

PETER KLUM, MIKAEL PETERSSON, MATTIAS VERONA



FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 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, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

Peter Klum, Mikael Petersson, Mattias Verona

Generisk modell av ett radarluftvärnsrobotsystem med Telekrig

Titel	Generisk modell av ett radarluftvärnsrobotsystem med Telekrig
Title	Generic model of a radar controlled SA-missile system with Electronic Warfare
Rapportnr/Report no	FOI-R--2537--SE
Rapporttyp Report Type	Metodrapport Methodology report
Sidor/Pages	58 p
Månad/Month	Juni
Utgivningsår/Year	2008
ISSN	ISSN 1650-1942
Kund/Customer	Försvarsmakten
Forskningsområde Programme area	6. Telekrig och vilseledning 6. Electronic Warfare
Delområde Subcategory	61 Telekrigföring med EM-vapen och skydd 61 Electronic Warfare including Electromagnetic Weapons and Protection
Projektnr/Project no	E7015
Godkänd av/Approved by	Mikael Sjöman
FOI, Totalförsvarets Forskningsinstitut	FOI , Swedish Defence Research Agency
Avdelningen för Ledningssystem	Command and Control Systems
Box 1165	
581 11 Linköping	SE-581 11 Linköping

Sammanfattning

Dokumentet beskriver en generisk modell av ett radarluftvärnsrobotsystem utvecklad för simuleringsramverket *EWSim*. Modellen är där utformad för att kunna användas i större simuleringsscenarier med telekrig. I modellen styrs luftvärnsroboten med kommandon att följa en eldledningsradars siktlinje mot målet (kollimationsstyrning realiserad med kommandostyrning). Eldledningsradarn in visas från en spaningsradar. Dokumentet ger en kort introduktion till radarstyrda luftvärnsrobotsystem samt telekrig i ett luftvärnsrobotsystem men är i övrigt på en teknisk nivå. Komponenter i modellen, deras arbetssätt, hur de kan påverkas av telekrig samt hur de konfigureras och används i *EWSim* presenteras.

En generell meddelandestruktur har utvecklats för kommunikation mellan de olika plattformarna i luftvärnsrobotsystemet. Validering av radarsystemet har genomförts och en avfyrd robot beter sig rimligt i de avfyrningsfall som har testats. Dokumentet avslutas med en diskussion kring den utvecklade modellen samt med några förslag till fortsatt arbete. Bland annat föreslås en vidareutveckling av radarmodellen, i syfte att möjliggöra modellering av semiaktivt styrda radarrobotar.

Nyckelord: modellering, simulering, värdering, telekrig, TK, radar, radarstörning, luftvärnsrobot, kollimationsstyrning, syftbäringsstyrning, kommandostyrning, semiaktiv målsökarstyrning

Summary

The document describes a generic model of a surface to air missile system (SAM system) using radar guidance. The model is developed for the simulation framework *EWSim* and is able to perform in a large scenario employing electronic warfare. The missile in the system is command guided to follow a tracking radar's line of sight towards the target (line of sight guidance with remote command guidance). The target to track is assigned by an acquisition radar. The document gives a short introduction to SAM systems and electronic warfare in a SAM system, but is besides that on a technical level. The components in the model, their method of working, the effect on the system by electronic warfare, how they are configured and how they are used in *EWSim* is presented.

A general message structure has been developed for communication between the platforms in the SAM system. The radar model has been validated and the missile model performance is estimated to be reasonable in the launch cases that have been tested. A discussion about the developed model and some proposals for further development concludes the document. One proposed further development is to modify the radar model to allow simulation of semi-active radar guided missiles.

Keywords: Electronic Warfare, EW, radar, radar jamming, surface to air missile, line of sight guidance, proportional navigation guidance, command guidance, semi-active guidance

Innehållsförteckning

1	Inledning	7
1.1	Kort om <i>EWSim</i>	7
1.2	Läsanvisningar.....	8
1.3	Tack till.....	9
2	Radarstyrda Ivr-system	11
2.1	Komponenter.....	11
2.2	Operation.....	12
2.2.1	Startfas.....	12
2.2.2	Anflygningsfas.....	12
2.2.3	Målstyrningsfas.....	12
2.3	Förbandsorganisation.....	17
2.4	Telekrig i ett Ivr-förband.....	19
2.5	Spridning.....	20
3	Design, modellering och implementation	23
3.1	Modellerade komponenter.....	23
3.1.1	TAR.....	23
3.1.2	TTR.....	24
3.1.3	TEL.....	25
3.1.4	Robot.....	25
3.2	Telekrig i den implementerade modellen.....	26
3.3	Designförslag semiaktivt styrd robot.....	28
4	Konfiguration	29
4.1	Konfigurering av TAR.....	29
4.2	Konfigurering av TTR.....	30
4.3	Konfigurering av TEL.....	32
4.3.1	TEL samlokaliserad med TTR.....	33
4.3.2	TEL som separat plattform.....	33
5	Användning	35
5.1	Sekventiell avfyrning med TAR och TTR.....	35
5.2	Direkt avfyrning från TAR ledningsvy.....	36
6	Verifiering och validering	39

7	Diskussion och rekommendationer till fortsatt arbete	41
7.1	Diskussion	41
7.2	Fortsatt arbete	41
8	Appendix: "Modifieringar av radarmodellen"	43
8.1	Trunkeringslober	43
8.2	Expanding av söksvepet mellan HLA-uppdateringar	44
9	Appendix: "Klasser"	47
9.1	Klassen EWTTMRMSICtrlUnit	47
9.2	Övriga klasser	48
10	Appendix: "Meddelanden"	51
11	Appendix: "Parametrar"	53
12	Referenser	57

Om omslagsbilden

Bilden på omslaget visar en avfyrande SA-11 "GADFLY". Bilden är renderad i Bryce 4.

1 Inledning

Detta dokument beskriver en generisk modell av ett radarluftvärnsrobotsystem (radar lvr-system) som har implementerats i simuleringsramverket *EWSim* (Electronic Warfare Simulation interface model). *EWSim* och radar lvr-systemmodellen utvecklas inom FoT-projektet *Duellsimulering framtida telekriginsatser*. Modellarbetet har syftat till att utveckla en lvr-systemmodell som kan nyttjas i ett större scenario med telekrig.

Den beskrivna radar lvr-systemmodellen kan ses som ett naturligt fortsatt steg i vidare-utvecklingen av radarmodellerna som beskrivs i [1], och som med fördel kan studeras då detta dokument läses.

Den lvr-systemmodell som har implementerats består av en spaningsradar, en eldledningsradar med robotlavett eller med robotlavett på ett separat fordon. Roboten som avfyras från lavetten är kollimationsstyrd (siktlinjesstyrd) realiserad med kommandostyrning från eldledningsradarn. Exempel på lvr-system som använder kollimationsstyrning realiserad med kommandostyrning är bland annat RBS 23 Bamse och SA-8 "Gecko"¹.

En kort introduktion till lvr-system samt telekrig i ett lvr-system ges i dokumentet. Dokumentet är i övrigt på en teknisk nivå och vänder sig till dem som vill få en inblick i hur lvr-systemmodellen är uppbyggd. Dokumentet beskriver bland annat modellens komponenter, deras arbetssätt, hur de konfigureras och används i *EWSim* samt hur modellen har verifierats och validerats. Avslutningsvis diskuteras den utvecklade modellen samt förslag på fortsatt arbete ges.

1.1 Kort om *EWSim*

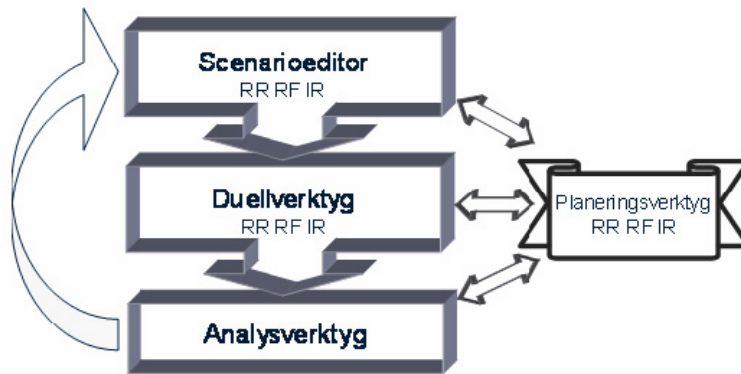
EWSim är ett ramverk för distribuerade telekrigssimuleringar [2,3,4]. Ramverket tillhåller ett bibliotek med modeller av t ex plattformar, sensorer, signaturer, varnare, motmedel och vapensystem, vilka kan kopplas ihop modulärt för att generera detaljerade simulerings-scenarier. *EWSim* använder simuleringsstandarden HLA (High Level Architecture) [5] för att koppla samman simuleringsmodeller (federater²) till en större simuleringsmodell (federation). Med detta följer även att det är möjligt att sätta upp distribuerade simuleringar över ett datornätverk, där olika modeller kan simuleras på olika datorer.

Ramverket består av tre delar, se Figur 1:

- *EWPlan*: scenarioplanering och konfiguration, där olika plattformar (t ex stridsvagnar och helikoptrar) konfigureras med olika komponenter (t ex sensorer och vapen). Dessutom kan räckviddsberäkningar utföras. För mer information om *EWPlan*, se referens [6].
- *Duellverktyg*: realtidssimulering av ingående delar.
- *EWLogger*: utvärdering där dueller kan återuppspelas och loggade data analyseras.

¹ NATO beteckning.

² En federat är benämningen på en deltagande simuleringsmodell i en simulering som nyttjar HLA. Alla federater samlas i en federation.



Figur 1 Beskrivning av hur *EWSim* konceptuellt är uppbyggt.

EWSim innehåller telekrigsmodeller inom områdena radar, kommunikation och optronik. Dessa kan simuleras integrerat i dueller eller större scenarier ("några mot några"), vilket medger systemvärdering ur ett multispektralt telekrigsperspektiv.

1.2 Läsanvisningar

Kapitel 2 "Radarstyrda lvr-system" behandlar översiktligt komponenter i ett generellt radarstyrt luftvärnsrobotsystem, olika radarinmättningsprinciper och robotstyrprinciper, exempel på olika radarstyrda luftvärnsrobotsystem samt förbandsorganisation och telekrig i ett lvr-förband.

Kapitel 3 "Design, modellering och implementation" ger en teknisk beskrivning av hur den kollimationsstyrda robotmodellen har designats, modellerats och implementerats samt hur telekrigsaspekter hanteras. I slutet av kapitlet presenteras designförslag för en semiaktivt styrd robot. Kapitlet förutsätter en viss kännedom om programmering.

Kapitel 4 "Konfiguration" beskriver hur ett radarstyrt luftvärnsrobotsystem konfigureras i *EWPlan*. Parametrar till de olika objekten återfinns i Appendix: "Parametrar".

Kapitel 5 "Användning" beskriver hur luftvärnsrobotsystemet används i *Duellverktyget* i *EWSim*.

Kapitel 6 "Verifiering och validering" beskriver vilket verifierings- och valideringsarbete som har gjorts.

Kapitel 7 "Diskussion och rekommendationer till fortsatt arbete" behandlar just det.

Appendix: "Modifieringar av radarmodellen" ger även en översikt över de modifieringar som har gjorts på radarmodellen.

Appendix: "Klasser" beskriver de klasser som har utvecklats för lvr-modellen.

Appendix: "Meddelanden" beskriver ledningssystemmeddelanden som skickas mellan de olika komponenterna i lvr-modellen vid målinvisning och vid robotavfyrning.

Appendix: "Parametrar" beskriver de parametervärden som kan användas för att konfigurera en fungerande lvr-systemmodell, referens till kapitel 4.

Text med typsnittet Courier används för att beteckna programkod i modellen.
(exempel: `EWTRMSlCtrlUnit::preUpdate()`).

Text i *kursiv stil* refererar till klasser i *EWPlan* eller begrepp införda i dokumentet (exempel: *MslCtrlUnit*, *EWSim*).

Text inom citationstecken anger menyval eller parametrar i användargränssnitt (exempel: "Fire Radar Missile", "LOSadvance").

1.3 Tack till

Vi som har skrivit rapporten vill tacka Christer Wigren och Mats Tjäder för deras bidrag till modellen.

(Tom sida)

2 Radarstyrda lvr-system

Kapitlet fungerar som en kortare introduktion till radarstyrda luftvärnsrobotsystem och behandlar översiktligt komponenter i ett generellt radarstyrt lvr-system, olika radar-inmättningsprinciper och robotstyrprinciper, exempel på olika system, förbandsorganisation samt telekrigs möjliga påverkan på systemet.

2.1 Komponenter

Fysiska komponenter i ett radarstyrt lvr-system är typiskt:

- En spaningsradarplattform (TAR, Target Acquisition Radar)
- En eldledningsradarplattform (TTR, Target Tracking Radar)
- En eller flera robotavfyrningsplattformar (TEL, Transporter Erector Launcher)
- En eller flera robotar per robotavfyrningsplattform

Förkortningarna TAR, TTR och TEL kommer att användas frekvent i det fortsatta dokumentet.

Ett radarstyrt lvr-system kan vara konfigurerat med alla komponenter på en plattform eller med komponenterna geografiskt spridda på olika plattformar. Mobiliteten för system ökar då komponenterna samlas till en plattform. En plattform som bär en radar men även kan avfyra robotar kallas TELAR (Transporter Erector Launcher and Radar). Det ryska lvr-systemet SA-8 "Gecko"³/9K33 "Osa"⁴ består av en TELAR som samlar alla komponenter på en och samma plattform och är därför mycket mobilt (se Figur 2).



Figur 2 Det korträckviddiga (< 13 km) ryska lvr-systemet SA-8 "Gecko".

Det ryska lvr-systemet SA-6 "Gainful"³/2K12 "Kub"⁴ är däremot ett större, geografiskt utspjutt och därmed mindre mobilt system (se Figur 3). TAR och TTR är samlade till ett fordon medan robotarna vanligtvis avfyras från 4 stycken geografiskt utspridda TEL. Se kapitel 2.3 där en presentation görs hur ett sovjetiskt luftvärnsregemente var organiserat.

³ NATO-beteckning

⁴ Rysk beteckning



Figur 3 Det mediumräkviddiga (<28 km) ryska lrb-systemet SA-6 "Gainful". Bilden till vänster visar TAR/TTR. Den högra bilden visar en TEL med 3 robotar.

2.2 Operation

TAR har som uppgift att avsöka lufthavet kring robotsystemet och invisa TTR mot mål att bekämpa. TTR följer målet i vinkel (sida och höjd) samt i avstånd och/eller hastighet. På kommando från TTR avfyras en eller flera robotar från TEL mot målet.

Då roboten har avfyrats genomlöper den en startfas, en eventuell anflygningsfas och en målstyrningsfas. I målstyrningsfasen styr roboten mot målet med målinformation erhållen från egen målinmätning (inre målinmätning) eller t ex från avfyrande TTR (yttre målinmätning).

2.2.1 Startfas

Under startfasen som varar under några sekunder styrs vanligtvis inte roboten utan den skjuts ballistiskt under det att startsteg "booster" brinner för att få upp roboten i styrbara hastigheter.

2.2.2 Anflygningsfas

Anflygningsfasen syftar till att flytta roboten till ett läge i rymden där målstyrning kan inledas. Styrning under anflygningsfasen benämns navigering eller ej målrelaterad styrning. Speciellt långräkviddiga system nyttjar ofta navigering. Navigering kan ske t ex med tröghetsnavigering (TN) eventuellt stöttad med GPS.

2.2.3 Målstyrningsfas

För att få roboten att träffa målet används någon målinmättningsprincip och någon styrprincip. Målinmätningen kan ske utanför roboten och kallas då yttre målinmätning. Sker målinmätningen i roboten benämns den inre målinmätning. Styrprincip för yttre målinmätning är vanligtvis kollimationsstyrning. Styrprinciper för inre målinmätning kan vara hundkurvestyrning (fast målsökare) eller syftbäringsstyrning (rörlig målsökare). Yttre målinmätning implementeras med kommandostyrning eller ledstrålestyrning och inre målinmätning med passiv, aktiv eller semiaktiv målsökarstyrning⁵. Vid passiv målsökarstyrning har roboten endast en mottagare och roboten styrs mot den signal som målet själv genererar. Vid semiaktiv målsökarstyrning belyses målet av en belysarsändare. Roboten har en mottagare som detekterar den reflekterade signalen från målet och styrs mot denna signal. Vid aktiv

⁵ Kallas även Homing.

målsökarstyrning har roboten både en sändare och mottagare och fungerar därför som en flygande radar som leder sig själv mot målet (fire-and-forget). Tabell 1 sammanställer styrprinciper, realiseringar samt exempel på några radarstyrda lvr-system.

Inmätning	Robottyp	Princip	Realisering/exempelsystem
Yttre	Utan målsökare	Kollimationsstyrning (Siktlinjesstyrning, Line of Sight, LOS)	Kommandostyrning (CLOS) • Manuell (MCLOS) • Halvautomatisk (SACLOS) • Automatisk (ACLOS) ○ RBS 23 Bamse ○ SA-8 ○ SA-15
			Ledstrålestyrning (Beam Riding, LOSBR) (Ovanlig i radarluftvärnssammanhang)
Inre	Fast målsökare	Hundkurvestyrning (Pursuit Guidance)	Styr så att målet ligger mitt i den fasta målsökarens synfält. (Ovanlig i radarluftvärnssammanhang)
	Rörlig målsökare	Syftbäringsstyrning (Proportional Navigation)	Målsökarstyrning • Passiv (Ovanlig i radarluftvärnssammanhang) • Semiaktiv (SARH) ○ SA-6 ○ SA-11 ○ SA-17 • Track-Via-Missile (TVM) ○ SA-10 ○ SA-12 ○ MIM-104 Patriot • Aktiv ○ SA-5 ○ NASAMS (Norwegian Advanced Surface to Air Missile System) ○ SLAMRAAM

Tabell 1 Inmätningssprincip, robottyp, styrprincip, realisering och exempel på radarluftvärnsrobotsystem.

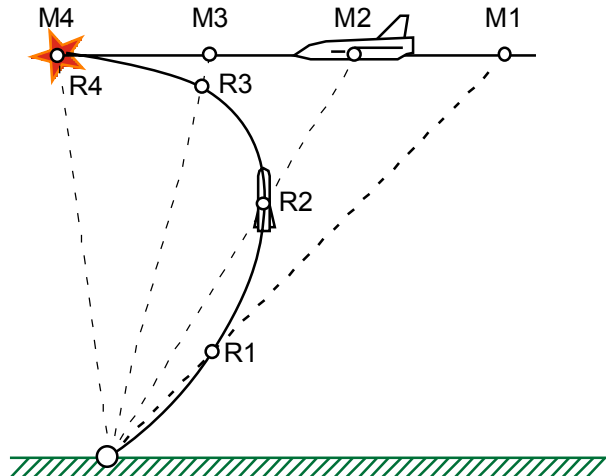
Se [7] och/eller [8] för beskrivning av luftvärnsrobotsystemen i Tabell 1.

I luftvärnssammanhang är kollimationsstyrning realiserad med kommandostyrning samt syftbäringsstyrning realiserad med semiaktiv målsökarstyrning vanligast. Dessa systemtyper kommer därför att presenteras översiktligt i kommande kapitel. En mer detaljerad beskrivning av styrprinciper ges i [9].

Modellen av den radarstyrda luftvärnsroboten som beskrivs från och med kapitel 3 representerar en kollimationsstyrd robot realiserad med kommandostyrning från en TTR.

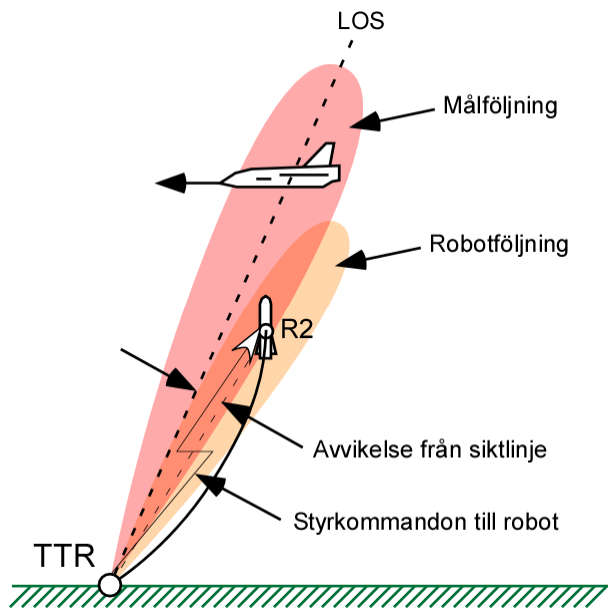
2.2.3.1 Kollimationsstyrning - kommandostyrning

Vid kollimationsstyrning (Line of Sight, LOS) eftersträvas styrning så att roboten hela tiden befinner sig på siktlinjen mellan styrutrustning och mål, se Figur 4.



Figur 4 Kollimationsstyrning. Roboten styrs till att ligga på linjen mellan styrutrustning och mål.

Vid kollimationsstyrning realiserad med kommandostyrning (CLOS, Commanded Line Of Sight) i radarsammanhang så följer TTR både målet och roboten, vilken vanligtvis följs med en separat robotföljning. Robotens vinkelavvikelse från siktlinjen TTR – mål mäts och styrorder beräknas och sänds via länk till roboten för att styra den mot siktlinjen, se Figur 5.



Figur 5 Kommandostyrning. Robotens avvikelse från siktlinjen mäts och styrkommandon beräknas och sänds till roboten för att styra den mot siktlinjen.

Fördelar respektive nackdelar med kollimationsstyrning realiserad med kommandostyrning är:

- + Roboten blir relativt enkel. Endast mottagare för styrorder samt utrustning för att omvandla styrorder till roderutslag (styrautomat).
- + Relativt störfast. Störningen måste ske mot mål- och/eller robotföljningen. Länken till roboten är svår att störa då mottagarantennerna i roboten är riktad mot styrkommandosändaren på marken. I många fall kan en människa stötta systemet om det störs.
- Krökta banor ger tämligen stora vinkelaccelerationer hos roboten och lång tid till träff.

- Onoggrannhet vid stora avstånd.
- Målföljning och styrsignalgenerering måste ske från markstationen under hela målstyrningsfasen.

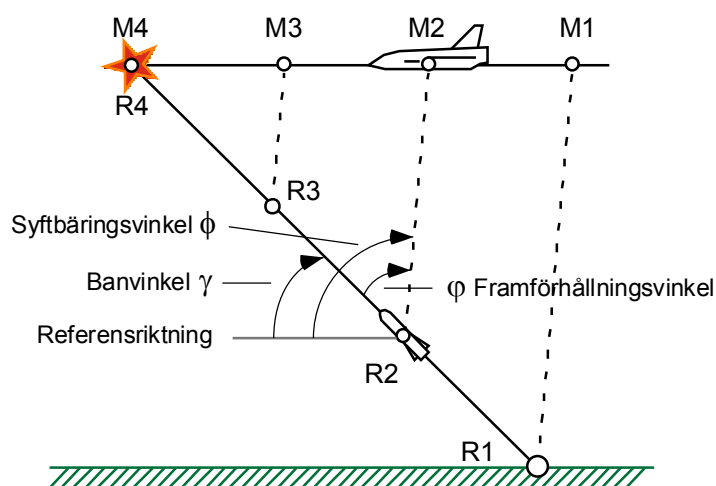
Det tidigare nämnda systemet SA-8 använder kollimationsstyrning realiserad med kommandostyrning, se Figur 6. I figuren ses att systemet har två robotföljeantennor och därför kan följa två robotar samtidigt. Dock kan det endast ske mot ett mål.



Figur 6 SA-8 "Gecko" med TAR-antenn, TTR-antenn, robotföljeantennor och 6 stycken robotar.

2.2.3.2 Syftbäringsstyrning - semiaktiv målsökarstyrning

Vid syftbäringsstyrning (Proportional Navigation, PN) eftersträvas en bana som går direkt mot den beräknade kollisionspunkten (framförpunkten). Principen bygger på uppmätning av vinkelhastigheten hos den så kallade syftbäringsvinkeln ϕ , se Figur 7.

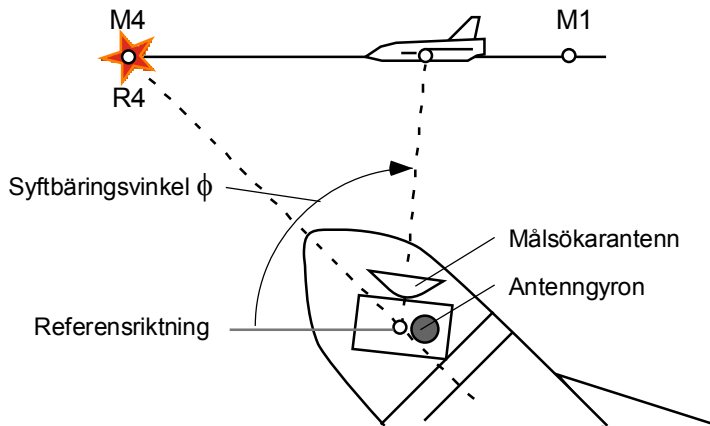


Figur 7 Syftbäringsstyrning.

Vid syftbäringsstyrning utnyttjas en målsökare med rörlig antenn. Med hjälp av denna och en styrdator i roboten styrs roboten så att syftbäringsvinkelhastigheten till målet går mot noll. Vid styrningen läggs en banvinkeländring på som är proportionell mot syftbäringsvinkelhastigheten:

$$\dot{\gamma} = c \cdot \dot{\phi}$$

Där $\dot{\gamma}$ är banvinkelhastigheten, c syftbäringskonstanten ($c > 1$) och $\dot{\phi}$ är syftbäringsvinkelhastigheten. $\dot{\phi}$ mäts i roboten med ett ratogyro i målsökaren, se Figur 8.



Figur 8 Syftbäringsvinkelhastigheten mäts med ett tungt gyro i målsökarantennen.

Vid konstanta hastighetsförhållanden och vid en rak kurs hos målet kommer roboten att svänga in till rak kollisionskurs (som visas i Figur 7). Målsökarens axel kommer därför att avvika från referensriktningen med den konstanta syftbäringsvinkeln.

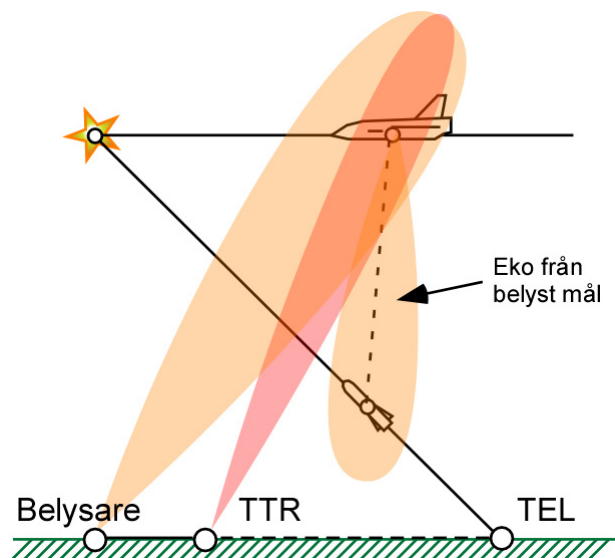
Ett lämpligt värde på syftbäringskonstanten c i styrlagen ovan beror på det taktiska skjutfallet och speciellt på närmandehastigheten, V_{rel} , mellan mål och robot. V_{rel} kan mätas ur ett eventuellt dopplerskift i radarsignalen eller skattas. Då man ofta vill kommendera roboten en tväracceleration a_M kan styrlagen ovan omformas till:

$$a_M = \alpha \cdot V_{rel} \cdot \dot{\phi}$$

Där:

- a_M kommenderad acceleration i tvärsled
- α navigeringskonstanten (vanligtvis i intervallet 3-5)
- V_{rel} närmandehastigheten mellan mål och robot, skattad eller mätt ur dopplerskift
- $\dot{\phi}$ syftbäringsvinkelhastigheten som mäts med ratogyro i målsökaren.

Syftbäringsstyrning implementeras ofta i robotar med semiaktivt målstyrda robotar. Vid semiaktiv målsökarstyrning följer en TTR målet som slavar en målbelysare att belysa målet. Roboten har en radarmottagare och följer ekot från det belysta målet. Se Figur 9.



Figur 9 Semiaktiv målföljning. Målet följs av TTR som slavar en belysningssändare att belysa målet. TTR kommenderar TEL att rikta in roboten mot en predikterad träffpunkt och avfyrar den. Robotens målsökare detekterar ekot från det belysta målet och vrider målsökarantennen för att följa målet. Syftbäringsvinkeln mäts och används för att styra målet mot en träffpunkt. I detta fall har både målet och roboten en konstant hastighet varför robotbanan blir rak.

Fördelar respektive nackdelar vid syftbäringsstyrning med semiaktiv målföljning är:

- + Höga sändareffekter kan tillåtas i belysaren, vilket ökar störfastheten hos systemet. Den mottagna signalen (ekot från belysarsignalen) ökar då roboten närmar sig målet vilket även ökar störfastheten och träffnoggrannheten.
- + Syftbäringsstyrning ger rakbanor i slutfasen.
- + God verkan mot snabba mål.
- Relativt komplicerad robot. Roboten innehåller en radarmottagare, en rörlig antennplattform, rategyro, styrdator samt vanligtvis en antenn i aktern för att kunna motta signalen direkt från belysaren för att kunna extrahera dopplerskiftet från målets rörelse relativt roboten.
- Målföljning och belysning måste ske från markstationen under hela målstyrningsfasen.

2.3 Förbandsorganisation

Ett luftvärnsförband kan bestå av ett flertal andra komponenter utöver TAR, TTR och TEL. Sådana kan vara ledningsfordon, fjärrspaningsradar (långräckviddig radar för förvarning), omladdningsfordon, förplägnadsfordon (kantin) samt andra luftvärnsvapen (t ex MANPADS⁶). Ett ledningssystem förbinder komponenterna för ordergivning och rapportering.

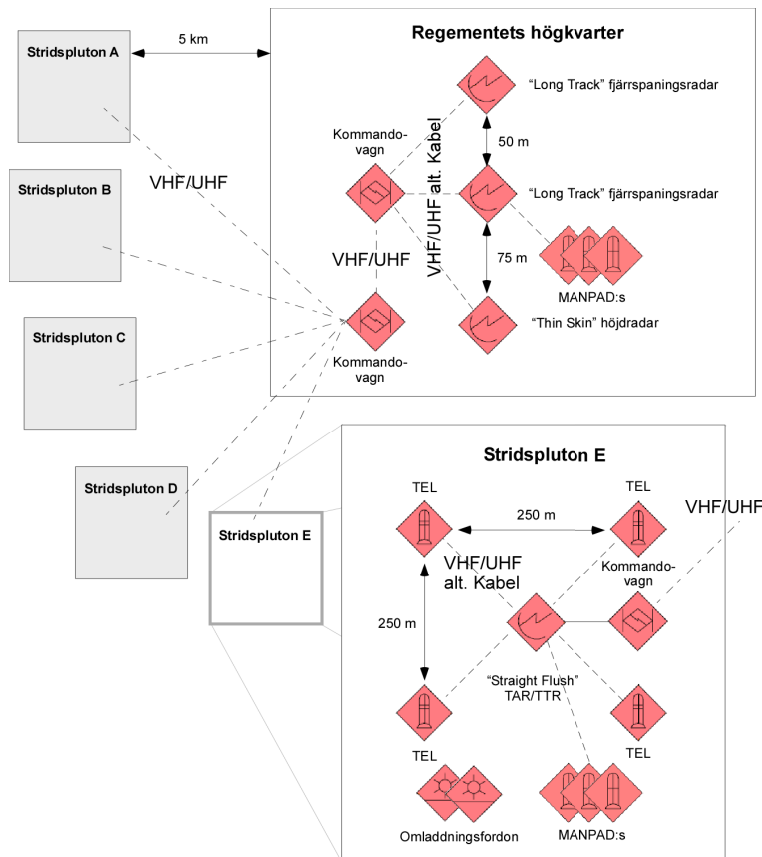
Som exempel bestod under Sovjettiden ett SA-8-luftvärnsregemente av bland annat [8]: 20 stycken SA-8 TELAR, 10 stycken fordon för robotlogistik, en ”Thin Skin B” och två ”Long Track”-radarer för fjärrspaning samt totalt 450 personer för all bemanning. Utöver SA-8-robotarna fanns även 21 stycken SA-7/SA-14 MANPADS spridda inom regementet. Invisning av dessa MANPADS kan stödjas av radarsystemen inom regementet. De stridande komponenterna i en normal pluton bestod av ett ledningsfordon, 4 stycken SA-8 TELAR samt

⁶ MANPADS, MAN-Portable Air-Defense Systems, [se http://en.wikipedia.org/wiki/MANPAD](http://en.wikipedia.org/wiki/MANPAD)

två robotomladdningsfordon. Kommunikationen inom plutonen skedde med VHF-radio. Kommunikationen på regementsnivå skedde med radio på HF-, VHF- och UHF-banden.

Ett SA-6-regemente bestod under Sovjettiden av bland annat [8]: 20 stycken TEL, 5 stycken TAR/TTR "Straight Flush", 10 stycken fordon för robotlogistik samt innehöll samma fjärrspaningsresurser som SA-8-systemet. Även fördelningen av MANPADS var samma som för SA-8 systemet. En stridande SA-6-pluton bestod av två ledningsfordon, en TAR/TTR "Straight Flush", 4 stycken TEL:ar samt 2 omladdningsfordon. Kommunikation mellan enheterna skedde på samma sätt som för SA-8-systemet. Kommunikation mellan TAR/TTR och TEL kan dock ske över kabelförbindelse.

Figur 10 visar organisation, geografisk placering och kommunikationssätt i ett sovjetiskt SA-6-regemente.



Figur 10 Organisation, geografisk placering och kommunikationer i ett sovjetiskt SA-6-regemente. Alla enheter visas ej.

Som visas i Figur 10 är det sovjetiska SA-6-förbandet komplext med många enheter. SA-6 (och SA-8, m.fl.) har exporterats till ett flertal länder, se kapitel 2.5. Många av dessa exportärenden har endast handlat om utrustning för stridsplutoner ("Straight Flush", TEL:ar, omladdningsfordon och MANPADS) [8]. I dessa länder kan därför avsevärt mindre enheter operera.

En stridsplutonenheter i SA-6-systemet är tämligen sårbart. Genom att slå ut "Straight Flush"-radarn i SA-6-systemet (med t ex telekrig) försvinner plutonens förmåga att avfira robotar. SA-8-systemet däremot är mer skyddat då varje stridspluton består av fyra fullständiga bekämpningssystem med egen TAR, TTR och TEL (4 stycken TELAR) som därmed kan operera mer självständigt.

Organisation och ledning för svenskt luftvärn beskrivs i [10].

2.4 Telekrig i ett lvr-b-förband

Telekrig mot ett luftvärnsförband kan rikta sig mot ledningsfunktionaliteten eller mot sensorsystemen.

Mot ledningssystemet kan telekrigåtgärder påverka kommunikationsnätet så att förbindelser bryts eller störs i den omfattningen att det blir obrukbart. Detta kan ske mot ett radiokommunikationsnät med kommunikationsstörning eller mot datanätet i ett modernt ledningssystem med hjälp av trojaner, som är planterade inom ledningssystemet och som med sin ”signalering” mättar⁷ kommunikationsnoder i nätet. Med falsk radiosignalering eller falska datameddelanden kan även vilseledande information spridas inom ledningssystemet. Störtåligheten ökar med ett robust system som t ex kommunicerar via nedgrävd kabel mellan enheterna och har inträngsdetektering för detektion av trojaner, virus, spyware, m.fl.

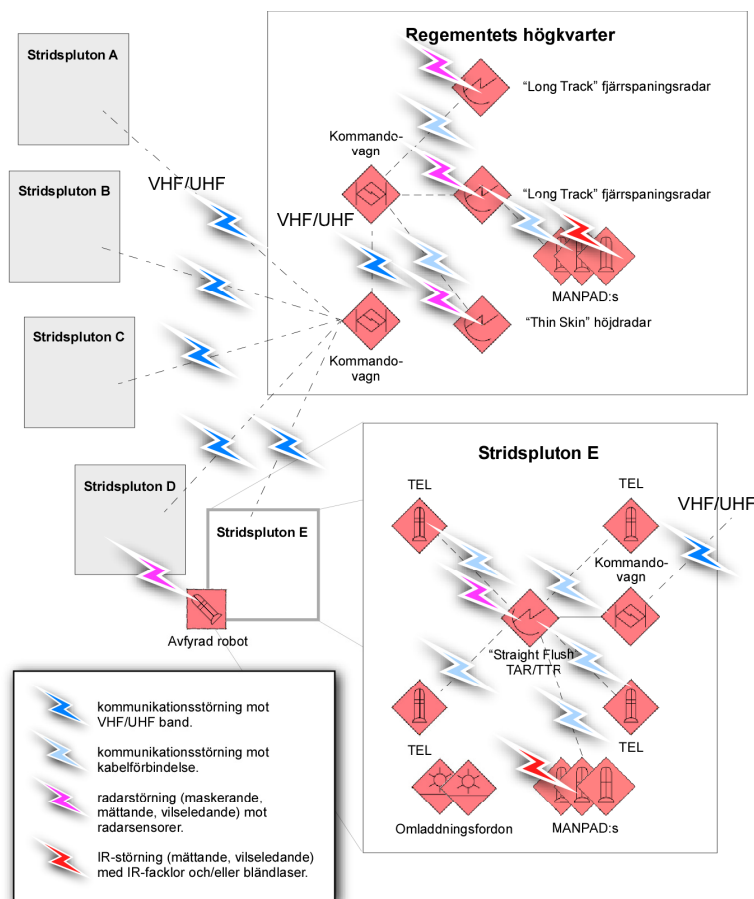
Störformer mot sensorsystemen kan vara av typen maskering, mättande eller vilseledande. Med maskering höjs bakgrundsbruset, med hjälp av brusstörning eller remskorridorer, för att dölja uppdraget. Med mättande störning skapas mängder av skenmål, med aktiv störning och/eller med preventiva rems- och fackelfällningar, som mättar sensorsystemet och/eller ökar felaktigt handlande i sensorsystemet. Med vilseledande störning avhakas sensorsystemets målföljning (t ex TTR:en tappar sin följning av målet). Vilseledande störning kan ske med t ex radarremsor och/eller aktiv störning (hastighets-, avstånds- och vinkelavhakning) och med t ex facklor och/eller bländlaser i IR-fallet. Mot en avfyrad luftvärnsrobot används vilseledande störning mot dess sensorsystem för att få den att missa sitt mål.

Figur 11 visar de störmöjligheter som kan finnas mot ett SA-6-luftvärnsregemente, se även Figur 10. Ledningssystemet bygger på stor del på VHF/UHF-kommunikation och är därför möjligt att störa med kommunikationsstörning. På de kortare sträckorna inom regementets högkvarter samt inom stridsplutonen mellan TAR/TTR och TEL:ar kan kommunikationen ske över kabel vilket ökar störtåligheten.

Maskering, mättande och vilseledande radarstörning med skenmål kan ske mot de tre radarsystemen i regementets högkvarter samt mot TAR/TTR i stridsplutonerna, för att dölja flygföretag eller skapa (onödig och uttröttande) påläsningsaktivitet inom förbandet. Avhakande störning kan ske mot TAR/TTR i stridsplutonerna för att haka av etablerad målföljning och därmed förhindra robotavfyrning. Vid avfyrate robotar kan avhakande störning ske mot robotens målsökare (mottagare) för att förhindra robotträff.

Som skydd mot de MANPADS som finns utspridda inom förbandet kan fackelfällning och/eller bländlaserstörning användas.

⁷ Denial-of-Service attack (DoS attack), se http://en.wikipedia.org/wiki/Denial-of-service_attack



Figur 11 Störmöjligheter mot ett sovjetiskt SA-6-regemente.

För en mer utvecklad beskrivning av telekrigföring inom luftvärnet, se [10].

2.5 Spridning

Radarluftvärnssystem finns exporterade vida över jorden. Vart olika radarluftvärnssystem har exporterats, har sammanställts från öppna källor [7, 11] och sorterats mot den subregionindelning [12] som definieras enligt FN. Figur 12 visar subregionerna.



Figur 12 Subregioner enligt FN.

Tabell 2 visar för några radarluftvärnssystem om dessa har exporterats till subregionerna. Tabellen visar ej huruvida systemen är operativa eller ej inom subregionen. I en del länder

kan systemen ha fallit för ålderssträcket (eller av annan anledning tagits ur tjänst), vissa system har uppdaterats och är fortfarande reella hot medan andra är nyligen⁸ exporterade.

Region	SA-2	SA-3	SA-4	SA-6	SA-8	SA-10	SA-11	SA-12	SA-15	2S6	ZSU-23/4	Crotale	HAWK
Northern America													X
Central America													
South America	X	X							X		X	X	
Caribbean	X	X		X	X						X		
Northern Europé		X					X					X	X
Western Europé					X	X						X	X
Southern Europé	X	X		X	X	X	X		X		X	X	X
Eastern Europé	X	X	X	X	X	X	X	X	X	X	X		
Western Asia	X	X	X	X	X	X	X				X	X	X
Central Asia	X	X			X						X		
Southern Asia	X	X		X	X	X	X	X	X	X	X	X	X
Eastern Asia	X	X	X	X	X	X	X	X	X		X	X	X
Southeastern Asia	X	X		X							X	X	X
Northern Africa	X	X		X	X						X	X	X
Western Africa		X									X		
Middle Africa	X	X		X	X						X		
Eastern Africa	X	X		X							X		
Southern Africa												X	

Tabell 2 Export av radarluftvärnssystem till länder inom olika subregioner.

Se [7, 8] för beskrivning av luftvärnsrobotsystemen i Tabell 2.

Som ses i Tabell 2 är de listade radarluftvärnssystemen exporterade till länder i södra och östra Europa, hela Asien och i mellersta och norra Afrika. Dessa subregioner ligger inom den aktionsradie som den Europeiska Unionens snabbinsatsstyrka har att kunna verka inom. Det är därför viktigt att kunna värdera luftvärnshotet och möjligheten att bekämpa detta med bland annat telekrig, för att kunna utveckla effektiva skydd för t ex strategiskt (C-17) och taktiskt transportflyg (C130-Hercules).

⁸ T ex enligt <http://www.aljazeera.com/news/newsfull.php?newid=131047> har Iran nyligen (2008) köpt SA-15 av Ryssland.

(Tom sida)

3 Design, modellering och implementation

Detta kapitel ger en teknisk beskrivning av hur det radarstyrda lvr-systemet har designats, modellerats och implementerats. Kapitlet beskriver de olika komponenterna i systemet samt den ledningssysteminteraktion som har definierats mellan komponenterna. I slutet av kapitlet presenteras designförslag för ett lvr-system med en semiaktivt styrd robot.

De klasser som har utvecklats för lvr-modellen beskrivs i Appendix: "Klasser". Appendix: "Meddelanden" beskriver de ledningssysteminteraktioner som sker i lvr-systemet från målinvisning till robotavfyrning.

Modellen är utvecklad i C++ (Microsoft Visual Studio 2008) och är ämnad att exekvera i simuleringsramverket *EWSim*. Den modell som beskrivs är revision 4779 i SubVersion⁹.

Kapitlet förutsätter en viss kännedom om programmering.

3.1 Modellerade komponenter

De komponenter som har definierats för att modelleras i lvr-systemet är TAR, TTR, TEL och robotar, se kapitel 2.1. Uppdelningen i TAR, TTR och TEL har gjorts för att göra strukturen generell. Denna uppdelning är vanlig i lvr-system där semiaktiv målsökarstyrning används. Vid kollimationsstyrning är det vanligare att TTR och TEL är en och samma plattform. Orsaken till detta är att roboten får göra stora manövrar för att styra in på eldledningsradarns siktlinje, om den avfyras från en annan plattform. I kapitel 4.3 visas hur TEL konfigureras på separat plattform samt samlokaliseras med TTR.

Komponenterna kommunicerar med varandra och skickar ordrar och rapporter som meddelanden över ett ledningssystem. Appendix: "Meddelanden" beskriver de meddelanden som skickas mellan komponenterna vid målinvisning och robotavfyrning. För en detaljerad beskrivning av meddelandehantering i *EWSim* se [13, 14].

3.1.1 TAR

Den TAR som används i lvr-systemet är sedan tidigare implementerad i *EWSim*, se [1]. Mindre vidareutvecklingar har skett i radarmodellen, se Appendix: "Modifieringar av radarmodellen".

TAR:en består av en plattform med grundläggande egenskaper som t ex position, orientering, maximal hastighet, maximal rotationshastighet. Som utrustning på plattformen finns en spaningsradar, en spaningsradarantenn samt en ledningsfunktion (C2) för att sända ordrar och motta rapporter. I kapitel 4.1 presenteras hur en TAR konfigureras.

Spaningsradarn har ett användargränssnitt i form av ett PPI¹⁰. Från användargränssnittet kan användaren spana av omgivningen, markera mållägen och invisera eldledningsradarn mot dessa. Invisningen sker genom att en order skickas (ORDER_TRACK_POS) från TAR till TTR med målpositionen som argument, se Appendix: "Meddelanden". Invisning från användargränssnittet innebär inte att roboten avfyras. Med hjälp av spaningsradarns ledningsvy ("Command and Control"-vy) kan däremot TTR:en invisas och roboten avfyras direkt. I kapitel 5 presenteras hur spaningsradarn används för att invisera eldledningsradarn samt hur hela kedjan från invisning till avfyrning kan ske från spaningsradarns ledningsvy.

⁹ SubVersion är det versionskontrollsystem som används i *EWSim*, se <http://subversion.tigris.org/>.

¹⁰ PPI - planpolär indikator

Den modellerade spaningsradarn har 3D-kapacitet men simuleras att endast ha 2D-kapacitet i denna konfiguration, vilket innebär att den ej har höjdinformation till målet. Efter invisningen måste därför eldledningsradarn själv söka efter målet i höjddled (elevation).

3.1.2 TTR

Den TTR som används i lvr-systemet är sedan tidigare implementerad i *EWSim*, se [1]. Mindre vidareutvecklingar har skett i radarmodellen, se Appendix: ”Modifieringar av radar-modellen”. Nyutveckling har däremot skett för att hantera interaktioner med den avfyrande plattformen samt för styrningen av roboten.

TTR:en består av en plattform med samma grundläggande egenskaper som för TAR:en. Som utrustning på plattformen finns en eldledningsradar, en ledningsfunktion (C2), en utrustning för att styra robotar mot målet (*TTRMslCtrlUnit* - TTR Missile Control Unit), en enkel modell av en operatör¹¹ samt ett i azimut roterbart torn. Detta torn har ett vridbord rörligt i azimut och elevation som håller radarns antenn. Tornet kan även ha ett eldrör (*Barrel*) som är rörligt i elevationsled och som i sin tur kan ha en lavett (*RadarMslLauncher* – Radar Missile Launcher) för att avfyr robotar. En hierarkisk konfiguration plattform – torn – eldrör – lavett – robot är den normala konfigurationen i ett kollimationsstyrt lvr-system. I kapitel 4.2 presenteras hur en TTR konfigureras.

Enheterna *RadarMslLauncher* samt *TTRMslCtrlUnit* är de viktigaste funktionerna för avfyrning av en robot. *RadarMslLaunchers* plats på eldröret definierar varifrån roboten avfyras (lavetten). Maximalt antal robotar, aktuellt antal robotar på lavetten samt tid för omladdning definieras och hanteras i *RadarMslLauncher*. *TTRMslCtrlUnit* är, i det kollimationsstyrda robotsystemet, alltid placerad på TTR:en och har som uppgift att styra roboten mot siktlinjen och har därför parametrar för robotstyrningen. Se kapitel 4.3.1 för konfigurering av robotavfyrning från TTR.

Då målkoordinater tagits emot från spaningsradar går radarn till påläsningsmod varvid dess piedestal vrids in i målbäringen och dess antenn börjar en nickande rörelse i elevation för att hitta målet (spaningsradarn simuleras att ej skicka målets höjd). Om målet detekteras övergår radarn till följemod och börjar följa. Följningen sker med hjälp av tyngdpunktsföljning, vilket är en förenklad modell av radarföljning, se [15].

Roboten avfyras från eldledningsradarn med menyalternativet ”Fire Radar Missile” i eldledningsradarns ”Objectlist”-meny. Menyalternativet skapar en kedja av ordrar och rapporter i ledningssystemet mellan TTR och TEL som slutligen leder till att en robot avfyras. Avfyrningskommandot leder till att:

1. En order skickas från TTR:en till alla underställda TEL:ar att de ska rapportera huruvida de kan avfyr en robot mot målet.
2. De TEL:ar som har möjlighet att avfyr rapporterar sina positioner till TTR:en.
3. TTR beräknar vilken av TEL:arna som har bäst möjlighet att bekämpa målet.
4. En order skickas till den valda TEL:en att vrida in sin *RadarMslLauncher* i riktning mot målet.
5. TEL:en rapporterar till TTR:en när *RadarMslLauncher* är inriktad mot målet.
6. TTR:en beordrar TEL:en att avfyr roboten.
7. TTR:en styr roboten från *RadarMslLauncher* mot målet längs siktlinjen, se kapitel 3.1.4 för beskrivning av robotstyrningen.

¹¹ Radaroperatörsmodellen definierar yttre och inre skjutgränser, fördröjningar vid avfyrning, m fl.

Se kapitel 9.1 i Appendix: ”Klasser”, som beskriver processen samt Appendix: ”Meddelanden”, som beskriver meddelandena. I kapitel 5 presenteras hur eldledningsradarn används för att avfyr roboten.

Som nämnts i kapitel 3.1.1 kan roboten även avfyras direkt från spaningsradarns ledningsvy. I detta fall så skickas `ORDER_TRACK_POS` från TAR till TTR och meddelandekedjan, i föregående stycke, startas direkt när TTR:en övergår till följemod.

3.1.3 TEL

Normalt så används inte en TEL i ett kollimationsstyrt robotsystem, utan TEL-funktionaliteten samgrupperas med den plattform som tillhandahåller siktlinjen (vanligtvis TTR:en), se kapitel 3.1.2. För att göra strukturen generell och bädda för framtida utvecklingar så tillåter designen att en separat TEL-plattform kan skapas och roboten avfyras därifrån.

TEL:en består av en plattform med samma grundläggande egenskaper som för TAR:en och TTR:en. Som utrustning på plattformen finns en ledningsfunktion (C2) och ett i azimut roterbart torn. Detta torn har ett eldrör (*Barrel*) som är rörligt i elevationsled. Utrustning på eldröret är lavetten (*RadarMslLauncher*) för avfyring av robotar. *RadarMslLaunchers* plats på eldröret definierar varifrån roboten avfyras. Maximalt antal robotar, aktuellt antal robotar på lavetten samt tid för omladdning definieras och hanteras i *RadarMslLauncher*. Se kapitel 4.3.2 för konfigurering av TEL.

TEL:en har inget användargränssnitt, utan all hantering sker via ordrar och rapporter mot TTR:en. Från det att TTR:en beordrar avfyring av en robot skickas meddelanden mellan TTR och TEL över ledningssystemet. Se Appendix: ”Meddelanden” för beskrivning av dessa meddelanden.

3.1.4 Robot

Roboten som nyttjas i robotsystemet bygger på en tidigare utvecklad trådstyrd IR-robot, se [3].

Roboten skapas dynamiskt vid avfyrningsögonblicket på TEL:ens lavett (*RadarMslLauncher*) och styrs mot målet längs eldledningsradarns siktlinje (LOS), det vill säga i dess antennriktning. Den algoritm som används för att styra roboten mot siktlinjen är implementerad i medlemsfunktionen `calculateSteerSignal` i klassen `EWTRMslCtrlUnit`.

Roboten styrs mot siktlinjen med en styrsignal i Yaw (sida) och en i Pitch (höjd). Styrningen i Yaw och Pitch är identiska varför endast styrningen i Yaw beskrivs nedan.

Styrsignalen i Yaw definieras som:

$$\psi_{steer} = \psi_{Antenna} - \psi_{Missile} + \beta$$

Där:

$\psi_{Antenna}$ är eldledningsradarns antennutvridning i sida.

$\psi_{Missile}$ är robotkroppens orientering i sida.

β är en vinkel som svarar mot robotens avvikelse från LOS. β är beroende av robotens avstånd till LOS samt parametern `LOSadvance`, se kapitel 4.2. β beräknas ur:

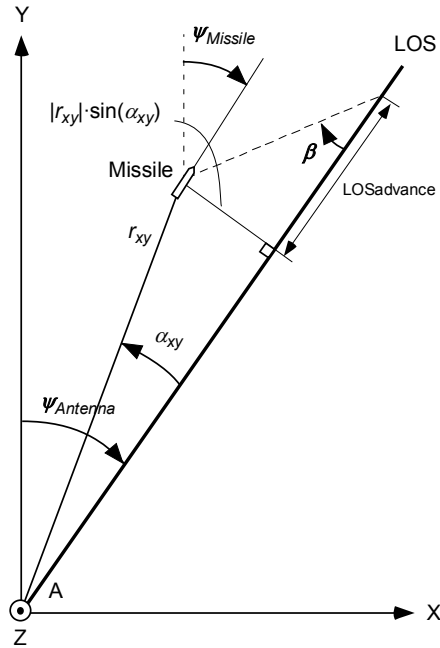
$$\beta = -\arctan\left(\frac{|r_{xy}| \cdot \sin(\alpha_{xy})}{LOSadvance}\right)$$

Där:

$|r_{xy}|$ är avståndet till roboten i xy-planet.

α_{xy} är vinkeln mellan LOS och vektorn till roboten i xy-planet.

β är den faktor som får roboten att närma sig LOS. Utan denna skulle roboten kunna flyga parallellt med siktlinjen. Ett stort värde på parametern $LOS_{advance}$ minskar inverkan av β , dvs leder till en långsammare insvängning mot siktlinjen.



Figur 13 Styrsignalparametrar i Yaw-led.

Styrsignalen påverkar robotens vinkelhastighet om styrsignalen ligger utanför ± 0.1 radianer. Är styrsignalen utanför detta vinkelintervall så styrs roboten med bang-bang styrning genom att maximal vinkelhastighet appliceras på robotkroppen mot siktlinjen. $LOS_{advance}$ samt maximal vinkelhastighet ($MslYawDotMax$ respektive $MslPitchDotMax$) sätts i `TTRMslCtrlUnit` i `EWPlan`, se kapitel 4.2.

I koden motsvaras `steerSignalYaw` av ψ_{steer} , `AntennaYaw` av $\psi_{Antenna}$, `MslYaw` av $\psi_{Missile}$ och `beta_xy` av β .

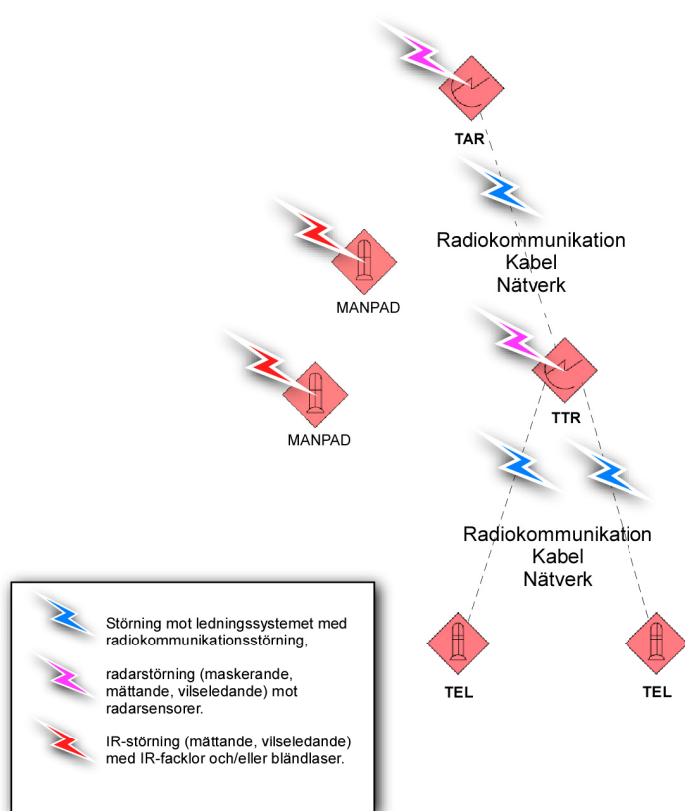
3.2 Telekrig i den implementerade modellen

Den utvecklade modellen av ett radarstyrt lrb-system kan påverkas av telekrig på ett flertal sätt.

- Mot TAR:en kan repeterstörning ske för att generera falskmål eller brusstörning för att dölja riktiga mål i den förhöjda "brusbakgrunden". Passiva störtekniker som t ex remsor kan skapa skenmål som ökar belastningen för TAR-operatören. Med en effektiv störning kommer TAR-operatören att misslyckas med att invisa mål till underställda TTR:er.
- Eventuell radiokommunikation mellan TAR och TTR kan störas med kommunikationsstörning så att invisningsordern från TAR ej uppfattas i TTR:en.

- Infångningsförloppet i TTR:en kan störas med avhakande repeterstörning och/eller proaktiv (påhakande) remsfällning som skapar skenmål som radarn felaktigt låser på. Störningarna kan leda till att förnyade invisningar får begäras från TAR:en.
- Följefasen i TTR:en kan störas med avhakande repeter- och brusstörning samt avhakande remsfällning.
- Eventuell radiokommunikation mellan TTR och TEL kan störas så att hela eller delar av meddelandekedjan för avfyrning av roboten störs så att roboten ej kan avfyras. En sådan störning blir dock mycket svår i fallet där TTR och TEL är samgrupperade (TELAR).
- Den kollimationsstyrda robotens flykt mot målet kan störas med störning som påverkar TTR:ens siktlinje till målet, se störning av TTR:ens följefas ovan.

Lvr-systemet kan ingå i ett större scenario i *EWSim* där även optiska sensorer (t ex MANPADS) och telekrig mot dessa simuleras, se Figur 14.



Figur 14 Exempel på störningsmöjligheter mot lvr-systemmodellen samt övriga luftvärnssystem inom ett mindre förband.

Styrkommandolänken mellan TTR och roboten är svårstörd och simuleras för närvarande inte. Robotens zonrör samt påverkan på det från zonrörsstörning (radarzonrör) eller från remsstörning simuleras ej heller i nuläget.

3.3 Designförslag semiaktivt styrd robot

Parallellt med utvecklingen av den kollimationsstyrda roboten har designarbetet för den syftbäringsstyrda roboten påbörjats. Mycket av designen är gemensam för de båda typerna av robotsystem. De ingående komponenterna är i stort sett samma, men är placerade annorlunda. Interaktionerna mellan komponenterna sker på ett snarligt sätt:

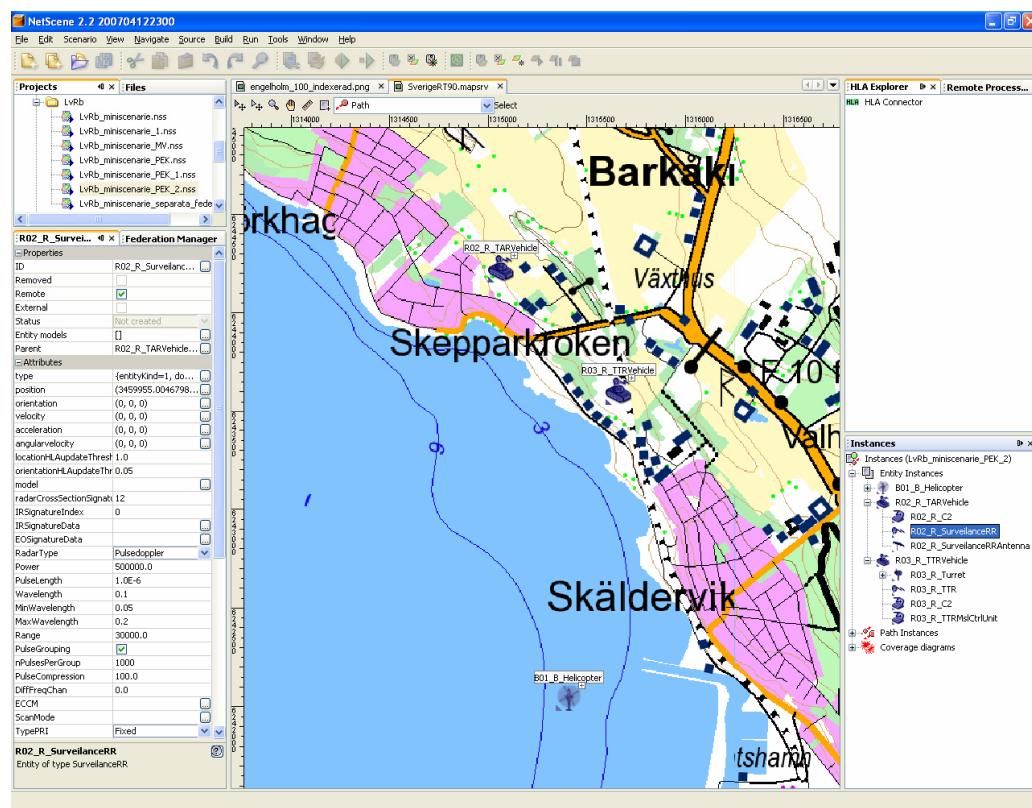
1. TAR visar in TTR som upprättar följning av målet.
2. TTR frågar om tillgängliga TEL:ar. TEL och TTR är i det semiaktiva fallet inte samlokaliserade i samma plattform.
3. TTR väljer mest lämplig TEL och beordrar inriktning, inväntar kvittens, beordrar avfyrning.

Den stora skillnaden mot ett kollimationsstyrt robotsystem är hur roboten styrs. I det semiaktiva fallet är den vanligaste styrprincipen syftbäringsstyrning och målinformationen tas från en mottagarantenn som sitter på roboten. Detta kräver av radarmodellen att bistatiska fall kan hanteras, vilket inte är fallet i nuläget. Inriktningen av lavetten innan avfyrning sker mot en predikterad träffpunkt, som beror på hur målet rör sig. Detta måste modelleras. Målsökaren (mottagarantennen) är hela tiden låst mot målet.

En följd av att styrlogiken i ett semiaktivt robotsystem sitter i roboten, är att klassen `EWTRMSlCtrlUnit` bör delas i två klasser, en som ligger under roboten och hanterar styrning av roboten och en som ligger på TTR:en och hanterar kommunikation mellan de olika komponenterna av robotsystemet innan avfyring.

4 Konfiguration

Detta kapitel ska ses som en vägledning hur ett lvr-system med en kollimationsstyrd robot realiserad med kommandostyrning konfigureras i *EWPlan*. Figur 15 visar scenariot som konfigureras i *EWPlan*.



Figur 15 EWPlan. Ett scenario med en TAR, en TTR samt en helikopter som mål.

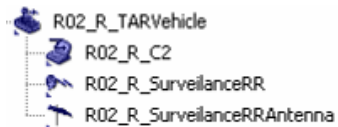
Beskrivningen av konfigurationerna i detta kapitel visar plattformarna då ideal kommunikation (trådbunden) nyttjas inom lvr-systemet. Ska kommunikationen ske över simulerad radio måste en radiokanal (typ *RadioChannel*), ett radionät (typ *RadioNet*) samt en radio (typ *RadioNode*) som underenhet till plattformens C2-enhet definieras i *EWPlan*. Simuleringen måste även inkludera en *ComNet*-federat som simulerar radiotrafiken.

Plattformarna är uppbyggda hierarkiskt där grundobjektet bär övrig utrustning och ger möjlighet till förflyttning. Utrustning sitter placerad relativt föräldraobjektet¹².

4.1 Konfigurering av TAR

TAR används för att visa in TTR mot målet. Figur 16 visar ett exempel på en TAR konfigurerad i *EWPlan*.

¹² Förälderobjekt är objekt närmast ovanför i den hierarkiska strukturen.



Figur 16 Exempel på en TAR konfigurerad i EWPlan. Prefixet "R02_R_" anger federatnamnet (R02) samt den sida plattformen tillhör (R=röd sida).

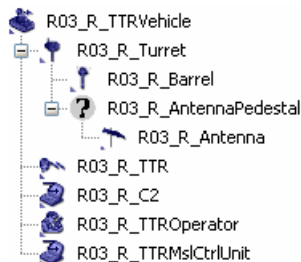
- *R02_R_TARVehicle* (av typen *FightingRadarVehicle*)
definierar plattformen med bland annat dess position, orientering, maximala hastighet, maximala rotationshastighet. Dess "type" definierar dess signatur samt vilken 3D-modell som används i simuleringen för att representera plattformen.
Utrustning på *TARVehicle* är:
 - *R02_R_C2* (typ *C2*)
Definierar ledningsfunktionaliteten (Command and Control) på plattformen. Då meddelanden skickas mellan plattformarna är det *C2*-objektet som har att skicka och ta emot meddelanden från andra federater, se Appendix: "Meddelanden". *C2*-objektet är föräldraobjekt till en *RadioNode* om radiokommunikation simuleras.
 - *R02_R_SurveillanceRR* (typ *SurveillanceRR*)
Definierar radarparametrarna hos spaningsradarn.
 - *R02_R_SurveillanceRRAntenna* (typ *SurveillanceRRAntenna*)
Definierar spaningsradarns antenn.

Parametrar (attribut) tillhörande TAR och dess utrustning visas i Appendix: "Parametrar", Figur 37 – Figur 40. För beskrivning av parametrarna se [1].

4.2 Konfigurering av TTR

I ett kollimationsstyrt radarstyrt lvr-system har TTR som uppgift att tillhandahålla siktlinjen (LOS) samt vid kommandostyrning att beräkna och sända styrorder till roboten så den styr mot siktlinjen. Siktlinjen fås av antennens pekriktning under målföljningen och styrorder beräknas i underenheten *TTRMslCtrlUnit* enligt kapitel 3.1.4. Kommunikationen till roboten simuleras ej.

Figur 17 visar ett exempel på en TTR konfigurerad i *EWPlan*.



Figur 17 Exempel på en TTR konfigurerad i EWPlan. Prefixet "R03_R_" anger federatnamnet (R03) samt den sida plattformen tillhör (R=röd sida).

- *R03_R_TTRVehicle* (av typen *FightingRadarVehicle*)
Definierar plattformen med bland annat dess position, orientering, maximala hastighet, maximala rotationshastighet. Dess "type" definierar dess signatur samt vilken 3D-modell som används i simuleringen för att representera plattformen.
Utrustning på *TTRVehicle* är:

- *R03_R_Turret* (typ *Turret*)
Definierar ett torn på TTR som kan rotera i azimutled med parametrar för bland annat rotationshastighet och vinkelbegränsningar (i azimut).
Utrustning på Turret är:
 - *R03_R_Barrel* (typ *Barrel*)
Definierar ett eldrör eller snarare en i elevation rörlig mekanisk konstruktion. *Barrel* kan utrustas med en *ProjectileLauncher* (för att avfyr projektiler) och/eller *RadarMslLauncher* (för att avfyr radarstyrda robotar). Hur *Barrel* konfigureras med en *RadarMslLauncher* för att skapa en samlokaliserad TTR/TEL presenteras i kapitel 4.3.1.
 - *R03_R_AntennaPedestal* (typ *AntennaPedestal*)
Definierar ett vridbord rörligt i azimut och elevation.
Utrustning på *AntennaPedestal* är:
 - *R03_R_Antenna* (typ *Antenna*)
Definierar eldledningsradarns antenn. Samma antenn används både vid sändning och vid mottagning.
- *R03_R_TTR* (typ *ZSU23 TTR*)
Definierar radarparametrarna hos eldledningsradarn.
- *R03_R_C2* (typ *C2*)
Definierar ledningsfunktionaliteten (Command and Control) på plattformen. Då meddelanden skickas mellan plattformarna är det *C2*-objektet som har att skicka och ta emot meddelanden från andra federater, se Appendix: ”Meddelanden”. *C2*-objektet är föräldraobjekt till en *RadioNode* om radiokommunikation simuleras.
- *R03_R_TTROperator* (typ *TTROperator*)
Definierar en förenklad radaroperatör.
- *R03_R_TTRMslCtrlUnit* (typ *TTRMslCtrlUnit*)
I TTR är *TTRMslCtrlUnit* den enhet som har att styra roboten mot siktlinjen. Är radarn störd kommer siktlinjen att avvika från linjen TTR – mål och roboten att förmodligen missa. I enheten definieras parametrar för roboten som armeringsavstånd, briseringsavstånd, livstid, brinntid, robothastighet samt parametrar för styrningen som ”LOSadvance”, ”MslYawDotMax” och ”MslPitchDotMax”, se Figur 18.

Attributes	
type	{entityKind=0, domain=0, c...
position	(3448070.228189689, 7940...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
ArmDistance	30.0
DetonateDistance	10.0
LifeTime	20.0
EngineBurnTime	18.0
Speed	200.0
Acceleration	-27.0
LOSadvance	10.0
MslYawDotMax	5.0
MslPitchDotMax	5.0

Figur 18 Ett exempel på konfigurering av TTRMslCtrlUnit.

Parametrar (attribut) tillhörande TTR och dess utrustning visas i Appendix: ”Parametrar”, Figur 41 – Figur 49. För beskrivning av parametrarna se [1].

4.3 Konfigurering av TEL

Robotavfyrningen kan konfigureras att ske antingen från ett eldrör (*Barrel*) på den målföljande TTR:en eller från ett eldrör på en separat plattform (TEL). Som nämnts är det förstnämnda alternativet vanligast vid avfyrning av kollimationsstyrda robotar.

Enheten *RadarMslLauncher* är den utrustning som definierar robotavfyrningsplatsen (lavetten). RadarMslLauncher måste positioneras på en *Barrel* som tillhandahåller position samt orientering för roboten vid avfyringen. Figur 19 visar parametrar för *RadarMslLauncher*.

Attributes	
type	{entityKind=0, domain=0, co...
position	(3459889.8286885708, 7877...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
Number of missiles	4
Reload time	300.0
Max speed	5.0

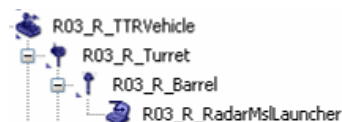
Figur 19 Ett exempel på konfigurering av RadarMslLauncher.

”Number of missiles” definierar hur många robotar som maximalt kan laddas på lavetten. ”Reload time” är den tid som en omladdning från tom till fulladdad lavett tar. Under simuleringen räknas en räknare ned för varje avfyrd robot. Då den sista roboten på lavetten

avfyras påbörjas omladdning. En TEL kan avfyras om lavetten är laddad med minst en robot, siktlinjen till målet är fri samt att TEL:en färdas i en hastighet lägre än "Max speed", se `osgRadarMissileLauncher::ableToFire()` i Appendix: "Klasser", kapitel 9.2.

4.3.1 TEL samlokaliserad med TTR

TEL samlokaliseras enkelt med TTR:en genom att skapa en *RadarMslLauncher* som underobjekt till TTR:ens eldrör (*Barrel*-objektet) i *EWPlan*, se Figur 20. Vid avfyring kommer roboten att orienteras och positioneras som *Barrel*. Då roboten lämnar *RadarMslLauncher* tar *TTRMslCtrlUnit* (se kapitel 4.2) över och styr roboten mot TTR:ens siktlinje.



Figur 20 Exempel med *RadarMslLauncher* konfigurerad som underenhet på TTRVehicle:s *Barrel*. Prefixet "R03_R_" anger federatnamnet (R03) samt den sida plattformen tillhör (R=röd sida).

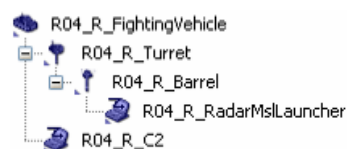
Övriga enheter på TTR-plattformen visas i Figur 17 och beskrivs i kapitel 4.2. Parametrar för *RadarMslLauncher* visas i Figur 19.

Meddelanden, se Appendix: "Meddelanden", mellan TTR och "TEL" går i detta fall internt på TTR-plattformen.

4.3.2 TEL som separat plattform

För att göra strukturen generell kan TEL definieras på en separat plattform. Vid semiaktiv målföljning är det vanligt att målet följs från en plattform och roboten avfyras från en annan. I simuleringen blir då TEL:en en egen federat.

Figur 21 visar ett exempel på en TEL konfigurerad som en separat plattform i *EWPlan*.



Figur 21 Exempel på en TEL konfigurerad som en separat plattform i *EWPlan*. Prefixet "R04_R_" anger federatnamnet (R04) samt den sida plattformen tillhör (R=röd sida).

- *R04_R_FightingVehicle* (av typen *FightingVehicle*)
Definierar plattformen med bland annat dess position, orientering, maximala hastighet, maximala rotationshastighet. Dess "type" definierar dess signatur samt vilken 3D-modell som används i simuleringen för att representera plattformen. Utrustning på *FightingVehicle* är:
 - *R04_R_Turret* (typ *Turret*)
Definierar ett torn på TTR som kan rotera i azimutled med parametrar för bland annat maximal rotationshastighet och vinkelbegränsningar (i azimut). Utrustning på *Turret* är:
 - *R04_R_Barrel* (typ *Barrel*)
Definierar ett eldrör eller snarare en i elevation rörlig mekanisk konstruktion. *Barrel* utrustas med en:

- *R04_R_RadarMslLauncher* (typ *RadarMslLauncher*)
Definierar plats för avfyrning av en radarstyrd robot.
Parametrar för *RadarMslLauncher* presenteras i Figur 19.
- *R04_R_C2* (typ *C2*)
Definierar ledningsfunktionaliteten (Command and Control) på plattformen.
Då meddelanden skickas mellan plattformarna är det *C2*-objektet som har att skicka och ta emot meddelanden från andra federater, se Appendix: "Meddelanden". *C2*-objektet är föräldraobjekt till en *RadioNode* om radiokommunikation simuleras.

Parametrar (attribut) tillhörande TEL och dess utrustning visas i Appendix: "Parametrar", Figur 50 – Figur 54.

Om en kollimationsstyrd robot ska avfyras, så måste TEL och TTR stå tämligen nära varandra för att roboten inom rimlig tid ska hinna styra in mot siktlinjen och träffa målet.

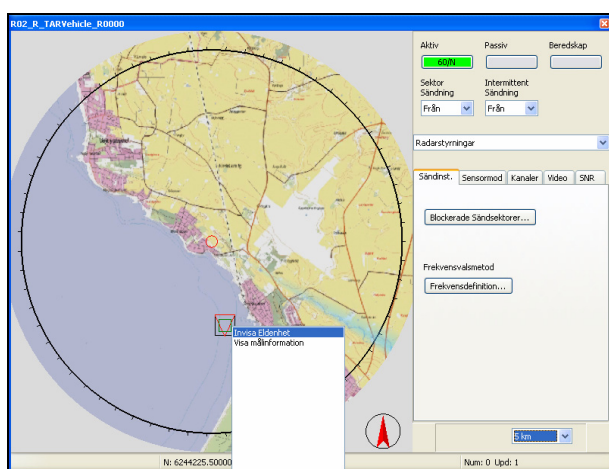
5 Användning

Avfyrning av roboten i *EW Sims duellverktyg* kan ske på två sätt:

- sekventiellt med invisning av TTR från TAR och avfyrning från TTR.
- direkt från spaningsradarns ledningsvy.

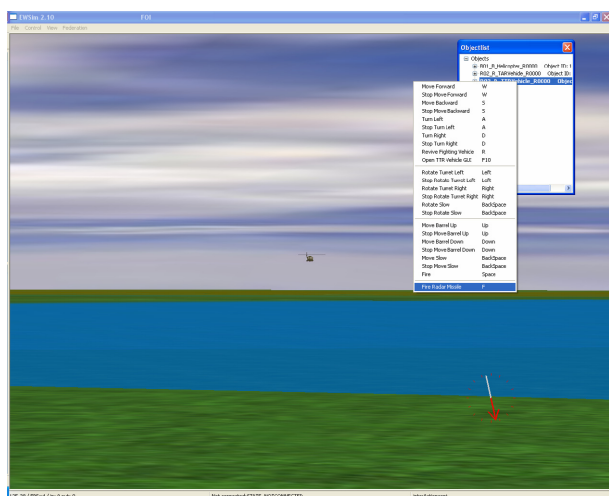
5.1 Sekventiell avfyrning med TAR och TTR

1. Spaningsradarn aktiveras om ej aktiverad. Mållägen presenteras i spaningsradarns användargränssnitt.
2. Mål att bekämpa markeras i spaningsradarns gränssnitt. Med menyvalet (högerklick) ”Invisa eldenhet” på det markerade målet, invisas TTR:en. Se Figur 22.



Figur 22 Användargränssnittet i TAR. Spaningsradarn (ring) är aktiverad och ett mål (helikoptern) är markerat. Eldledningsradarn invisas mot målet med "Invisa Eldenhet" i popup-menyn.

3. Med menyvalet "Fire Radar Missile" i eldledningsradarns "Objectlist"-meny inrikts TEL och robot avfyras. Se Figur 23.

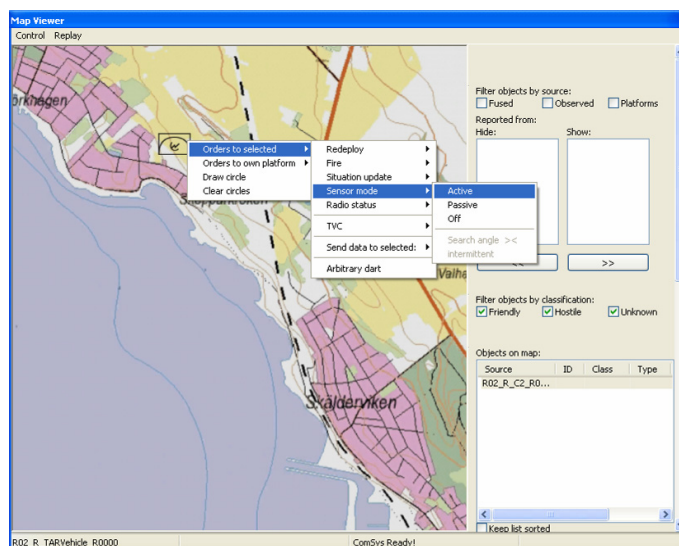


Figur 23 Vy från eldledningsradarns antenn (sett i dess boresight-riktning) efter att målet har hittats. TTR:en följer målet och användaren väljer att avfyras en robot mot den genom att i "Objectlist"-menyn välja "Fire Radar Missile".

5.2 Direkt avfyrning från TAR ledningsvy

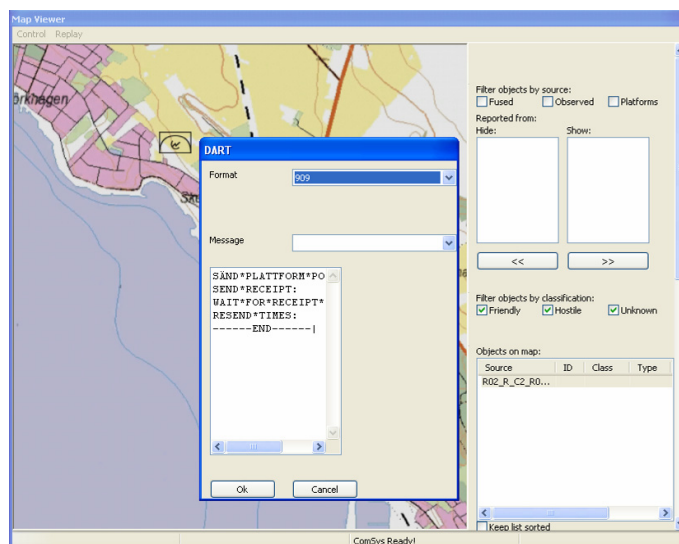
För beskrivning av ledningsvyn, se [14].

1. Spaningsradarn aktiveras om ej aktiv antingen direkt i TAR:ens gränssnitt eller från TAR:ens ledningsvy ("Command and Control"-vy) med menyvalet "Orders to selected ► Sensor mode ► Active" på spaningsradarns symbol, se Figur 24.



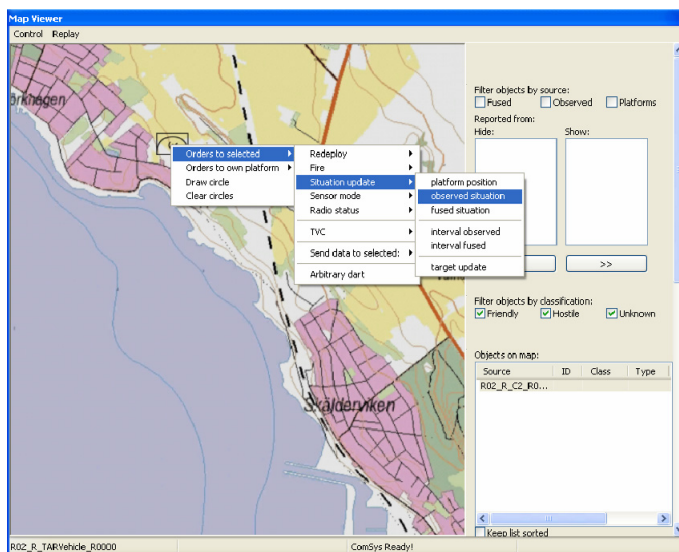
Figur 24 Spaningsradarn aktiveras från TAR:ens ledningsvy.

2. Order skickas till alla på egen sida att rapportera in sina positioner till lägesbilden med menykommandot "Control ► Send FFM to all team ► DART kod 909", se Figur 25.



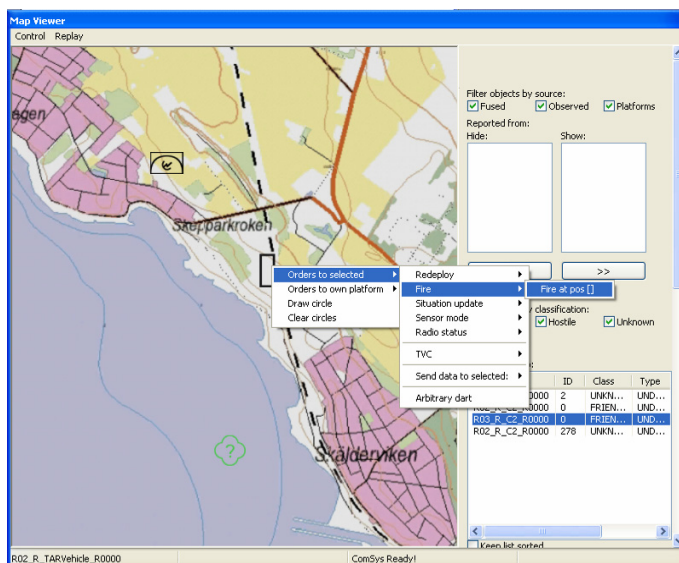
Figur 25 Order skickas till alla lagmedlemmar att rapportera in sin position till ledningsvyn.

3. Order skickas till TAR att uppdatera lägesbilden med menyvalet "Orders to selected ► Situation update ► observed situation", se Figur 26.



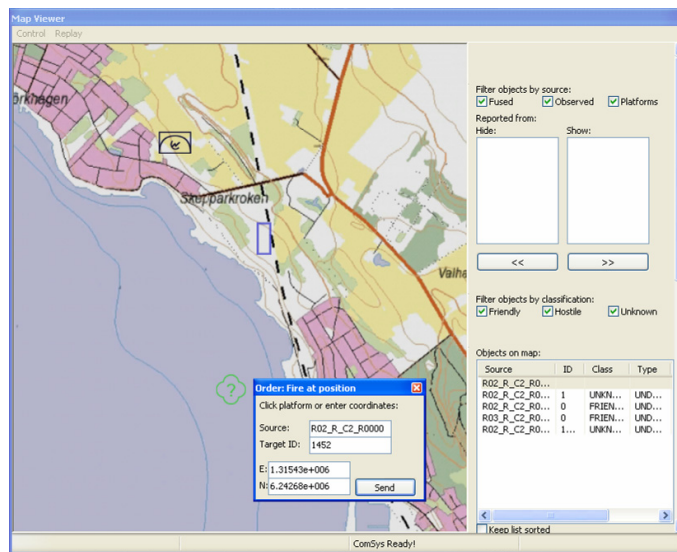
Figur 26 Order skickas till TAR:en att uppdatera lägesbilden i ledningsvyn.

4. TTR markeras och menyvalet "Orders to selected ► Fire ► Fire at pos []" görs, se Figur 27.



Figur 27 Den markerade TTR väljs att avfira robot.

5. En dialog visas, se Figur 28. Målet markeras och ordern skickas till den valda TTR:en. Eldledningsradarn visas, TEL väljs och inriktas varefter robot avfyras.

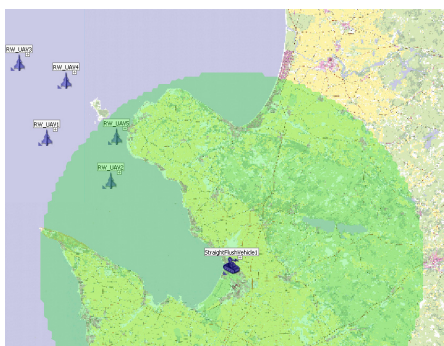


Figur 28 Målkoordinaterna anges genom att målet markeras. Med knappen "Send" skickas koordinaterna till eldledningsradarn. Eldledningsradarn visas, TEL väljs och inriktas varefter robot avfyras.

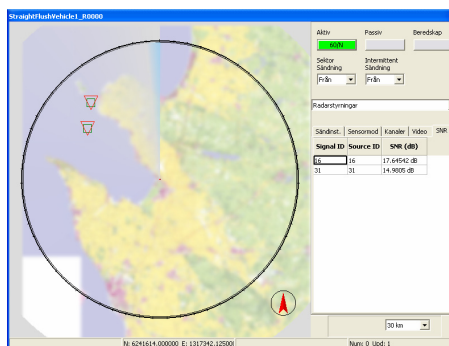
6 Verifiering och validering

I verifierande¹³ syfte har koden i modellen gått igenom och ett flertal skjutfall mot kommande, gående och hovrande mål på olika avstånd och i olika geometriska förhållanden relativt TAR/TTR har testats för att hitta konstigheter i modellen. Modellen har även testats i ett större scenario med HLA och fungerar där utan anmärkning. Utvecklade konventioner har följts vad gäller modellens interagerande med simulerad radiotrafik, vilket därför bör fungera, men har ej explicit testats.

I validerande¹⁴ syfte har ett flertal tester gjorts, där radarmodellens resultat har jämförts mot radarekvationen [16]. Tester av räckviddsprestanda har genomförts i både *EWSim* (TAR:ens användargränssnitt) samt i *EWPlan*. Figur 29 visar ett scenario och täckningsdiagrammet från en TAR i *EWPlan*. Figur 30 visar motsvarande scenario i TAR:ens användargränssnitt i *EWSim*. Teoretiskt ska räckvidden (med aktuella parametrar) bli 22,5 km vilket stämmer med det täckningsdiagram *EWPlan* visar och de mål som detekteras i *EWSim*.

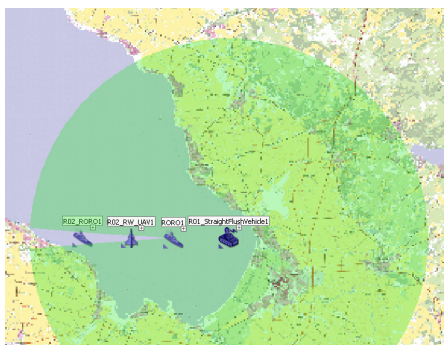


Figur 29 Avståndsmätning i *EWPlan*. Ingen terränghänsyn har tagits i beräkningen.

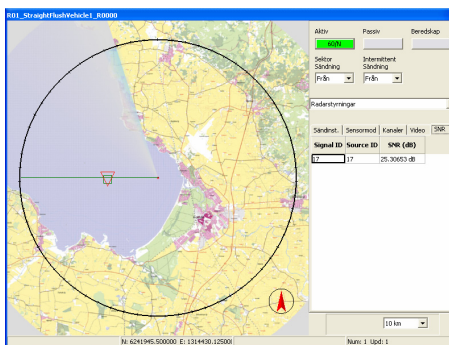


Figur 30 Radarinmätning i *EWSim* med samma scenario som Figur 29.

Detektionsmöjlighet vid störda fall har även studerats och visat sig korrekta. Ett test visas i Figur 31 och Figur 32.



Figur 31 Störsignal på R02_RW_UAV1 påverkar radarn. Radarn kan detektera fartygen i det gröna området.



Figur 32 Motsvarande störning i *EWSim*. Enbart det närmsta fartyget är synligt för radarn. Störbåringen är utträd i PPI:et.

Variationer av ett flertal parametrar i radarmodellen har gjorts, vilket visar att resultat från radarmodellen stämmer med radarteoretiska grunder.

¹³ Verifiering är en process som avgör om en modellimplementation representerar utvecklarens specifikationer (dvs gör modellen vad man tänkt sig att den ska göra?).

¹⁴ Validering är en process som avgör till vilken grad en modell eller simulering är en god representation av verkligheten med hänsyn till modellens syfte (dvs hur bra är modellen för det den ska användas till?).

Följeprestanda hos TTR-systemet har testats och fungerar men behöver förbättras framöver, vad avser dess hantering av följefilter och följedynamik. Denna delmodell beter sig rimligt men har ej direkt validerats mot något känt TTR-system.

Robotens beteende är rimligt och den styr på ett bra sätt längs siktlinjen som TTR:en tillhandahåller om roboten avfyras i närheten av siktlinjen. Huruvida den styrprincip som beskrivs i kapitel 3.1.4 kan parametersättas för att realistiskt beskriva ett verkligt system behöver dock fortsättningsvis studeras.

Delsystemvalideringar enligt ovan är förhållandevis lätta att genomföra, dock är det avsevärt svårare att validera hela modellen av radarluftvärnsrobotsystemet då jämförelsedata kan vara svåra att hitta. Valideringen av systemet får därför ske genom validering av delmodellerna. Avslutningsvis kan dock sägas att hela systemet beter sig robust och rimligt från målupptäckt till målbekämpning.

7 Diskussion och rekommendationer till fortsatt arbete

7.1 Diskussion

Modellen beskriver på ett rimligt sätt ett kollimationsstyrt radarluftvärnsrobotsystem och är validerad på delsystemnivå. En validering på högre nivå skulle innebära att anpassa modellens parametrar för att efterlikna något verkligt system och studera likheter/skillnader.

Genom att modellen är implementerad i simuleringsramverket *EWSim* kommer ramverkets förmåga att simulera telekrig att nyttjas i simuleringar med lvr-systemet. Störning mot sensordelarna i lvr-systemet kan simuleras tillsammans med störning mot systemets ledningsfunktionalitet i scenarier med flera ("några mot några") simulerade enheter.

Vår ambition har varit att hitta en gemensam design för både ett kommandostyrt och ett semiaktivt styrt lvr-system. Vi anser att vi har hittat en bra och generell meddelandestruktur som troligtvis fungerar bra även för semiaktiva system. Vi har dock insett att modellering av ett semiaktivt system kräver större modifieringar av radarmodellen. En vidareutveckling av radarmodellen, för att hantera bistatiska fall, är därför nödvändig innan ett radarluftvärns-robotsystem med semiaktivt styrd robot kan utvecklas.

7.2 Fortsatt arbete

Nedan listas några arbeten som vi anser bör göras fortsättningsvis:

- Parametrisera modellen för att efterlikna något verkligt system och genomföra jämförande tester.
- Vidareutveckling av radarmodellen för att kunna simulera semiaktivt styrda robotar (hantera bistatiska fall).
- Utveckla radarrobotmodell med syftbäringsstyrning och semiaktiv målsökarstyrning.
- Införa bättre och robustare simulering av följodynamiken i radarn.
- Modellera robotanflygningsfaser. För att kunna simulera robotens start- och anflygningsfas, se kapitel 2.2.
- Införa störskyddslogik (anti-VGPO¹⁵, HOJ¹⁶, etc.) i radarn/roboten.
- PKill-optimering som väljer optimal TEL för bekämpning.

¹⁵ VGPO – Velocity Gate Pull Off, Hastighetsvilsledning.

¹⁶ HOJ – Home On Jam, Brusföljning.

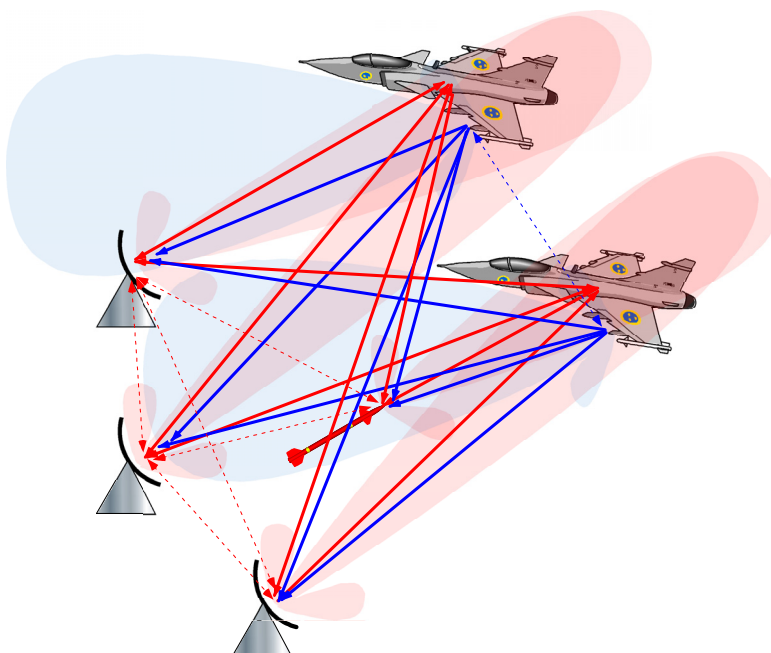
(Tom sida)

8 Appendix: "Modificeringar av radar-modellen"

Nedan beskrivs några av de modifieringar som har gjorts på radarmodellen som tidigare har beskrivits i [1].

8.1 Trunkeringslober

Ett problem man möter då man i ett scenario vill simulera ett flertal objekt som sinsemellan kan påverka varandra, är att antalet beräkningar ökar snabbt med antalet objekt.



Figur 33 Tre spanings-/eldledningsradarer registrerar/följer flygplanen. En radarstation belyser "ett" av flygplanen för att bekämpa det med en semiaktiv robot. Flygplanen, som är utrustade med radarvarnare, upptäcker att de är belysta i olika riktningar. De är utrustade med radarstörare och väljer i detta läge att störsända. Radarstationerna kommer att registrera ett eller flera mål beroende på mottagen effekt, frekvens, pulsgivning, etc. Bidrag ifrån telekonflikter visas med streckade linjer i figuren.

Som visas i Figur 33 blir den elektromagnetiska interaktionen mellan de olika objekten i simuleringen snabbt komplext. För att kunna hantera detta (speciellt då modellerna ska användas i realtidssammanhang), är det viktigt att kunna tillämpa olika "knep" för att minska antalet interaktioner. Sådana knep kan t ex vara att begränsa signaler i en radars mottagare utifrån:

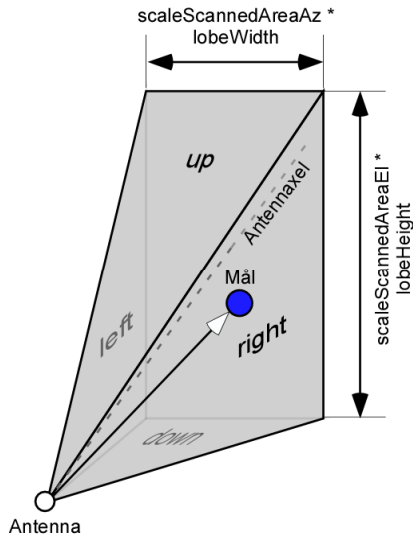
1. Ger signalen tillräcklig effekt i mottagaren för att detekteras?
2. Har signalen rätt frekvens för att kunna mottas?
3. Är signalens pulsgivning korrekt för att målet ska accepteras i mottagaren?

I *EW Radar Administrator* i *EW Sim* minskas sedan tidigare [1] antalet interaktioner genom att endast de signaler som överlappar mottagarens bandbredd hanteras i mottagaren (punkt 2 ovan).

För att ytterligare begränsa interaktionerna mellan objekten har trunkeringslober införts. En trunkeringslob motsvarar ett idealt antenndiagram som trunkerar bort alla mål utanför dess

huvudlob. I modellen kan trunkeringsloben skalas relativt lobbredden i sida och höjd ($\text{scaleScannedAreaAz}$ respektive $\text{scaleScannedAreaEl}$).

Trunkeringen görs genom att fyra polygoner skapas centrerade kring antennaxeln (bildar en pyramid), se Figur 34. Skalärprodukten för varje polygons normal mot positionsvektorn för målet beräknas. Om alla skalärprodukter > 0 så befinner sig målet i pyramiden och ska behandlas.

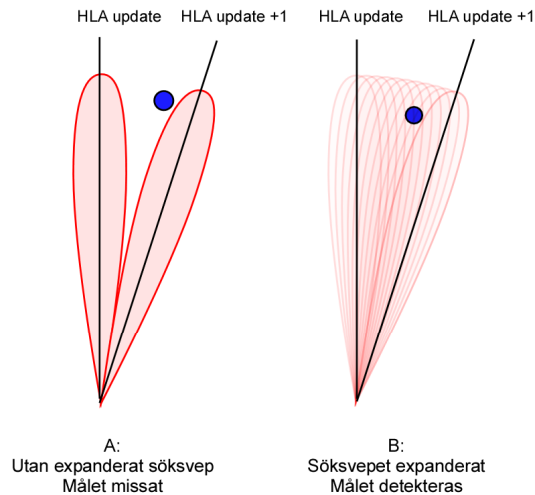


Figur 34 Trunkeringslob som fyra polygoner bildande en pyramid centrerad på antennaxeln.

Trunkeringslobber är ett grovt sätt att begränsa interaktionerna mellan objekten och bör användas med försiktighet då objekt som skulle ha kunnat påverka simuleringen kan sorteras bort.

8.2 Expanding av söksvepet mellan HLA-uppdateringar

Då TAR och TTR beräknar målträffar i sitt antenndiagram vid HLA-uppdateringar finns risk att mål missas då söksvepet rör sig snabbt. Att målet missas kan bero på att målet trunkeras bort (se kapitel 8.1) och/eller att målsignalen är svag på grund av till exempel att målet har en liten signatur eller befinner sig långt bort. Se figur A i Figur 35.



Figur 35 Vy av antenndiagrammet sett vinkelrätt mot sökriktningen.
A: målet missas vid HLA-uppdateringen. B: målet detekteras i den skannade arean

Genom att skapa en volym motsvarande den volym som antennen har avsökt mellan HLA-uppdateringar och göra måldetektion i denna volym undviks missade mål. Den avsökta volymen skapas genom att antalet antennlobbredder (i sökriktningen) mellan HLA-uppdateringar beräknas. Antennlägen skapas för att fylla volymen mellan HLA-uppdateringar med det beräknade antalet antennlobbredder. För att förbättra möjligheten att detektera svaga mål så görs stegen i sökriktningen till en tiondel av antennens lobbredd, vilket ger fler och tätare antennloblägen, se figur B i Figur 35.

Funktionerna `EWRadars::updateScannedArea()` för TAR samt `EWTTTR::updateScannedArea()` för TTR har modifierats i detta syfte.

(Tom sida)

9 Appendix: "Klasser"

Detta appendix beskriver översiktligt nya klasser som har utvecklats i lvr-b-modellen. Klassen `EWTRMslCtrlUnit` som är den viktigaste vid robotavfyrningen beskrivs i ett eget kapitel.

9.1 Klassen `EWTRMslCtrlUnit`

Klassen kontrollerar avfyrningen av roboten samt styr roboten under dess flykt. Speciellt intresse ägnas åt medlemsfunktionen `EWTRMslCtrlUnit::preUpdate()`.

`class EWTRMslCtrlUnit`

`EWTRMslCtrlUnit::preUpdate()` är en medlemsfunktion som exekveras vid varje HLA-uppdatering och som är central under robotavfyrningen. I funktionen underhålls en lista `m_listTEL` som anger de TEL:ar som TTR:en interagerar med under avfyrningen.

Innan roboten avfyras så anropas i `EWTRMslCtrlUnit::preUpdate()` i följande ordning:

`EWTRMslCtrlUnit::sendFireRequest()` skapar meddelandet `ORDER_SEND_TEL_MISSILEDATA` med följd målposition som argument till TEL:ar för att dessa ska rapportera huruvida de kan avfyr mot målpositionen. De TEL:ar som rapporterar tillbaka inom en sekund fyller `m_listTEL`.

`EWTRMslCtrlUnit::selectOptimalTEL()` ska ur `m_listTEL` välja den TEL som har bäst möjlighet att bekämpa målet, dvs den TEL som ger maximal PKill¹⁷. Vidareutvecklingsmöjligheter finns för denna funktion då den i dagsläget endast väljer den första TEL:en i listan.

`EWTRMslCtrlUnit::alignTEL()` skapar meddelandet `ORDER_TEL_ALIGN` som skickar följd målposition till den valda TEL:en för att inrikta dess eldrör i riktning mot målpositionen. Funktionen anropas kontinuerligt (på grund av att målet kan röra sig) tills meddelandet `SENDING_TEL_ALIGNED` skickas från TEL:en som kvittens på att eldröret slutligen är inriktad.

`EWTRMslCtrlUnit::fireRadarMissile()` anropas då den valda TEL:ens eldrör är riktad mot målet. Funktionen skapar dynamiskt en robot av klassen `osgRadarMissile`, parametersätter den samt kopplar en rörelsemodell av klassen `CRadarMissileMotion` till den. Funktionen anropar `EWTRMslCtrlUnit::sendFireToTEL()` som skapar meddelandet `ORDER_TEL_FIRE_MISSILE` som gör att TEL:en räknar ned antalet robotar på sin *RadarMsILauncher* med ett.

Då roboten är avfyrad, anropas i `EWTRMslCtrlUnit::preUpdate()` för varje HLA-uppdatering:

`EWTRMslCtrlUnit::calculateSteerSignal()` och styrprincipen som beskrivs i kapitel 3.1.4 exekveras. Rutinen returnerar kommenderade vinkelhastigheter för roboten i en rotationsmatris (quaternion). Rotationsmatrisen appliceras på robotens aktuella orientering för att ge den en ny orientering. Klassen `CRadarMissileMotion` (se nästa kapitel) beräknar robotens nya position ur dess nya orientering och hastighet.

¹⁷ PKill, Probability of Kill - sannolikheten för att en avfyrad robot kommer att bekämpa målet.

9.2 Övriga klasser

Övriga klasser som har utvecklats eller vidareutvecklats för lvr-systemet listas nedan.

class EWTRMslCtrlUnitEventHandler

Klassen registrerar och hanterar den händelse (event) som genereras då användaren väljer "Fire Radar Missile" i eldledningsradarns "Objectlist"-meny.

Medlemsfunktionen `EWTRMslCtrlUnitEventHandler::handleEvent()` sätter den logiska flaggan `EWTRVehicle::m_bFireOnTrack` vid avfyrning från menyn. Avfyrningssekvensen i `EWTRMslCtrlUnit::preUpdate()`, som beskrivs i Appendix: "Klasser", kapitel 9.1, initieras om flaggan är satt och TTR:en är i moden TRACK.

class EWAActionHandlerTTRVehicle

Hanterar EWAAction-objekt som har skapats ur mottagna meddelande i C2-enheten på TTR:en. De EWAAction-objekt som berör lvr-modellen och som hanteras i `EWAActionHandlerTTRVehicle::handle()` härstammar från meddelandena:

- ORDER_TRACK_POS från TAR
- SENDING_TEL_MISSILEDATA från TEL:ar
- SENDING_TEL_ALIGNED från vald TEL
- ORDER_FIRE från TAR

Meddelandet ORDER_FIRE skapas då roboten avfyras direkt från TAR:ens "Command and Control"-vy, se kapitel 5. Flaggan `EWTRVehicle::m_bFireOnTrack` sätts direkt vid mottagandet av ordern vilket leder till att roboten avfyras automatiskt då TTR:en går till moden TRACK, se klassen `EWTRMslCtrlUnitEventHandler` ovan.

För beskrivning av meddelandena, se Appendix: "Meddelanden".

class EWAActionHandlerRadarMissileLauncher

Hanterar EWAAction-objekt som har skapats ur mottagna meddelande i C2-enheten på TEL:en. De EWAAction-objekt som berör lvr-modellen och som hanteras i `EWAActionHandlerRadarMissileLauncher::handle()` härstammar från meddelandena:

- ORDER_SEND_TEL_MISSILEDATA från TTR
- ORDER_TEL_ALIGN från TTR
- ORDER_TEL_FIRE_MISSILE från TTR

För beskrivning av meddelandena, se Appendix: "Meddelanden".

class EWTRVehicle

Håller bland annat medlemsvariabeln `m_bFireOnTrack`, se klassen `EWTRMslCtrlUnitEventHandler` ovan. Mindre vidareutveckling av befintlig klass.

class osgRadarMissile

Klassen för den skapade roboten. Är en mindre vidareutveckling av den befintliga klassen `osgRemoteMissile` som används för trådstyrda robotar.

class CRadarMissileMotion

Klassen för robotens rörelsemodell. Är en vidareutveckling av den befintliga klassen `CMissileMotion` för optiskt siktlinjesstyrda robotar.

class osgRadarMissileLauncher

Hanterar funktionaliteten hos *RadarMslLauncher*. Viktiga medlemsfunktioner är:

`osgRadarMissileLauncher::ableToFire()`

Anropas som resultat på meddelandet `ORDER_SEND_TEL_MISSILEDATA` från en TTR. Avgör huruvida en robot kan avfyras från *RadarMslLauncher*. Villkor för avfyring är att *RadarMslLauncher* har fri sikt till den följda positionen, har robotar laddade på lavetten samt att TEL-plattformen färdas långsammare än en maximal hastighet, se kapitel 4.3. Uppfylls dessa villkor så skickas meddelandet `SENDING_TEL_MISSILEDATA` tillbaka till TTR:en.

`osgRadarMissileLauncher::alignBarrel()`

Anropas som resultat på meddelandet `ORDER_TEL_ALIGN` från en TTR. Kommenderar eldröret att ställa sig i riktning mot den följda målpositionen.

`osgRadarMissileLauncher::preUpdate()`

Exekveras varje HLA-uppdatering. Kontrollerar om eldröret är tillräckligt inriktat (mindre än 3° avvikelse) mot den följda målpositionen. Då detta inträffar skickas meddelandet `SENDING_TEL_ALIGNED` till TTR som därefter kan beordra avfyring.

För beskrivning av meddelandena, se kapitel 10.

(Tom sida)

10 Appendix: "Meddelanden"

TAR, TTR och TEL kan vara skilda federater i en federation varför den struktur som utvecklats för att skicka meddelanden över HLA nyttjas i simuleringen. Ett antal meddelanden (ordrar och rapporter) har definierats för kommunikationen mellan TAR, TTR och TEL.

`ORDER_TRACK_POS(dx, dy, dz)`

Skickas från TAR till TTR då menyalternativet "Invisa eldenhet" görs för en markerad målposition i TAR:ens användargränssnitt. Syftet är att TTR ska invisas till positionen som anges i argumenten. I dagsläget skickas meddelandet till alla TTR på den egna sidan. I framtiden bör den TTR som ska motta meddelandet kunna adresseras.

`ORDER_SEND_TEL_MISSILEDATA(dx, dy, dz)`

Skickas från TTR till TEL:ar på den egna sidan då "Fire Radar Missile" väljs i eldledningsradarns "Object list"-meny. Argumentet är positionen på det mål som TTR:en följer. Meddelandets syfte är att de TEL:ar som har möjlighet att avfyr ska skicka sin position till TTR:en, se `osgRadarMissileLauncher::ableToFire()`, i kapitel 9.2. För att kommunikationen ska ske endast till de TEL:ar som är underställda TTR:en bör ett eget radionät skapas inom förbandet. Meddelandet inleder en kedja av meddelanden som i slutändan avfyrar en robot.

`SENDING_TEL_MISSILEDATA(ID, ID_C2, dx, dy, dz)`

Skickas från de TEL som har möjlighet att avfyr robotar till den TTR som sände `ORDER_SEND_TEL_MISSILEDATA`. Argumenten som sänds är ID för *RadarMslLauncher*, ID för plattformens C2-enhet samt *RadarMslLaunchers* position. TTR sammanställer rapporteringen från tillgängliga TEL och väljer den TEL som ger bästa träffsannolikhet (optimal TEL).

`ORDER_TEL_ALIGN(dx, dy, dz)`

Skickas från TTR till vald optimal TEL för att vrida in dess eldrör mot målet. Argumenten är följd målposition.

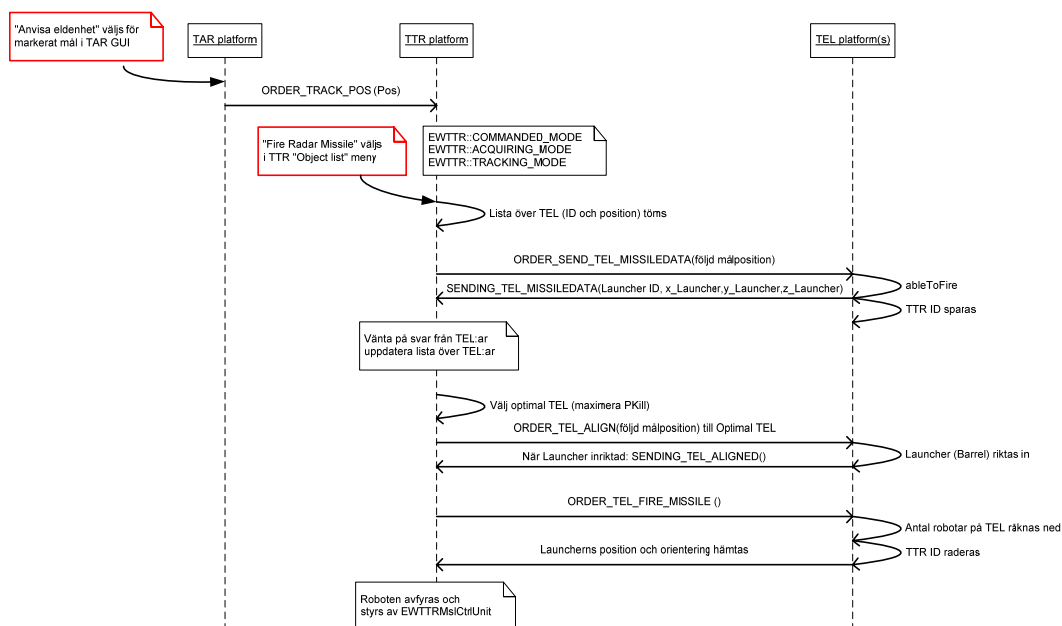
`SENDING_TEL_ALIGNED()`

Skickas från vald TEL till TTR, då TEL:ens eldrör är invridet mot målet.

`ORDER_TEL_FIRE_MISSILE()`

Skickas från TTR till vald TEL vid avfyrningen. TEL:ens *RadarMslLauncher* räknar ner antal tillgängliga robotar med ett. TTR skapar en robot positionerad och orienterad som TEL:ens *RadarMslLauncher* och styr den därefter mot målet från `EWTTMRMslCtrlUnit`.

Figur 36 visar meddelandena mellan TAR, TTR och TEL vid invisning i TAR:en och vid avfyrning i TTR:en.



Figur 36 Sekvensdiagram med invisning av TTR från TAR samt avfyrning av robot från TEL på order från TTR. Rödmarkerade boxar anger interaktion från användaren.

C2-enheterna på de olika plattformarna har att förmedla meddelanden mellan plattformarna. `CommandnControl::analyzeAction()` konverterar meddelandena till `EWAction`-objekt. `Handler`-klasser som `EWTRTRMslCtrlUnitEventHandler` och `EWActionHandlerRadarMissileLauncher`, se Appendix: "Klasser", har att agera på dessa `EWActions`.

11 Appendix: "Parametrar"

Utgående från konfigurationsbeskrivningen i kapitel 4 och de konfigurationsdialoger i *EWPlan* som visas i detta Appendix kan ett fungerande radarstyrt lrb-system konfigureras.

Figur 37 - Figur 40 visar attribut för R02_R_TARVehicle och dess utrustning, se Figur 16. Properties och eventuella Container Parameters visas ej i nedanstående figurer. För beskrivning av parametrarna se [1].

Attributes	
type	{entityKind=1, domain=1, co...
position	(3459976.166569464, 78655...
orientation	(-0.094256803393364, 0.564...
velocity	(0, -0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	12
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
MaxSpeed	20.0
MaxRotationSpeed	30.0
MaxR	0.0
TerrainLimitsFile	
ReactionTime	0.0
SightID	
RadioID	
RadarID	

Figur 37 Parametrar för R02_R_TARVehicle

Attributes	
type	{entityKind=98, domain=98, ...
position	(3459976.166569464, 78655...
orientation	(-0.094256803393364, 0.564...
velocity	(0, -0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
C2Conditions	
trusted orders	
trusted reports	
DatafusionInterval	2.0
BearingAcceptance	1.0
PositionAcceptance	150.0
UnassociatedTime	5.0
RadioID	
SendReceipt	☒
ControlState	0
ShowMessages	
Count and Log Objects	0

Figur 38 Parametrar för R02_R_C2

Attributes	
type	{entityKind=1, domain=1, c...
position	(3459976.166569464, 786...
orientation	(-0.094256803393364, 0.5...
velocity	(0, -0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	0
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
RadarType	Pulsedoppler
Power	500000.0
PulseLength	9.999999974752427E-7
Wavelength	0.1
MinWavelength	0.05
MaxWavelength	0.2
Range	30000.0
PulseGrouping	☒
nPulsesPerGroup	1000
PulseCompression	100.0
DiffFreqChan	0.0
ECCM	
ScanMode	
TypePRI	Fixed
PRIList	0.001
ThresholdDetection	1.000000031710769E-28
SNR	2.0000000298023224
FAP	0.0
SideLobeSuppression	9.99999995904E11
BackLobeSuppression	9.99999995904E11
RangeResolution	100.0
AzimuthResolution	0.009999999776482582
ElevationResolution	0.009999999776482582
MeasurementStdDevRange	0.0
MeasurementStdDevEl	0.0
MeasurementStdDevAz	0.0
InnerRegion	5000.0
OuterRegion	20000.0
MinOperationalFrequency	5000.0
MaxOperationalFrequency	5000.0
TargetTracking	☒

Figur 39 Parametrar för R02_R_SurveillanceRR

Attributes	
type	{entityKind=1, domain=1, co...
position	(3459976.6937991753, 7865...
orientation	(-0.2818463742733, 0.52043...
velocity	(0, -0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	0
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
AntennaDiagram	Antenna - Exponential lobe (b...

Figur 40 Parametrar för R02_R_SurveillanceRRAntenna

Figur 41 - Figur 49 visar attribut för R03_R_TTRVehicle och dess utrustning, se Figur 17. Properties och eventuella Container Parameters visas ej i nedanstående figurer. För beskrivning av parametrarna se [1].

Attributes	
type	{entityKind=1, domain=1, co...
position	(3460305.835337398, 78762...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angular velocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	13
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
MaxSpeed	10.0
MaxRotationSpeed	30.0
MaxR	0.0
TerrainLimitsFile	
ReactionTime	0.0
SightID	
RadioID	
RadarID	

Figur 41 Parametrar för R03_R_TTRVehicle

Attributes	
type	{entityKind=1, domain=1, co...
position	(3460305.7793373982, 7876...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angular velocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	13
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
BarrelID	
MaxRotationSpeed	20.0
RotationLimit	☒
MaxAzimuthAngle	360.0
MinAzimuthAngle	-360.0

Figur 42 Parametrar för R03_R_Turret

Attributes	
type	{entityKind=1, domain=1, co...
position	(3460305.835337398, 78762...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angular velocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	13
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
Max elevation	60.0
Min elevation	0.0
MaxElevationSpeed	10.0
FireTime	0.0

Figur 43 Parametrar för R03_R_Barrel

Attributes	
type	{entityKind=1, domain=1, co...
position	(3459987.1878469493, 7881...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angular velocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05

Figur 44 Parametrar för R03_R_AntennaPedestal

Attributes	
type	{entityKind=1, domain=1, co...
position	(3459987.1878469493, 7881...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angular velocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	13
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
AntennaDiagram	

Figur 45 Parametrar för R03_R_Antenna

Attributes	
type	{entityKind=98, domain=98, ...}
position	(3435801.616755459, 79206...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angular velocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
C2Conditions	
trusted orders	
trusted reports	
DatafusionInterval	2.0
BeamingAcceptance	1.0
PositionAcceptance	150.0
UnassociatedTime	5.0
RadioID	
SendReceipt	☒
ControlState	0
ShowMessages	
Count and Log Objects	0

Figur 47 Parametrar för R03_R_C2

Attributes	
type	{entityKind=1, domain=1, c...
position	(3460017.2999163265, 788...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angular velocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	13
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
RadarType	Pulse
Power	1000000.0
PulseLength	1.0E-6
Wavelength	0.1
MinWavelength	0.05
MaxWavelength	0.2
Range	30000.0
PulseGrouping	☒
nPulsesPerGroup	1000
PulseCompression	100.0
DiffFreqChan	0.0
ECCM	
ScanMode	
TypePRI	Fixed
PRIList	0.001
ThresholdDetection	1.0E-28
SNR	2.0
FAP	0.0
SidelobeSuppression	1.0E12
BackLobeSuppression	1.0E12
RangeResolution	100.0
AzimuthResolution	0.01
ElevationResolution	0.01
MeasurementStdDevRange	0.0
MeasurementStdDevEl	0.0
MeasurementStdDevAz	0.0
InnerRegion	1.0
OuterRegion	200000.0
MinOperationalFrequency	10000.0
MaxOperationalFrequency	10000.0
RangeGateLength	100.0
ServoParameters	
LockOnPattern	

Figur 46 Parametrar för R03_R_TTR

Attributes	
type	{entityKind=1, domain=1, co...
position	(3444911.375289572, 79104...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	Rotation [rad] resulting in an update to HLA
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
MaxRotationSpeed	10.0
MaxElevationSpeed	10.0
FOV	40.0
ActionsOnAlert	LockOnBrightestSpot
PointingAccuracy	2.0
ArmTime	1.0
LockTime	1.0
FireTime	1.0
TargetHeight	5000.0
TargetDistance	6000.0

Figur 48 Parametrar för R03_R_TTROperator

Attributes	
type	{entityKind=0, domain=0, c...
position	(3448070.228189689, 7940...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
ArmDistance	30.0
DetonateDistance	10.0
LifeTime	20.0
EngineBurnTime	18.0
Speed	200.0
Acceleration	-27.0
LOSadvance	10.0
MslYawDotMax	5.0
MslPitchDotMax	5.0

Figur 49 Parametrar för R03_R_TTRMslCtrlUnit

Figur 50 - Figur 54 visar attribut för R04_R_FightingVehicle (separat TEL) och dess utrustning, se Figur 21. Properties och eventuella Container Parameters visas ej i nedanstående figurer. För beskrivning av parametrarna se [1].

Attributes	
type	{entityKind=1, domain=1, co...
position	(3460293.6500102123, 7877...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	12
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
MaxSpeed	1000.0
MaxRotationSpeed	30.0
MaxR	0.0
TerrainLimitsFile	
ReactionTime	0.0
SightID	
RadiolID	
RadarID	

Figur 50 Parametrar för R04_R_FightingVehicle

Attributes	
type	{entityKind=1, domain=1, co...
position	(3459572.72970393, 787689...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	0
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
BarrelID	
MaxRotationSpeed	20.0
RotationLimit	
MaxAzimuthAngle	360.0
MinAzimuthAngle	-360.0

Figur 51 Parametrar för R04_R_Turret

Attributes	
type	{entityKind=1, domain=1, co...
position	(3459318.4993700203, 7877...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
model	
radarCrossSectionSignatureIndex	0
IRSignatureIndex	0
IRSignatureData	
EOSignatureData	
Max elevation	70.0
Min elevation	-30.0
MaxElevationSpeed	10.0
FireTime	0.0

Figur 52 Parametrar för R04_R_Barrel

Attributes	
type	{entityKind=0, domain=0, co...
position	(3459889.8286885708, 7877...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
Number of missiles	4
Reload time	300.0
Max speed	5.0

Figur 53 Parametrar för R04_R_RadarMslLauncher

Attributes	
type	{entityKind=98, domain=98, ...}
position	(3460006.8805539887, 7874...
orientation	(0, 0, 0)
velocity	(0, 0, 0)
acceleration	(0, 0, 0)
angularvelocity	(0, 0, 0)
locationHLAupdateThreshold	1.0
orientationHLAupdateThreshold	0.05
C2Conditions	
trusted orders	
trusted reports	
DatafusionInterval	2.0
BearingAcceptance	1.0
PositionAcceptance	150.0
UnassociatedTime	5.0
RadioID	
SendReceipt	☒
ControlState	0
ShowMessages	
Count and Log Objects	0

Figur 54 Parametrar för R04_R_C2

12 Referenser

- 1 Morgan Nordin, Mikael Petersson, Peter Klum,
”Generisk radarmodell för Telekrigssimulering”
FOI-R--2320--SE, Metodrapport 2007
- 2 Lars Tydén, Hanna Andersson, Kjell Andersson, Lars Berglund, Leif Festin, Fredrik Forslund, Mikael Petersson, Mats Tjäder, Peter Klum, Christer Wigren,
”Lägesrapport 2007 Duellsimulering framtida telekriginsatser”
FOI-R--2365--SE, Användarrapport 2007
- 3 Lars Tydén, Hanna Andersson, Ulrika Andersson, Leif Festin, Fredrik Forslund, Daniel Haavisto, Peter Klum, Morgan Nordin, Sebastian Olsson, Mikael Petersson, Christer Wigren,
”Lägesrapport Duellsimulering framtida telekriginsatser”,
FOI-R--2136--SE, Användarrapport, December 2006
- 4 Lars Tydén, Hanna Andersson, Niclas Appleby, Joakim Bergman, Leif Festin, Daniel Haavisto, Carl Hedberg, Sebastian Olsson, Mikael Petersson, Morgan Nordin, Pär Svensson, Christer Wigren,
”Slutrapport Duellsimulering telekrig”,
FOI-R--1775--SE, Användarrapport, November 2005
- 5 <https://www.dmsomil/public/transition/hla>
- 6 Lars Tydén, Leif Festin, Mikael Petersson, Morgan Nordin, Sebastian Olsson, Peter Klum, Christer Wigren, Hanna Andersson,
”Planeringsverktyg EWPlan med yttäckningsberäkningar för kommunikation, radar och optronik”,
FOI-R-2129--SE, Metodrapport, December 2006
- 7 http://en.wikipedia.org/wiki/List_of_surface-to-air_missiles
- 8 Steven J Zaloga
Soviet Air Defence Missiles - Design, Development and Tactics
Jane's Information Group, ISBN 0-7106-0589-7, 1989
- 9 Ching-Fang Lin
Modern navigation, guidance, and control processing. Volume II.
Prentice Hall 1991, ISBN 0-13-596230-7
- 10 Per Gerdle
Lärobok i telekrigföring för luftvärnet - Radar och radartaktik
Försvarsmakten, M7741-850101 Telekrig Rr Tak Lv, 2004
- 11 <http://missile.index.ne.jp/cgi/misearch.cgi>
- 12 <http://en.wikipedia.org/wiki/Subregion>
- 13 Lars Tydén, Leif Festin, Sebastian Olsson, Christer Wigren
”Systembeskrivning LKS V2 Federationsdesign”,
FOI-R--2274--SE, Teknisk rapport, April 2007

- 14 Lars Tydén, Hanna Andersson, Fredrik Forslund, Leif Festin, Fredrik Mörnstedt, Sebastian Olsson, Mikael Petersson, Christer Wigren
”Systembeskrivning LKS V2 teknikplattform”,
FOI-R--2275--SE, Användarrapport, Maj 2007
- 15 Peter Klum, Lars Bergström, Morgan Nordin, Kjell Rixon,
”Tyngdpunktsföljning - en förenklad modell av radarföljning”,
FOI-R--0938--SE, Metodrapport, September 2003
- 16 Simon Kingsley, Shaun Quegan,
Understanding Radar Systems,
McGraw-Hill, 1992, ISBN 0-07-707426-2