 11.



**Figur 11** Användargränssnitt som visar en översiktsbild med kameran positionerad bakom och ovanför det styrda objektet (kamerans synfält är centrerat kring objektet). Det styrda objektet är utrustat med en IRST. Till höger har IRST:n aktiverats och en IRST-bild dyker upp i nederkanten av användargränssnittets fönster.

IRST-vyn som visas har gula + som markerar upptäckta målkandidater. Målkandidater, som resulterar i en varning genererar ett rött + tills denna varningsposition är överlämnad till plattformens VMS (se avsnitt 2.11). En grön vertikal linje markerar moderfordonets orientering i sidled och en horisontell linje på denna markerar moderfordonets orientering i vertikalled. Liknande gröna linjer dyker upp om fordonet har ett kanontorn med eldrör. Dessa visar kanontornets orientering i sidled och eldrörets orientering i höjddled.

Det går att stänga av IRST-vyn men ändå behålla dess funktionalitet. Samtidig tangenttryckning 'Ctrl'+ 'y' växlar mellan visning och icke visning av IRST-vyn när IRST:n är aktiverad.

### 2.2.3 Modellbeskrivning

#### I Generering av IRST-bild

En IRST har ofta en sensor som sveper över det totala synfältet på 360° i horisontalled. Simuleringsmodellen håller reda på vilket tidssteg,  $\Delta t$ , som används i simuleringen och ställer, för simuleringens skull, in ett horisontellt synfält,  $HFOV$ , som motsvarar detta tidssteg ( $HFOV = ScanSpeed \cdot \Delta t$ ). Bilden från en kamera med detta synfält flyttas sedan över nederkanten av bilden allt eftersom tiden går och IRST:ns sensor roterar för att täcka ett synfält på 360°. Att bygga upp den totala IRST bilden på detta sätt gör att ett objekts position i bilden bara uppdateras en gång per varv precis som för ett verkligt system.

## II Detektion av objekt

För att upptäckt av ett objekt skall kunna ske måste ett antal kriterier vara uppfyllda:

1. Objektet måste ligga i IRST:ns momentana synfält i horisontalled.
2. Objektet måste ligga i IRST:ns vertikala synfält.
3. Objektet får inte vara dolt bakom terrängen.
4. Objektet får inte vara dolt bakom rök.
5. Objektet får inte skymmas av andra objekt.
6. Den registrerade intensiteten från objektet måste överstiga bakgrunden med en viss faktor (*Threshold*).

Ett kriterium som står senare i listan kollas bara om tidigare kriterier är uppfyllda.

### II.1 Kriterium 1 - Objektet måste ligga i IRST:ns momentana synfält i horisontalled

Den horisontella riktningen till varje objekt beräknas och jämförs med den nuvarande horisontella riktningen på IRST:ns kamera. Kriteriet anses uppfyllt om skillnaden är mindre än halva det momentana synfältet.

### II.2 Kriterium 2 - Objektet måste ligga i IRST:ns vertikala synfält

Den vertikala riktningen till varje objekt beräknas och jämförs med den vertikala riktningen på IRST:n. Om IRST:n inte är stabiliserad i vertikalled går det inte att använda den momentana riktningen på IRST:n eftersom det simulerade momentana synfältet i horisontalled beror på simuleringens tidssteg och vid stora tidssteg kan det momentana synfältet också bli stort och därmed kan det bli stora skillnader i vinkeln i vertikalled i centrum av synfältet och i kanterna (i extremfallet när syfältet är 360° i sidled kan t ex IRST:n vara riktad uppåt i mitten av synfältet men neråt i kanterna). I stället använder vi IRST:ns uppåtriktning och jämför med enhetsvektorn som går mellan IRST:n och objektet.

$$\Delta\varphi = \frac{\pi}{2} - \arccos(\hat{r}_{IrstToObject} \cdot \hat{r}_{up})$$

### II.3 Kriterium 3 - Objektet får inte vara dolt bakom terrängen

För att göra en test av detta kriterium skapar vi en vektor med riktning som går från IRST:n till objektet. Sedan använder vi funktioner som finns inbyggda i den grafikmotor som vi använder (OpenSceneGraph) för att se hur långt borta terrängen ligger i denna riktning. Om avståndet till objektet är mindre än till terrängen betyder det att vi kan se målet.

### II.4 Kriterium 4 - Objektet får inte vara dolt bakom rök

Rökmoln som skapas i simuleringsprogrammet är speciella i det avseendet att vi inte kan använda vanliga grafikrutiner för att studera om rökmolnet ligger mellan sensor och mål. Rökmolnet simuleras som ett partikelsystem och när detta kriterium kontrolleras studeras alla partiklars position och radie tills det har konstaterats om någon partikel skymmer målets centrum.

### II.5 Kriterium 5 - Objektet får inte skymmas av andra objekt

Det enklaste hade förmodligen varit att kontrollera detta kriterium samtidigt som kriterium 3 men för att få bästa resultat kontrolleras dock alla objekt (plattformar) utom det som IRST:n sitter på samt det objekt som vi vill upptäcka, vilket innebär att det krävs lite mer beräkningar. Vi vill därför inte göra dessa om terräng eller rök redan skymmer objektet. Kontrollen för



detta kriterium består av att samla en grupp med objekt som består av alla 3D-objekt i simuleringen utom huvudföräldrar, till IRST:n och det sökta objektet. Sedan används grafikmotorns funktioner för att studera om något objekt ligger mellan IRST:n och det sökta objektet på samma sätt som gjordes för terrängen i kriterium 3.

## II.6 Kriterium 6 - Den registrerade intensiteten från objektet måste överstiga bakgrund med en viss faktor (Threshold)

Signalbrusförhållandet ( $S/N$ ) beräknas på följande sätt:

Signal,  $S$ :

$$I_P = I_R A \Omega \approx I_R A \left( \frac{\phi^2 / 2}{4\pi d} 4\pi \right) = I_R A \frac{\phi^2}{2d^2}$$

där

$\Omega$	rymdvinkel (i ovanstående ekvation har små vinklar antagits)
$I_P$	mottagen effekt [W]
$I_R$	radians från målet [ $\text{W m}^{-2} \text{sr}^{-1}$ ]
$A$	area på mål [ $\text{m}^2$ ]
$\phi$	diameter på insamlande optiken [m]
$d$	avstånd [m]

Brus,  $N$

$$P_{RMS} = NEI \frac{\phi^2}{2}$$

där

$P_{RMS}$	bruseffekt [W]
$NEI$	noise equivalent irradiance [ $\text{W m}^{-2}$ ]
$\phi$	diameter på mottagande apertur [m]

Signal till brusförhållande,  $S/N$ , med hänsyn tagen till Beer's lag  $\{\tau = \exp(-\alpha d / 1000)\}$ , där  $\alpha$  är extinctionskoefficient [ $\text{km}^{-1}$ ], och  $d$  är avstånd [m]}:

$$\boxed{\frac{S}{N} = \frac{I_P}{P_{RMS}} \exp\left(-\frac{\alpha d}{1000}\right) = \frac{I_R A}{d^2 NEI} \exp\left(-\frac{\alpha d}{1000}\right)} \quad (1)$$

Strålningsintensiteten,  $I_R A$ , fås ur tabell som funktion av våglängd, IR-signaturindex, azimut- och elevationsvinkel. Våglängd och IR-signaturindex är fixa och kräver inga ytterligare beräkningar, däremot måste vi beräkna vilken azimut- och elevationsvinkel som gäller för tillfället. Vinklarna kan beräknas ur vektorn som går från objektet till IRST:n men de vinklar som avses är relativt ett objektsfast koordinatsystem.

När  $I_R A$  har erhållits görs beräkningar enligt (1). Om  $S/N$  enligt denna beräkning är större än det tröskelvärde, som är ett ingångsvärde till IRST-modulen från NetScene, görs ytterligare beräkningar för att korrigera  $S/N$  så att signalvärdet är signalen över bakgrundsnivån. Detta

görs genom att ”mäta” bakgrunds-nivån i fyra punkter runt målet. Dessa fyra punkter i bakgrunden väljs så att de linjer som går från IRST:n till dessa punkter tangerar en sfär (tangeringspunkter är lodrätt upp och ner samt horisontellt till vänster och till höger från centrum av målet) som precis täcker in hela objektet. Typen av bakgrund kollas för dessa punkter och bakgrunds-nivåer erhålles från *TerrainSignatureData*. Om ingen terräng påträffas på kortare avstånd än 10 km används nivån för *SkySignature*. Bakgrunds-nivån,  $I_{R,background}$  sätts till medelvärdet av de fyra bakgrunds-nivåerna. En korrigeringsfaktor räknas ut,  $k_{Background}$ :

$$k_{Background} = \frac{I_R A - I_{R,background} A_{Background}}{I_R A}$$

där  $A_{Background}$  är den area som IRST:n ser i en pixel. En förenkling är gjord i ovanstående ekvation eftersom denna antar att bakgrunden är på samma avstånd som objektet. I simuleringsmodellen används också avståndet till objektet för att beräkna  $A_{Background}$ :

$$A_{Background} = IFOVH \cdot IFOVV \cdot d^2$$

Det korrigerade signalbrusförhållandet,  $S/N_{Background}$  beräknas som  $S/N \cdot k_{Background}$ .  $S/N_{Background}$  jämförs sedan med samma tröskel som tidigare för att avgöra om upptäckt är möjlig eller inte.

### III Varning

IRST:n kan användas som varnare. I detta fall kontrolleras de, enligt ovan, upptäckta målkandidaterna med avseende på intensitets- och vinkelförändringar. IRST:n skall markera alla målkandidater men bara varna för snabba mål (t ex robotar) som är på väg mot den egna plattformen. En intensitetsökning talar om att ett mål närmar sig och avsaknad av vinkelförändringar talar om att målet är på väg mot den egna plattformen. För att dessa två kriterier skall kunna studeras måste objektet uppfylla upptäcktskriterierna ovan i två konsekutiva tidssteg. Det tar därför minst två rotationsvarv för IRST:n att varna för ett objekt.

Ett mål som har gett en varning kommer inte att resultera i en ny varning förrän målet först har försvunnit ur listan med upptäckta målkandidater under minst *RevolutionsBeforeNewWarning* antal varv.

Alla varningar ligger kvar tills de har blivit efterfrågade av varnar- och motverkanssystemet. Riktningar i varningslistan uppdateras inte under denna tid. Däremot uppdateras vinklar till upptäckta objekt.

## 2.2.4 Beskrivning av parametrar

IRST:n bygger på parametrar från *BaseEntity*, se avsnitt 2.1.1. I IRST:ns id bör *ModulTyp* vara IRST, och dess typ är {entityKind=6, domain=1, countryCode=1, category=1, subcategory=1, specific=1, extra=3}. Typen kan komma att ändras och det rekommenderas att användaren konsulterar aktuell lista med typer för olika system vid uppsättning av simulering. De parametrar som inte kommer från *BaseEntity*, utan är typiska för IRST:n listas i Tabell 4.

**Tabell 4** Beskrivning av IRST:ns parametrar.

Parameternamn	Default-värde	Beskrivning
RelativePosition		Anger position relativt moderobjektet. [m]
WavelengthRange	IR	Går att välja visuell eller IR. Har ingen betydelse för beräkningar men påverkar hur IRST-bilden visas för användaren.
VFOV	5,0	Vertikalt synfält. (Det horisontella synfältet anges inte utan antas vara 360°). [°]
IFOVH	0,001	Horisontellt synfält för ett bildelement. [rad]
IFOVV	0,001	Vertikalt synfält för ett bildelement. [rad]
ScanSpeed	360,0	IRST:ns horisontella avsökningshastighet [°/s]
Stabilize	<i>true</i>	Checkbox som bestämmer om IRST:n är vertikalt stabiliserad eller om den lutar på samma sätt som moderobjektet.
NEI	$0,1 \cdot 10^{-6}$	Noise equivalent irradiance - Vilken effekttäthet som behövs in mot sensorns optik för att motsvara brusnivån. [W/m <sup>2</sup> ]
ExctinctionCoefficient	0,4	Atmosfärens dämpfaktor inom IRST:ns våglängdsområde. [km <sup>-1</sup> ]
Threshold	8,0	Det signalbrus- eller signalklotterförhållande som krävs för upptäckt.
TerrainSignatureData		<p>Signaturdata för olika terrängtyper. Skall vara på formen:</p> <pre>Terrängtyp1  signatur1  beskrivning1 Terrängtyp2  signatur2  beskrivning2 ...</pre> <p>Där <i>Terrängtyp#</i> är ett index som motsvarar olika terrängtyper, <i>signatur#</i> är signatur [W/m<sup>2</sup>/sr] och <i>beskrivning#</i> är en beskrivning av terrängtypen eller någon annan valfri kommentar. Rader som börjar med ! räknas som kommentarrader. Utöver rader med terrängtyp kan det finnas en rad som börjar med ordet <i>default</i> och följs av ett signaturvärde och en eventuell kommentar. Denna signatur används i</p>

		de fall inget signaturvärde finns för terräng-typen. Om inget defaultvärde anges används 0,0 som default.
SkySignature	0,0	Signaturvärde för himmelsbakgrund (eller i de fall då terrängen ligger på längre avstånd än 10000 meter). [ $\text{Wm}^{-2}\text{sr}^{-1}$ ]
RevolutionsBeforeNewWarning	1,5	Om ett upptäckt mål slutar ge varning (p.g.a. att det är dold bakom terräng, annat objekt eller rök, att mottagen intensitet från målet blir för lågt eller att kriterier från varning inte längre är uppfyllda) ges ej ny varning för samma objekt förrän IRST:n har roterat minst det antal varv som anges av denna parameter.
StoNIncreaseThreshold	8,0	<p>Varningar ges bara för objekt som är på väg mot den egna plattformen med tillräckligt hög hastighet. En kontroll av den registrerade intensiteten för det upptäckta objektet svarar på frågan om målet närmar sig tillräckligt snabbt. Denna parameter anger hur hög denna intensitetsökning måste vara i %/s för att en varning skall ges. Under antagandet att objektets signatur inte ändrar sig och att målet inte är upplöst ges intensitetsökningen av:</p> $\Delta \frac{s}{N} = \left( \frac{2}{d} + \frac{\alpha}{1000} \right) v 100 \quad [\% / s],$ <p>där <math>d</math> är avståndet i m, <math>\alpha</math> extinctionskoefficienten i <math>\text{km}^{-1}</math> och <math>v</math> är närmande hastigheten i m/s. Värdet på denna parameter bör väljas så högt att varning inte ges p g a att den egna plattformen kör mot objektet i fråga. Om varning önskas för alla upptäckta objekt (som också uppfyller kriteriet nedan) kan denna parameter sättas till ett negativt värde. [%/s]</p>
MaxDetectionDeltaAngle	$5 \cdot 10^{-3}$	Den maximala registrerade vinkelförändringen per sekund för att objektet skall anses vara på väg mot den egna plattformen och därmed resultera i en varning. Välj värdet på denna parameter $\geq IFOV * 360 / ScanSpeed$ . Ett större värde kan väljas om varning önskas för alla upptäckta objekt (kriteriet ovan måste också vara uppfyllt). [rad/s]

## 2.3 Optikspanare

### 2.3.1 Allmänt om optikspanare

Optiska sikten och sensorer som samlar inkommande strålning, ger vid belysning upphov till en retroreflex [IREO]. Figur 12 visar en retroreflex från en kikare belyst med låg lasereffekt på två km avstånd.



**Figur 12** Retroreflex från en fältkikare på två km avstånd [STRFF].

En optikspanare använder sig av en laserstråle och en sensor som registrerar eventuella retroreflexer inom sitt synfält. Både riktning och avstånd till upptäckta mål kan bestämmas med god noggrannhet. Upptäckta hot kan alltså snabbt störas, exempelvis med bländlaser. Avsökningsmönster kan utformas på olika sätt. I vår modell används en spaltformad laserstråle som avsöker varvet runt på en viss tid. Andra varianter är t ex sektorvis avsökning. Figur 13 visar två exempel på hur system för optikspaning kan se ut.



**Figur 13** Exempel på optikspanare Stingray (källa [FAS]).

### 2.3.2 Optikspanare i EWSim

En optikspanare konfigureras i NetScene och kopplas där till en plattform som sedan styrs i EWSim. I simuleringarna kan optikspanaren användas för att upptäcka objekt som innehåller optik och också fungera som varnare. För upptäckt krävs att den optiska sensorn hos målet ger en tillräckligt stark retroreflex vid belysning med laser samt den inte skymms av terräng, andra objekt eller rök. Dessutom kan man välja att ett objekt måste ha en viss minsta närmandehastighet för att varning ska genereras. Figur 14 visar alla parameterinställningar som kan göras i NetScene. Dessa behandlas också utförligare i avsnitt 2.3.4.

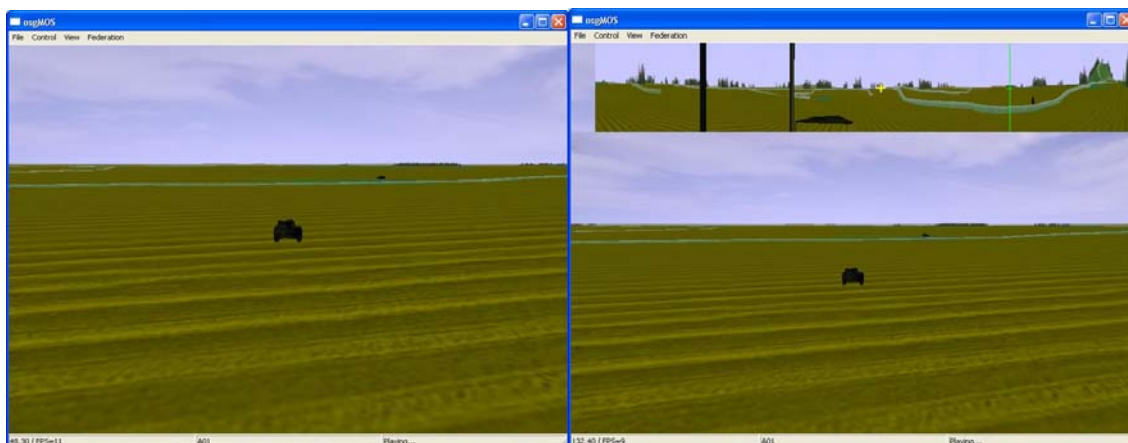
Properties of A01_B_OpticsSurveillance_2	
Properties	
ID	A01_B_OpticsSurveillance_2
Removed	<input type="checkbox"/>
Remote	<input checked="" type="checkbox"/>
Status	Not created
Entity models	[]
Parent	A01_B_FightingVehicle_2#OpticsSurveillanc...
Attributes	
type	{entityKind=6, domain=1, countryCode=1, categ...
position	(3468203.255058401, 780854.729793902, 5...
orientation	(-2.676903486251831, -0.573542594909668...
velocity	(0.008629942312837, -0.059908896684647, ...
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
RelativePosition	(0, 0, 2.5)
ScanSpeed	360.0
Stabilize	<input checked="" type="checkbox"/>
RetroThreshold	-1.0
RevolutionsBeforeNewWarning	2.0
LaserIntensity	2.0
MinApproachSpeed	null

**Figur 14** Parameterinställningar i NetScene.

För att fungera som det är tänkt behöver optikspanaren ha tillgång till en *OpticalSensor*, se avsnitt 2.10. Detta konfigureras i NetScene genom att en optik placeras in med optikspanaren som förälder. Denna sensor ges sedan parametervärden som är lämpliga för att kunna registrera retroreflexer i det tänkta scenariet. Av en optisk sensors parametrar behöver endast *ApertureDiameter* och *Transmission* sättas, resten sköts internt i optikspanaren.

## I Användargränssnitt i EWSim

En optikspanare startas och stoppas genom att den som kontrollerar moderobjektet trycker på tangenten 'h'. Då optikspanaren startas aktiveras en vy överst i bilden som visar vad optikspanaren ser, se Figur 15. Man kan sedan välja om denna vy ska synas eller inte genom att samtidigt trycka på 'Ctrl'+ 'h'.



**Figur 15** Användargränssnitt som visar en översiktsbild från en position snett bakom det kontrollerade fordonet. Fordonet är utrustat med en optikspanare som på den högra bilden är aktiverad. Det gula +:et markerar ett möjligt mål.

I optikspanarvyn visas upptäckt optik som gula +. Optik som genererar varning visas med ett rött + tills varningen är vidarebefordrad till ett varnare- och motverkanssystem (VMS). En

grön vertikal linje markerar moderfordonets orientering i sidled och en horisontell linje på denna markerar moderfordonets orientering i vertikalled.

### 2.3.3 Modellbeskrivning

#### I Skanning

En optikspanare skannar av ett område genom att ett system med laser och sensor tillåts svepa över det totala synfältet på 360° i horisontalled. För att inga luckor ska uppstå under ett varv håller simuleringsmodellen reda på vilket tidssteg,  $\Delta t$ , som används i simuleringen och ställer in ett horisontellt synfält, *HFOV*, som motsvarar detta tidssteg och inställd svephastighet ( $HFOV = ScanSpeed \cdot \Delta t$ ). Bilden från en sensor med detta synfält flyttas över överkanten av bilden allt eftersom tiden passerar och optikspanarens sensor roterar för att täcka in ett synfält på 360°. Bilden som syns på skärmen uppdateras alltså bara en gång per varv precis som för ett verkligt system.

#### II Detektion

För att upptäckt av ett objekt skall kunna ske måste ett antal kriterier vara uppfyllda:

1. Objektet måste ligga i optikspanarens momentana synfält i horisontalled.
2. Objektet måste ligga i optikspanarens vertikala synfält.
3. Objektet får inte vara dolt bakom terrängen, andra objekt eller rök.
4. Den registrerade retroreflexen från objektet måste överstiga en viss tröskelnivå (*RetroThreshold*).

Ett kriterium som står senare i listan kollas bara om tidigare kriterier är uppfyllda. Alla upptäckta målkandidater placeras in i en objektslista hos optikspanaren.

##### II.1 Kriterium 1 - Objektet måste ligga i optikspanarens momentana synfält i horisontalled

Den horisontella riktningen till varje objekt beräknas och jämförs med den nuvarande horisontella riktningen hos optikspanaren. Kriteriet anses uppfyllt om skillnaden är mindre än halva det momentana synfältet.

##### II.2 Kriterium 2 - Objektet måste ligga i optikspanarens vertikala synfält

Riktningen i vertikalled till varje objekt beräknas och jämförs med den vertikala riktningen på optikspanaren. Om optikspanaren inte är stabiliserad i vertikalled går det inte att använda den momentana riktningen på optikspanaren eftersom det simulerade momentana synfältet i horisontalled beror på simuleringens tidssteg. Vid stora tidssteg kan det momentana synfältet också bli stort och därmed kan det bli stora skillnader i vinkeln i vertikalled i centrum av synfältet och i kanterna (i extremfallet när synfältet är 360° i sidled kan t ex optikspanaren vara riktad uppåt i mitten av synfältet men neråt i kanterna). I stället jämför vi optikspanarens uppåtriktning med vektorn som går mellan optikspanaren och objektet.

##### II.3 Kriterium 3 - Objektet får inte vara dolt bakom terrängen, andra objekt eller rök

För att göra en test av detta kriterium skapar vi en vektor med riktning som går från optikspanaren till objektet. Sedan använder vi funktioner som finns inbyggda i grafikmotorn (OpenSceneGraph) för att se hur långt borta terrängen respektive övriga objekt ligger i denna riktning. Om avståndet till objektet är mindre än till terrängen betyder det att målet är synligt. Rökmoln simuleras som ett partikelsystem. För att kontrollera om rök skymmer objektet tittar man på alla partiklars position och radie för att se om någon av dem skymmer målet.

### III Varning

Optikspanaren används ofta som varnare som del i ett varnare- och motverkanssystem (VMS). För att avgöra om varning ska skickas kontrolleras alla upptäckta målkandidater i objektslistan med avseende på positionsförändringar. Optikspanaren skall markera alla målkandidater men bara varna för mål kommer emot den egna plattformen med en närmandehastighet större än *MinApproachSpeed* (t ex robotar). Eftersom skanningen utförs med hjälp av laser fås förutom vinkelangivelse även ett avstånd till målobjektet. Närmandehastigheten kontrolleras med hjälp av förändringen i avstånd till målet mellan två mätningar med en tidsskillnad  $\Delta t$ . Vill man varna för ett objekt placeras detta i en varningslista. Alla varningar ligger kvar tills de har blivit efterfrågade av varnare- och motverkanssystemet. Riktningar och positioner i varningslistan uppdateras inte under tiden. Däremot uppdateras dessa på upptäckta objekt som lagts till i objektslistan.

Ett mål som har gett en varning kommer inte att resultera i en ny varning förrän målet först har försvunnit ur listan med upptäckta målkandidater under minst *RevolutionsBeforeNewWarning* antal varv.

#### 2.3.4 Beskrivning av parametrar

Optikspanaren bygger på parametrar från *BaseEntity*, se avsnitt 2.1.1. I optikspanarens id bör *ModulTyp* vara *OpticsSurveillance*, och dess typ är {entityKind=6, domain=1, countryCode=1, category=1, subcategory=1, specific=1, extra=5}. Typen kan komma att ändras och det rekommenderas att användaren konsulterar aktuell lista med typer för olika system vid uppsättning av simulering. De parametrar som inte kommer från *BaseEntity*, utan är typiska för optikspanaren listas i Tabell 5.

Tabell 5 Beskrivning av optikspanarens parametrar.

Parameternamn	Default-värde	Beskrivning
RelativePosition		Anger position relativt moderobjektet. [m]
ScanSpeed	360,0	Optikspanarens horisontella avsöknings-hastighet. [°/s]
Stabilize	<i>true</i>	Checkbox som bestämmer om optikspanaren är vertikalt stabiliserad eller om den lutar på samma sätt som moderobjektet.
RetroThreshold		Tröskelvärde för mottagen effekt hos retro-reflexen för att varning ska ges. [W]
RevolutionsBeforeNewWarning	2,5	Om ett mål slutar ge varning (för att kriterier för upptäckt eller varning inte längre är uppfyllda) ges ej ny varning för samma objekt förrän optikspanaren har roterat minst det antal varv som anges av denna parameter.
LaserIntensity		Intensiteten hos den utsända laserstrålningen. [W/sr]
MinApproachSpeed	0,0	Minsta närmandehastighet hos ett upptäckt objekt som ger varning. [m/s]



## 2.4 UV-varnare

### 2.4.1 Allmänt om UV-varnare

UV-varnare utnyttjar ett våglängdsintervall runt 0,2-0,3  $\mu\text{m}$ . I detta intervall är transmissionsförhållandena i atmosfären gynnsamma och det finns nästan inget bakgrundsbrus, eftersom ozonskiktet absorberar UV-strålning från solen. En robotflamma emitterar strålning i UV-området, och UV-varnaren kan alltså upptäcka en robot i dess framdrivningsfas. Robotar med lång räckvidd som avfyras bortom fri sikt upptäcks däremot inte. Svetsning eller kraftiga elektriska urladdningar kan ge upphov till UV-signaturer. Dessa kan dock diskrimineras genom signalbehandling, då de till skillnad från en robot inte rör sig. Figur 16 visar ett exempel på hur en UV-varnare kan se ut.



Figur 16 UV-varnare [STRFF].

### 2.4.2 UV-varnare i EWSim

En UV-varnare konfigureras i NetScene och kopplas till en plattform som sedan styrs i EWSim. I simuleringarna kommer UV-varnaren att varna för missiler som är nyligen avfyrade och alltså har en robotflamma.

#### I Beskrivning av parametrar

UV-varnaren bygger på parametrar från *BaseEntity*, se avsnitt 2.1.1. I dess id bör *ModulTyp* vara *UVWarner*, och dess typ är {entityKind=6, domain=1, countryCode=1, category=1, subcategory=1, specific=2, extra=1}. Typen kan komma att ändras och det rekommenderas att användaren konsulterar aktuell lista med typer för olika system vid uppsättning av simulering. Parametrar som inte kommer från *BaseEntity*, utan är typiska för UV-varnaren, omfattar endast *RelativePosition* som bestämmer positionen relativt moderobjektet.

#### II Användargränssnitt i EWSim

En UV-varnare startas och stoppas genom att den som kontrollerar moderobjektet trycker på tangenten 'u'. Det finns ingen display som visar vad UV-varnaren ser, men varningar kommer att skickas om robotar hamnar i siktlinjen inom två sekunder efter avfyrning. För tillfället finns ingen diskriminering för vinkelrörelser, d v s UV-varnaren kontrollerar inte om det upptäckta objektet är på väg mot den egna plattformen eller ej.

### 2.4.3 Modellbeskrivning

#### I Upptäckt

De objekt som kan ge upphov till UV-signaturer är missiler i sin uppstartsfas. För att ett objekt ska upptäckas måste följande kriterier vara uppfyllda:

1. Objektet måste vara en missil som avfyrares för max två sekunder sedan.
2. Objektet får inte vara dolt bakom terrängen eller andra objekt.

##### I.1 Kriterium 1 – Objektet måste vara en missil som avfyrares för max två sekunder sedan

För samtliga objekt i scenen kontrolleras om det är en missil. Om det är det hämtas information om när den avfyrares och jämförs med nuvarande tidpunkt.

##### I.2 Kriterium 2 – Objektet får inte vara dolt bakom terrängen eller andra objekt

För att göra en test av detta kriterium skapar vi en vektor med riktning som går från UV-varnaren till objektet. Sedan använder vi funktioner som finns inbyggda i den grafikmotor som vi använder (OpenSceneGraph) för att se hur långt borta terrängen respektive övriga objekt i scenen ligger i denna riktning. Om avståndet till objektet är mindre än till terrängen betyder det att vi kan se målet.

#### II Varning

UV-varnaren används som del i ett varnare- och motverkanssystem (VMS). Upptäckta målkandidater placeras i en varningslista. Där ligger de kvar tills de blivit hämtade av VMS-logiken. Under tiden uppdateras ingen information i varningslistan.

## 2.5 Laservarnare

### 2.5.1 Allmänt om laservarnare

Laservarnare varnar för inkommande laserstrålning som kan komma från avståndsmätare, laserutpekare och laserledstrålestyrda robotar. Varnare av denna typ har ofta möjlighet att ge information om typ av laser och vinkeln till laserkällan. Figur 17 visar exempel på hur laservarnare kan se ut.



Figur 17 Exempel på laservarnare [STRFF].

### 2.5.2 Laservarnare i EWSim

En laservarnare konfigureras i NetScene och kopplas till en *PhysicalEntity* som sedan styrs i EWSim. Laservarnaren varnar bara för laserstrålar som träffar laservarnaren eller dess plattform.

#### I Beskrivning av parametrar

Laservarnaren bygger på parametrar från *BaseEntity* se avsnitt 2.1.1. I dess id bör *ModulTyp* vara *LaserWarner* och dess typ är {entityKind=9, domain=1, countryCode=0, category=5, subcategory=0, specific=0, extra=0}. Typen kan komma att ändras och det rekommenderas att användaren konsulterar aktuell lista med typer för olika system vid uppsättning av simulering. Parametrar som inte kommer från *BaseEntity*, utan är typiska för laservarnaren, omfattar endast *WarnIfLaserOnPlatform* som bestämmer om lasern måste träffa laservarnarens position eller om det räcker med att träffa den plattform som lasern sitter på.

#### II Användargränssnitt i EWSim

En laservarnare behöver inte startas för att fungera utan fungerar under förutsättning att den konfigureras på ett riktigt sätt.

### 2.5.3 Modellbeskrivning

Laservarnaren kommer med automatik att placeras i centrum av dess förälderobjekt. EWSim ger laservarnaren information om att en laserstråle är skapad och laservarnarmodulen ser efter om lasern träffar varnaren eller dess plattform (laserkällan skall finnas inom line-of-sight från varnaren och riktningen på strålen skall vara rätt). Dessutom kontrolleras om den registrerade lasereffekten överskrider en tröskelnivå (för närvarande satt till 0,002 W, vid behov kan NetScene och EWSim lätt ändras så att denna parameter kan justeras av användaren). Om laserstrålens centrumlinje inte träffar varnaren eller dess plattform studeras även om kanten på laserstrålen träffar eller om reflekterad strålning har sådan intensitet att den kan trigga laservarnaren (reflektionskoefficienten är satt till 0,01). Ingen kontroll görs huruvida laserns våglängdsområde stämmer överens med laservarnarens utan det antas att varnarens våglängdsområde är tillräckligt brett.

## 2.6 Missil

En missil är en styrd projektil där styrning antingen sker med hjälp av en intern målsökare eller genom att fjärrkommandon ges. Den interna målsökaren kan vara helt autonom (en s.k. *fire-and-forget* missil) eller stöttas av t ex en laserutpekare. Kommandostyrningen kan ske på flera sätt t ex via tråd eller genom att missilen befinner sig i en laserledstråle. I IR-fallet finns det i EWSim förberett för ett flertal av dessa missiltyper även om det i skrivandets stund bara är missiler av typen *fire-and-forget* som fungerar fullt ut.

### 2.6.1 IR-missil i EWSim

I en simulering med en målsökande IR-missil skall missilen kunna styra mot ett mål, generera en signatur som kan användas av varnare för att varnarens moderobjekt skall kunna vidta motåtgärder och när motåtgärder är vidtagna skall roboten reagera på ett rimligt sätt. Målsökaren i en *IR-fire-and-forget*-missil i EWSim använder en genererad IR-bild av vad roboten ser för att extrahera en målposition. Denna målposition, som kan vara en verklig målposition eller en position som målsökaren tror är den rätta men som kanske i verkligheten motsvarar facklor eller naturliga falskmål, används sedan för att styra roboten. För att missilen skall kunna interagera med t ex varnare måste roboten generera en egen signatur: för en IR-varnare (t ex av typen IRST) beskrivs robotens IR-signatur och för fallet optikspanare beskrivs målsökarens optiska sensor. Alla dessa bitar kan konfigureras i NetScene i *MissileLauncher*-modulen. Figur 18 visar de parameterinställningar som kan göras i NetScene.

[-] Properties	
ID	A02_R_MissileLauncher_1
Removed	<input type="checkbox"/>
Remote	<input checked="" type="checkbox"/>
Status	Not created
Entity models	[]
Parent	A02_R_Barrel_1#MissileContainer
[-] Attributes	
type	{entityKind=2, domain=0, countryCode=0, category=1, subc...
position	(3468303.86233181, 781614.5991698288, 5277631.387...
orientation	(-2.329997539520264, -0.516315400600433, 2.829647...
velocity	(0.002326423767954, -0.061981953680515, 0.0076001...
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
model	null
radarCrossSectionSignatureInd...	21
IRSignatureIndex	12
IRSignatureData	null
WavelengthRange	IR
ImageSize	128
FOV	2.0
Image	[EWImage];Colorband : GL_RGBA = 0; GL_LUMINANCE ...
Tracker	[EWSeeker];Type : CORRELATION = 0; CENTROID = 1; R...
Motion	[EWMotion];Type : GENERAL = 0; JAVELIN = 1; ACSL = ...
OpticalSensor	[OpticalSensor]ApertureDiameter=0.05Transmission=1.0...
[-] Container Parameters	
relativePosition	(-77.9330464201048, -122.12572434579488, 68.49033...
relativeOrientation	(-1.452648878097534, -1.560869130305946, -3.141592...

Figur 18 Parametrar i NetScene för en *MissileLauncher*.

## 2.6.2 Beskrivning av parametrar

Tabell 6 visar alla parametrar som kan ställas in i NetScene för *MissileLauncher*.

**Tabell 6** Beskrivning av parametrar för *MissileLauncher*.

Parameternamn	Default-värde	Beskrivning
IRSignatureIndex		Detta index överförs på missilen när den avfyras och används t ex när en IRST beräknar om missilen kan upptäckas. Denna parameter finns för alla objekt som är av typen <i>PhysicalEntity</i> .
IRSignatureData		Överförs på missilen när den avfyras. Denna parameter finns för alla objekt som är av typen <i>PhysicalEntity</i> .
WavelengthRange	IR	Bilden som målsökaren använder kan vara en IR eller visuell bild.
ImageSize	128	Anger storleken (antal bildelement) på målsökarens genererade IR-bild (den bild som används av målsökaren för att extrahera målpositioner).
FOV	5	Målsökarbildens synfält. [°]
Image Tracker Motion OpticalSensor		Textbaserad initiering av parametrar i målsökarens bildmodul ( <i>Image</i> ), målsökarens optiska sensor ( <i>OpticalSensor</i> ), målsökaren ( <i>Tracker</i> ) och missilens rörelsedynamik ( <i>Motion</i> ). <i>OpticalSensor</i> beskrivs i separat avsnitt (se avsnitt 2.10), övriga beskrivs nedan.

### I.1 Textbaserad initiering av moduler - allmänt

Parametrar som hör till en modul måste komma efter det aktuella modulnamnet (mellan hakparenteser - [*Modulnamn*]) och innan nästa modulnamn (också det mellan hakparenteser). Rader som börjar med ; räknas som kommentarrader. Värden till parametrar för modulen skrivs in genom att skriva *keyword=värde*. Alla keywords som är tillåtna måste inte finnas med och keywords som inte är tillåtna men ändå finns med ignoreras.

### I.2 Bildmodulen

Bildmodulen läser in en bild som grafikkortet skapat och omvandlar den till rätt format för målsökarmodulen. Modulnamnet skall vara *EWImage*. Tillåtna keywords finns listade i Tabell 7.

**Tabell 7** Beskrivning av bildmodulens parametrar.

Parameter (keyword)	Default-värde	Beskrivning
Enabled	<i>true</i>	
Colorband	2	Bestämmer hur bilden läses in från grafikkortet: 0 = RGBA (den färgbild som visas på datorskärmen)

		1 = luminansbild 2 = röd (den röda komponenten i färgbilden som visas på datorskärmen) 3 = grön 4 = blå 5 = avståndskomponent (avståndsinformation till varje bildelement i bilden)
RadialDepth	<i>true</i>	När avståndskomponenten används kan avståndet (i meter) till varje pixel beräknas eller också kan datorns interna representation användas - noggrannare avstånd (dvs. kortare avståndsintervall) nära än långt bort.
ROI	<i>false</i>	ROI = region of interest. Om <i>true</i> utnyttjas bara den del av målsökarens bildyta för målsökning som anges av <i>ROIx</i> , <i>ROIy</i> , <i>ROIwidth</i> , <i>ROIheight</i> (se nedan).
ROIx	0	Offset från vänsterkanten på bildytan i antal bildelement - bestämmer del av bildyta som utnyttjas för målsökaren.
ROIy	0	Offset från nederkanten på bildytan i antal bildelement - bestämmer del av bildyta som utnyttjas för målsökaren.
ROIwidth	512	Storlek på den del av bildytan som utnyttjas för målsökaren - bredd i antal bildelement. (Kan aldrig bli bredare än den bild som visas).
ROIheight	512	Storlek på den del av bildytan som utnyttjas för målsökaren - höjd i antal bildelement. (Kan aldrig bli högre än den bild som visas).
Save	<i>false</i>	Den visade målsökarbilden kan sparas till disk (innan målsökaren ritar sin information i den).
StartFrame	5	Börjar spara efter att detta antal bilder har visats.
EndFrame	15	Slutar spara efter att detta antal bilder har visats.
Key	<i>false</i>	Det finns stöd för att implementera funktioner för att knapptryckningar skall starta sparning. Antal bilder som sparas blir fortfarande <i>EndFrame - StartFrame</i> .
SaveFilename		Filnamn till vilken bildsekvensen sparas. Filformat som för närvarande stöds är .bmp, .dds, .dx, .jp2, .jpc, .jpeg, .jpg. Om filen redan existerar ändras namnet genom att <i>_#</i> läggs till på slutet, där <i>#</i> är en siffra.
m_bIrScale	<i>false</i>	Används för att omvandla gråskala med 255 nivåer till radiansskala (flyttal) där skalan kan ha två olika lutningar.
m_nBreakPoint	200	Under denna gråskalenivå används lutning 1 över används lutning 2.
m_dLowRad	0	Radiansnivå ut vid 0 intensitet (gråskala) in.
m_dBreakRad	0,5	Radiansnivå ut vid <i>m_nBreakPoint</i> intensitet (gråskala) in.
m_dHighRad	5,1	Radiansnivå vid 255 intensitet (gråskala) in.

Exempel på initieringsdata:

```
[EWImage]
;Colorband      : GL_RGBA = 0; GL_LUMINANCE = 1; GL_RED = 2;
;               GL_GREEN = 3; GL_BLUE = 4
Enabled         =true
Colorband       =2
ROI             =false
ROIx            =0
ROIy            =0
ROIwidth        =512
ROIheight       =512
Save            =true
StartFrame      =5
EndFrame        =15
Key             =false
SaveFilename    =C:\test.bmp
Convfact        =false
fConvfact       =1500.000000
m_bIrScale      =false
m_nBreakPoint   =200
m_dLowRad       =0
m_dBreakRad     =0.5
m_dHighRad      =5.1
```

### I.3 Målsökarmodulen

Initieringsdata till målsökarmodulen kan innehålla flera modulnamn (ett för huvudmodulen som bestämmer vilken typ av målsökare som skall användas och ett för varje typ av målsökare som kan vara aktuell). Om roboten inte byter typ av målsökare under simuleringens gång påverkar bara två moduler simuleringen - för huvudmodulen och för den aktuella målsökaren. Däremot kan det för att underlätta modifiering av ett scenario vara värt att definiera flera olika målsökartyper i samma textmassa (målsökartypen kan sedan lätt bytas genom att ändra värdet på *Type*-parametern under modulnamnet *EWSeeker*, se nedan).

**Tabell 8** Beskrivning av parametrar under modulnamnet *EWSeeker*.

Parameter (keyword)	Defaultvärde	Beskrivning
Type	1	Typ av målföljare: 0 = korrelation 1 = centroid 2 = retikel 3 = rosett 4 = kryssdetektor 5 = fjärrstyrd

**Tabell 9** Beskrivning av parametrar under modulnamnet *CorrelationSeeker* (se [CORR] för mer detaljer).

Parameter (keyword)	Default-värde	Beskrivning
Enabled	<i>true</i>	
m_nCorrelationMode	2	<p>Korrelationsmode (hur korrelationen räknas ut). <math>I</math> är intensiteten från bildelement i bilden, <math>R</math> är bildelement från referensbilden, <math>\bar{X}</math> betecknar medelvärde av <math>X</math>.</p> <p>0 = <math>I \cdot R</math></p> <p>1 = <math>\frac{I \cdot R}{\sqrt{\sum I^2}}</math></p> <p>2 = <math>\frac{I \cdot R}{\sqrt{\sum I^2} \cdot \sqrt{\sum R^2}}</math></p> <p>3 = <math>\frac{(I - \bar{I}) \cdot (R - \bar{R})}{\sqrt{\sum (I - \bar{I})^2} \cdot \sqrt{\sum (R - \bar{R})^2}}</math></p> <p>4 = <math> I - R </math></p> <p>5 = <math>(I - R)^2</math></p>
m_dLambda	0,9	Anger hur mycket referensbilden uppdateras. (0,0 = helt nytt, 1,0 = ingen uppdatering).
m_nShift	0	<p>Anger vilken färg i en färgbild (RGB) som utnyttjas (används bara om <i>Colorband</i> är satt till RGBA i bildmodulen).</p> <p>0 = röd</p> <p>1 = grön</p> <p>2 = blå</p>
m_nFrameUpdate	1	Anger hur ofta referensbilden uppdateras. (1 = varje bild, 2 = varannan, osv)
m_nKernelSize	9	Storlek på referensbilden i bildelement (= m_nKernelSize x m_nKernelSize).
m_nRoi	200	Målsökaren söker bara i ett område som är m_nRoi x m_nRoi stort och centrerat kring senast beräknad målposition.
m_bSaveOriginal	<i>true</i>	Sparar en bildsekvens med ingångsbilder till målsökaren.
m_strOriginalFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).
m_bSaveCorrelation	<i>false</i>	Sparar en bildsekvens med korrelationsbilder.



m_strCorrelationFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).
m_bSaveKernel	<i>false</i>	Sparar en bildsekvens med referensbilder.
m_strKernel-Filename		Namnet på sparad sekvens (bör sluta på .pic eller .des).

**Tabell 10** Parametrar under modulnamnet *CentroidSeeker* (se [CEN] för mer detaljer).

Parameter (keyword)	Default-värde	Beskrivning
Enabled	<i>true</i>	
TrackMode	1	Målsökarens följemode: 0 = surveillance 1 = centroid 2 = edge 3 = correlation 5 = land 9 = ship 10 = dead
IntegrationMode	0	Integrationsmode: 0 = area balance 1 = center of mass 2 = intensitet
TargetContrastMode	1	0 = mörk, 1 = ljus
ECCM	<i>false</i>	En enkel motmotmedelsfunktion.
SaveOriginal	<i>false</i>	Sparar en bildsekvens med ingångsbilder till målsökaren.
SaveOriginalFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).
SaveAgc	<i>false</i>	Sparar en bildsekvens med bilder som är processade med en AGC (automatic gain control) funktion.
SaveAgcFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).
SaveThreshold	<i>false</i>	Sparar en bildsekvens med bilder (AGC) som är processade med en tröskelnivå.
SaveThresholdFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).

**Tabell 11** Parametrar under modulnamnet *ReticleSeeker* (se referens [RET] för mer detaljer).

Parameter (keyword)	Default-värde	Beskrivning
Enabled	<i>true</i>	
fRps	100,0	Antal varv retikeln snurrar (eller nuterar) per sekund. [Hz]
dTimeConstant	0,001	Används till en AGC (automatic gain control) funktion [s]
fMaxSteer	1,0	Maximal beräknad felvinkel (1,0 motsvarar halva synfältet).
nAngles	256	Antal vinkelsteg som används när retikelsignalen skapas för ett retikelvarv.
fOffsetAngle	0,0	Målposition ges av två vinklar, dels en felvinkel och dels en polärvinkel (där 0 är till höger, $\pi/2$ (90°) är upp, $\pi$ (180°) är till vänster och $3\pi/2$ (270°) är ner). Signalbehandlingen kan resultera i att en offset måste läggas till för att resultatet skall bli rätt.
bNutate	<i>true</i>	Anger om retikeln är nuterande.
bRotate	<i>false</i>	Anger om retikeln är roterande (ignoreras om bNutate är true).
bUseFFT	<i>true</i>	Bestämmer om FFT används för att beräkna resultatet.
fNutationRadius	0,5	Radien på nutations-varvet. 1,0 motsvarar halva synfältet.
fBandPass1CenterFrequency	1000,0	Filterfrekvens vid signalbehandling. [Hz]
fBandPass1BandWidth	500,0	Filterfrekvens vid signalbehandling. [Hz]
fHighPassFrequency	0,0	Filterfrekvens vid signalbehandling. [Hz]
fLowPassFrequency	200,0	Filterfrekvens vid signalbehandling. [Hz]
fBandPass2CenterFrequency	100,0	Filterfrekvens vid signalbehandling. [Hz]
fBandPass2BandWidth	20,0	Filterfrekvens vid signalbehandling. [Hz]
m_nShift	0	Anger vilken färg i en färgbild (RGB) som utnyttjas (används bara om <i>Colorband</i> är satt till RGBA i bildmodulen).  0 = röd 1 = grön 2 = blå
m_bFixImageSize	<i>false</i>	Lås storleken på den behandlade bilden genom att vid behov fylla på eller dra ifrån

		en ram kring ingångsbilden från <i>Image</i> -modulen.
m_nFixedImageSizeX m_nFixedImageSizeY	128	Storlek på behandlad bild i horisontal- och vertikalled om <i>m_bFixImageSize</i> är satt till <i>true</i> .
nReticleType	0	Typ av retikel (retikeln skapas automatiskt, se nedan, eller läses från fil).  0 = Rising sun 1 = FM1 2 = NutAm1 3 = retikeln läses från fil
m_strReticleImageFilename		Filnamn på retikelbild vid retikeltyp 3.
nSpokes	7	Antal ekrar när retikeltyp 0-2 skapas, se ovan.
fRadius	0,9	Radien på retikeln som skapas (retikeltyp 0-2). 1,0 motsvarar halva synfältet. Observera att radien bör vara så liten att retikelbilden inte sticker utanför synfältet. För en nuterande retikel innebär detta att nutationsradien + denna radie inte bör överstiga 1,0.
fInnerRadius	0,224	Används för att skapa retikeltyp 1.
fRadiusOfInnerStructure	0,069	Används för att skapa retikeltyp 1.
fDistanceFromCenterToInnerStructure	0,138	Används för att skapa retikeltyp 1.
nRadialFields	3	Används för att skapa retikeltyp 2. Måste vara 3 eller mindre.
nSpokesRadialFields1	2	Används för att skapa retikeltyp 2.
nSpokesRadialFields2	4	Används för att skapa retikeltyp 2.
nSpokesRadialFields3	8	Används för att skapa retikeltyp 2.
m_bDrawReticleImage	<i>true</i>	En bild på retikeln överlagras den visade målsökarbilden (visas endast efter pålåsning men innan avfyring).
m_bDrawSignalBeforeFilters	<i>true</i>	Ritar retikelsignalen i målsökarbilden (efter pålåsning).
m_bSaveOriginal	<i>false</i>	Sparar en bildsekvens med ingångsbilder till målsökaren.
m_strOriginalFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).
m_bSaveResult	<i>false</i>	Sparar en bildsekvens med resultatbilder. När FFT och nuterande retikel används är resultatbilden = originalbilden faltad med retikelbilden. Utan FFT sparas en

		bildsekvens där man ser originalbilden maskad med retikelbilden som roteras eller nuteras. Vid FFT och rotation sparas ingen bildsekvens.
m_strResultFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).
m_bSaveIntensityBeforeFilter	<i>false</i>	Det är förberett för att spara en bildsekvens med retikelsignaler.
m_strIntensityBeforeFilterFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).
m_bSaveIntensityAfterFilter	<i>false</i>	Det är förberett för att spara en bildsekvens med retikelsignaler.
m_strIntensityAfterFilterFilename		Namnet på sparad sekvens (bör sluta på .pic eller .des).

**Tabell 12** Parametrar under modulnamnet *RemoteSeeker*.

<b>Parameter (keyword)</b>	<b>Default-värde</b>	<b>Beskrivning</b>
Enabled	<i>true</i>	
TrackMode	0	0 = tråd 1 = laserledstråle 2 = ostyrd

## 2.7 Facklor

### 2.7.1 Allmänt om facklor

Facklor används för att störa optroniska målsökare så att missilen som styrs av målsökaren ska missa målet. Facklor fungerar genom att de avger en stor mängd strålningseffekt i det våglängdsband som missilens målsökare är känslig inom. Om målsökaren styr mot det starkast lysande målet, kommer den därmed att styra mot facklan istället för målet. Det är viktigt att för en given missil ha rätt typ av fackla då olika missiler kan vara känsliga inom olika våglängdsband.

#### I Flygfallet

Facklor i flygfallet finns eller kommer att finnas till HERKULES, JAS, och den framtida helikopter 14. Facklorna är viktiga för att skydda sig mot t ex manburna luftvärnssystem s k MANPADS t.ex. SA-7 b. Fackelfällningen kombineras ibland med undanmanöver för att uppnå resultat.

#### II Sjöfallet

I sjöfallet så genererar facklorna i allmänhet stora volymer och kan vara mycket lika rök d v s de kan fungera genom att avskärma målet från målsökaren så att målsökaren inte kan se målet.

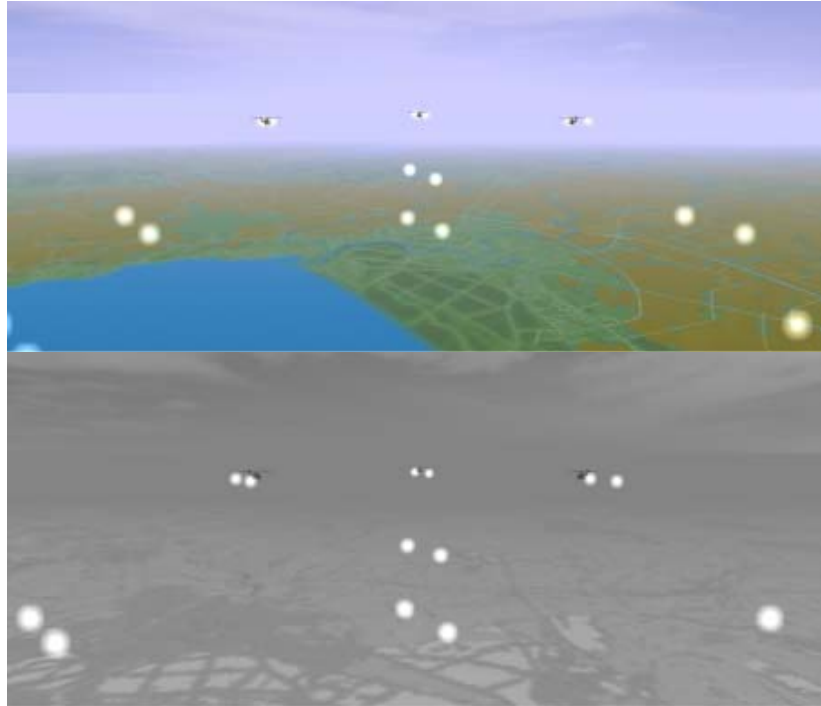
### 2.7.2 Fackelfällning i EWSim

Fackelfällningen konfigureras med en fil eller text som anges i NetScene och kopplas till ett markfordon, en flygande plattform eller till ett fartyg som kan styras i EWSim. När ett objekt har ett fackelfällarsystem kan detta aktiveras i EWSim genom att den som styr moderobjektet trycker på tangenten 'o' varvid fackelfällningsprogram nummer ett aktiveras, se t ex Figur 19. Alternativt kan facklorna i modellen kastas ut automatiskt via VMS-modulen (se avsnitt 2.11) när varning erhålls.

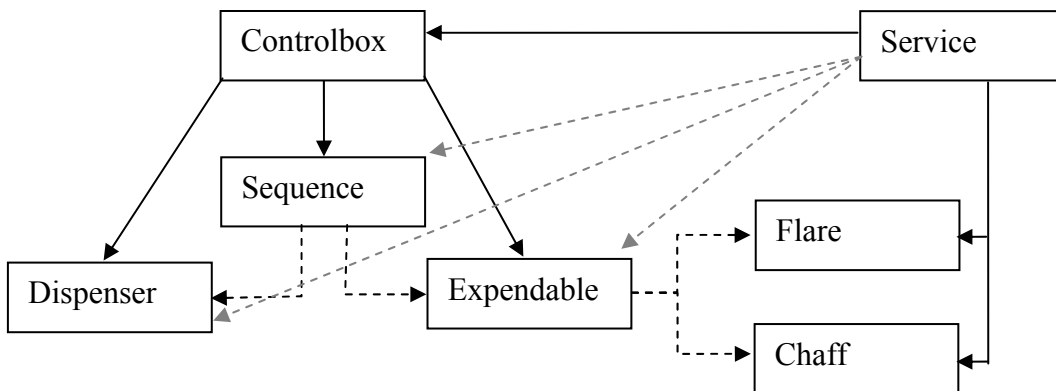
### 2.7.3 Modellbeskrivning

Fackelfällningen simuleras av två delar där den ena genererar data för hur facklorna ser ut vid ett antal tidpunkter. Dessa data tas sedan om hand i den andra delen och placeras ut i 3D-scenen så att det genereras facklor i IR och den visuella bilden. De fysikaliska formler som använd finns i [EWSIM].

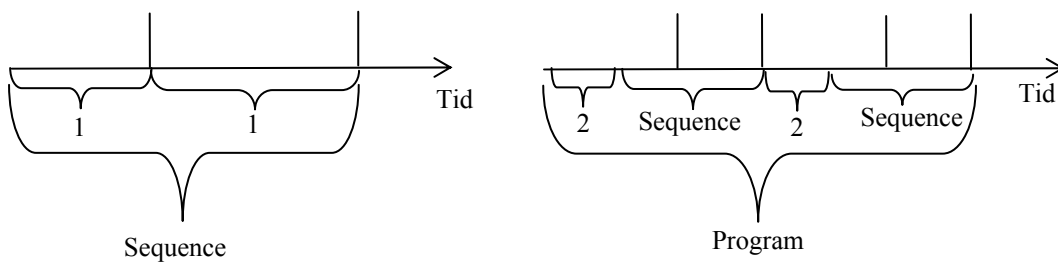
Fällningsprogrammet konfigureras genom att ett antal konfigurerbara objekt kombineras. Dessa objekt med relationer finns i Figur 20. Det finns ett objekt som har tillgång till alla de andra objekten kallat *Service*. När facklorna ska kastas ut kommer *Service*-modulen att ange vilket motmedelsprogram som ska köras och hur många gånger. Ett fällprogram är uppbyggt av att man kör en viss sekvens av fällning ett antal gånger. Detta program ligger i *Controlbox*-objektet. *Sequence*-objektet anger hur en sekvens av facklor (eller remsor) ska fällas i tid samt från vilka fackelfällare. En *Expendable* består av en eller flera subladdningar vilket i flygfallet oftast är en. *Flare* konfigureras med data om hur en fackla brinner spatialt, temporalt och spektralt. *Chaff* beskrivs inte i denna rapport då det hör till radarområdet.



**Figur 19** Fackelfällning, överst en visuell bild och underst en IR-bild av hur tre helikoptrar faller facklor.



**Figur 20** De moduler som kan konfigureras och tillsammans skapa ett fällprogram.



**Figur 21** En *Sequence* är uppbyggd av ett antal tider mellan fackelfällningar samt en initial tid. Programmet är uppbyggt av ett antal sekvenser. Siffrorna i figuren refererar till parameterbeskrivningen.

## 2.7.4 Beskrivning av parametrar

Tabell 13 - Tabell 18 visar de parameterinställningar för fackelfällning som kan göras i NetScene. Alla modulkonfigurationer har ett id angivet efter respektive modulnamn. Detta index refereras till från andra modulkonfigurationer.

**Tabell 13** Beskrivning av parametrar för *Service*.

Parameternamn	Default-värde	Beskrivning
NumberOfControlBoxs	0	Antal <i>Controlbox</i> -konfigurationer.
NumberOfFlares	0	Antal <i>Flare</i> -konfigurationer.
NumberOfChaffs	0	Antal <i>Chaff</i> -konfigurationer.

**Tabell 14** Beskrivning av parametrar för *Controlbox*.

Parameternamn	Default-värde	Beskrivning
m_switchProgram	1	Anger om programmet ska köras manuellt = 0, hel = 2 eller halv-automatiskt = 1.
NumberOfPrograms	0	Anger hur många program som finns i sekvensen. Dessa kommer sedan att räknas upp i en indexerad lista, se nästa parameter.
m_programTable_seq [n]	0	Anger <i>Sequence</i> -id som går i programmet med n som löpnummer.
m_programTable_num [n]	0	Hur många gånger sekvensen skall köras i programmet n.
m_programTable_tIntervall [index] (2 i figur)	0	Tidsintervall mellan körning av sekvenser. [s]
m_programTable_totTid	0	Total tid för att köra programmet n. [s]
m_programTable_num_expendable 0..7	0	Anger hur mycket av <i>Expendable</i> 0-7 som går åt för att köra programmet n.

**Tabell 15** Beskrivning av parametrar för *Sequence*.

Parameternamn	Default-värde	Beskrivning
NumberOfRows	0	Anger hur många <i>Expendable</i> :s som finns i sekvensen. Dessa kommer sedan att räknas upp i en indexerad lista, se nästa parameter.
m_FunkTable_time [n] (1 i figur)	0,0	Fälltid för <i>Expendable</i> med givet index. [s]
m_FunkTable_expendableid [n]	0	Refererar till ett <i>Expendable</i> -id.
m_FunkTable_Dispenserid [n]	0	Refererar till ett <i>Dispenser</i> -id.
m_TrigDelayTime	0,0	Tid tills sekvensen kan startas. [s]

**Tabell 16** Beskrivning av parametrar för *Dispenser*.

Parameternamn	Default-värde	Beskrivning
m_dir_theta	0,0	Anger kastarens riktning i eulervinkel theta i plattformens koordinater.
m_dir_phi	0,0	Anger kastarens riktning i eulervinkel phi i plattformens koordinater.
m_posx	0,0	Anger kastarens x-position i plattformens koordinater. [m]
m_posy	0,0	Anger kastarens y-position i plattformens koordinater. [m]
m_posz	0,0	Anger kastarens z position i plattformens koordinater. [m]
NumberOfRowsFlare	0	Anger hur många typer av <i>Expendable</i> med <i>Flare</i> i som kastaren har. Dessa kommer sedan att räknas upp i en indexerad lista, se nästa parameter.
m_LoadTableFlareNum [n]	0	Lista med antalet <i>Expendable</i> med <i>Flare</i> i där n anger typen och parametern antalet va den typen.
NumberOfRowsChaff		Anger hur många typer av <i>Expendable</i> med <i>Chaff</i> i som kastaren har dessa kommer sedan att räknas upp i en indexerad lista se nästa parameter.
m_LoadTableChaffNum [n]		Lista med antalet <i>Expendable</i> med <i>Chaff</i> i där index anger typen och parametern antalet.



**Tabell 17** Beskrivning av parametrar för *Expendable*.

Parameternamn	Default-värde	Beskrivning
NumberOfRows	0	Anger hur många <i>Chaff</i> och/eller <i>Flare</i> som finns i en <i>Expendable</i> . Dessa kommer sedan att räknas upp i en indexerad lista, se nästa parameter.
m_FunkTable_time [n]	0,0	Tid tills att en <i>Chaff</i> eller <i>Flare</i> skall explodera efter att <i>Expendable</i> :n har skjutits ut för ett givet index. [s]
m_FunkTable_flareid [n]	-1	Refererar till ett <i>Flare</i> -konfigurationsid.
m_FunkTable_chaffid [n]	-1	Refererar till ett <i>Chaff</i> -konfigurationsid.
m_FunkTable_restitution-Koefficient	0	Överföringskonstant för rörelseenergi mellan <i>Expendable</i> och <i>Flare</i> eller <i>Chaff</i> .
m_Aerodyn_area	0	Aerodynamisk area. [m <sup>2</sup> ]
m_Aerodyn_Cd	0	Luftmotståndskoefficient.
m_Aerodyn_mass		Massa. [kg]
m_Aerodyn_stid		Tid tills <i>Expendable</i> :n försvinner efter avfyrning. [s]
m_Aerodyn_vpoints		Antal punkter som beräknas under en fällning. I varje punkt beräknas position.

**Tabell 18** Beskrivning av parametrar för *Flare*.

Parameternamn	Default-värde	Beskrivning
m_name		Filnamn eller sträng som ger intensitet och storlek som funktion av tiden.
m_Aerodyn_area	0	Aerodynamisk area. [m <sup>2</sup> ]
m_Aerodyn_Cd	0	Luftmotståndskoefficient.
m_Aerodyn_mass	0	Massa. [kg]
m_Aerodyn_stid		Tid tills facklan försvinner efter avfyrning. [s]
m_Aerodyn_vpoints		Antal punkter som beräknas under en

		fällning. I varje punkt beräknas position storlek och intensitet.
m_Aerodyn_rho	0	Luftdensitet. [kg/m <sup>3</sup> ]
m_Aerodyn_zspeed	0	Om nästa variabel är satt till true så är detta fallhastigheten. [m/s]
m_Aerodyn_esign	0	Parameter som väljer om fallhastigheten ska beräknas med aerodynamiska ekvationer eller till den givna hastigheten i föregående variabel.

För *Chaff*-modulen (remsor) finns liknande parametrar som för facklor men med RCS-information (Radar Cross Section) om remsor istället för facklor. Detta beskrivs inte i denna rapport då det ej hör till EO-området utan radar.

## 2.8 Rök

### 2.8.1 Allmänt om rök

Rök är ett motmedel som kan användas för att dölja och kamouflera mål så att de ej kan upptäckas eller identifieras av optroniska sensorer. Det finns olika typer av rök, dels rök som är avsedd att täcka in stora områden, oftast inte verksam i det infraröda våglängdsområdet, då rökpartiklarna är för små, dels finns det rök som främst är för egenskydd och döljer målet under en kortare stund. Rök som skall förvillra målsökare bör ha olika interna kontraster så att målsökaren inte kan in visas genom att styra mot rökens centrumpunkt.

#### I Markfallet



**Figur 22** Rökkastning från Strf 9040C.

Rökkastare finns t ex på Strf 9040C (se Figur 22) och Strv122. Att ha kastare som är rörliga i förhållande till tornet gör att röken kan läggas i rätt riktning i förhållande till hot och vind.

#### II Sjöfallet

Rök används till ett flertal svenska fartyg och en variant som fungerar bra i det visuella är dieslrök som dock inte fungerar i infrarött.

### 2.8.2 Rök i EWSim

Rök kan konfigureras i NetScene och kopplas till ett markfordon, en flygande plattform eller till ett fartyg som kan styras i EWSim. Rök kan i modellen användas som motåtgärd som aktiveras automatiskt eller manuellt. När ett objekt har ett röksystem kan detta aktiveras manuellt i EWSim genom att den som styr moderobjektet trycker på tangenten 'e'. I detta fall aktiveras röken och syns i bilden. Röken kan även aktiveras automatiskt av VMS-modulen, se avsnitt 2.11.

### 2.8.3 Modellbeskrivning

Röken simuleras genom att ett system med partiklar bestående av texturer med transparens-lager för respektive våglängdsband placeras ut i 3D-scenen. Placeringen i scenen sker genom att radien på rökmolnet sätts och partiklar placeras därefter ut slumpmässigt inom denna sfär under den tid som rökmolnet utvecklas. Efter att en partikel placerats ut rör sig denna dels med vinden, dels med ett litet slumpmässigt avstånd som adderas för varje tidsintervall samt att den faller neråt med en depositions-hastighet. Depositionshastigheten beräknas m h a luftens viskositet samt partikelstorleken och massan.

## 2.8.4 Beskrivning av parametrar

Tabell 19 visar de parameterinställningar för rök som kan göras i NetScene.

**Tabell 19** Beskrivning av parametrar för rök, utöver de som finns i basklassen för *PhysicalEntity*.

Parameternamn	Defaultvärde	Beskrivning
Time to explosion	1,0	Tid till avfyrning av rök från rökkastare. [s]
TimeToDisappear	20,0	Tid tills röken försvinner. [s]
TimeToDevelop	5,0	Tid tills röken är fullt utvecklad. [s]
Wind	1,0	Vindhastighet. [m/s]
WindDirection	0	Vindriktning. [grader]
RelativePosition	{5,5,5}	Anger rökens position relativt moderobjektet som en vektor. [m]
DispenserFixedToParentObject	<i>true</i>	Anger om kastare är fixerade eller roterbara.

## 2.9 DIRCM

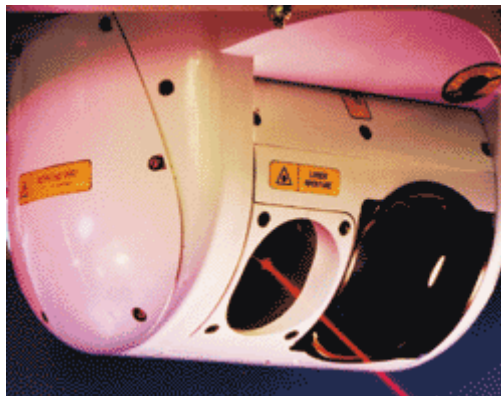
### 2.9.1 Allmänt om DIRCM

Riktade lasermotmedel (DIRCM – Directed Infrared Countermeasures) får en allt mer framträdande roll inom telekrigföring i skyddet av den egna plattformen mot hot inom IR-området. Genom att belysa den angripande roboten med riktad strålning av hög intensitet (laser) kan ett DIRCM-system störa ut robotens IR-målsökare genom vilseledning (blinkstörning), bländning eller förstörande verkan.

Ett av dagens största hot i olika typer av konflikter utgörs av manburna robotsystem, såsom MANPADS (MAN Portable Air Defense Systems). Internationellt pågår stora forskningsinsatser att utveckla DIRCM-system för att skydda framför allt flygplan och helikoptrar från detta hot. Denna typ av riktade motmedel kan även komma att spela en viktig roll för civilt flyg, som är sårbart mot terroristattacker nära flygplatser vid start och landning [OM].

Mer traditionella motmedelsinsatser som facklor kan med framgång användas mot den äldre generationens IR-målsökare, såsom retikelmålsökare. Den nya generationens målsökare kan dock ofta diskriminera facklor genom att använda flera våglängdsband eller avancerad bildbehandling. I detta sammanhang kommer DIRCM att spela en viktig roll, i bekämpningen av morgondagens mer avancerade hot.

I dagsläget finns ett DIRCM-system i produktion för militärt bruk, Northrop Grummans AN/AAQ-24(V) Nemesis, som används av både Storbritannien och USA för skydd av flygande plattformar [NEMESIS] (se Figur 23).



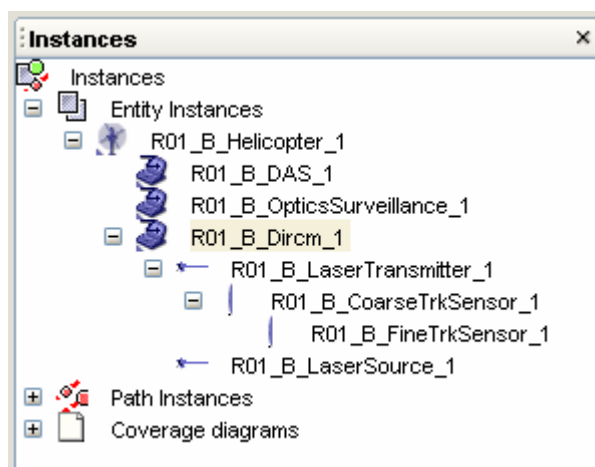
**Figur 23** Nemesis, ett DIRCM-system utvecklat av Northrop Grumman.

### 2.9.2 DIRCM i EWSim

DIRCM konfigureras i NetScene och kopplas där till en valfri plattform som ska skyddas, på land, till sjöss eller i luften. Plattformen behöver dessutom kompletteras med varningssystem för att kunna upptäcka och sortera bland de hot som dyker upp. Parametrarna för DIRCM beskrivs i avsnitt 2.9.4.

Figur 24 visar strukturen för en typisk DIRCM-konfiguration. Helikoptern (R01\_B\_Helicopter\_1) har ett DIRCM-system (R01\_B\_Dircm\_1) som underobjekt. Under DIRCM-objektet finns en laserkälla (R01\_B\_LaserSource\_1) samt en lasertransmitter (R01\_B\_LaserTransmitter\_1) som anger riktningen i vilken laserstrålningen sänds ut. Lasertransmittern har i

detta fall två optiska sensorer (R01\_B\_CoarseTrkSensor\_1 och R01\_B\_FineTrkSensor\_1) som används vid inriktningen av lasern. Det är m h a dessa som hotet observeras, både passivt (IR) och aktivt (genom retroreflexen i hotsensorn). Vill man bara följa hotet passivt kan man ta bort optiken för finföljning.



**Figur 24** Strukturen för ett DIRCM-system, konfigurerat i NetScene.

Under simulering i EWSim handlar DIRCM i stort sett autonomt; varningssystemen upptäcker inkommande hot och aktiverar DIRCM-systemet som därvid automatiskt påbörjar störning. Genom att trycka 'x' under simulering kan denna funktionalitet slås av och på, vilket kan liknas vid en stor strömbrytare för hela systemet. Hur en IR-målsökare störs av laserstrålningen från ett DIRCM-system demonstreras i avsnitt 3.4, Figur 66. Den störda målsökarvyn visas under den ostörda.

### 2.9.3 Modellbeskrivning

Detta avsnitt ger en kort översikt av DIRCM-modellen som är implementerad i EWSim. En utförligare modellbeskrivning återfinns i en tidigare rapport utgiven på FOI [DIRCM].

Så snart varningssystemet upptäcker ett hot och beslutar om motåtgärd i form av laserstörning riktar DIRCM-systemet automatiskt in sig i hotets riktning och aktiverar lasern. Till en början sker styrning mot den hotriktning som tillhandahölls av varningssystemet. När inriktningen är tillräckligt god aktiveras lasern. Om invisningen är tillräckligt noggrann kommer lasern omedelbart att träffa det angripande hotet.

I de allra flesta fall utgörs hotet av en IR-sökande robot, utrustad med optik för att följa sitt mål. Optiken spelar en viktig roll även för den försvarande sidan. Då ett optiskt system belyses med laser ger den ifrån sig en kraftig signatur till följd av retroreflexen [DIRCM], som kan användas för att DIRCM-systemet noggrannare ska kunna följa hotet (aktiv följning). Dynamiken för laserns pekriktning beskrivs av reglerade vinkelservon [DIRCM].

Hur störd en målsökare blir av laserbelysningen beror dels på optikens egenskaper och dels på målsökarens signalbehandling. Den viktigaste egenskapen hos optiken i detta sammanhang är ströljus [DIRCM][STRÖ] som först och främst har en bländande effekt. Ett vanligt exempel på hur man kan störa signalbehandlingen i en målsökare är blinkstörning mot retikel. Optiken är mycket viktig vid laserstörning och beskrivs under ett särskilt avsnitt, 2.10.

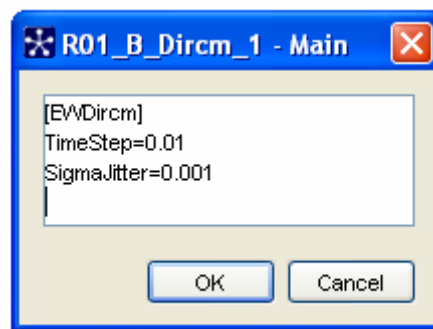
Sammanfattningsvis: DIRCM-systemet arbetar autonomt för att störa ut optiska hot. Till stöd finns på plattformen ett varningssystem för att upptäcka hot och koordinera bekämpningen av dessa. För att ett hot ska kunna bekämpas med DIRCM krävs att det har en optik.

#### 2.9.4 Beskrivning av parametrar

Parametrarna för DIRCM sätt i NetScene i form av textsträngar med parametrarnas namn kopplat till dess värde. Hur textsträngarna ser ut framgår i beskrivningen för varje delenheter nedan (jmf. också avsnitt 2.6.2). Utelämnas en given parameter blir den defaultinitierad, d.v.s. den ges ett förbestämt värde. Dessa värden ges i tabellerna nedan.

##### I DIRCM (huvudenhet)

Huvudenheten i DIRCM-systemet (R01\_B\_Dircm\_1 i Figur 24) innehåller ett fåtal övergripande parametrar som beskrivs i Tabell 20. Parametrarna anges i en textsträng i NetScene och ska läggas under modulnamnet [*EWDircm*] enligt Figur 25.



**Figur 25** Parametrar för DIRCM anges under modulnamnet [*EWDircm*].

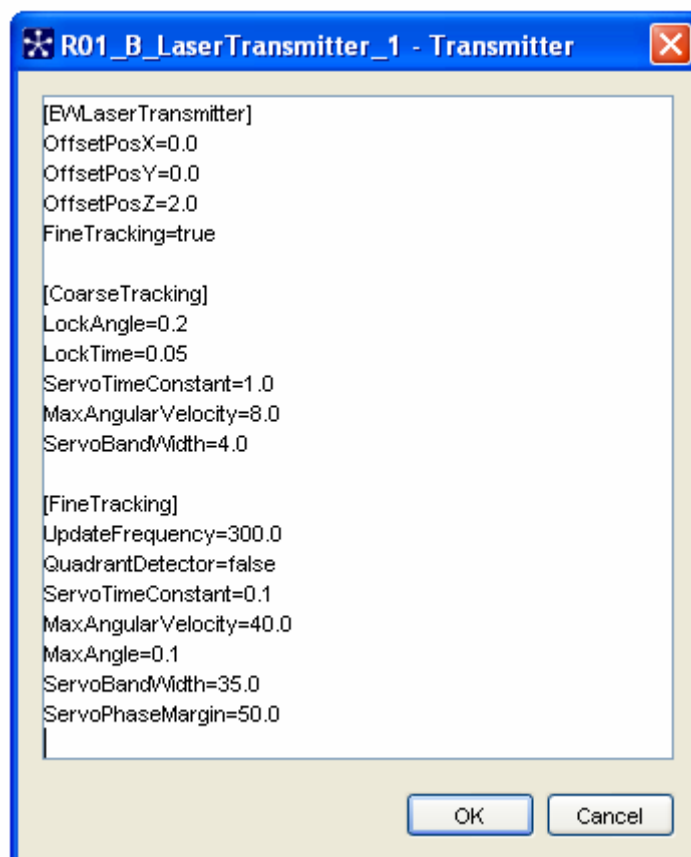
**Tabell 20** Beskrivning av övergripande parametrar för DIRCM.

Parameter (keyword)	Default-värde	Beskrivning
TimeStep	0,05	Det tidssteg som DIRCM-systemets övergripande funktioner arbetar med. I början av varje nytt tidssteg sker en uppdatering av allt som inte har med själva inriktningen av lasern att göra, såsom prioritering och val av hot samt kommunikation med varningssystemen. [s]
SigmaJitter	0,0	Beskriver vibrationerna som påverkar DIRCM-systemet. Dessa härstammar vanligen från plattformen och beskrivs som ett vinkeljitter. Värdet anger standardavvikelsen för jittret som är gaussiskt fördelat. [rad]

##### II Lasertransmitter

Parametrarna för lasertransmittern är indelade i tre delgrupper. Den första anges med modulnamnet [*EWLaserTransmitter*] och innehåller parametrar av övergripande karaktär. De andra två parametergrupperna är kopplade till respektive följesystem; grovföljning [*Coarse-Tracking*] och finföljning [*FineTracking*] (se Figur 26). Notera att parametrarna för finföljning kan utelämnas om DIRCM-systemet inte är utrustat med detta. I Tabell 21 ges två

defaultvärden för en given parameter i de fall den kan anges för både grov- och finföljning. Det vänstra värdet svarar då mot grovföljning och det högra mot finföljning.



**Figur 26** Parametrar för lasertransmittern anges under modulnamnen [*EWLaserTransmitter*], [*CoarseTracking*] och [*FineTracking*].

**Tabell 21** Beskrivning av parametrar för lasertransmittern.

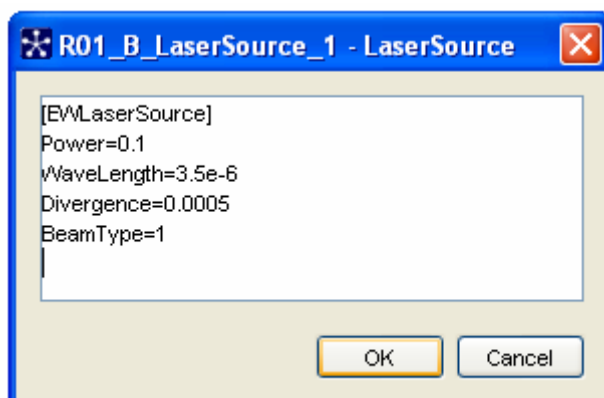
Parameter (keyword)	Default-värde	Beskrivning
OffsetPosX OffsetPosY OffsetPosZ	0,0 0,0 0,0	Lasertransmitterns placering i förhållande till plattformens centrum. [m]
ForwardDirectionX ForwardDirectionY ForwardDirectionZ	0,0 1,0 0,0	Anger transmitterns utgångsriktning i förhållande till plattformen, d.v.s. laserns riktning då följevinklarna (pitch och yaw) är noll.
FineTracking	<i>false</i>	Anger om transmittern styrs av grov- och finföljning ( <i>true</i> ) eller bara grovföljning ( <i>false</i> ). Grovföljningen är passiv och styrs mot robotens IR-signatur. Finföljningen är aktiv och styrs mot retroreflexen i hot-sensorn under laserbelysning. Observera att tillhörande optik till ett givet följesystem måste skapas i NetScene. Optiken för grovföljning ska ligga under transmittern och finföljarens optik ska ligga under



		grovföljarens. Notera också att om <i>AutoTracking</i> sätts till <i>true</i> så är denna parameter redundant – styrsystemet använder i detta fall ingen finföljning.
AutoTracking	<i>false</i>	Anger om transmittern automatiskt ska följa det utvalda hotet (idealt, d.v.s. inget pekfel).
TrackingAngle	1,0	Det pekfel relativt hotriktningen där grovföljning påbörjas. [°]
LockAngle	0,1	Den felvinkel relativt hotet inom vilken grovföljaren anses vara låst. Denna vinkel bör anpassas till laserns divergensvinkel så att lasern träffar hotet så snart lasern aktiveras (se även <i>LockTime</i> ). [°]
LockTime	0,1 / 0,1	Den tid som respektive följesystem måste vara låst på målet innan nästa steg i hotbekämpningen påbörjas. Grovföljaren anses låst inom felvinkeln <i>LockAngle</i> . Finföljaren anses låst då retrosignal registreras från hotet. Efter låsning av grovföljaren aktiveras lasern och ev. finföljning. Finföljningen i sin tur följs av störning av hotet (efter ev. klassificering). [s]
UpdateFrequency	50 / 200	Följesystemens uppdateringsfrekvens, d v s den frekvens med vilken signalen från transmitters sensorer, och därmed uppskattningen av hotets position, uppdateras. Finföljningen är i regel snabbare. [Hz]
QuadrantDetector	<i>true</i>	Anger hur retrosignalen från hotsensorn detekteras för att ge en uppskattning hotets position, som används vid finföljningen. Detektion kan ske med en kvadrantdetektor ( <i>true</i> ) eller med en array av detektorer och intensitetsviktning för att beräkna hotpositionen ( <i>false</i> ) [DIRCM].
ServoTimeConstant	1,0 / 0,1	Anger tidskonstanten (stigtiden) för servot till respektive styrsystem [DIRCM]. [s]
MaxAngularVelocity	10 / 40	Maximal vinkelhastighet som respektive servo kan ge. [rad/s]
MaxAngle		Maximal utvridning för respektive servo i yaw och pitch. Ett värde över $2\pi$ betyder att hela rymden täcks in (default). I annat fall täcks ett mindre område kring den riktning som anges av <i>ForwardVector</i> ovan. [rad]
ServoBandWidth ServoPhaseMargin	3 / 60 30 / 60	Anger önskad snabbhet och stabilitet för styrningen av respektive servo [DIRCM]. Bandbredd anges i [Hz], fasmarginal i [°].

### III Laserkälla

Parametrarna för laserkällan beskriver strålningens egenskaper som effekt, våglängd och divergens (se Figur 27 och Tabell 22).



**Figur 27** Parametrar för laserkällan anges under modulnamnet [*EWLaserSource*].

**Tabell 22** Beskrivning av parametrar för laserkällan.

Parameter	Default-värde	Beskrivning
Power	1,0	Laserns effekt. [W]
WaveLength	$3,0 \cdot 10^{-6}$	Laserns våglängd. [m]
Divergence	0,001	Laserstrålens divergensvinkel. Räknas från strålcenrum, d v s "halva vinkeln". [rad]
RadiusOfCurvature	1,0	Laserstrålens krökningsradie. Beskriver bl a om den är konvergent, divergent eller kollimerad [TURB].
InfiniteRadiusOfCurvature	<i>true</i>	Används för att ange en laserstråle med oändlig krökningsradie, vilket innebär att strålen är kollimerad.
BeamType	0	Beskriver laserns strålprofil, d v s effekt-fördelningen i utbredningsplanet. 0 = gaussisk. 1 = homogen.

### IV Optisk sensor

För att DIRCM-systemet ska kunna följa hoten måste det utrustas med optiska sensorer enligt vad som tidigare beskrevs. Parametrarna för en allmän optik ges i detalj i avsnitt 2.10.2. Av dessa behöver endast följande sättas för att erhålla ett välfungerande DIRCM-system: *ApertureDiameter*, *Transmission*, *FovX*, *FovY*, *SizeX* och *SizeY*.

## 2.10 Optik

I optroniska system utgör optiken en central del. Det är optiken som registrerar den elektromagnetiska strålningen från miljön och som t ex gör det möjligt för en IR-målsökare att följa ett mål. Olika typer av optiska system förekommer i en mängd tillämpningar inom telekrigföring, bl a i sikten, sensorer och robotar.

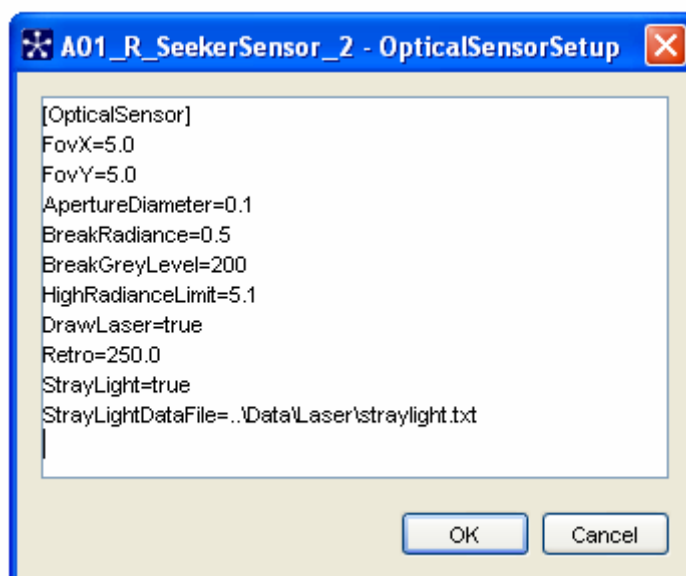
### 2.10.1 Optik i EWSim

I EWSim används optik för att beskriva en rad viktiga detaljer som påverkar utgången av en telekrigsduell. Framför allt utgör optiken en gränsyta mellan de optiska hoten (målsökande robotar och fientliga sikten) och dess motmedel (t.ex. DIRCM och optikspaning). Viktiga egenskaper hos optikerna i detta sammanhang är lasermålarea (retroreflektion) och ströstrålning.

Optikerna kopplas i många modeller (t.ex. målsökare och sikten) till en kamera i EWSim. Kameran ger en vy av terrängen och övriga objekt under simulering.

### 2.10.2 Beskrivning av parametrar

Figur 28 visar hur en optik konfigureras i NetScene. Parametrarna och deras önskade värden anges i en textsträng och de parametrar som utelämnas defaultinitieras (till det värde som anges i Tabell 23).



**Figur 28** Parametrar för optik anges under modulnamnet [*OpticalSensor*].

**Tabell 23** Beskrivning av parametrar för optik.

Parameter	Default-värde	Beskrivning
FovX FovY	3,0 3,0	Optikens synfält. [°]
ArraySizeX ArraySizeY	128 128	Upplösningen i kameran som används för att betrakta miljön. Anger antalet pixlar i den registrerade bilden. Kan även ses som antalet detektorelement i optiken (stirrande array).
ApertureDiameter	0,05	Optikens aperturdiameter. Cirkulär apertur antas. [m]
Transmission	1,0	Optikens transmissionskoefficient, d v s den andel av den infallande strålningsintensiteten som registreras av optiken.
BreakRadiance BreakGreyLevel LowRadianceLimit HighRadianceLimit	1,0 200 0,0 10,0	Anger hur terrängens och objektens gråskalenivåer översätts till radians. Funktionen anges som en styckvis linjär funktion med en brytpunkt. Linjär interpolation mellan punkterna:  (0, <i>LowRadianceLimit</i> ), ( <i>BreakGreyLevel</i> , <i>BreakRadiance</i> ) och (255, <i>HighRadianceLimit</i> ).  Radianserna anges i [ $\text{Wm}^{-2}\text{sr}^{-1}$ ].
NearPointExclusion FarPointExclusion	5,0 5,0	Områden som undantas vid kontroll av fri sikt. Dessa anges som radien på två sfärer, dels runt optiken, dels runt den punkt i miljön som optiken betraktar. Kan vara användbart om man vill bortse från att de egna plattformarna kan skymma sikten. [m]
Retro	100,0	Optikens lasermålarea. Anger den retroreflekterade intensiteten som funktion av irradiansen som faller in mot optiken [DIRCM]. [ $\text{m}^2/\text{sr}$ ]
DrawLaser	<i>false</i>	Anger om laserstörning ska ritas in i sensorns vy (då den är kopplad till en kamera).
Turbulence	<i>false</i>	Anger om bildegradering till följd av atmosfärsturbulens ska beräknas för den laserradians som når optiken [LASER].
ImageDancing	<i>false</i>	Image dancing innebär att den registrerade bilden hos en optik hoppar runt i fokalplanet till följd av atmosfärsturbulens [LASER]. Denna parameter anger om image dancing ska beräknas för laserstrålningen i optikens synfält, vilket innebär att lasern kommer hoppa runt i bilden.
Blobs	1	Används för att medelvärdesbilda flera laserfläckar som rör på sig under image dancing. Med hjälp av

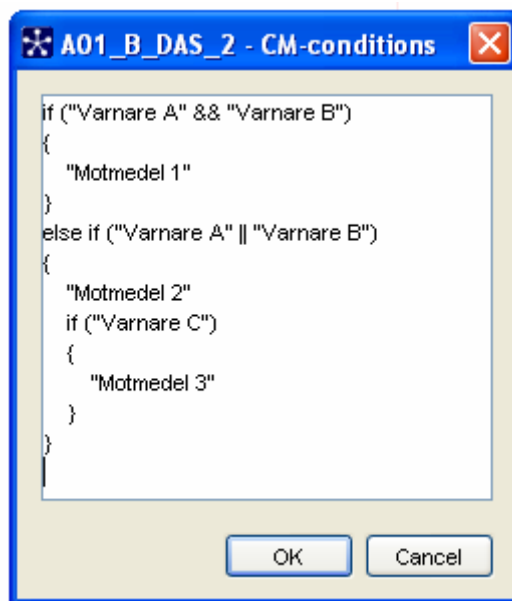
		denna parameter kan man därmed välja integreringstid (eller tidsskala) i optiken, i förhållande till turbulensens tidsskala. Ett högre <i>Blobs</i> -värde innebär längre integreringstid i optiken och därmed mindre effekt av image dancing.
StrayLight	<i>false</i>	Anger om ströljus i optiken ska påverka bildgenereringen av laserstrålningen som träffar optiken. Ströljuset anges i form av en PSF (Point Spread Function) som anger hur en punktformig laserkälla avbildas av optiken [DIRCM]. Ströljuset gör det bl a möjligt att blända optiken. Utan ströljus (och turbulens) ritas lasern upp som en punkt i optikens synfält då radiansen är fokuserad i en enda infallsriktning.
StrayLightDataFile		<p>Sökvägen till en textfil som anger optikens ströljus (PSF). Filen ska vara en textfil med siffror som löpande anger vinkel (offset från infallsvinkeln) [rad] följt av ströljusbidrag för denna vinkel. Enklast i form av två kolumner. Exempel:</p> <pre> 0      1 0.01  0.2 0.02  0.05 0.04  0.01 0.06  0.003 0.1   0.001 0.2   0.0001 0.4   0 </pre> <p>Den vänstra kolumnen anger vinkeln och den högra det tillhörande PSF-värdet. Linjär interpolation sker sedan mellan de angivna punkterna.</p>

## 2.11 VMS-logik

### 2.11.1 VMS i EWSim

Detta avsnitt handlar om sammankopplingen av varnare och motmedel vilka tillsammans utgör den aktuella plattformens VMS (varnare- och motverkanssystem). Olika typer av varnare och motmedel har beskrivits tidigare i rapporten. Till dessa kan också läggas sådana från andra våglängdsband som radar och radio. Något kortfattat kan man säga att varnarnas uppgift är att upptäcka de hot som dyker upp och registrera riktningen som de närmar sig från. Utifrån denna riktning bekämpas hoten efter bästa förmåga med de motmedel som finns att tillgå, t ex genom att kasta rök eller facklor, eller genom laserstörning.

Aktiveringen av motmedlen måste i de flesta fall ske mycket snabbt (högst ett fåtal sekunder) för att ge plattformen en rimlig överlevnadschans. Det finns mycket att vinna på en automatisk aktivering av motmedel som är anpassade till det givna hotet. I EWSim sker en logisk sammanvägning av registrerade varningar för att generera en lämplig motåtgärd. Denna logik är konfigurerbar i NetScene, i en enhet som kallas *DAS* (efter engelskans Defensive Aids Suite). Logiken ges i form av en textsträng som kopplar ihop id för varnare med id för motmedel i en logisk satsstruktur (lik den i C++). Ett exempel på detta ges i Figur 29.



**Figur 29** Varnare och motmedel kopplas ihop i en logisk struktur.

Satserna avkodas med speciella nyckelord:

`if, else if, else, { }, " ", &&, ||, !, ( )`

`if` och `else if` testar logiska uttryck som ska omges av parenteser och byggs upp av:

- Id för varnare som anges inom " ". När det logiska uttrycket evalueras omvandlas dessa till värdena *sant* eller *falskt*. Om varnaren finns på plattformen och har genererat en varning så genereras värdet sant, annars falskt.
- De logiska operatorerna OCH (&&), ELLER (||) och NOT (!), vilka är logiska operationer i enlighet med den boolska algebran.

- Parenteser som visar i vilken ordning uttryck ska evalueras. Om inga parenteser anges har ! högst prioritet, följt av && och sist | |.

I exemplet i Figur 29 är den första if-satsen sann om varnarna med id "Varnare A" och "Varnare B" tillhör den aktuella plattformen och har gett varning från samma riktning.

Om ett givet logiskt uttryck är sant genomförs operationerna inom klammrarna { } som ska följa direkt efter. I exemplet kommer motmedel med id "Motmedel 1" att aktiveras (såvida det finns ett sådant motmedel på den aktuella plattformen). På samma sätt aktiveras "Motmedel 2" om någon av "Varnare A" eller "Varnare B" gett varning. "Motmedel 3" aktiveras om dessutom "Varnare C" varnat.

### 3 Exempel på simulerade scenarier

Det här avsnittet syftar till att ge några exempel på hur EWSim kan användas. De uppsatta scenarierna är relativt enkla, men innehåller trots det ett stort antal moduler som måste samverka med varandra på ett korrekt sätt. Det är sedan inte svårt att utvidga dessa scenarier till att omfatta fler fordon, hot, varnare, motmedel och kombinationer däremellan. Det blir där-  
emot svårare att presentera resultatet av ett mer omfattande scenario.

I merdelen av exemplen är programkörningen uppdelad på två stycken federater, så att styrningen av blått och rött lag sköts av var sin exekvering. Det finns ingen begränsning på hur många federater man kan använda. Distribuering av olika delar till olika datorer är därför ett bra sätt att öka simuleringshastigheten.

Förutom de rena optronikmodulerna som finns presenterade i kapitel 2 används i exemplen modeller för moteld, automatisk eld och varning via radio. Moteld är en typ av motmedel som finns tillgänglig för olika plattformar och som kan triggas av VMS-logiken när den får en varning. Vid aktivering ställs aktuellt sikte i rätt riktning. Är t ex plattformen av typen stridsfordon vrids tornet in. Har plattformen tillgång till en laseravståndsmätare, mäts så avståndet till hotet, innan slutligen ett skott avlossas i hotets riktning.

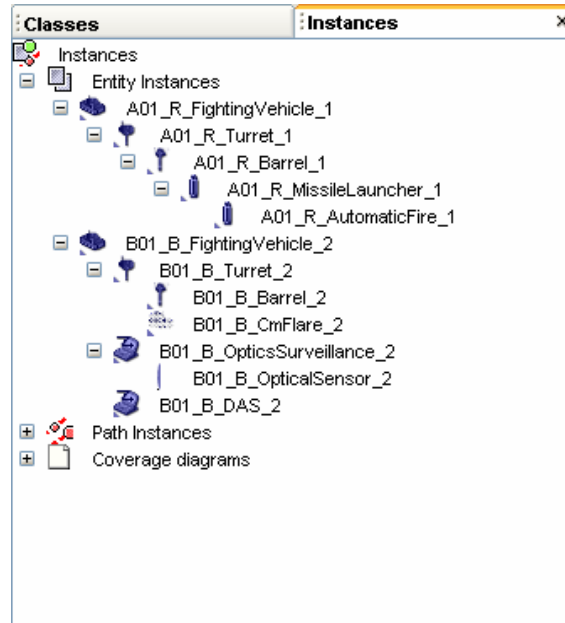
Automatisk eld syftar till att möjliggöra ett icke-operatörstyrt hot och finns tillgänglig för stridsfordon. Vid fri sikt till en plattform som inte tillhör det egna laget avfyras automatiskt antingen en missil eller en projektil, beroende på vad stridsfordonet är utrustat med. Finns laseravståndsmätare föregås projektilskott av avståndsmätning.

Varning via radio ger oss möjligheten att simulera gruppoperationer. Finns radiovarningsmodulen aktiverad kan ett fordon med varningssystem vidarebefordra varningar till andra lagmedlemmar. För den som är mottagare av varning via radio fungerar det som vilken annan varnare som helst. Via inställningar av VMS-logiken kopplas radiovarning till en vald specifik motåtgärd, som förhoppningsvis ger fordonet möjlighet att undkomma hotet.



### 3.1 Scenario 1: Bildalstrande missil mot ett stridsfordon

Scenario 1 skildrar duellen mellan ett stridsfordon i blått lag och en missil i rött lag. Missilen, som har en centroidmålsökare, sitter på ett stridsfordon och avfyras automatiskt då det blåa stridsfordonet blir synligt. Det blåa stridsfordonet är utrustat med VMS som använder sig av en optikspanare för att upptäcka och varna för optiken i missiler. En varning triggar rök som motmedel. Figur 30 visar mer utförligt vilka delar som scenariet består av.



**Figur 30** Scenarioupställning 1.

Vid start är stridsfordonet och missilen placerade ca 800 meter ifrån varandra, på var sin sida av en skog vilket gör att de inte har fri sikt mellan sig.

#### 3.1.1 Valda parametrar

Exemplen nedan är gjorda med följande parameterinställningar, med namn enligt Figur 30.

##### I A01\_R\_MissileLauncher\_1

Missilen har en centroidmålföljare som arbetar i det visuella våglängdsområdet. Den använder areabalansering och har en ROI på 128x128 pixlar. Missilen rör sig med en hastighet på 300 m/s och avfyras automatiskt vid line of sight.

##### II B01\_B\_DAS\_2

VMS-logiken kopplar ihop optikspanare med rök så att en varning alltid ger utkastning av rök.

##### III B01\_B\_OpticsSurveillance\_2

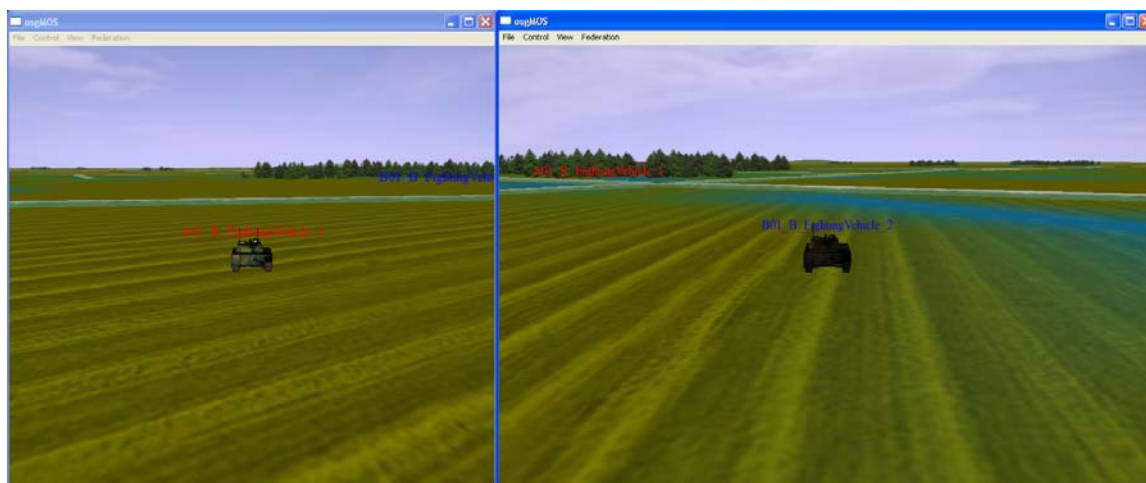
Optikspanaren har en svephastighet på ett varv per sekund, är stabiliserad i pitchled och skickar varning för optik som närmar sig med en hastighet högre än 50 m/s. Den har en mycket låg tröskel för effekten hos retroreflexen som ska ge varning.

#### 3.1.2 Simulering

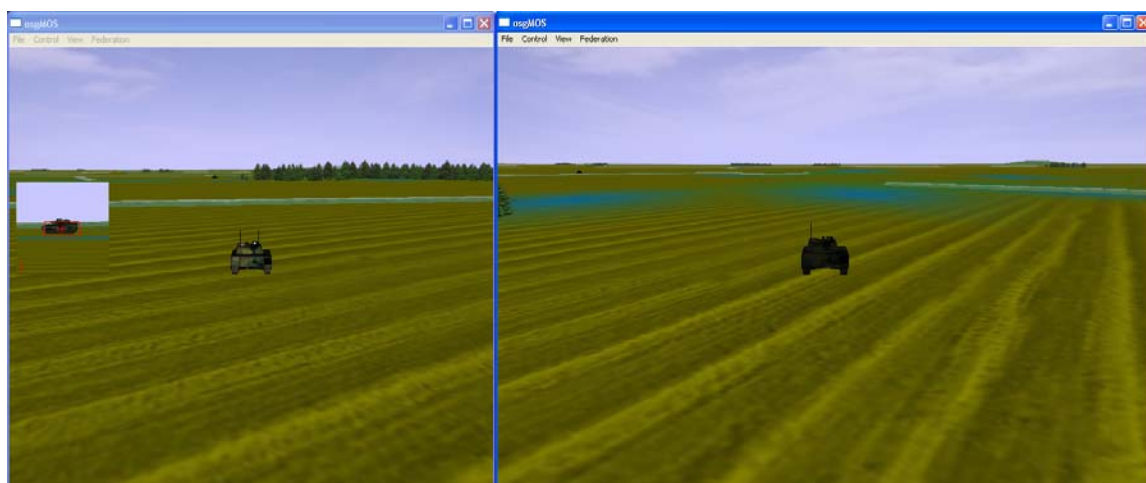
Vi låter missilen stå still på sin startposition medan stridsfordonet rör sig runt skogen. När det blir fri sikt mellan de båda fordonen avfyras missilen. I det här avsnittet visas bilder på en del olika resultat av simuleringarna. I första fallet använder det blåa stridsfordonet inget

motmedel, Figur 31 - Figur 33. Sedan används en optikspanare för att ge varning, Figur 34 - Figur 38. Dels varnar den enbart för snabbt närmande mål, dels även för stillastående mål.

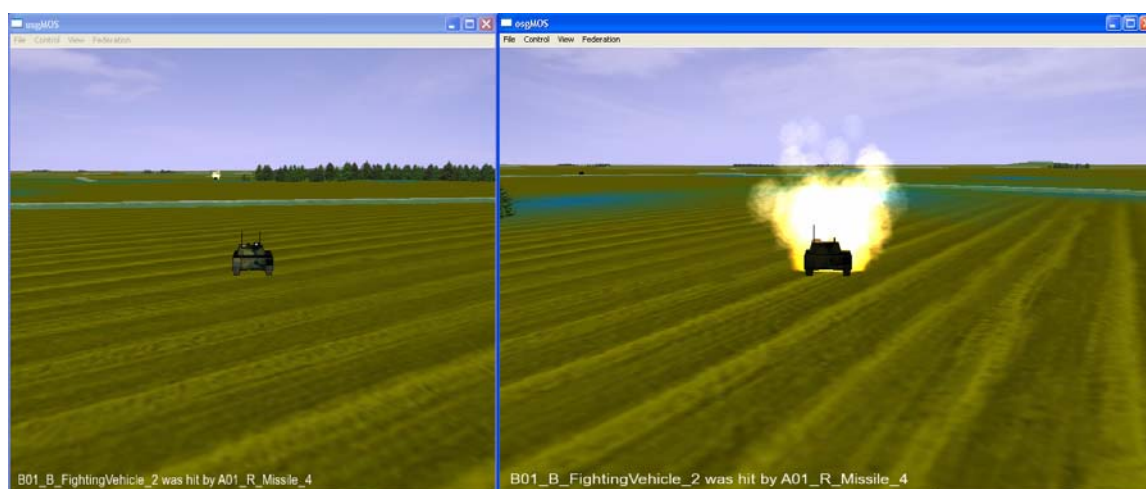
## I Utan motmedel



**Figur 31** I startpositionerna.

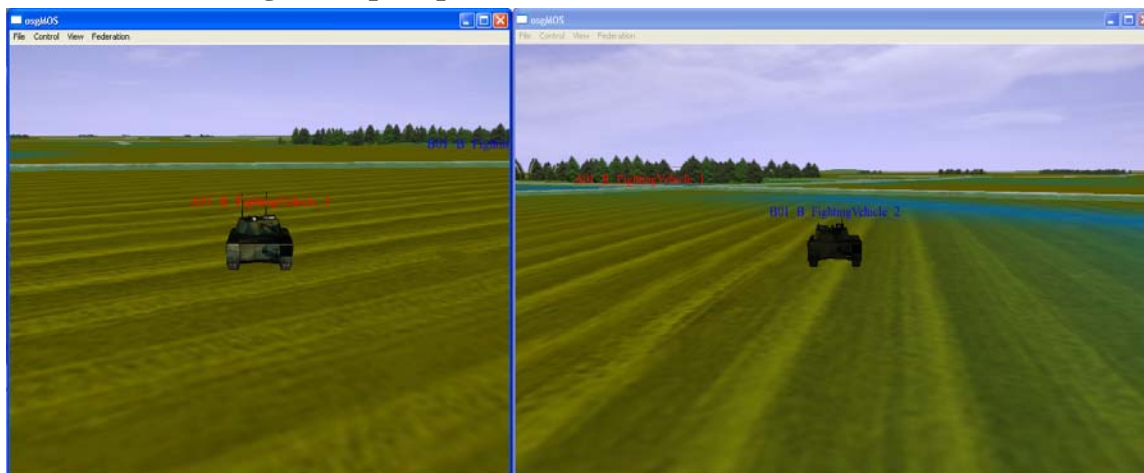


**Figur 32** Stridsfordonet kommer i siktlinjen och missilen avfyras.

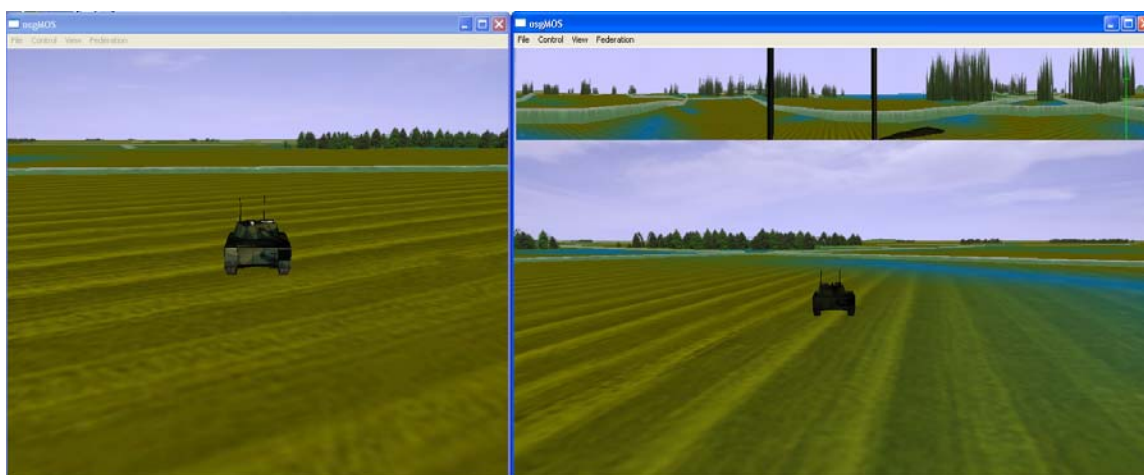


**Figur 33** Stridsfordonet träffas av missilen.

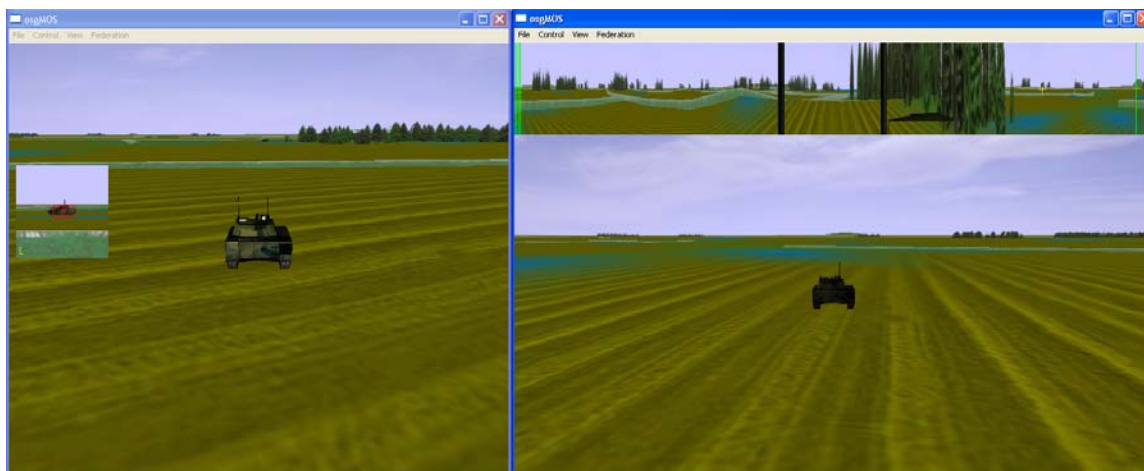
## II Med varning från optikspanare och rök som motmedel



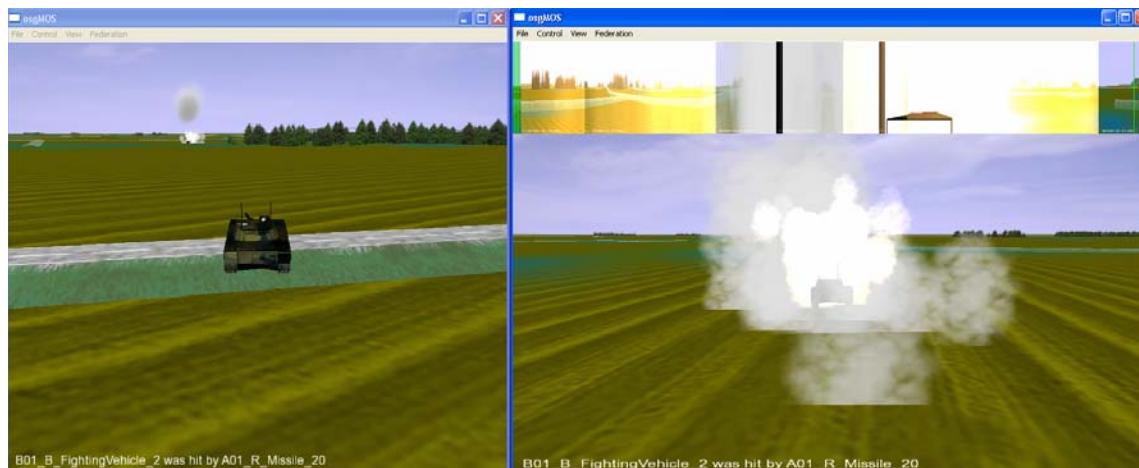
**Figur 34** Fordonen i startposition.



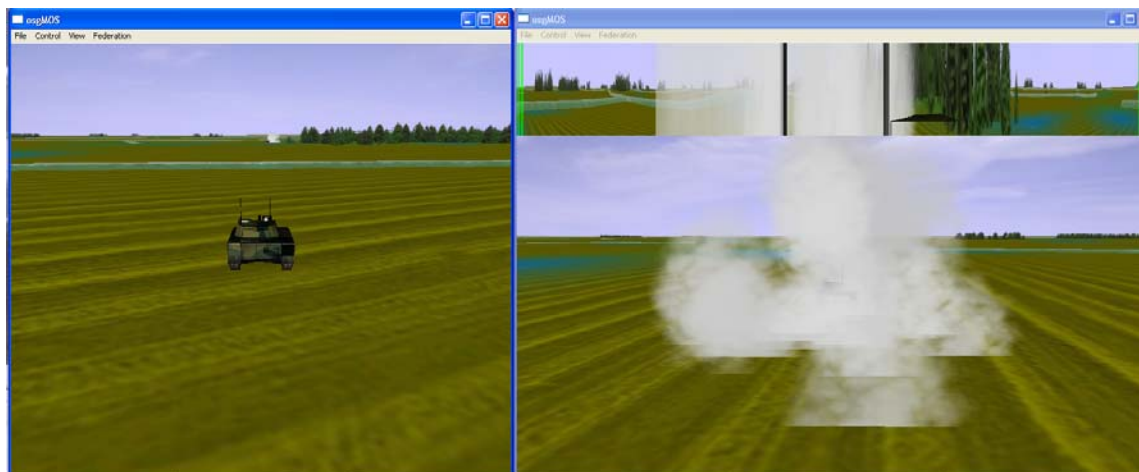
**Figur 35** Optikspanaren har aktiverats och dess display visas i skärmens överkant. Den gröna vertikala linjen markerar det egna fordonets orientering i horisontalled och den lilla horisontella linjen därpå visar orienteringen i vertikalled.



**Figur 36** Stridsfordonet kommer in i synfältet och missilen låser på. Optiken i missilen upptäcks av optikspanaren och markeras med ett gult + i dess display.



**Figur 37** Aktiveras varningen endast av optik som snabbt närmar sig det egna fordonet, kommer inte röken hinna utvecklas på det här korta avståndet och stridsfordonet blir träffat.

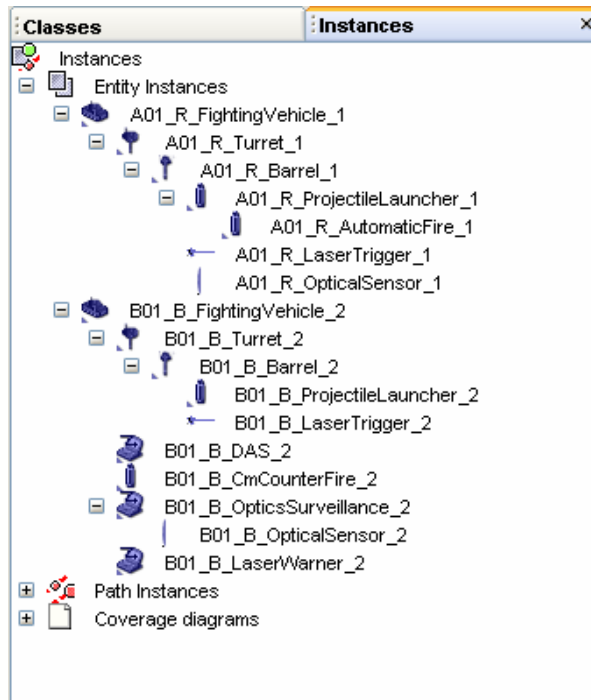


**Figur 38** Låter man optikspanaren generera varning även för stillastående optik kan man upptäcka siktet på det fientliga stridsfordonet och röken kan läggas ut när man vill. Detta gör i det här fallet att missilen missar, men åtgärden kräver ett noggrannare val av motåtgärd eftersom alla sikten registreras.



## 3.2 Scenario 2: Stridsfordon mot stridsfordon

Scenario 2 skildrar duellen mellan två stridsfordon. Fordonet i det röda laget är utrustat med projektil, laserinmätare samt automatisk eld. Vid fri sikt aktiveras den automatiska elden, som vrider in siktet och mäter in avståndet till målet med laser. När detta är gjort avfyras en projektil mot målet. Stridsfordonet i det blåa laget är utrustat med VMS som använder sig av antingen laservarnare eller optikspanare för att upptäcka eventuella hot och generera varning. En varning triggar motmedel i form av moteld föregående av laserinmätning. Figur 39 visar mer utförligt vilka delar som scenariet består av.



**Figur 39** Scenarioupställning 2.

Vid start är stridsfordonet och missilen placerade ca 1000 meter ifrån varandra, utan fri sikt sinsemellan.

### 3.2.1 Valda parametrar

Exemplen nedan är gjorda med följande parameterinställningar.

#### I **A01\_R\_AutomaticFire\_1**

Tidsfördröjningar mellan inriktning av torn, laserinmätning och avfyrning varieras mellan olika simuleringar. Dessa beskrivs mer utförligt för varje simulering.

#### II **B01\_B\_DAS\_2**

VMS-logiken kopplar ihop antingen laservarnare eller optikspanare med moteld så att en varning alltid ger moteld med föregående inmätning av avståndet till hotet.

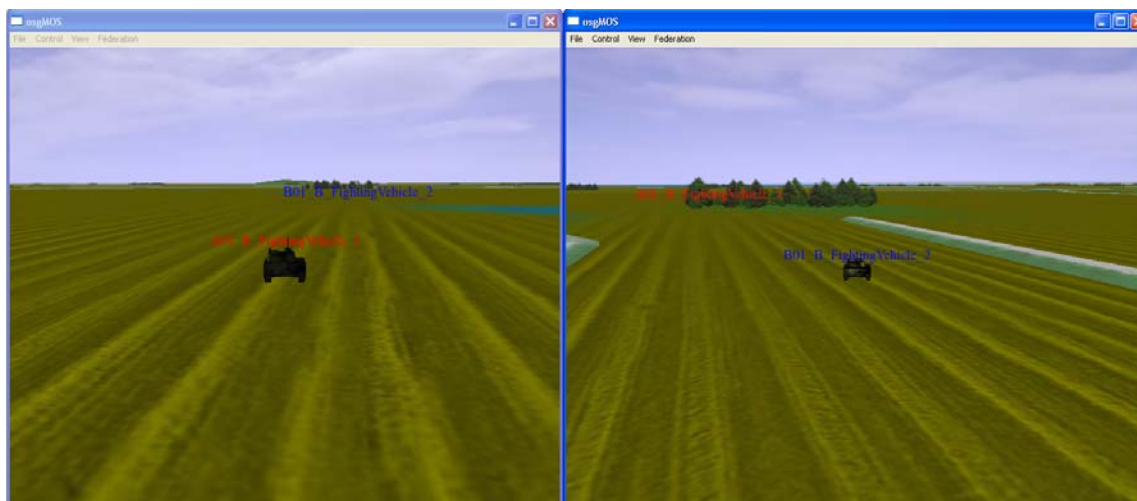
#### III **B01\_B\_OpticsSurveillance\_2**

Optikspanaren har en svephastighet på ett varv per sekund, är stabiliserad i pitchled och skickar varning för all optik som inte avlägsnar sig från stridsfordonet. Tröskeln för effekten hos retroreflexen som ska ge varning ligger på  $1 \cdot 10^{-7}$  W.

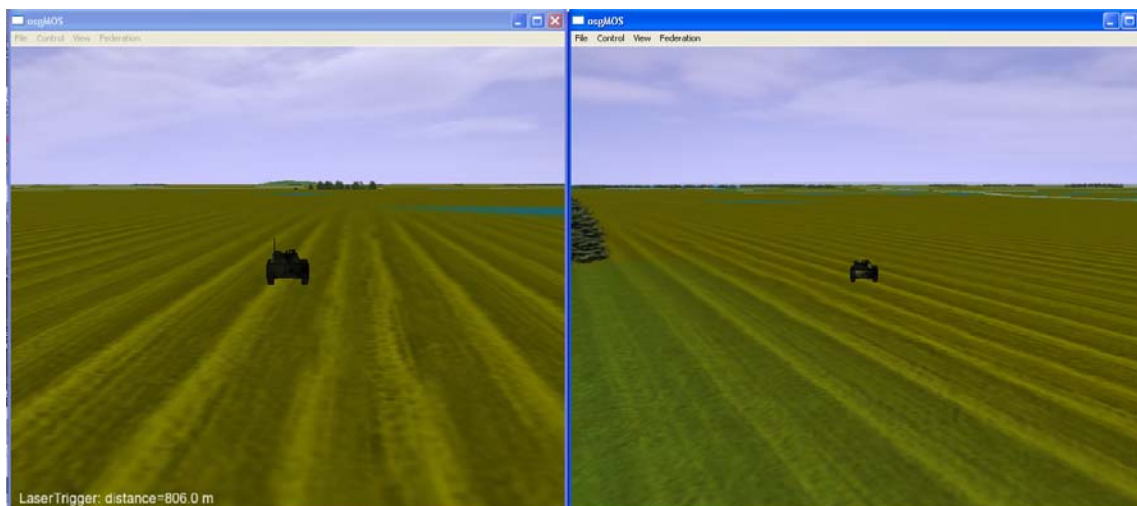
### 3.2.2 Simulering

Vi låter det blåa stridsfordonet röra sig tills det blir fri sikt mellan fordonen. I det här avsnittet visas bilder på en del olika resultat av simuleringarna. I första fallet använder det blåa stridsfordonet inget motmedel, Figur 40 - Figur 42. I nästa fall används en laservarnare för att ge varning, Figur 43 - Figur 45. Till slut används en optikspanare för att ge varning, Figur 46 - Figur 51. Här visas vad som händer om det blåa stridsfordonet får varning dels precis då det röda fordonet gör sin laserinmätning, dels då det röda stridsfordonet har det blåa i siktet, innan det har bestämt sig för att skjuta.

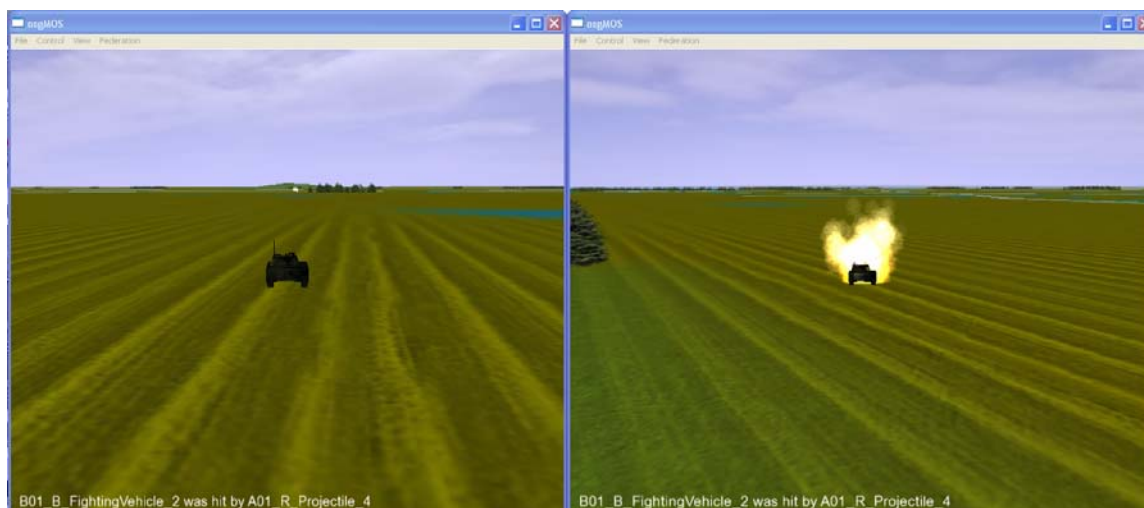
#### I Utan motmedel



**Figur 40** I startpositionerna.

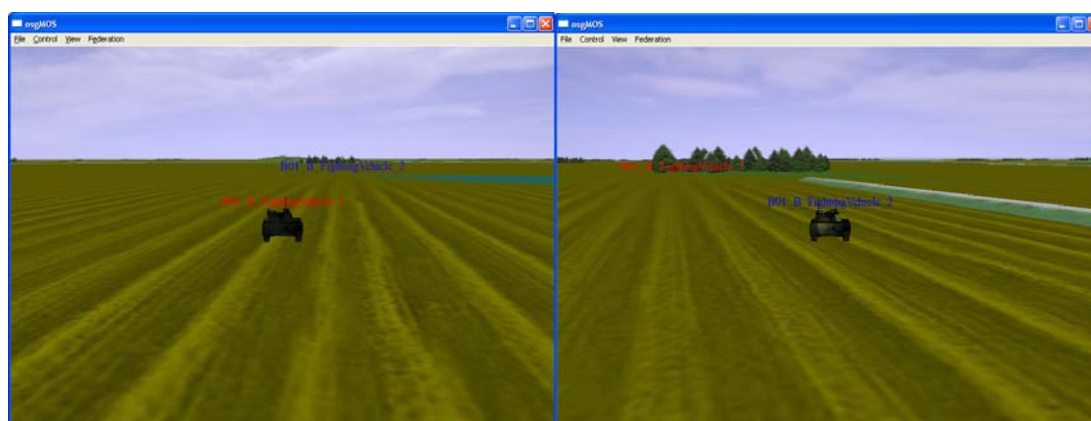


**Figur 41** Det blåa stridsfordonet kommer i siktlinjen för det röda stridsfordonet som vrider in tornet och mäter in avståndet till sitt mål med laser.

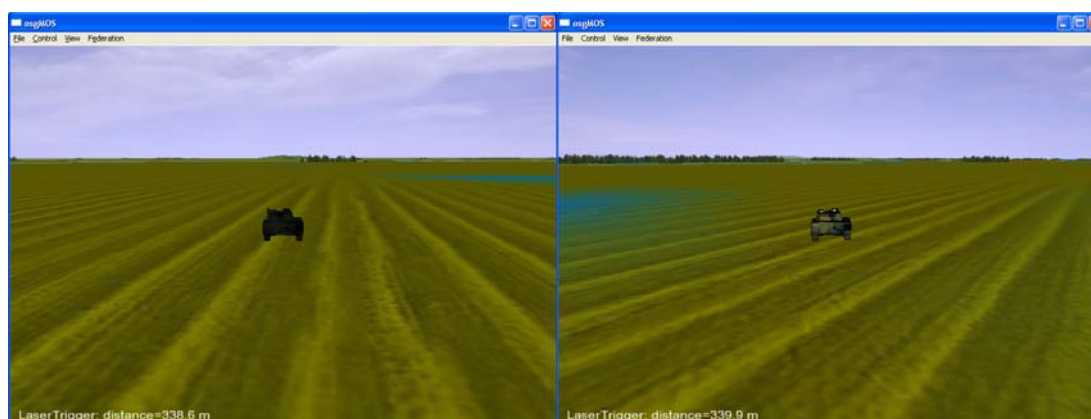


**Figur 42** Det röda stridsfordonet avfyrar en projektil som träffar det blåa stridsfordonet.

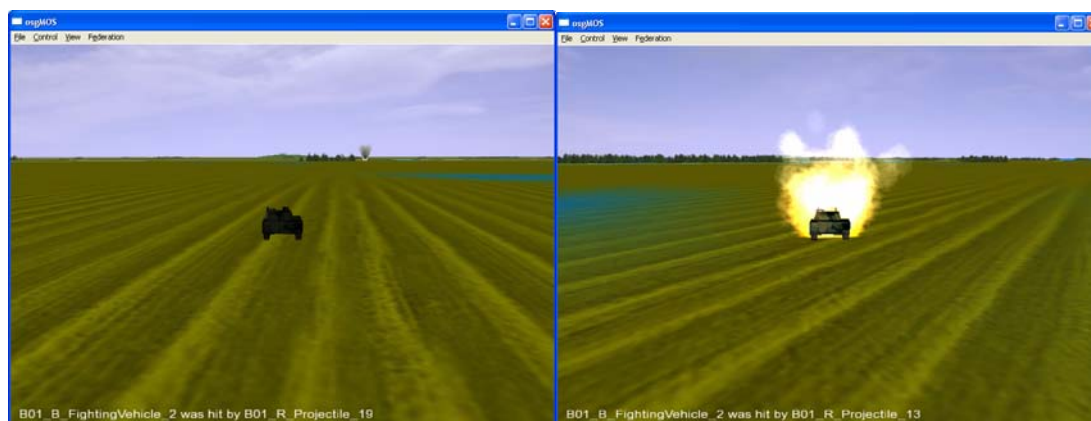
## II Med varning från laservernare och moteld som motverkan



**Figur 43** I startpositionerna.

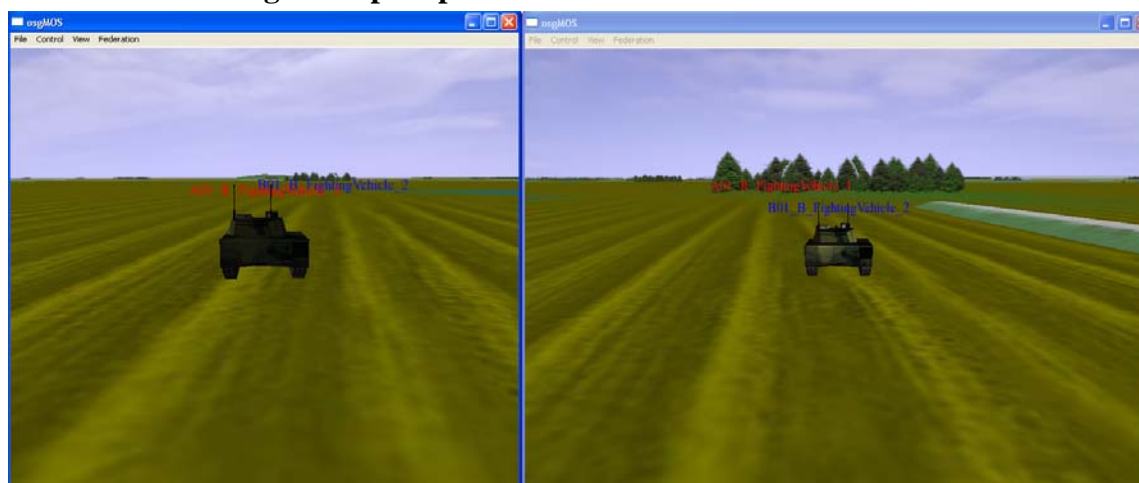


**Figur 44** Det blåa stridsfordonet kommer in i det röda stridsfordonets siktlinje, varvid det röda stridsfordonet mäter in avståndet med laser.

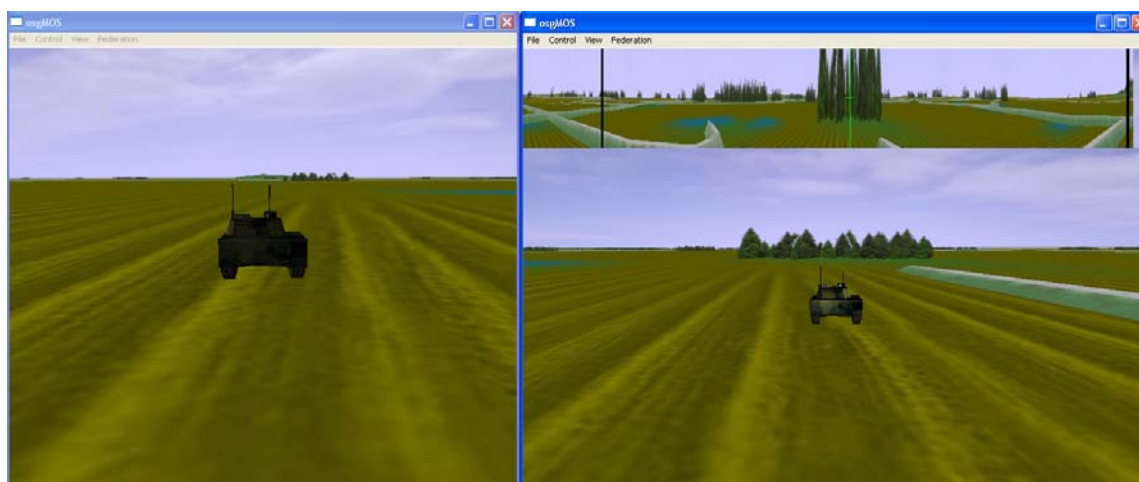


**Figur 45** Det blåa fordonet registrerar en varning från sin laservarnare men hinner knappt börja vrida in tornet innan det blir träffat.

### III Med varning från optikspanare och moteld som motverkan

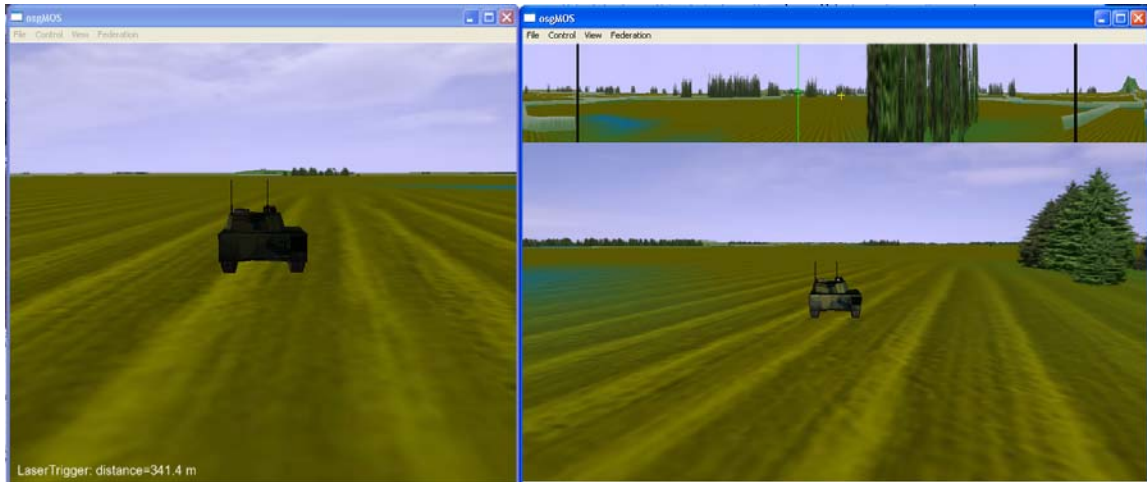


**Figur 46** I startpositionerna.

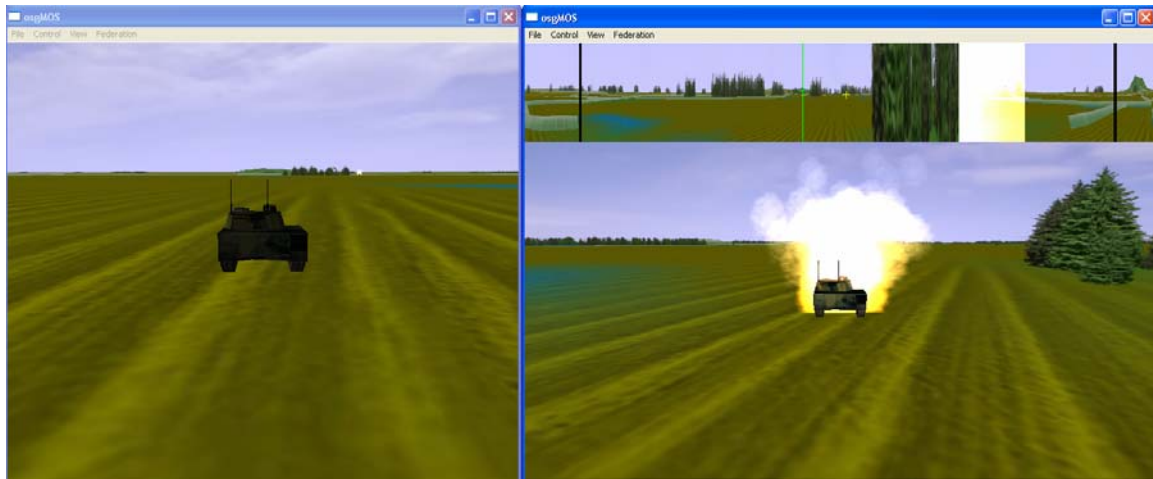


**Figur 47** Det blåa stridsfordonet har optikspanaren påslagen, vilket syns i överkant på skärmen. Det egna fordonets orientering syns som en grön vertikal linje (yawled) med en kort horisontell linje på (pitchled).

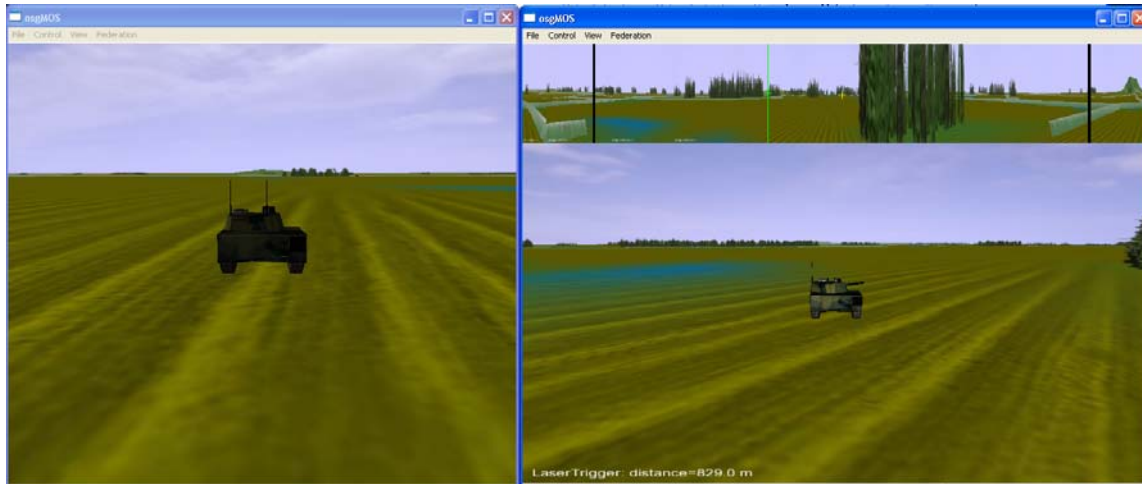




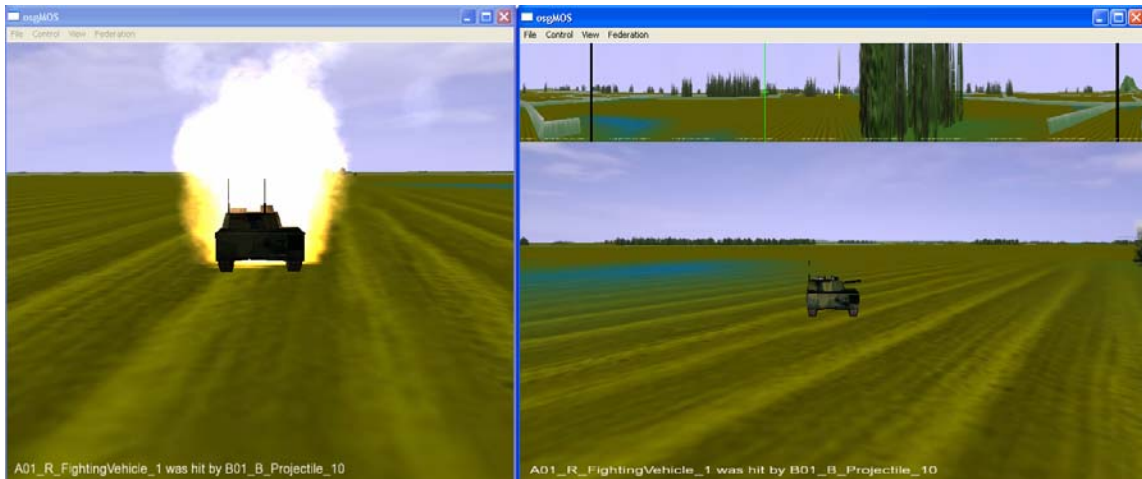
**Figur 48** Det röda stridsfordonet står redo när det blåa kommer i siktlinjen och riktar in sitt torn och mäter avståndet samtidigt som optiken i dess sikte dyker upp som en varning från optikspanaren hos det blåa stridsfordonet.



**Figur 49** Det blåa stridsfordonet börjar rikta in sitt torn för att ge moteld, men hinner inte klart utan träffas av projektilen från det röda stridsfordonet.



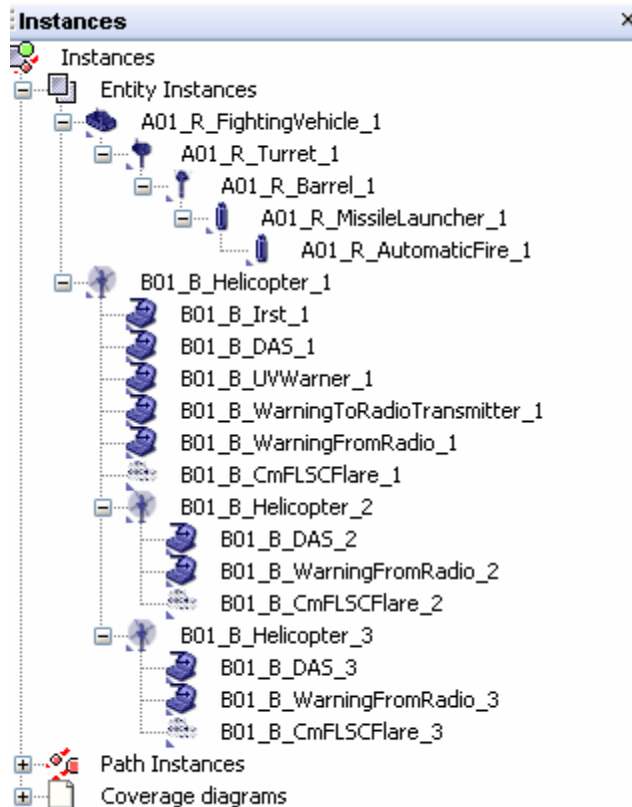
**Figur 50** Får det blåa stridsfordonet i stället mer tid på sig innan det röda fordonet avfyra sin projektil hinner den både rikta in tornet och mäta avståndet till det röda fordonet...



**Figur 51** ...och blir den som skjuter först.

### 3.3 Scenario 3: En bildalstrande IR missil mot tre Hkp

Scenario 3 skildrar duellen mellan tre helikoptrar i blått lag och tre missiler i rött lag. Missilerna, som har en centroidmålsökare, sitter på ett stridsfordon och avfyras automatiskt då de blåa helikoptrarna blir synliga. Ledarhelikoptern är utrustad med VMS som använder sig av en UV-varnare för att varna för robotskott. En varning skickas ut med radio till de två följande helikoptrarna, vilka triggar facklor som motmedel. Figur 52 visar mer utförligt vilka delar som scenariet består av.



**Figur 52** Scenarioupställning 3.

Vid start är helikoptrarna och missilerna placerade ca 1400 meter ifrån varandra.

#### 3.3.1 Valda parametrar

Exemplen nedan är gjorda med följande parameterinställningar.

##### I A01\_R\_MissileLauncher\_1

Missilen har en centroidmålföljare som arbetar i IR-våglängdsområdet. Den använder area-balansering och har ett ROI på 128x128 pixlar. Missilen rör sig med en hastighet på 300 m/s och avfyras automatiskt vid line of sight.

##### II B01\_B\_DAS\_1

VMS-logiken i ledarhelikoptern kopplar ihop UV-varnaren med ”varning över radio” som ger varning till de följande helikoptrarna, samt ger utkastning av facklor enligt ett förvalt program, i detta scenario en sekvens bestående av 2-2 facklor som kastas två gånger med en sekunds mellanrum. VMS-logiken i de följande helikoptrarna kopplar ihop radiovarning med facklor så att en varning alltid ger utkastning av facklor.

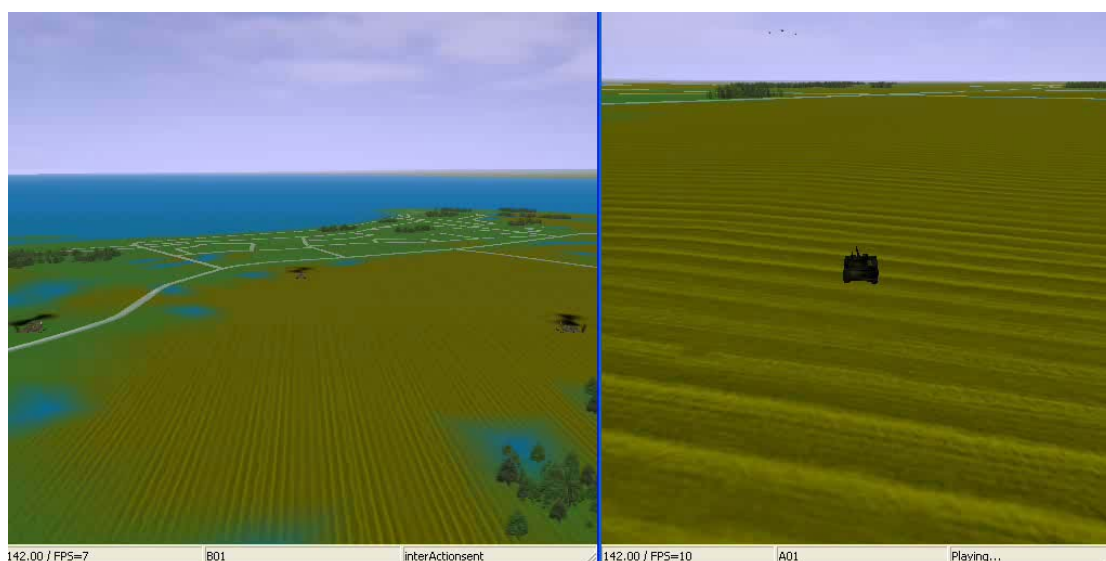
### III B01\_B\_UVWarner\_1

UV-varnaren ger varning av robotskott vid fri sikt av avfyrningsplatsen, i intervallet 0-2 sekunder efter avfyrningen.

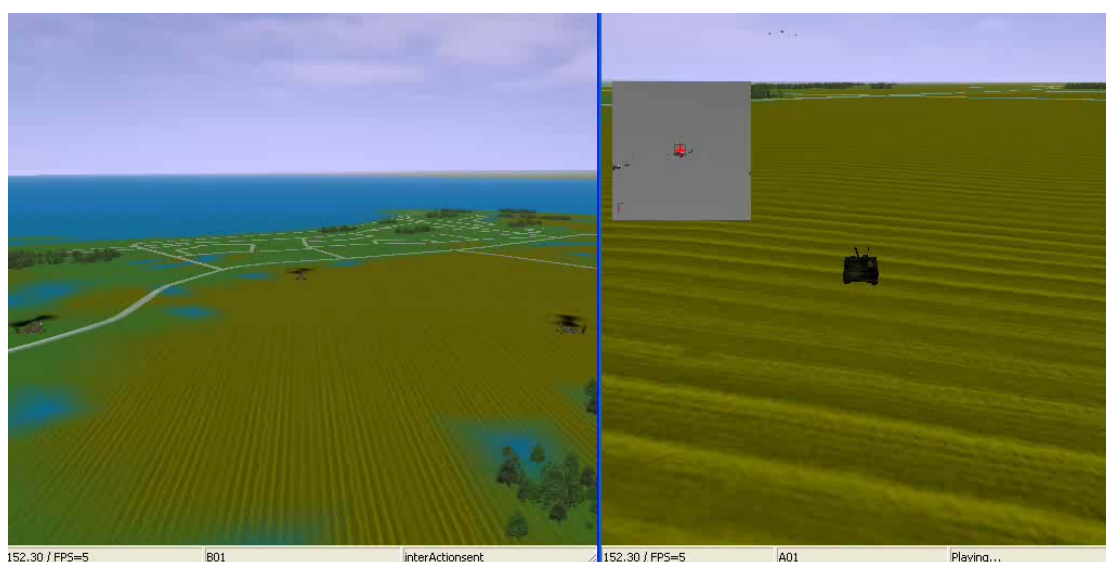
#### 3.3.2 Simulering

Vi låter missilen stå still på sin startposition och helikoptrarna hovrar på avståndet 50 m från varandra. I det här avsnittet visas bilder på en del olika resultat av simuleringarna. I första fallet använder helikoptrarna inget motmedel, Figur 53 - Figur 55. Sedan används en UV-varnare för att ge varning till ledarhelikoptern, som vidarebefordrar varningen till de andra helikoptrarna via radio, Figur 56 - Figur 60, i detta fall med mycket kort tidsfördröjning.

#### I Utan motmedel

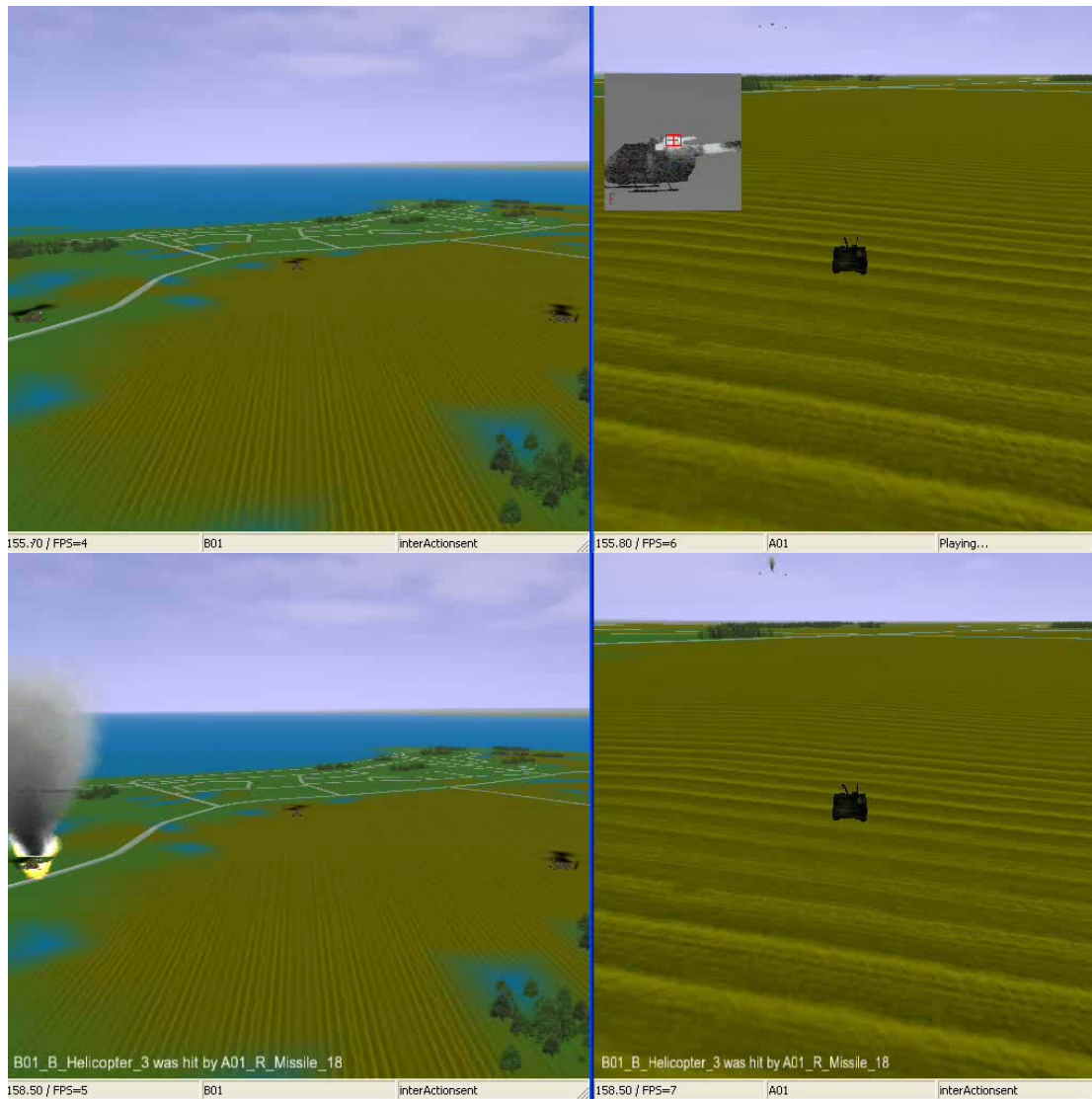


Figur 53 I startpositionerna.



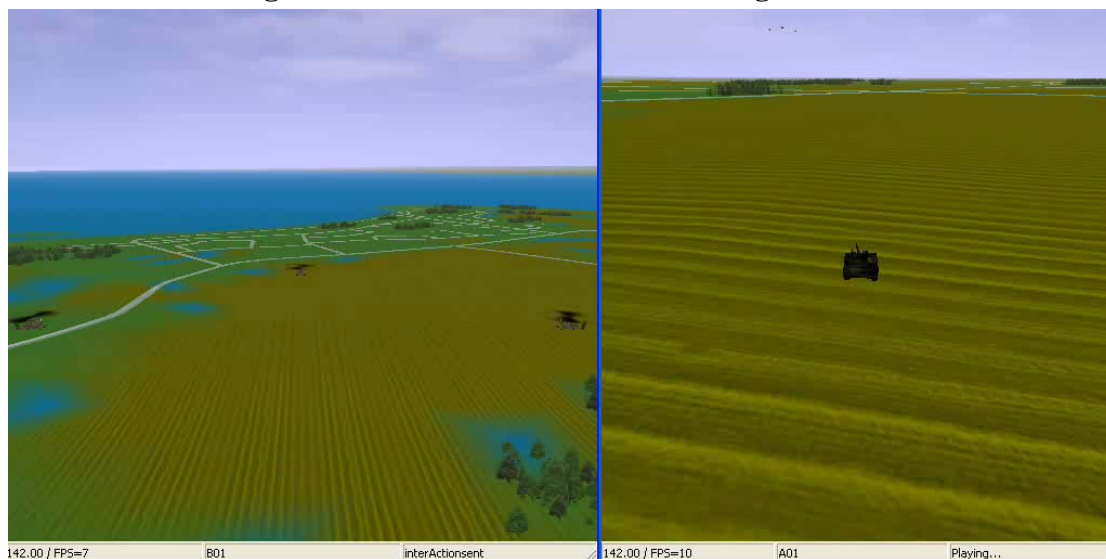
Figur 54 Helikoptrarna upptäcks och missilen avfyras.



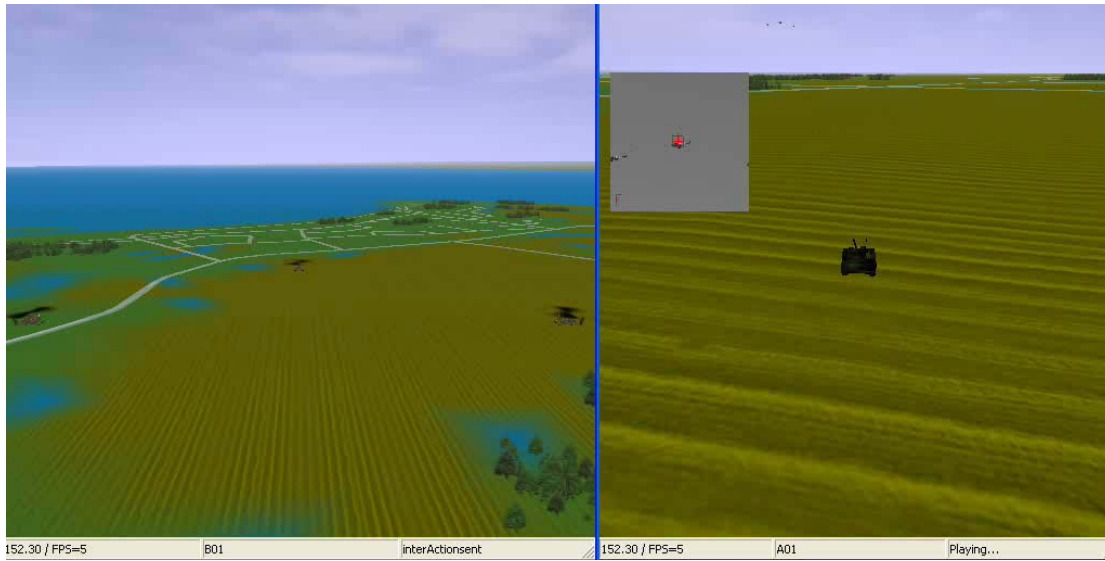


**Figur 55** Helikoptern träffas av missilen och exploderar.

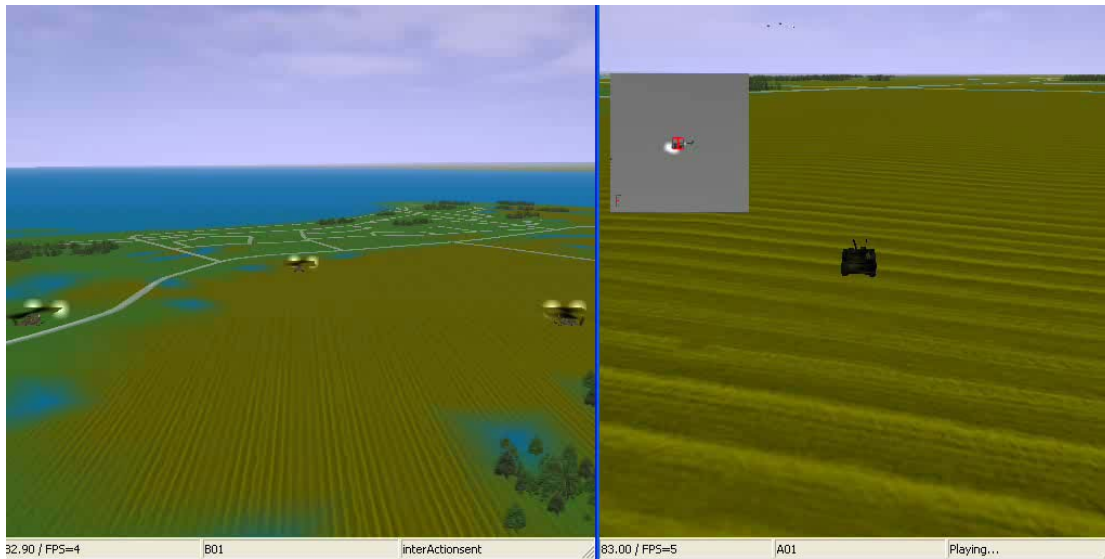
## II Med varning från UV-varnare och fackelfällning



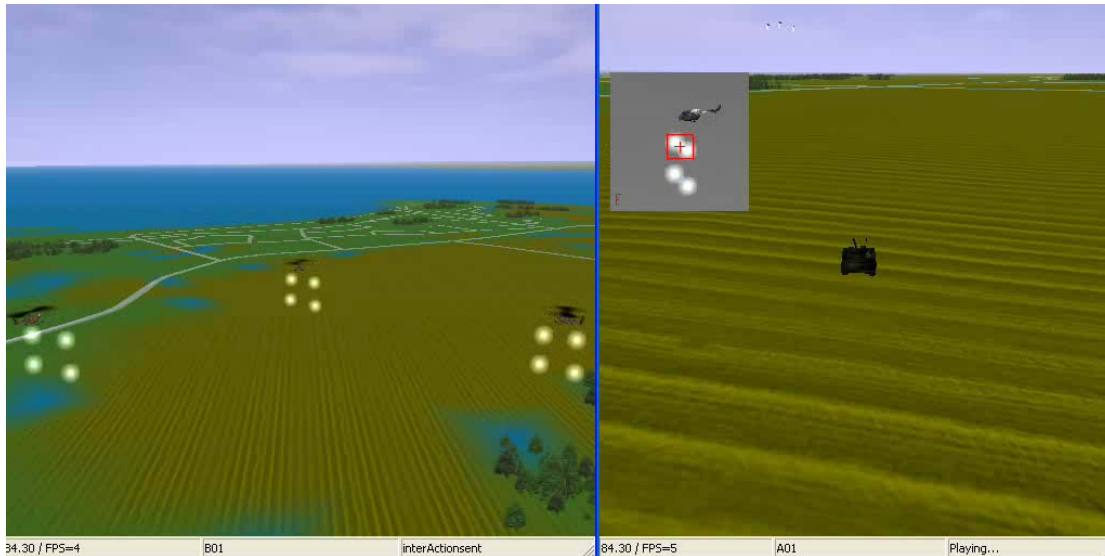
**Figur 56** I startpositionerna. UV-varnaren är aktiverad.



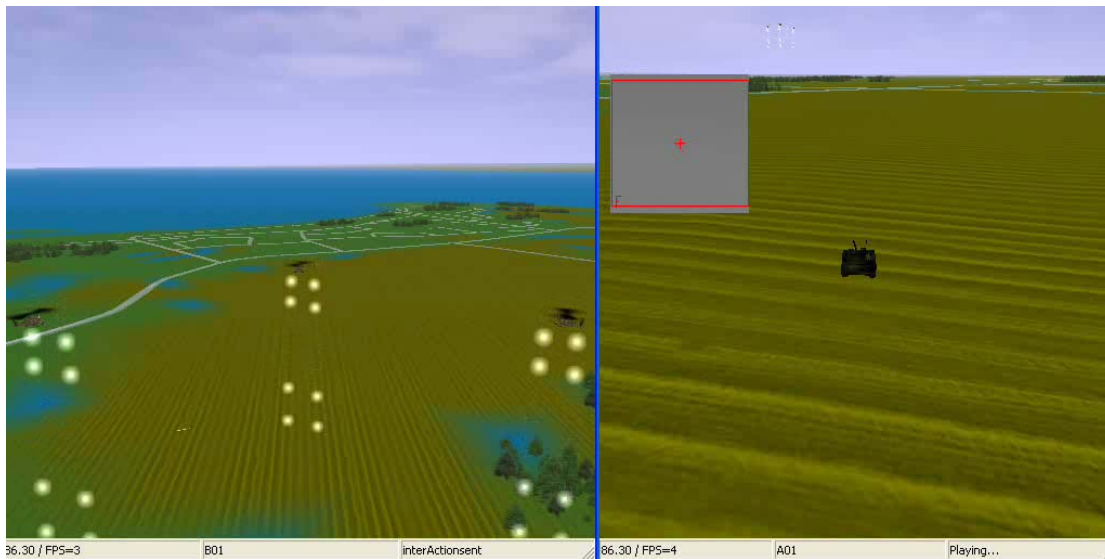
**Figur 57** Helikoptrarna upptäcks och missilen avfyras.



**Figur 58** UV-varnaren har varnat för robotskott och facklor utlöses.



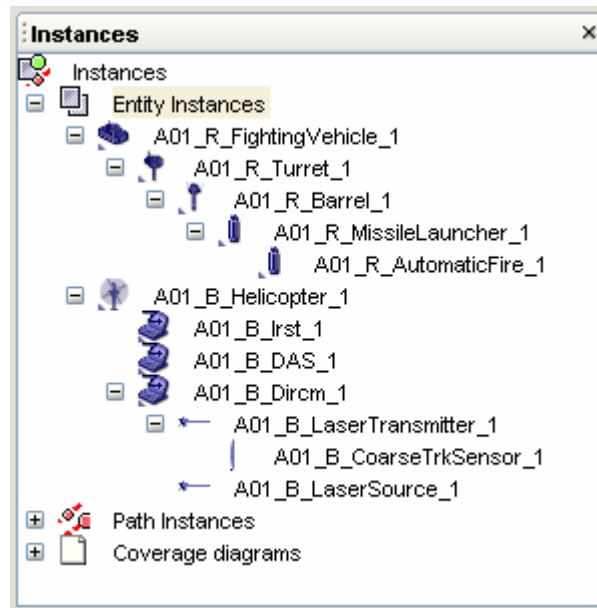
**Figur 59** Missilen låser på facklorna som sedan faller nedanför helikoptern.



**Figur 60** Helikoptern överlever då missilen fortsätter igenom facklorna.

### 3.4 Scenario 4: En retikelmålsökare mot IRST och DIRCM

I scenario 4 skildras duellen mellan det röda lagets stridsfordon, som avfyrar en värmesökande missil utrustad med retikelmålsökare, och det blåa lagets helikopter. Eldgivningen från stridsfordonet sker automatiskt, precis som i scenario 3. Helikoptern är i detta fall utrustad med VMS som består av IRST och DIRCM. IRST:n varnar för närmande missiler och triggar via VMS-logiken DIRCM som genom varningsriktningen kan lokalisera och störa ut hotet med blinkstörande laser. Figur 61 visar en vy från NetScene över de delar som ingår i scenariet.



Figur 61 Uppställning för scenario 4.

#### 3.4.1 Valda parametrar

Simuleringarna som redovisas nedan är körda med följande parametrar för missil, IRST och DIRCM.

##### I A01\_R\_MissileLauncher\_1

Missilen utrustas med en retikelmålföljare som arbetar inom IR-området. Målföljaren är av nuterande FM-typ (frekvensmodulering). Nutationsfrekvensen valdes till 100 Hz. Missilen rör sig med en hastighet på 300 m/s och avfyras automatiskt vid line of sight.

##### II A01\_B\_Irst\_1

IRST:n skannar av omgivningen med ett varv i sekunden och ger varning för alla hastigt närmande föremål med tillräcklig IR-signatur (se avsnitt 2.2.3).

##### III A01\_B\_Dircm\_1

Lasereffekten är 0,1 W och modulerad med frekvensen 101 Hz för att vilseleda retikelmålsökaren. Målsökaren kommer dessutom att bländas av lasern till följd av ströljus i optiken (se avsnitt 2.10.2).

#### 3.4.2 Simulering

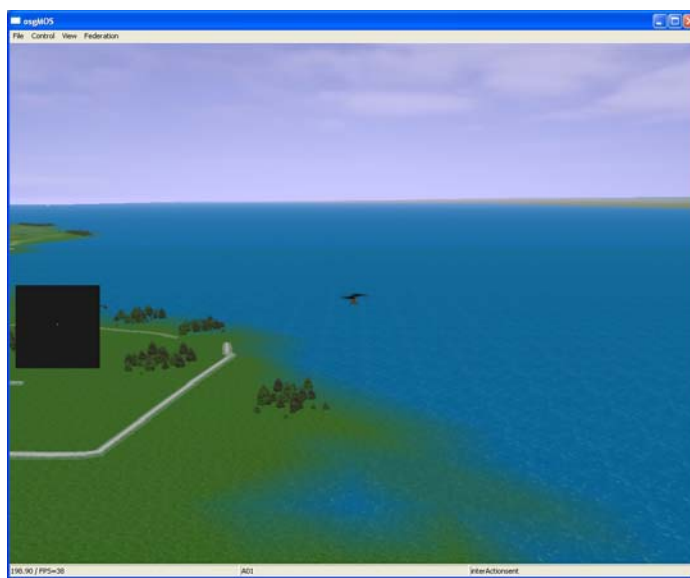
Simuleringarna som presenteras nedan körs på en enda federat. Operatören styr i detta fall helikoptern samtidigt som stridsfordonet styrs automatiskt på samma federat. Då scenariet



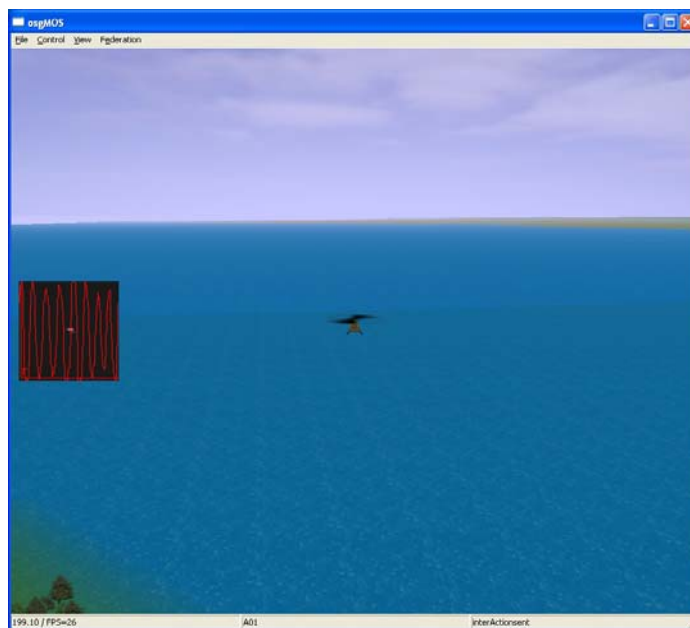
startar är helikoptern och stridsfordonet placerade ca 2000 m från varandra. Helikoptern manövreras under det att missilen låses och avfyras mot den.

I det första fallet som beskrivs använder helikoptern inga motmedel och blir därmed nedskjuten av missilen. I det andra fallet är helikopterns VMS, med IRST och DIRCM, aktiverad. Missilen upptäcks då av helikopterns IRST och störs sedan ut av DIRCM. Figur 62 - Figur 64 visar bilder från det ostörda scenariet, Figur 65 - Figur 67 visar bilder från det störda.

## I Utan motmedel



**Figur 62** Så snart stridsfordonet har fri sikt mot helikoptern armeras missilen. Målsökarens vy visar sig till vänster i bilden. Helikoptern är synlig i målsökarens synfält.

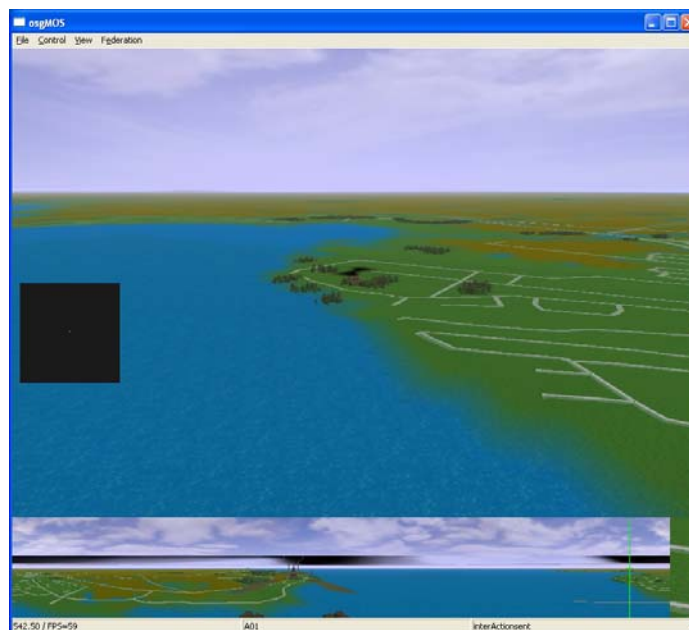


**Figur 63** Missilen är avfyrad och retikelmålsökaren följer helikopterns IR-signatur trots manövrering.

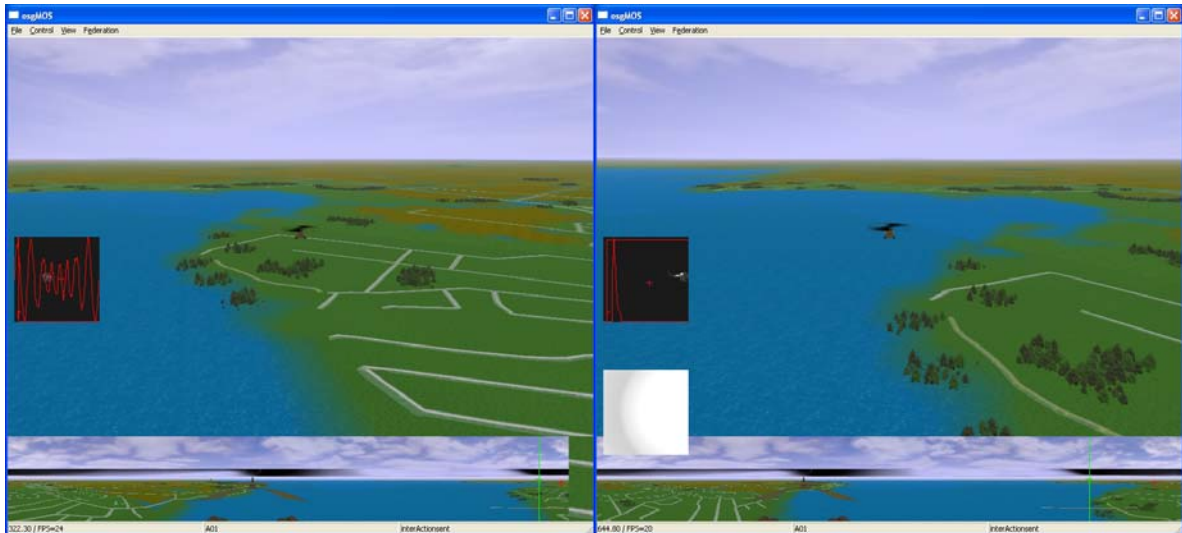


**Figur 64** Helikoptern träffas av missilen.

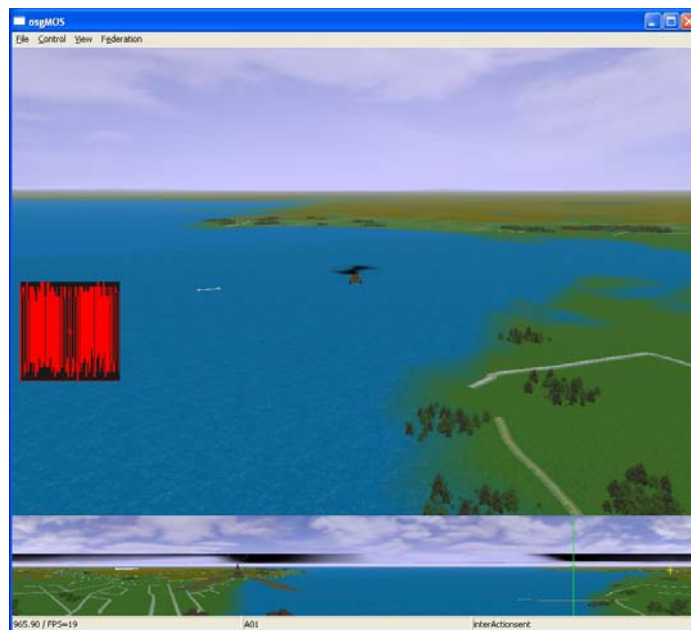
## II Med varning från IRST och laserstörning med DIRCM



**Figur 65** Precis som i föregående fall armeras missilen då stridsfordonet har fri sikt mot helikoptern. IRST:n upptäcker missilen i ett tidigt skede men ger ingen varning innan den avfyras.



**Figur 66** IRST:n varnar för den angripande missilen, vilket kan ta en sekund från det att missilen avfyras beroende på svephastigheten hos IRST:n (1 varv/s). Störning med DIRCM inleds och retikelmålsökaren blir förvirrad.



**Figur 67** Retikelmålsökaren har tappat bort helikoptern och flyger rakt fram. Helikoptern överlever angreppet.

## 4 Diskussion

Simulering är ett kraftfullt verktyg för värdering av telekrigdueller då dagens försvar har begränsade resurser för utbildning av förband i telekrigföring samt utveckling av teknik och taktik eftersom antalet tillfällen för övningar och praktiska prov är begränsade. Detta dokument beskriver ett antal moduler som kan kombineras för simulering och värdering av telekrigdueller, vilket kan stödja utveckling, kravsättning och förståelse för samverkande telekrigssystem. I rapporten visas exempel på scenarier som kan simuleras. Scenarierna kan tyckas vara ganska enkla men tittar man närmare på dessa så har de mellan fem och tjugo modeller av system som samverkar eller motverkar varandra. Modulerna kan kombineras och sättas på alla typer av plattformar såsom stridsfordon, helikoptrar, UAV:er, flygplan, båtar och fartyg. Detta gör att sammansatta förbands telekrigföringsförmåga samt sårbarhet för telekrig kan simuleras. Att simulera scenarier med alla dess ingående system ger den övergripande kontrollen på var svagheter och styrkor med enskilda system finns. Att sedan göra detaljerade studier av enskilda system är något som inte lämpar sig att köra i scenarier. Denna djupare analys utförs lämpligen med detaljerade modeller av ett specifikt system där modellen stimuleras med simulerade signaler som t ex varierar i en batchkörning. Om simuleringarna utförs på båda nivåerna ger det både de övergripande systemegenskaperna i ett sammanhang samt detaljerad förståelse för ett specifikt systems prestanda på apparatnivå.

En klar styrka med simuleringsramverket är att det är lätt att återanvända simulerade plattformar som redan är konfigurerade genom att dessa finns beskrivna i en fil vilken helt enkelt kan instansieras till nya objekt. Detta gör att det är enkelt att skala upp scenarier med flera farkoster som använder de utvecklade modulerna. Analys av resultat från simuleringar där komplexiteten är som i de simulerade scenarierna som visas i kapitel 3 är inte helt enkelt då det inträffar många olika händelser som är beroende av varandra och ger olika utfall av simuleringen. Detta är något som det nya FoT-projektet ”Duellsimulering framtida telekriginsatser” kommer att utveckla metodik för, bl a visualiseringar i grafer och återuppspelningar från loggade parametrar. Idag kan vissa parametrar loggas men eftersom all källkod finns tillgänglig finns möjligheten att logga allt, vilket är en förutsättning för att kunna värdera.

Att använda de framtagna simuleringsverktygen som de ser ut idag för värdering bör genomföras av personer med kunskap om modellernas begränsningar då risken annars finns att fel slutsatser kan dras. Tidigare versioner av några av modellerna har använts för att studera hur telekrigmateriel ska kunna utnyttjas taktiskt och vilka systemparametrar som begränsar det taktiska utnyttjandet hos system, vilket kunnat användas vid upphandling och kravställning. Detta går att göra som förut men nu på ett mer generaliserat sätt samt att man nu kan skala upp subscenarier till en större helhet och därmed få en mer fullständig bild av hur system kompletterar varandra eller skulle kunna komplettera varandra.

Med EWSim finns nu möjligheten att sätta upp scenarier som kan stödja värdering av några mot några i enskilt användande eller som värderingsduell där man kan bilda lag som kan duellera.

## 5 Fortsatt arbete

I och med att Sverige ska delta i internationella operationer så finns det många nya aspekter att ta hänsyn till, till exempel civila folkmassor och utländska geodata. Att utveckla en generisk modell av PV-robot är ett klart behov då detta är ett av de system som kan tänkas finnas vid internationella insatser. Då simuleringarna nu kan göras mer komplexa finns ett behov att kunna analysera data från komplexa händelseförlopp samt att presentera dessa data. Den modell av UV-varnare som idag finns modellerad behöver förbättras så att den mer efterliknar de system som kommer att finnas på Hkp 14 och Strf 90 så att denna kan simuleras på ett bra sätt vid värderingar. I planeringsverktyget och scenarioeditorn är räckviddsberäkningar en central del vilket endast finns delvis idag. Detta bör införas så att räckviddsberäkningar med störda EO-sensorer finns med. Att förbättra de existerande modellerna samt att verifiera och validera dem mot data som vi kan komma över är ett arbete som kontinuerligt pågår. Det fortsatta arbetet kommer att utföras i ett nytt projekt "Duellsimulering framtida telekriginsatser" som är ett treårigt FoT-projekt med slutår 2008.

## 6 Referenser

- [OPTSIM] C. Hedberg, L. Tydén, C. Wigren, ”Generell metod för simulering av elektro-optiska telekrigdueller”, FOA rapport FOA-R--99-01160-616--SE (1999).
- [EWSIM] C. Hedberg, L. Tydén, C. Wigren, ”EwSim - Electronic Warfare Real Time Simulation in the Visual and Infrared Wavelength Range”, FOI-R--0709--SE (2005).
- [ENUM] J. J. Brann, ”Enumeration and Bit Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications”, IST-CF-03-01, University of Central Florida, (2003).
- [IRSTMARK] <http://www.mss.mil.se/index.php?c=news&id=23561>
- [LEMUR] <http://www.army-technology.com/contractors/fire/bofors/index.html>
- [IRSTSJÖ] <http://www.kockums.se/pdf/visbyprospekt.pdf>
- [IROTISI] Proceedings of SPIE -- Volume 4820.
- [IROTIS] <http://www.canit.se/~griffon/aviation/text/otis.html>
- [IREO] J. S. Accetta, D. L. Shumaker, *The Infrared & Electro-Optical Systems Handbook*, Executive Editors SPIE Optical Engineering Press, (1993), Volym 7, sid 337-339.
- [STRFF] P. Klum, C. Wigren, M. Pettersson, G. Bolander, G. Olsson, ”VMS-funktion i nätverk för stridsfordonsförband, några typfall”, FOI-R--1328--SE (2004).
- [FAS] <http://www.fas.org>
- [CORR] L. Tydén, L. Berglund, ”Laborationshandledning Korrelationsmålföljare”, FOA-rapport FOA-D--97-00335-616--SE, (Augusti 1997).
- [CEN] L. Berglund, C. Hedberg, ”Laborationshandledning Centroidmålföljare”, FOA rapport FOA-D--97-00335-616--SE, (Augusti 1997).
- [RET] Olsson, ”Simulering retikelmålsökare Metodbeskrivning”, FOA rapport C 30685-8.3, (1992).
- [OM] L. Berglund, G. Kindvall (redaktörer), *FOI orienterar OM Telekrig*, Nummer 5 (2005).
- [NEMESIS] <http://www.globalsecurity.org/military/systems/aircraft/systems/an-aaq-24.htm>
- [DIRCM] M. Petersson, ”Implemetering, simulering och verifiering av en laserstörmodell (DIRCM) i ett realtidsramverk”. FOI-rapport, FOI-R--1337--SE (Sep 2004).
- [STRÖ] L. Uhlin, B. Ericson. ”Ströljus i optiska system – Karakterisering av känslighet för laserstörning”. FOA-rapport, FOA-R--95-00206-3.1--SE (Jan 1996).
- [TURB] L. Andrews, R. Phillips. *Laser Beam Propagation through Random Media*. SPIE Optical Engineering Press (1998).
- [LASER] C. Wigren, O. Gustafsson. ”Implementering av laserstörmodell”. FOI-rapport, FOI-R--0249--SE (Nov 2001).