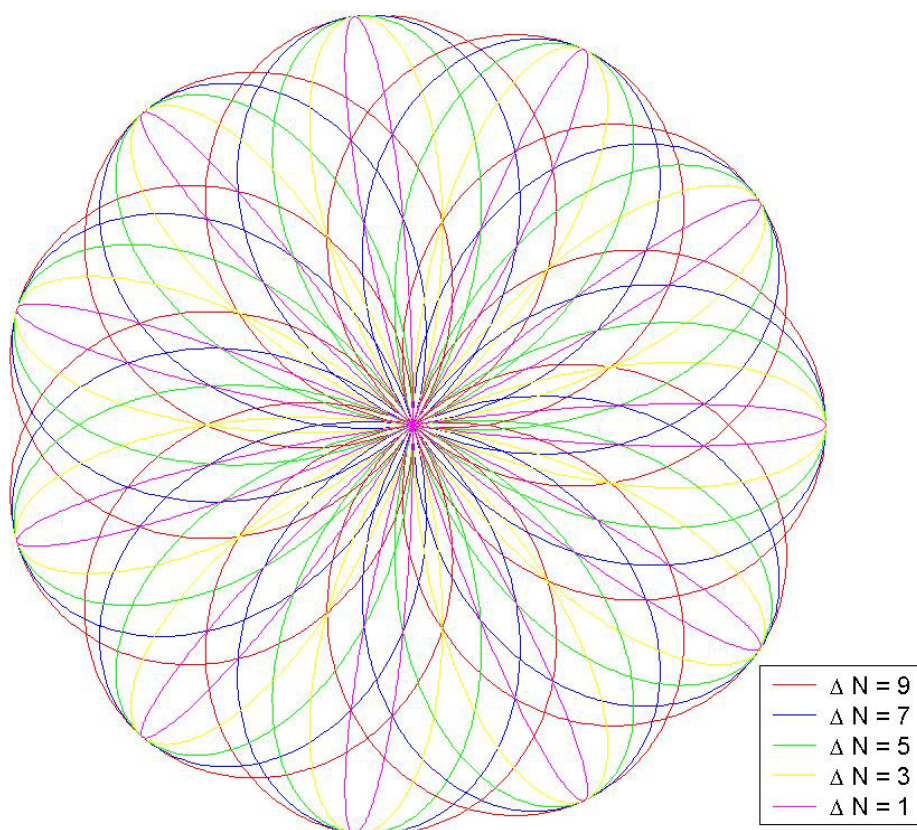


Ulrika Andersson

Rosettskannande målsökare - funktion och analys



TOTALFÖRSVARETS FORSKNING SINSTITUT

Ledningssystem

Box 1165

581 11 Linköping

FOI-R--0833--SE

Mars 2003

ISSN 1650-1942

Metodrapport

Ulrika Andersson

Rosettskannande målsökare - funktion och analys

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 Linköping	Rapportnummer, ISRN FOI-R--0833--SE	Klassificering Metodrapport
	Forskningsområde 6. Telekrig	
	Månad, år Mars 2003	Projektnummer E7015
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 61 Telekrigföring med EM-vapen och skydd	
Författare/redaktör Ulrika Andersson	Projektledare Lars Tydén	
	Godkänd av Mikael Sjöman	
	Uppdragsgivare/kundbeteckning	
	Tekniskt och/eller vetenskapligt ansvarig	
Rapportens titel Rosettskannande målsökare - funktion och analys		
Sammanfattning (högst 200 ord) En studie av en rosettskannande målsökare och dess egenskaper har genomförts. Rosettmönstret genereras med hjälp av två optiska element som roterar åt motsatta håll. Rotationsfrekvenserna hos dessa element är avgörande för rosettmönstrets utseende, såsom antalet öglor och vidden på öglorna. Rapporten beskriver hur rosettmönstret påverkas av det rosettskannande systemets parametrar och hur systemet arbetar för att plocka ut målets position från detektorsignalen. Rosettmålsökaren utnyttjar en enda liten cirkulär detektor som söker av ett område i ett rosettförformat mönster. Det område som detektorn täcker in utgör det momentana synfältet (IFOV). Om IFOV är litet blir scenen bättre upplöst, men systemets prestanda urartar om IFOV är så litet att inte hela synfältet täcks in. Beräkningar av det minsta IFOV som kan användas för att uppnå fullständig synfältstäckning beskrivs i rapporten. Två program har utvecklats under studiens gång. En modell har implementerats i Matlab och en annan i C++. I Matlab-programmet kan enskilda bilder studeras för att ge svar på grundläggande frågeställningar. I C++-programmet appliceras rosettmålsökaren på en sekvens av bilder och målföljning kan därmed simuleras. C++-modellen har dessutom integrerats med ramverket EwSim, som möjliggör simuleringar i realtid (eller nära realtid). Vidare avhandlas ett antal störskydd som en rosettmålsökare kan utrustas med för att minska sin känslighet för motmedel. Fem metoder beskrivs och fyra av dessa är implementerade i C++-modellen.		
Nyckelord rosett, infraröd, störskydd, IFOV, TFOV, synfält, datormodell		
Övriga bibliografiska uppgifter	Språk Svenska	
ISSN 1650-1942	Antal sidor: 31 s.	
Distribution enligt missiv	Pris: Enligt prislista	

Issuing organization FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	Report number, ISRN FOI-R--0833--SE	Report type Methodology
	Programme Areas 6. Electronic Warfare	
	Month year March 2003	Project no. E7015
	General Research Areas 5. Commissioned Research	
	Subcategories 61 Electronic Warfare including Electromagnetic Weapons and Protection	
Author/s (editor/s) Ulrika Andersson	Project manager Lars Tydén	
	Approved by Mikael Sjöman	
	Sponsoring agency	
	Scientifically and technically responsible	
Report title (In translation) Rosette scanning infrared seeker - function and analysis		
Abstract (not more than 200 words) A study of a rosette scanning infrared seeker (RSIS) and its characteristics has been made. The rosette pattern is generated by two counter rotating optical elements. The rotation frequencies determine the appearance of the rosette pattern, e.g. the number of petals and the width of the petal. The report describes how the parameters of the RSIS affect the pattern and how the system calculates the position of the target from the detector signal. The RSIS scans a single small detector across the total field of view (TFOV) in a rosette pattern. The area that the detector sees in every moment is the instantaneous FOV (IFOV). If the IFOV is small the system gets better resolution, but if the IFOV is too small the performance of the RSIS can deteriorate. Calculations to get the smallest IFOV which can be used to get full scan coverage in the TFOV are described in the report. Two programs have been developed during the study. The first model has been implemented in Matlab and the other in C++. In the Matlab program it is possible to study individual images to get answers to fundamental questions. In the C++ program the RSIS can be used on a sequence of images and it is therefore possible to simulate target tracking. The C++ model has also been integrated with the framework EwSim, which makes it possible to do real time simulations. Further some counter-countermeasures for the RSIS are discussed. Five methods are described and four of them are implemented in the C++ model.		
Keywords rosette, infrared, counter-countermeasures, IFOV, TFOV, field of view, model		
Further bibliographic information	Language Swedish	
ISSN 1650-1942	Pages 31 p.	
	Price acc. to pricelist	

1	INLEDNING	6
2	ROSETTSKANNANDE SYSTEM	6
2.1	Mönstergenerering	8
2.2	Fullständig synfältstäckning	9
2.3	Konventionell metod för att beräkna storleken på IFOV	11
2.4	Förbättrad metod för att beräkna storleken på IFOV	12
3	SIGNALBEHANDLING	13
3.1	Tyngdpunktsmetoden	13
3.2	Störskydd	13
4	UTVECKLADE PROGRAM	16
4.1	ros1 – ett program utvecklat i Matlab	16
4.2	RSS – ett program utvecklat i Visual C++	24
4.3	EwSeeker – en modul i Vega	28
5	SLUTSATSER	30
6	REFERENSER	31

1 Inledning

En infraröd (IR) målsökare, monterad på en termiskt följande missil, identifierar målets position genom att upptäcka och bearbeta den termiska energi som målet utstrålar. Informationen om målets position utnyttjas sen för att leda missilen till målet. Men det tilltänkta målet kan använda sig av motmedel, såsom facklor, för att störa missilens uträkning av målposition. För att kunna motstå effekten av detta, bör IR-målsökaren utrustas med störskydd.

I denna rapport kommer funktionen hos en rosettskannande IR-målsökare att studeras. Rosettmålsökaren utnyttjar en liten, cirkulär detektor som söker av ett område i ett rosettförmät mönster. Ju mindre momentant synfält som kan användas vid avsökningen av det totala synfältet, desto mindre känslig blir målsökaren för facklor och andra störkällor. Rosettmålsökaren kan ses som en pseudobildalstrande målsökare eftersom den inte bygger upp en bild som en riktig bildalstrande målsökare, men ändå har möjligheten att utnyttja spatiell information från skanningen för att diskriminera falska mål såsom facklor. Dessutom är det möjligt att rekonstruera en tvådimensionell bild från den information som målsökaren erhåller. En stor förtjänst hos skannande målsökare i allmänhet är att de kan utnyttja spatiell diskriminering för att förkasta oönskade källor. Ett exempel på ett system med rosettskannande avsökning är roboten Stinger-POST, som utöver IR-området även utnyttjar UV-området för att hitta ett lämpligt mål.

Vidare kommer ett antal olika störskydd att beskrivas och diskuteras. Till sist kommer de program som utvecklats under arbetets gång att gås igenom och exempel från simuleringar med dessa program att visas.

2 Rosettskannande system

Ett rosettskannande system utnyttjar en enda IR-detektor. Detektorelementet är cirkulärt. Det som detektorn ser i varje ögonblick utgör det momentana synfältet (IFOV). Detektorn söker av det totala synfältet (TFOV) i ett rosettmönster och bygger upp en tredimensionell signal bestående av intensitetsvärden samt motsvarande koordinater. Med hjälp av den tredimensionella signalen är det möjligt att reproducera en tvådimensionell bild av vad systemet ser. Om IFOV är litet blir scenen (den inkommande bilden) bättre upplöst. Men systemets prestanda urartar om IFOV är så litet att inte hela synfältet täcks in, eftersom målets position relativt rosettmönstret då blir avgörande för om målsökaren kan upptäcka målet eller inte. *Figur 3* visar hur TFOV och IFOV är definierade i denna rapport. I själva verket sträcker sig området som detektorn kan se utanför det markerade TFOV eftersom IFOV har en viss utbredning. Rosettmönstret erhålls med hjälp av två optiska element (speglar eller prismor) som roterar åt motsatta håll. En principskiss som visar ett möjligt sätt att generera ett rosettmönster i en rosettskannande robot visas i *Figur 1* [4]. Ett blockdiagram över rosettmålsökarens arbetsgång visas i *Figur 2*.

Rosettmönstret i cartesiska koordinater ges av:

$$x(t) = \frac{\rho}{2} \cdot (\cos 2\pi f_1 t + \cos 2\pi f_2 t) \quad \text{ekvation 1}$$

$$y(t) = \frac{\rho}{2} \cdot (\sin 2\pi f_1 t - \sin 2\pi f_2 t) \quad \text{ekvation 2}$$

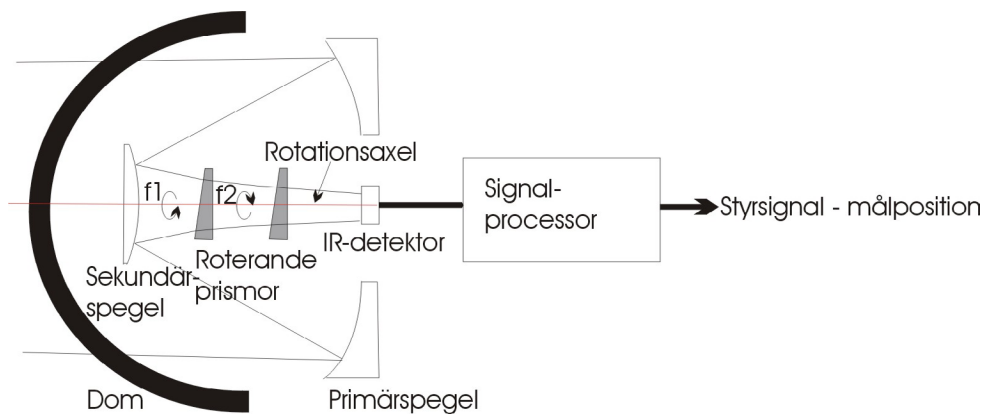
Rosettmönstret i polära koordinater ges av:

$$r(t) = \rho \cdot \cos \pi(f_1 + f_2)t \quad \text{ekvation 3}$$

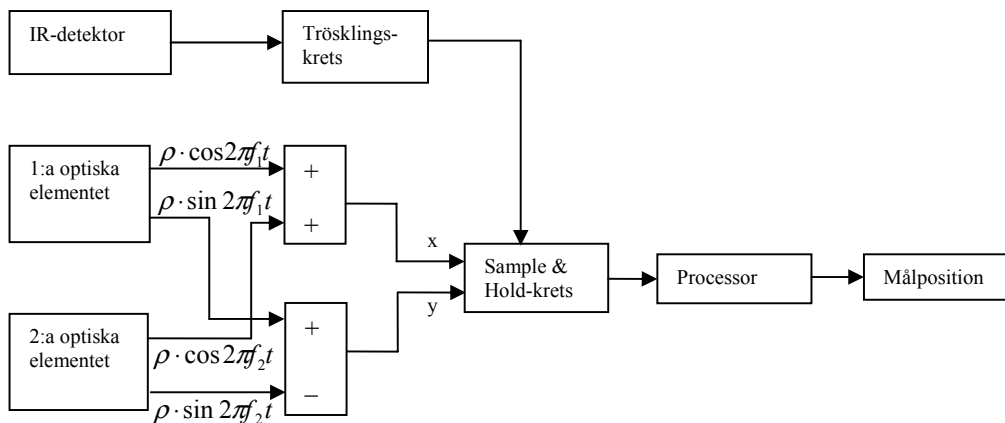
$$\theta(t) = \pi(f_1 - f_2)t \quad \text{ekvation 4}$$

f_1 och f_2 - prismornas rotationsfrekvens, $f_1 > f_2 > 0$

ρ - prismornas brytningsvinkel eller speglarnas lutning, samt även halva TFOV

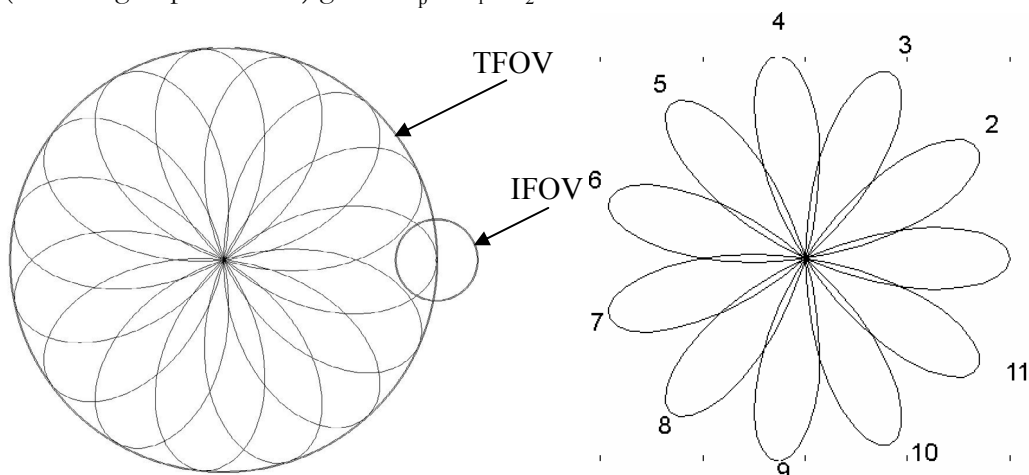


Figur 1. Principskiss över rosettmålföljarens uppbyggnad.



Figur 2. Blockdiagram över rosettmålsökarens arbetsgång.

Värdena på f_1 och f_2 bestämmer antalet öglor (eng. petals) i rosettmönstret och bredden på varje ögla. f_1 och f_2 har en största gemensam divisor f , sådan att $N_1 = f_1/f$ och $N_2 = f_2/f$ är positiva heltal. Antalet öglor i rosettmönstret ges av $N = N_1 + N_2$. Rosettperiodtiden ges av $T = 1/f = N_1/f_1 = N_2/f_2$. Öglefrekvensen (antalet öglor per sekund) ges av $f_p = f_1 + f_2$.

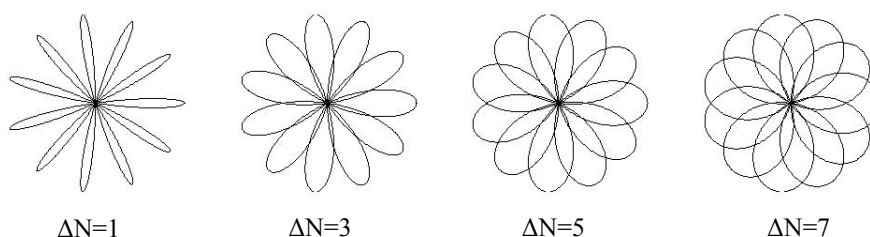


Figur 3. Rosettmönster med angivet TFOV respektive IFOV.

Figur 4. $N_1 = 7$, $N_2 = 4$.
Numrering av rosettens öglor motsols.

2.1 Mönstergenerering

Rosettmönstret består av ett antal öglor som utgår från ett gemensamt centrum. Antalet öglor ges av $N = N_1 + N_2$, där $N_1 = f_1/f$ och $N_2 = f_2/f$ och f är f_1 's och f_2 's största gemensamma divisor. f_1 och f_2 är de två optiska elementens rotationsfrekvens. Bredden hos öglorna är beroende av N och ΔN ($N_1 - N_2$) samt rosettens radie, ρ . Överlapp mellan öglor kan ses i relation till ΔN . Ju större värde på ΔN desto större överlapp mellan närliggande öglor (Figur 5).



Figur 5. Rosettmönstret som funktion av ΔN . Överlappet ökar med ökande ΔN . Antalet öglor $N = 11$ i samtliga bilder.

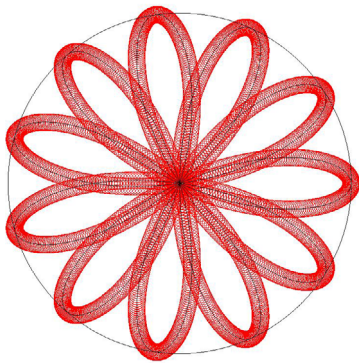
Om man numrerar öglorna motsols och startar med öglan vid $(x=\rho, y=0)$ som nummer 1 (Figur 4), så ges ordningen öglorna genereras av följande beräkning:

$$P_{n+1} = (N_1 \cdot n \% N) + 1 \quad n = 0, 1, 2, \dots, N-1$$

Första öglan utgår alltid från $(x, y) = (\rho, 0)$, d.v.s. $P_1 = 1$. P_2 är den andra öglan som skapas, P_3 den tredje osv. Bokstaven P syftar på engelskans *petal*, d.v.s. engelskans namn på öglorna i rosettmönstret.

2.2 Fullständig synfältstäckning

Ett litet IFOV gör att signalen påverkas mindre av störningar från bakgrund och detektorbrus. Men om IFOV är för litet, blir vissa områden inuti öglorna eller områden mellan öglorna osynliga för systemet (*Figur 6*). Om både den inre delen av en ögla och området mellan öglorna täcks in helt av IFOV, så erhålls fullständig synfältstäckning (eng. full scan coverage). Fullständig synfältstäckning uppnås alltså om samtliga delar inom TFOV (*Figur 3*) skannas över av detektorelementet.



Figur 6. Vita fält innanför den svarta cirkeln illustrerar vilka områden som ej kommer skannas när IFOV är för litet. Bredden på den röda linjen beror på storleken på IFOV.

Då alla öglor i rosettmönstret har samma storlek och form är det tillräckligt att betrakta en ögla och dess närmaste grannar. Låt oss betrakta ögla 1. Genom att jämföra diametern hos IFOV dels med den största vidden på öglan och dels med vidden på öglan vid den punkt där öglan skär sin närmsta grannes ögla, är det möjligt att avgöra om öglans inre delar kommer att bli synliga för systemet. Låt t_b vara tidpunkten då detektorn befinner sig vid den bredaste delen av första öglan och t_s tidpunkten då detektorn befinner sig vid skärningspunkten mellan första och andra öglan. Om $x(t_b) \leq x(t_s)$ kan metoderna nedan (avsnitt 2.3 och 2.4) användas för att beräkna IFOV. Kravet $(x(t_b) \leq x(t_s))$ uppfylls då $\Delta N > 3$, enligt [1]. Kravet kan uttryckas i polära koordinater, $r(t_b) \leq r(t_s)$, och gäller då för godtycklig ögla. Då $\Delta N < 3$ skär inte närliggande öglor varandra över huvud taget.

Låt oss anta att en ögla skär närliggande öglor. Då kan en cirkel genom alla dessa skärningspunkter ritas ut (*Figur 7*). Vinkeln mellan två skärningspunkter blir $2 \cdot \pi / N$. Anta att ett par öglor skär varandra vid tidpunkten t_s . Från de polära ekvationerna för rosettmönstret (ekvation 3 och ekvation 4), kan avståndet från rosettens centrum till denna skärningspunkt lösas ut. Avståndet blir:

$$r(t_s) = \rho \cdot \cos(\pi / \Delta N)$$

ekvation 5

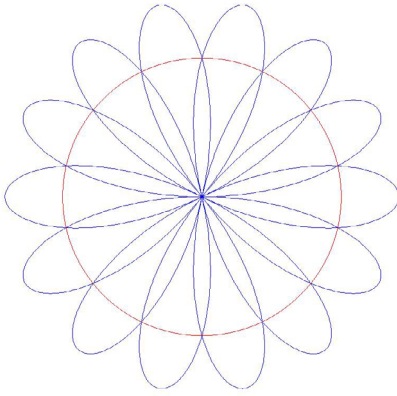
Mellan två närliggande skärningspunkter skapas en cirkelbåge, w . $w > 0$ eftersom cirkelns radie är större än 0. Cirkelbågen kan uttryckas med hjälp av vinkeln, v och radien, r . (Figur 8)

$$w = v \cdot r = \frac{2 \cdot \pi}{N} \cdot \rho \cdot \cos(\pi / \Delta N) > 0 \Rightarrow \cos(\pi / \Delta N) > 0$$

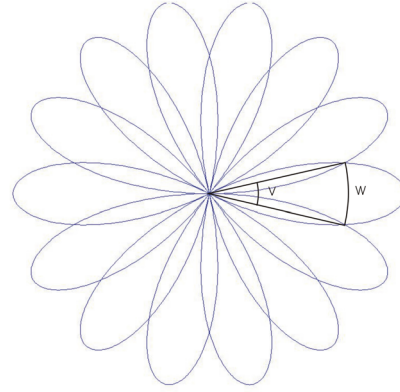
$$\Rightarrow -\pi/2 < \pi / \Delta N < \pi/2 \Rightarrow \Delta N > 2 \vee \Delta N < -2 \Rightarrow$$

$\Rightarrow$ andra lösningen förkastas då $\Delta N = N_1 - N_2$ alltid är större än 0.

$\therefore$ Om $\Delta N \leq 2$ så skär ejöglorna varandra och $r(t_s)$ kan ej uttryckas på ovanstående sätt (ekvation 5).

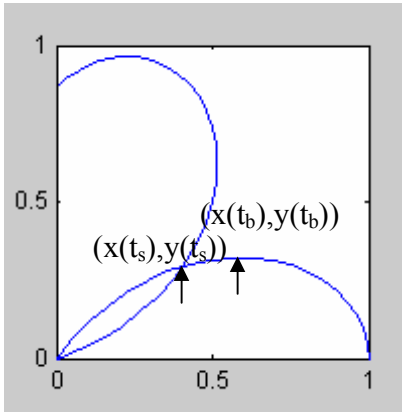


Figur 7. Rosettmönster med cirkel genom öglornas skärningspunkter.

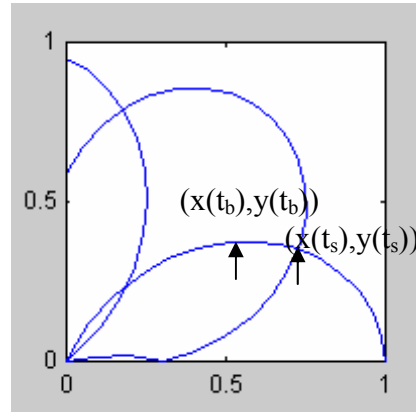


Figur 8. Rosettmönster med utritad cirkelsektor mellan två skärningspunkter.

Figur 9 a) visar ett exempel då $x(t_b) > x(t_s)$. I detta fall kommer inte full täckning inuti öglorna att uppnås om diametern på IFOV beräknas med någon av nedanstående metoder. För att hela den inre delen av öglan ska täckas in måste diametern av IFOV i detta fall vara lika stor som öglans bredaste del. I Figur 9 b) visas ett fall där de inre delarna täcks in helt när IFOV beräknas med metoderna nedan, d.v.s. $x(t_b) < x(t_s)$.



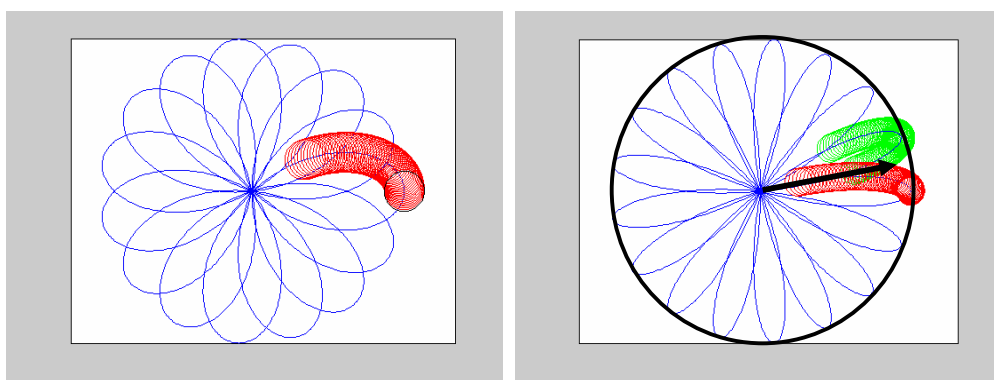
a)



b)

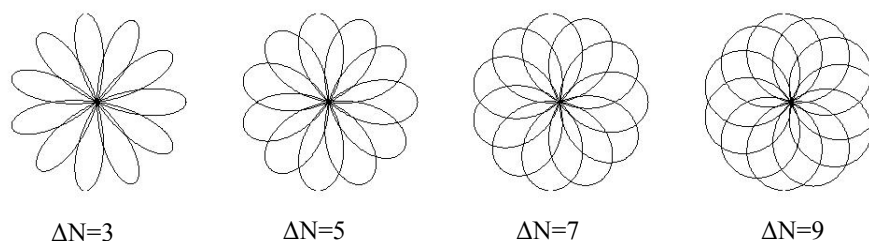
Figur 9. a) $\Delta N=3$, $N_1=4$, $N_2=1$: $x(t_b) > x(t_s)$ b) $\Delta N=5$, $N_1=6$, $N_2=1$: $x(t_b) < x(t_s)$

Nu har de inre delarna av rosetten undersökts. Låt oss nu betrakta området som uppstår utanför två närliggande öglor. Rosettmönstret, som visas i bilderna i detta dokument och som ges matematiskt av ekvation 1-4, visar endast var centrum av detektorelementet befunnit sig för varje tidpunkt. På grund av utbredningen hos detektorelementet (IFOV) så får linjen som beskriver mönstret en bredd lika stor som diametern hos IFOV (*Figur 10 a*). Betrakta den yttersta skärningspunkten mellan två sådana breda, närliggande öglor. I *Figur 10 b* pekar den tjocka, svarta pilen ut denna skärningspunkt. Om avståndet från rosettens centrum till denna skärningspunkt är mindre än halva TFOV så erhålls ej full täckning mellan närliggande öglor, utan det uppstår glapp, gropar mellan öglorna i de yttre delarna av rosettmönstret. Djupet på dessa gropar minskar då ΔN ökar (*Figur 11*).



Figur 10. a) $N_1=11$, $N_2=4$

b) $N_1=10$, $N_2=7$



Figur 11. Groparna mellan öglorna minskar då ΔN ökar. Antalet öglor $N = 11$ i samtliga bilder.

2.3 Konventionell metod för att beräkna storleken på IFOV

I den konventionella metoden ([4], [11]) approximeras diametern på IFOV med längden, ω_k , på cirkelbågen mellan p_1 och p_2 (*Figur 12 a*). p_1 och p_2 utgör skärningspunkterna mellan en ögla och dess närmaste grannar. r_1 och r_2 (*Figur 12*) utgör avstånden från rosettmönstrets centrum till respektive skärningspunkt, p_1 och p_2 . Avstånden r_1 och r_2 ($|\vec{r}_1| = |\vec{r}_2|$) kan beräknas ur rosettmönstrets polära ekvationer (ekvation 3 och ekvation 4) då vinkeln v , enligt *Figur 12*, är känd (se ekvation 6).

$$v = 2 \cdot \theta(t) = 2 \cdot \pi / N.$$

ekvation 6

Storleken på IFOV ges av ekvation 7:

$$\omega_k = v \cdot r = \frac{2 \cdot \pi}{N} \cdot \rho \cdot \cos\left(\frac{\pi}{\Delta N}\right)$$

ekvation 7

$$\text{där } r = r_1 = r_2, N = (f_1 + f_2)/f, \frac{N_2}{N_1} = \frac{f_2}{f_1}, \Delta N = (N_1 - N_2) \geq 3,$$

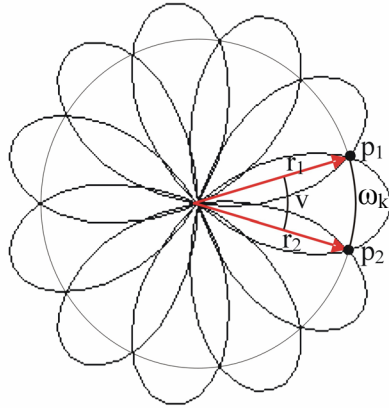
observera dock att då $\Delta N=3$ uppnås ej täckning inuti öglorna, se ovan i avsnitt 2.2.

Då $\Delta N < 3$ skär inte närliggande öglor varandra.

f_1 och f_2 = prismornas rotationsfrekvens, $f_1 > f_2$

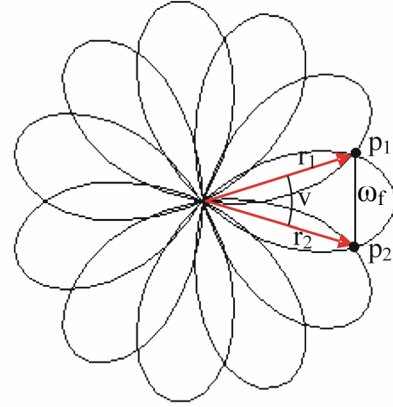
$f = f_1$:s och f_2 :s största gemensamma divisor,

ρ = prismornas brytningsvinkel



a)

Figur 12. a) Konventionell metod.



b)

Figur 12. b) Förbättrad metod.

2.4 Förbättrad metod för att beräkna storleken på IFOV

För att minska interferens från bakgrundssignaler och detektorbrus bör IFOV vara så liten som möjligt, men tillräckligt stor för att uppnå full täckning av det totala synfältet. Genom att definiera diametern på IFOV som det euklidiska avståndet, ω_f , mellan p_1 och p_2 (Figur 12 b), blir IFOV mindre än då den konventionella metoden används ([2], [3], [5], [6]). Avståndet kan beräknas med hjälp av cosinussatsen:

$$\omega_f = \sqrt{r_1^2 + r_2^2 - 2 \cdot r_1 \cdot r_2 \cdot \cos v} = \rho \cdot \cos\left(\frac{\pi}{\Delta N}\right) \cdot \sqrt{2 - 2 \cdot \cos\left(\frac{2 \cdot \pi}{N}\right)} \quad \text{ekvation 8}$$

där $\Delta N = N_1 - N_2$ och $\Delta N \geq 3$, observera dock att då $\Delta N=3$ uppnås ej täckning inuti öglorna, enligt avsnitt 2.2. Då $\Delta N < 3$ skär inte närliggande öglor varandra.

r_1 och r_2 ($|\bar{r}_1| = |\bar{r}_2|$) är avstånden från rosettens centrum till skärningspunkterna p_1 respektive p_2 .

v är vinkeln mellan $\bar{r}_1$ och $\bar{r}_2$.

3 Signalbehandling

3.1 Tyngdpunktsmetoden

Målets position bestäms genom att beräkna tyngdpunkten för de signalvärden som härrör från målet. Eliminering av brus och bakgrund från signalen (målet) sker genom tröskling. Trösklingsfunktionen ges av:

$B(n) = 1$, if $T_1 < S(n)$ else $B(n) = 0$. B är den trösklade signalen, S är

ursprungssignalen och n anger aktuellt sampel. Trösklingen sker för varje sampel i signalen. Tröskelvärde sätts strax under medelintensiteten av målsignalen,

$T_1 = a \cdot I_{avg}$, $0 < a < 1$.

När tyngdpunkten beräknas kan detta ske på två sätt. Antingen beräknas en tyngdpunkt endast utgående från den trösklade signalen (ekvation 9) eller så beräknas tyngdpunkten med hjälp av den trösklade signalen som mask i den ursprungliga signalen (ekvation 10).

$$\bar{X} = \frac{\sum_{n=1}^N X(n) \cdot B(n)}{\sum_{n=1}^N n \cdot B(n)} \quad \bar{Y} = \frac{\sum_{n=1}^N Y(n) \cdot B(n)}{\sum_{n=1}^N n \cdot B(n)} \quad \text{ekvation 9}$$

$$\bar{X} = \frac{\sum_{n=1}^N X(n) \cdot B(n) \cdot S(n)}{\sum_{n=1}^N n \cdot B(n) \cdot S(n)} \quad \bar{Y} = \frac{\sum_{n=1}^N Y(n) \cdot B(n) \cdot S(n)}{\sum_{n=1}^N n \cdot B(n) \cdot S(n)} \quad \text{ekvation 10}$$

X och Y innehåller de x- och y-koordinater som beskriver rosettmönstret. B är den trösklade signalen. S innehåller signalens intensitetsvärden. N är antalet sampel i signalen.

Eftersom rosettmönstret är icke linjärt blir formen och storleken av ett upptäckt objekt beroende av objektets position i TFOV och den beräknade tyngdpunkten kommer dras något mot mitten av rosettmönstret.

3.2 Störskydd

Storleken på TFOV har betydelse för hur känslig en målsökare är för störning. Ett stort TFOV ökar risken för att målsökaren ska påverkas negativt av brus och eventuella motmedel. Även avståndet mellan mål och målsökare har betydelse.

Om t.ex. ett mål, som befinner sig på långt avstånd från en målsökare, släpper en fackla så kommer facklan bli kvar inom TFOV under en lång tid och när målet är nära målsökaren så kommer facklan bara att befinna sig inom TFOV under en mycket kort tid.

Den rosettskannande målsökaren har en väsentlig fördel när det gäller störskydd eftersom den har ett IFOV som är mycket mindre än dess TFOV. Den tid som

detektorn uppehåller sig på målet är relativt kort och detta gör att den blir svår att störa ut med laser och andra direktriktade störkällor (eng. jamming), enligt [12].

För att ett rosettskannande system ska lyckas följa ett mål bör signalbehandlingen innefatta ett störskydd mot motmedel, såsom facklor. Nedan presenteras några möjliga metoder. Samtliga metoder, utom metod 5, förutsätter att inga motmedel finns i synfältet vid påläsning. Många metoder utnyttjar någon form av logik för att urskilja målet från bakgrund, brus, facklor och liknande. Logikens uppgift är att jämföra olika egenskaper som extraherats från pulserna, t.ex. skulle logik kunna upprättas som känner igen mål i förhållande till bakgrund genom storleken; bakgrund är generellt större än målet. I detta fall skulle alltså beslutet kunna baseras på pulslängden. En bred puls förkastas medan en smal puls accepteras. Mycket vanligt är det att betrakta pulsamplituden och förkasta för höga pulstoppar då dessa sannolikt härrör från facklor. Det är även möjligt att betrakta flera pulser sekventiellt innan beslut fattas, då kan information såsom pulsamplitud, pulslängd och position lagras i ett minne för att sedan utnyttjas i mönsterigenkännings-algorimer för att avgöra vilken puls som ska användas för följning.

3.2.1 Metod 1 –Tyngdpunktsteknik med två tröskelvärden

När IR-detektorn passerar över målet och facklorna skapas signalpulser. Då facklornas intensitet oftast är mycket högre än målets är det oftast möjligt att skilja dem från målet. Genom att utnyttja två tröskelvärden går det att eliminera de delar av signalen som har för låga intensitetsvärden (bakgrundsbrus) och för höga intensitetsvärden (facklor) och kvar blir den del av signalen som härrör från målet. På så sätt är det möjligt att minska effekterna av motmedlet. Trösklingsfunktionen ges av: $B(n) = 1$, if $T_1 < I < T_2$ else $B(n) = 0$. Från den kvarvarande signalen beräknas målets position med hjälp av tyngdpunktmetoden, enligt avsnitt 3.1. Två tröskelvärden används endast när det är troligt att systemet är utsatt för störning. För att avgöra detta jämförs intensitetsvärdena från nuvarande signal med intensitetsvärdena från föregående signal, en plötslig intensitetsstegring tyder på att systemet är utsatt för störning. I annat fall sker beräkning av målposition enligt avsnitt 3.1.

Förhållandet mellan de två tröskelvärdena, T_1 och T_2 , kan definieras som $T_2 = a \cdot T_1 = a \cdot b \cdot I_{avg}$, där I_{avg} är medelintensiteten för den signal som antogs motsvara målet i föregående rosett/bild, a är en konstant som anger hur mycket starkare intensitet som krävs för att signalen ska tolkas som en fackla, $a > 1$ och b är en konstant som anger hur mycket lägre intensitet som krävs för att signalen ska tolkas som bakgrundsbrus, $0 < b \leq 1$.

3.2.2 Metod 2 - Klusterteknik

Nackdelen med metod 1 är att facklornas intensitet varierar med tiden och vid den tidpunkt då facklornas intensitet ej skiljer sig nämnvärt från målets intensitet är det ej möjligt att separera signalerna, vilket medför att tyngdpunktsberäkningen kommer att ge ett missvisande resultat. Likaså misslyckas målföljningen om två liknande objekt/mål befinner sig inom synfältet.

Genom att dela upp signalvärdena i flera kluster beroende på avståndet mellan de spatiella koordinaterna hos signalvärdena i den trösklade signalen är det möjligt att separera mål från falskmål. Om det spatiella avståndet mellan två sampel är mindre än IFOV så kan inte målsökaren särskilja de två samplen. Storleken på IFOV är alltså avgörande för systemets upplösning. En rimlig tumregel för när det är möjligt att anta att två sampel härrör från olika mål, och alltså ska tillhöra två olika kluster, är när det spatiella avståndet i x-y-planet är större än IFOV. Men bara för att två sampel har ett större spatiellt avstånd än IFOV så behöver det inte betyda att de inte i slutändan hamnar i samma kluster, det kan finnas andra sampel som förenar dem. Efter att klusteruppdelningen skett beräknas tyngdpunkten för varje kluster. Utgående från egenskaper, såsom storleken på klustren, deras medelintensitet samt deras positioner, väljs det mest lämpliga målet.

3.2.3 Metod 3 – Kombination av metod 1 och 2

Om ett väldigt intensitetsstarkt föremål, t.ex. en fackla, dyker upp i metod 2 finns det risk för att övriga målkandidater blir undertryckta redan vid trösklingsmomentet. Bäst resultat erhålls om metod 1 kombineras med metod 2. Metod 1 kommer att eliminera facklor som befinner sig i ett tidigt skede med hög intensitet medan metod 2 lyckas eliminera dem även om de har samma intensitet som målet genom att betrakta storleken hos objekten. Facklorna antas vara mindre än målet. Storlek kan definieras som utsträckning i x- respektive y-led eller som antal pixlar.

3.2.4 Metod 4 – Sektordämpning

Genom att utnyttja tidsluckor, då systemet bortser från vad detektorn känner av, är det möjligt att minska systemets störkänslighet. Efter pålåsning centrerar systemet det valda målet till mitten av rosettmönstret. Därför kan det vara lämpligt att utesluta signalvärden som ligger utanför en given radie och därmed minska risken att målsökaren läser över på ett falskmål. D.v.s. rosettmålsökarens TFOV minskas efter att målval och centrering skett. Nackdelen är att detta kan medföra en sämre följeprestanda om målet gör en plötslig manöver.

3.2.5 Metod 5 - Multispektral teknik

Det finns flera tänkbara störskydd med diverse för och nackdelar. När facklan och målet har en liknande intensitet (inträffar någon gång under tidsförloppet eftersom facklan har en intensitet som varierar med tiden) är det med många metoder svårt att skilja mål och fackla åt. Då kan en lösning vara att betrakta flera våglängdsband, eftersom det finns många skillnader mellan mål och fackla när fördelningen av strålningsenergin över ett större våglängdsområde beaktas. Facklor har i allmänhet andra strålningsegenskaper än målet. Målet strålar i huvuddel inom IR-bandet medan facklor strålar både inom IR- och UV-bandet. Roboten Stinger-POST utnyttjar en IR-detektor, som arbetar i 3 – 5 μm -området, för att upptäcka intressanta objekt och en UV-detektor för att undvika att läsa över på facklor ([8], [9]). Både målet och facklorna har större strålningsintensitet än himlen i IR-bandet men i UV-bandet emitterar målet mindre energi än himlen, medan facklorna emitterar mer energi än himlen. Genom att utnyttja detta kan det korrekta målet väljas ut.

4 Utvecklade program

4.1 ros1 – ett program utvecklat i Matlab

Ett enkelt Matlab-program har utvecklats, där användaren har möjlighet att studera ett rosettmönsters egenskaper. Användaren kan välja hur många öglor rosettmönstret ska innehålla och därefter få reda på vilka kombinationer av N_1 och N_2 som därmed är möjliga. Användaren kan välja att se detektorelementets storlek i förhållande till TFOV och även flytta runt detektorelementet längs rosettbanan och därmed få förståelse för hur rosettmönstret genereras. En lämplig storlek på IFOV givet TFOV, N_1 och N_2 beräknas av programmet med hjälp av den konventionella metoden. Men användaren kan även ange en valfri storlek för att ha möjlighet att studera konsekvenserna av för stort eller för litet IFOV.

Användaren kan även skapa en enkel bild bestående av en svart bakgrund med ett eller två konvexa objekt i form av ellipser. Objekten är antingen homogena eller så ges de en gaussisk intensitetsfördelning. Det är även möjligt att lägga på brus på bilden. På den skapade bilden kan sen användaren applicera rosettmönstret och få fram en detektorsignal. Användaren väljer var i bilden rosettens centrum ska placeras samt rosettens diameter. Innan signalvärdena plockas fram ur bilden, faltas bilden med ett medelvärdesbildande filter med samma storlek och form som detektorelementet. På så sätt tas hänsyn till detektorelementets utbredning. Detta underlättar väsentligt när signalen ska plockas fram eftersom det då bara är att läsa av signalvärdena längs rosettspåret i den faltade bilden. Signalen plottas upp i en tvådimensionell graf. Programmet beräknar ett lämpligt tröskelvärde, men användaren kan även själv plocka ut ett lämpligt tröskelvärde ur grafen.

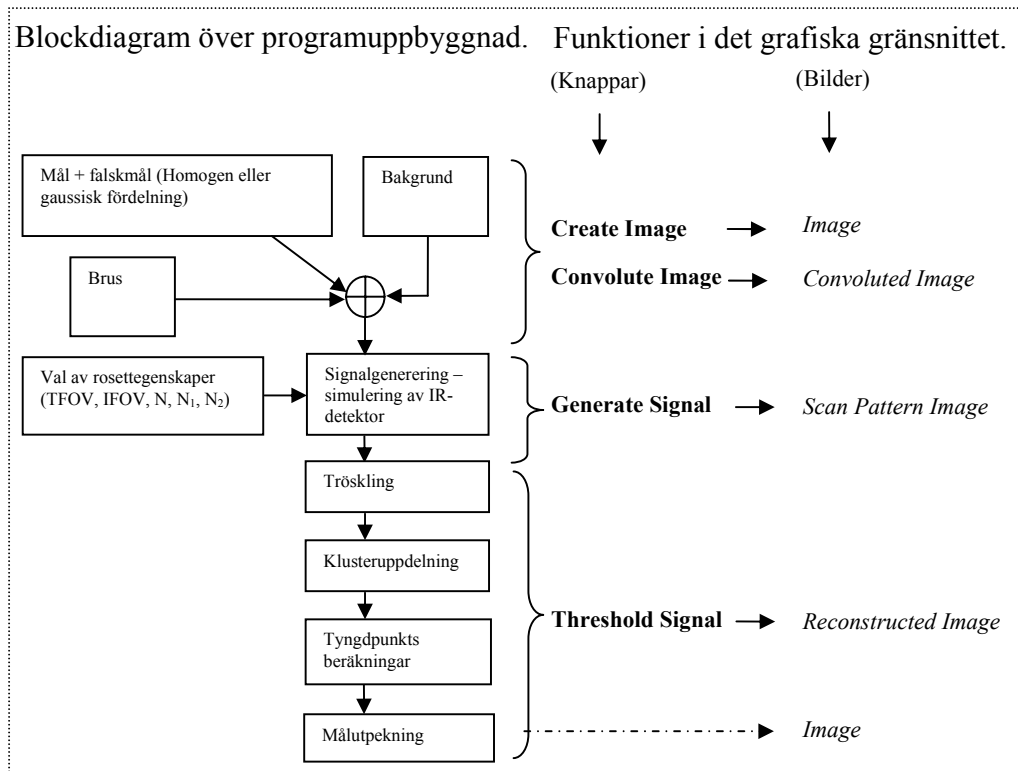
Programmet använder följande formel för att beräkna lämpligt tröskelvärde:

$$T = 0.5 \cdot (S_{\max} - S_{\text{medel}}) + S_{\text{medel}} \cdot S_{\max}$$

$S_{\max}$ är signalens maxvärde. S_{medel} är signalens medelvärde. Den trösklade signalen rekonstrueras sen till en tvådimensionell bild där användaren kan urskilja vad rosettmålsökaren ser. Utgående från signalen beräknas en position där målet antas befinna sig. Positionen beräknas med en tyngdpunktsberäkning och markeras i ursprungsbilden med en stjärna. I Matlab-programmet beräknas alltid tyngdpunkten med hjälp av intensiteten (enligt ekvation 10).

Programmet fungerar endast på bilder där målet har positiv kontrast relativt bakgrunden. Programmet kan bara användas på enskilda bilder och är skapat som ett verktyg som kan ge svar på grundläggande frågeställningar samt presentera rosettmönstrets egenskaper.

I *Figur 13* beskrivs programmets uppbyggnad i form av ett blockdiagram. Dessutom syns kopplingen mellan programmet och knapparna i det grafiska gränssnittet och bilderna som genereras vid de olika knapptryckningarna.



Figur 13 beskriver Matlab-programmets uppbyggnad och relation till det grafiska gränssnittet (som visas i figurerna till exemplen nedan).

Nedan följer fyra exempel från programmet.

4.1.1 Exempel 1

Exempel 1 visar ett scenario innehållande ett mål med gaussisk intensitetsfördelning på en homogen bakgrund. Exempel 1 återfinns i *Figur 14*. Den utpekade målpositionen stämmer väl överens med målets position i inbilden.

Längst ner i figuren visas en graf av detektorsignalen. Den blå kurvan utgör signalens intensitetsvärden. Den gröna kurvan visar detektorelementets avstånd från rosettens centrum. Den svarta kurvan visar detektorelementets vinkelposition. Den röda linjen visar det valda tröskelvärdet. Skalan på y-axeln anger intensitet och hör således ihop med den blå kurvan och den röda linjen. Programmets beräknade målposition visas som en cyan-färgad stjärna i bilden *Image*. Ovanstående stycke gäller för alla nedanstående exempel.

4.1.2 Exempel 2

Exempel 2 (*Figur 15*) visar ett scenario innehållande ett mål och en fackla på en homogen bakgrund. Facklan har 4 gånger högre intensitet än målobjektet och dess area är 2/9 av målets area. I detta fall hamnar den beräknade målpositionen närmare facklan än målet. I *Figur 16* visas en förstoring av den rekonstruerade bilden med den valda målpositionen utritad med en cyan-färgad stjärna. Dessutom finns två röda kryss i bilden. I detta fall har signalvärdena delats upp i flera kluster och de röda kryssen visar klustrens respektive tyngdpunkt. Det är alltså möjligt att

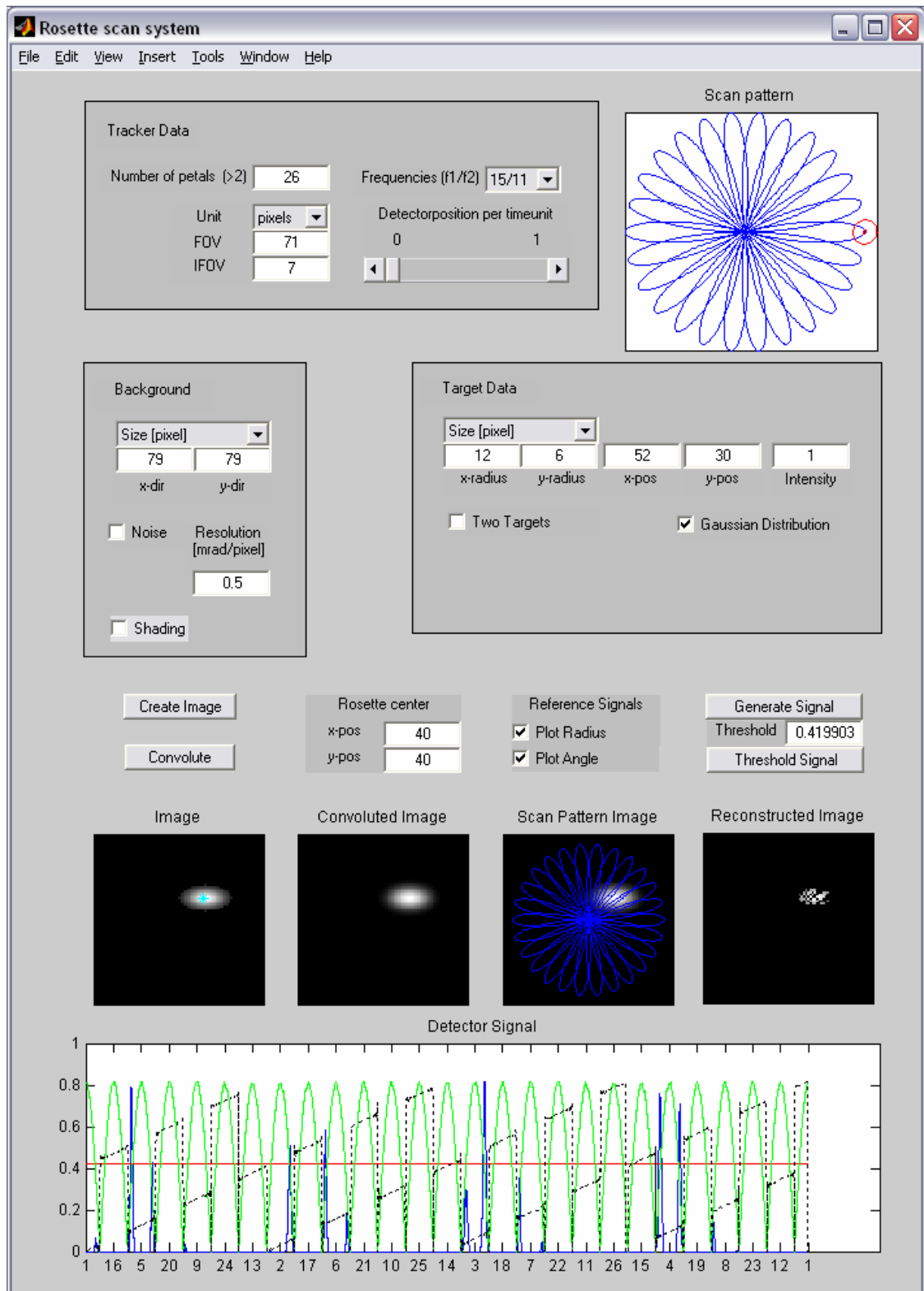
skilja målet från facklan, men sen krävs någon form av regelverk och logik för att avgöra vilket av klustren som ska väljas. Dessa beslutsregler är ej implementerade i Matlab-programmet. När Matlab-programmet delar upp signalvärdena i kluster utnyttjas diametern på IFOV som särskiljningsavstånd.

4.1.3 Exempel 3

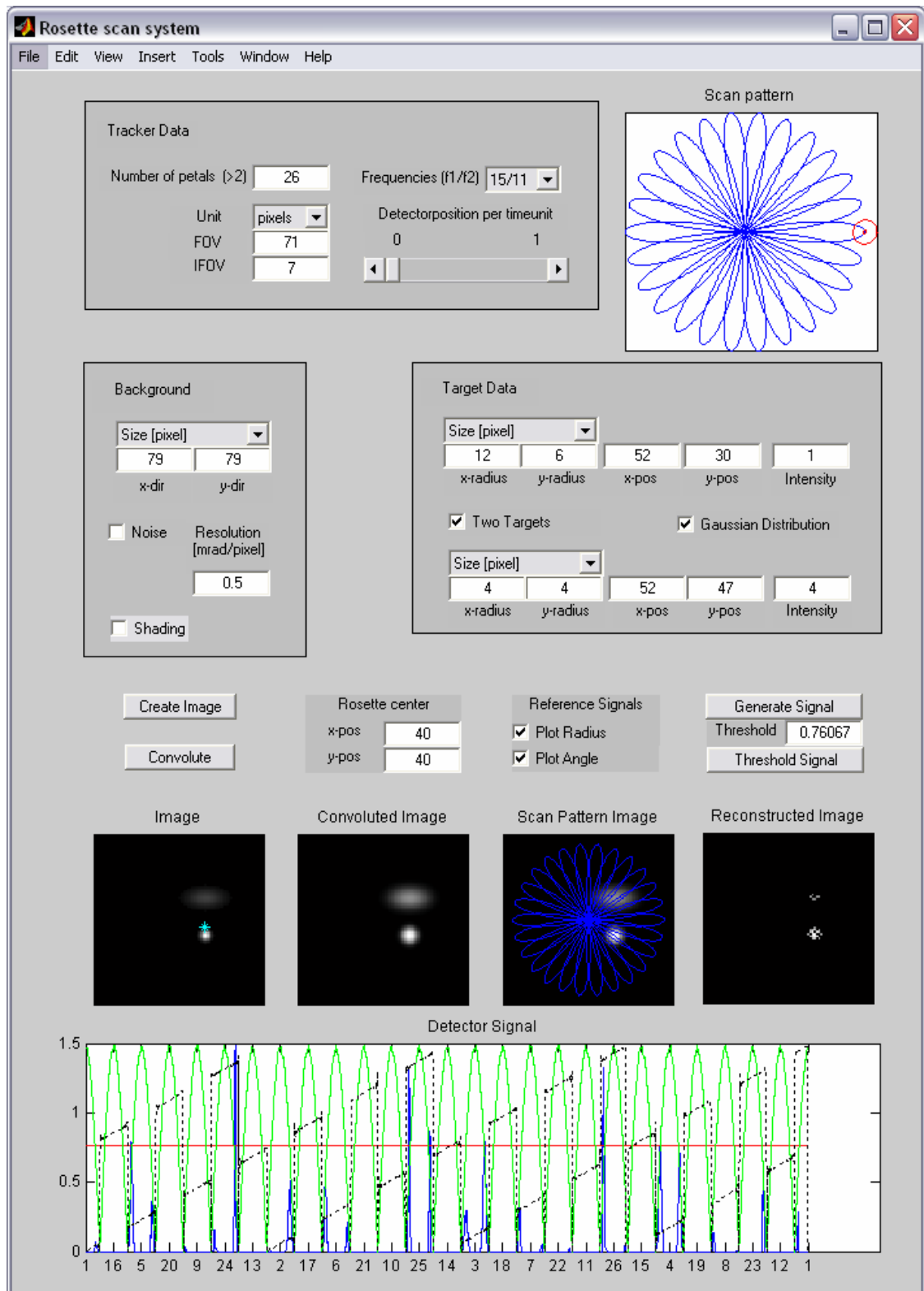
Exempel 3 visar samma scenario som exempel 2 med den skillnaden att objekten har flyttats, så att målet har centrerats i rosettens centrum. Detta för att påvisa rosettmönstrets icke linjära natur och dess konsekvenser. Formen på ett upptäckt objekt blir beroende av objektets position i TFOV. I detta fall hamnar den beräknade målpositionen närmare målet än facklan, vilket beror på att detektorn sveper oftare över rosettens centrum än dess kanter. Exempel 3 återfinns i *Figur 17*.

4.1.4 Exempel 4

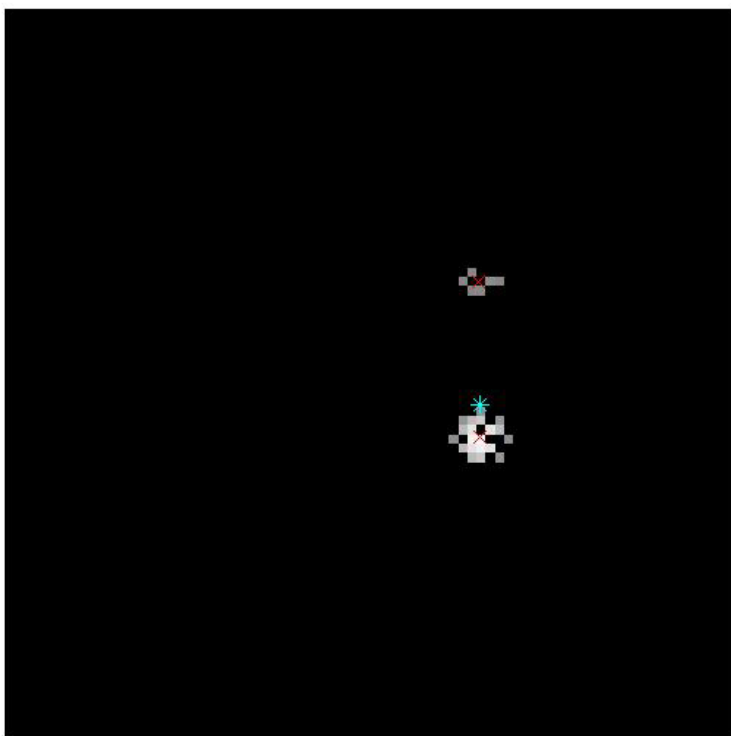
Exempel 4 visar samma scenario som exempel 2 men med brus med variansen 0.01 pålagt i bilden. Exempel 4 återfinns i *Figur 18*.



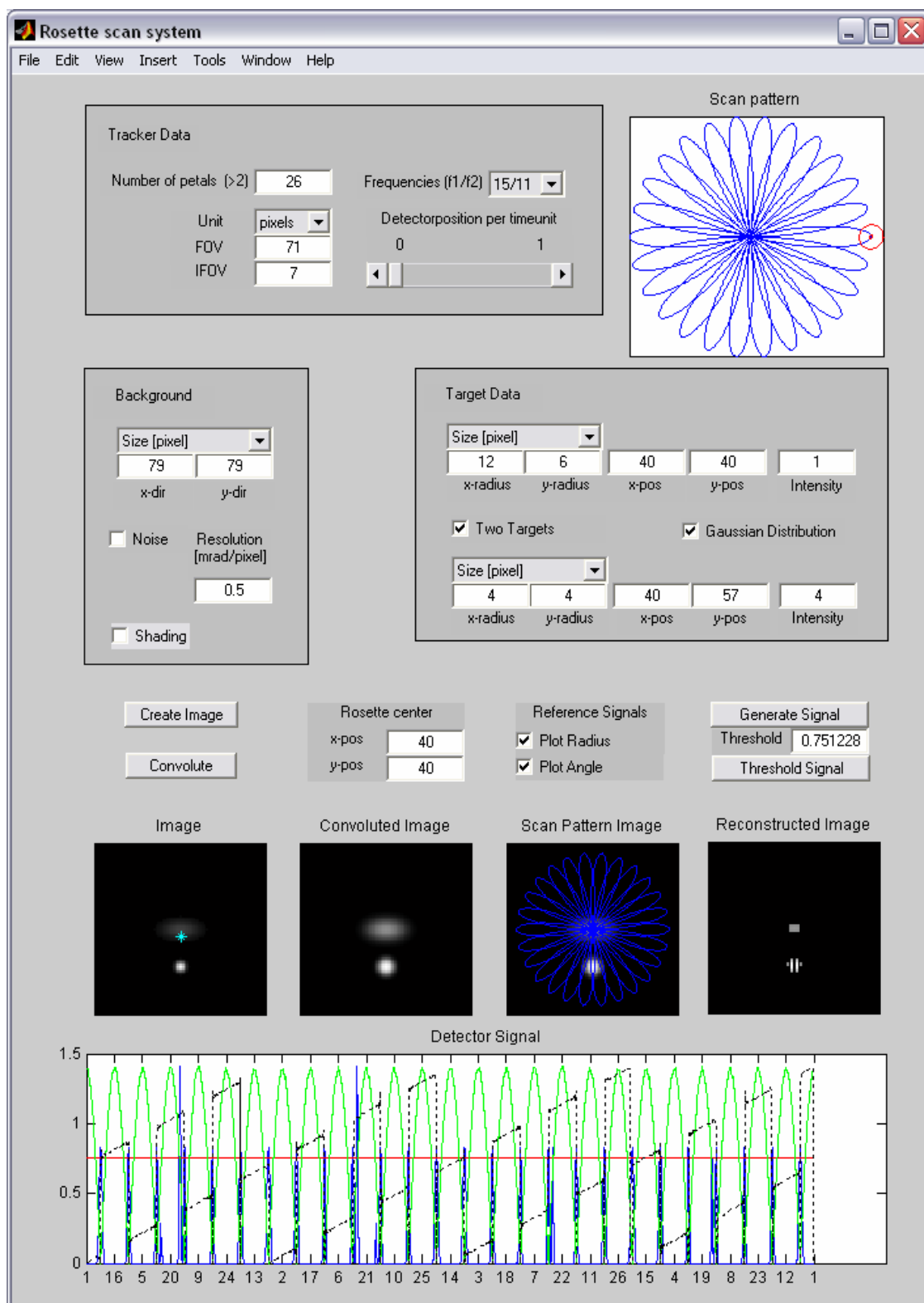
Figur 14. Exempel 1. Mål på homogen bakgrund.



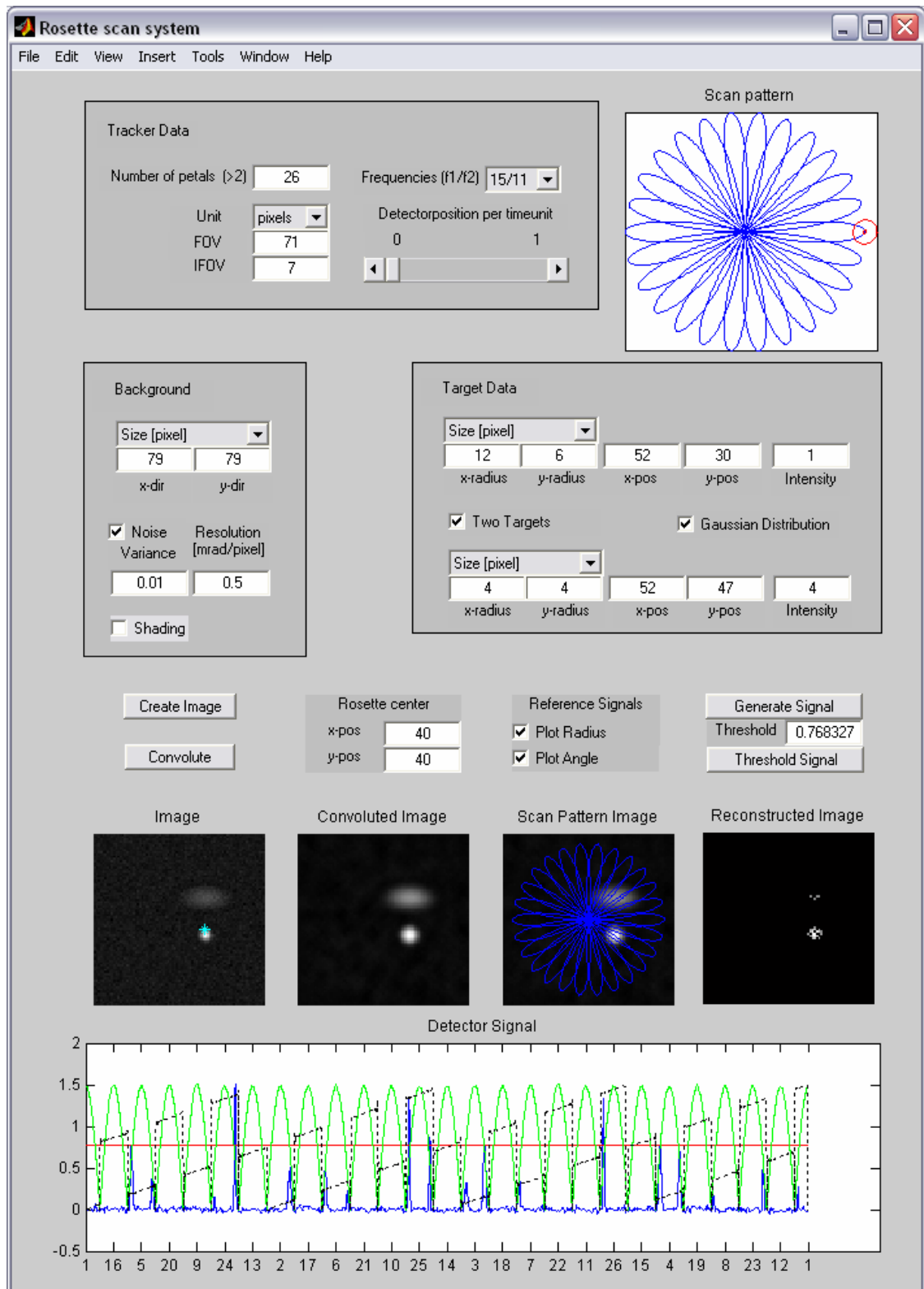
Figur 15. Exempel 2. Mål och fackla på homogen bakgrund.



Figur 16. Rekonstruerad bild från exempel 2. Vald målposition samt enskilda klusters tyngdpunkt utmärkta.



Figur 17. Exempel 3. Mål och fackla på homogen bakgrund. Målet centrerat i rosettmönstrets centrum.



Figur 18. Exempel 4. Samma som i exempel 2 men med brus med variansen 0.01 pålagt i bilden.

4.2 RSS – ett program utvecklat i Visual C++

Nästa steg i utvecklingen, av en datormodell av en rosettskannande målsökare, blev att utveckla ett program där det gick att utföra en målföljarsimulering på en sekvens av bilder. Detta resulterade i programmet RSS (Rosette Scanning System), som är ett program utvecklat i Visual C++. Programmet går att köra fristående mot en bildsekvens, speciellt lämpligt i de fall då målföljning på ett objekt i en verklig, inspelad sekvens ska testas. Men programmet går även att köra som en del i ramverket OPTSIM (OPTronic Simulation Interface Model) [13], vilket lämpar sig bra då användaren själv har skapat ett scenario som denne vill undersöka närmare.

4.2.1 Funktion och uppbyggnad

Programmet är uppbyggt på samma sätt som Matlab-programmet men med den skillnaden att RSS fortsätter att läsa in nya bilder att applicera rosettmålsökaren på, tills bildsekvensen är slut. Dessutom finns det fler valmöjligheter i RSS vad gäller val av olika beräkningsmetoder för att välja ut det bästa målet. I detta program finns även några störskydd implementerade.

När användaren kör simuleringen kan denne välja att betrakta flera vyer av scenariot; originalbilden, den faltade bilden och/eller den rekonstruerade bilden. Intensitetsvärdena i signalen kan plottas upp i en tvådimensionell graf. Den valda målpositionen kan plottas som ett rött kors i bilderna och i de fall då klustringsalgoritmen har använts kan användaren även välja att se positionerna för de olika målkandidaterna. Därutöver kan användaren välja att betrakta rosettmönstrets utseende och position i bilderna samt att rita ut rosettmålsökarens synfält i bilderna.

I inställningsfönstret (*Figur 19*) anger användaren vilka bilder i den inlästa sekvensen som ska användas i simuleringen.

Med hjälp av parametrarna f_1 , f_2 (betecknade $f1$ respektive $f2$ i *Figur 19*) och TFOV kan en lämplig storlek på IFOV beräknas. Två vanliga metoder för att beräkna IFOV är implementerade i programmet, dels den konventionella metoden och dels den förbättrade metoden. Användaren kan även välja att skriva in en egenhändigt vald storlek på IFOV, men måste i så fall avmarkera parametern "Automatic Parameter Update". I fall att "Automatic Parameter Update" är förbockad beräknas även lämpliga värden på sampeltiden och klustringsavståndet automatiskt, baserade på övriga parametervärden.

En rosettmålsökare väljer själv ut det mest lämpliga målet, beroende på vad som är synligt i målsökarens synfält. För att målsökaren ska lyckas göra ett lämpligt målval får synfältet inte innehålla flera tänkbara objekt vid påläsningsögonblicket. Vissa andra målsökare kräver att användaren pekar ut målet som ska följas och en sådan målsökare är inte lika känslig för störningar under påläsningsfasen.

Signalen skapas då detektorelementet flyttas runt rosettmönstret. Storleken på signalen, antalet sampel, bestäms av sampeltiden (tiden mellan varje sampel). Denna får inte vara så stor att glapp uppstår i mönstret. För att kunna applicera rosetten på en bild krävs att vinkelupplösningen, antalet mrad/pixel, för bilden är känd.

Setup

Frame start: 1
 Frame end: 100
 Frame increment: 1

☒ Automatic parameter update

f1: 11
 f2: 5
 Number of petals (>2): 16

TFDV [mrad]: 36.7
 IFDV [mrad]: 6.2406

☒ Conventional IFDV
☐ Improved IFDV

Sampling Time: 0.0005707

Angles resolution [mrad/pixel]: 0.5

Moment technique
☒ Binary Method
☐ Intensity Method

Counter-countermeasure
☐ Two Threshold Levels
☐ Sector Attenuation
☐ Clustering

Intensity Flare/Target: 3
 Radius [pixels]: 25
 Distance [pixels]: 13

x-coordinate: 0 y-coordinate: 0
 Rosette center: 0 0

OK Cancel

Figur 19. Inställningsfönstret för valbara parametrar inför simulering med RSS.

Under signalbehandlingen sker en tröskling av signalen. Tröskelvärden beräknas med hjälp av innehållet i signalen. För den trösklade signalen beräknas slutligen tyngdpunkten. Användaren väljer om signalvärdena ska tas med i beräkningen av tyngdpunkten eller om endast signalens koordinatinformation ska tas med.

En rosett appliceras på varje bild och för varje fullbordad rosett beräknas en tyngdpunkt. Detta innebär en begränsning. I de fall då rosettmålsökaren appliceras på en inspelad sekvens kan inte bildtakten anpassas till aktuella värden på f_1 och f_2 , utan simuleringen kommer i själva verket att motsvara en körning med värdena $f_1 = (N_1 \times \text{bildtakten hos sekvensen})$ och $f_2 = (N_2 \times \text{bildtakten hos sekvensen})$. Detta betyder att användaren lika gärna kan skriva in värdena på N_1 och N_2 istället för f_1 och f_2 när en inspelad sekvens används som indata. I de fall då programmet används via OPTSIM måste användaren manuellt gå in och ändra på bildtaktsparmetern, så att den stämmer överens med aktuella värden på f_1 och f_2 för rosettmålsökaren. Bildtakt/bildfrekvens = $(f_1 + f_2) / N = f$, där N är antalet öglor i rosettmönstret och f är f_1 :s och f_2 :s största gemensamma divisor.

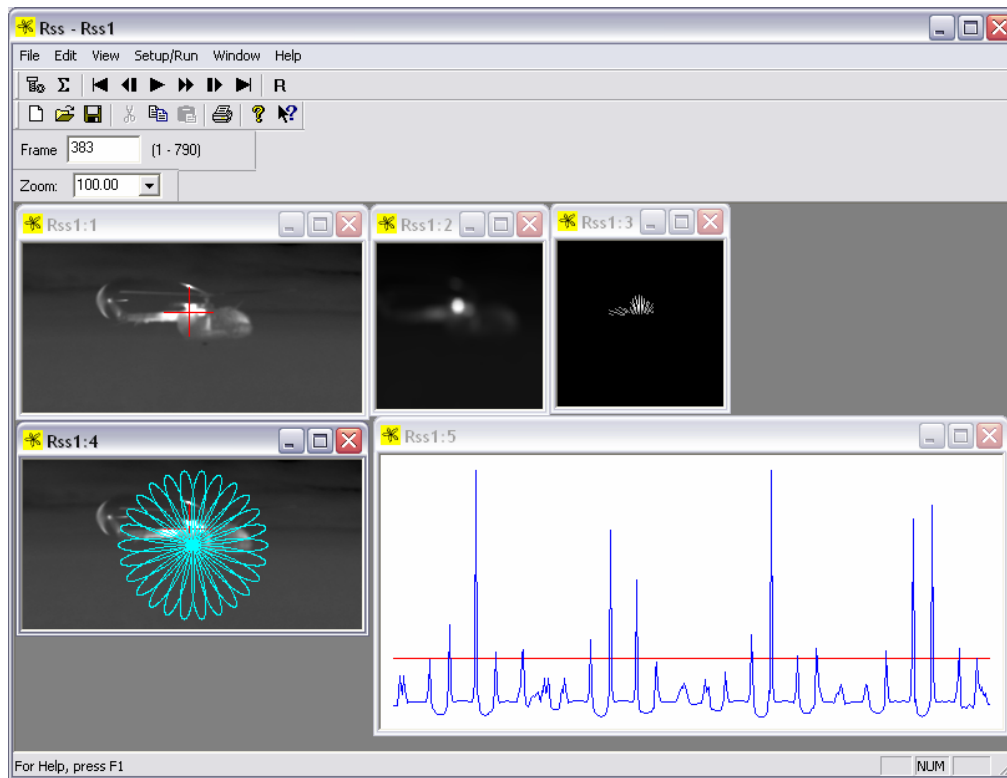
I programmet finns tre olika störskydd implementerade som underlättar för målsökaren att bibehålla läsning på målet även vid förekomst av facklor. De går att använda var för sig eller i valfri kombination. Samtliga förutsätter dock att synfältet bara innehåller ett tänkbart objekt vid påläsningsögonblicket

Ifall att en fackla dyker upp, som är betydligt mer intensitetsstark än målet, kommer målet att försvinna vid trösklingen och den trösklade signalen kommer alltså att beskriva facklan. Detta kan undvikas genom att två tröskelvärden används om intensiteten ökar kraftigt från en bild till en annan. Då tas dels låga signalvärden som härrör från brus och bakgrund och dels höga signalvärden som härrör från facklan bort.

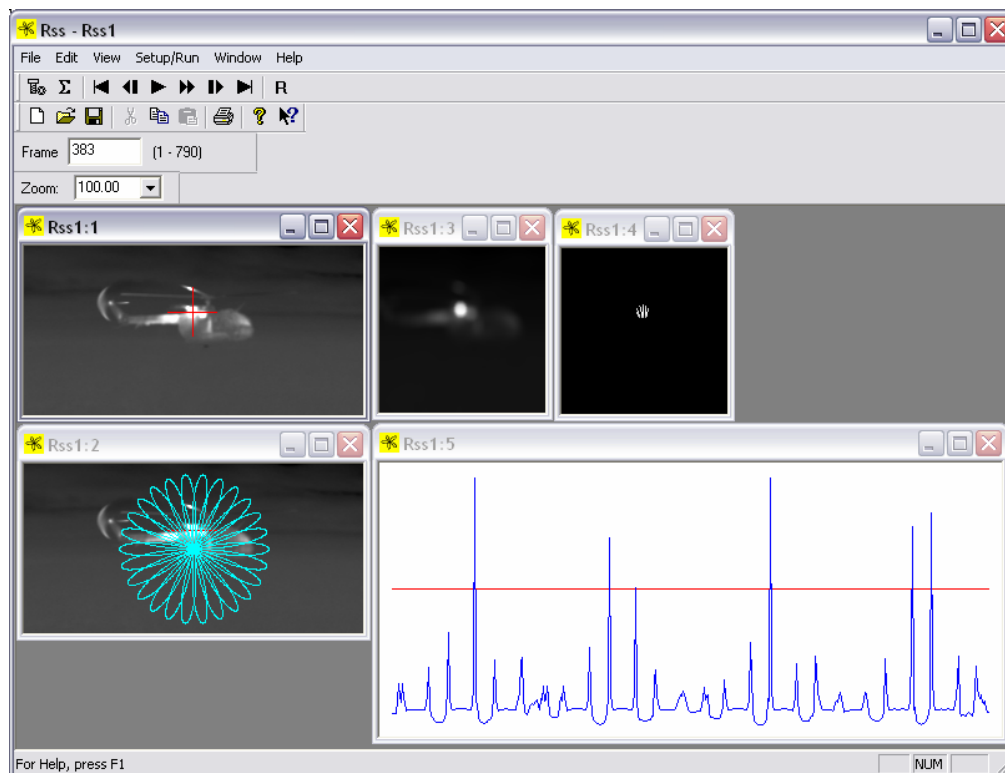
Under ideala, förenklade förutsättningar, då facklan har en homogen intensitetsfördelning, fungerar ovan nämnda störskydd mycket bra. Men eftersom detektorelementet har en utbredning, kommer det känna av en blandning mellan bakgrund och fackla när det sveper över facklan. Detta betyder att målsökaren upplever att facklan har en intensitet som avtar ut mot kanterna och därmed blir det svårt att tröscla bort hela facklan. Kanten på facklan blir kvar. Ett liknande problem uppstår då facklans intensitet avtagit så pass mycket att den knappt avviker från målets intensitet. I detta fall kommer signalen att innehålla både mål och delar av eller hela facklan, efter trösklingen. När sen tyngdpunkten beräknas kommer den att hamna mellan mål och fackla. För att undvika detta och kunna skilja på flera objekt, som ligger separerade spatialt, kan signalen delas upp i flera delar, kluster, beroende på hur samplen ligger i förhållande till varandra avståndsmässigt. Sen beräknas en tyngdpunkt för varje kluster och det kluster som mest liknar det valda målet i föregående bild väljs som mål. Som kriterier för att mäta likhet kan flera egenskaper extraheras från klustren och sen jämföras med motsvarande egenskaper för det valda målet i föregående bild. För närvarande betraktas klustrens medelintensitet och utbredning. Andra intressanta egenskaper skulle kunna vara klustrens position och antalet sampel i varje kluster.

Det sista implementerade störskyddet är sektordämpning. När systemet valt ett lämpligt mål försöker systemet att centrera målet så att det ligger mitt i målsökarens synfält. Att målet är centrerat kan sen utnyttjas genom att signalbehandlingen bortser från signalvärden som ligger för långt ifrån centrum, vilket minskar systemets störkänslighet. Men synfältet får inte minskas för mycket, eftersom det innebär en ökad risk för att systemet tappar målet. Detta störskydd är ej lämpligt att använda då programmet körs fristående mot en bildsekvens, eftersom det då inte finns någon dynamikmodul inkopplad som sköter centreringen av det valda målet.

Figur 20 och *Figur 21* visar två exempel från två olika simuleringar med RSS. Exempelen är likadana så när som på val av tröskelnivå. Första exemplet har en lägre tröskel än det andra exemplet. Tröskelvärdet är dock inget som användaren, för närvarande, själv enkelt kan gå in och ändra på (kan göras i Matlab-programmet). I figurerna visas det mesta som användaren kan välja att betrakta för att få förståelse för en simulering.



Figur 20. $N1=15$, $N2=11$. Rss1:1: Inbild med utritad målkoordinat. Rss1:2: Inbilden faltad med detektorelementet. Rss1:3: Rekonstruerad bild. Rss 1:4: Inbilden med utritat rosettmönster. Rss 1:5: Genererad signal (blå linje) med utritad tröskelnivå (röd linje).



Figur 21. Samma som Figur 20 men med en högre tröskelnivå.

4.3 EwSeeker – en modul i Vega

Slutligen har modellen av rosettmålsökaren integrerats med ramverket EwSim och blivit en del av modulen EwSeeker. Denna modul innehåller, sen tidigare, modeller av korrelations-, centroid- och retikelmålsökare. EwSim är ett ramverk som bygger på den kommersiella produkten Vega. Till produkten hör ett grafiskt användargränssnitt, LynX, som underlättar för användaren då denne definierar och sätter upp en Vega-applikation. För en mer detaljerad orientering, se [14].

Den integrerade rosettmålsökaren är i princip uppbyggd på samma sätt som RSS. Det som skiljer är bl.a. vilka parametrar som går att ändra under respektive inställningsfönster. *Figur 22* visar LynX, där användaren valt att betrakta modulen EwSeeker som innehåller en rosettmålsökare, *rosette_seeker*, och tillhörande parametrar. Vad rosettmålsökaren har för synfält i denna applikation bestäms av den inkommande bilden samt det angivna IFOV, enligt ekvation 11.

$$TFOV = Bildstorlek - IFOV + 1$$

ekvation 11

Därmed är det inte längre nödvändigt för användaren att ange var i bilden rosettmönstret ska appliceras, utan det appliceras mitt i den inkommande bilden.

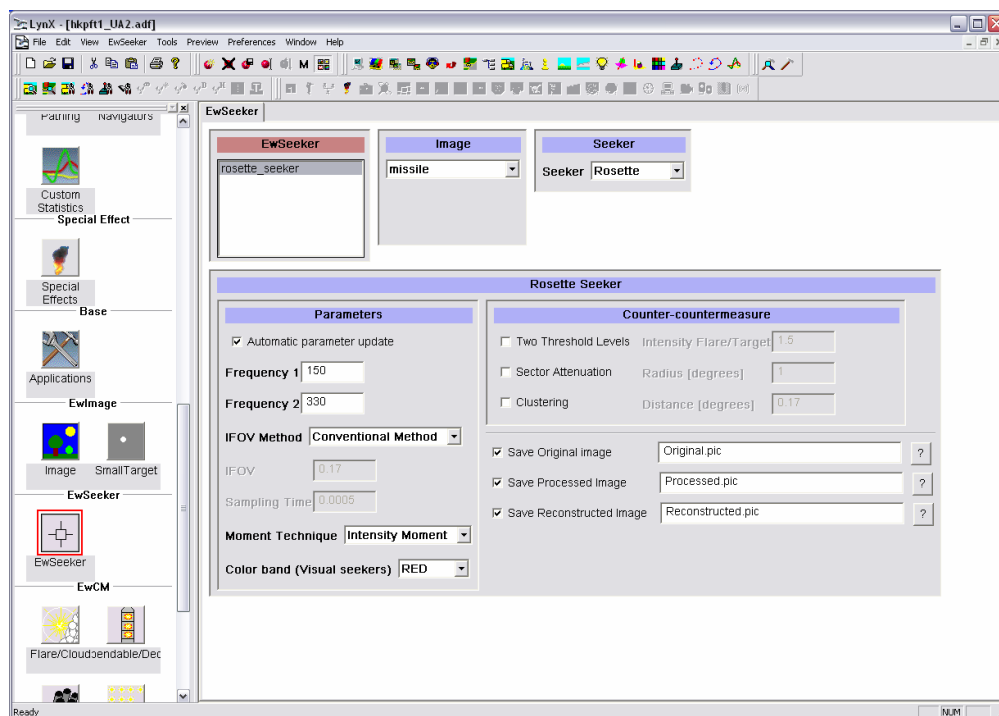
Nackdelen med att köra rosettmålsökaren via Vega är att användaren inte får samma inblick i vad rosettmålsökaren ser, i och med att den rekonstruerade bilden inte kan betraktas medan simuleringen körs. Dessutom visas inte utseendet på rosetten, positionerna för olika målkandidater, den faltade bilden m.m. Den

rekonstruerade bilden och den faltade bilden liksom originalbilden går däremot att spela in under simuleringen för att betraktas i efterhand.

Eftersom simuleringar med hjälp av ramverket EwSim sker på artificiella sekvenser som skapas med hjälp av Vega:s moduler kan programmet räkna ut vinkelupplösningen på den inkommande bilden, till skillnad från RSS där användaren fick ange denna. Tack vare att simuleringarna sker på artificiella sekvenser kan programmet generera nya bilder i den takt som krävs för givna frekvenser, f_1 och f_2 (betecknade *Frequency 1* och *Frequency 2* i Figur 22). Bildtakt/bildfrekvens = $(f_1 + f_2)/N = f$, där N är antalet öglor i rosettmönstret och f är f_1 :s och f_2 :s största gemensamma divisor.

I RSS valde användaren start- och slutbild för simuleringen och pålåsningen skedde i första bilden. När man kör rosettmålsökaren via Vega är det användaren som väljer när missilen ska apteras, låsas och avfyras mot ett mål. På så sätt kan användaren göra en ny låsning om föregående misslyckats. Den möjligheten har inte användaren av RSS.

Den stora fördelen med modulen EwSeeker och ramverket EwSim är att de möjliggör målsökarsimuleringar i nära realtid, vilket bidrar till att en operatör kan medverka i simuleringen. Därmed blir programmen även lämpliga som utbildningsverktyg. Ett program som arbetar i nära realtid är dessutom lämpligt för mängdkörningar.



Figur 22. Grafiska gränssnittet LynX med modulen EwSeeker aktiverad.

5 Slutsatser

IR-missiler kan använda en mängd olika störskydd för att upptäcka och förkasta facklor. Det rosettskannande systemet har fördel av ett litet IFOV som ger det möjlighet att särskilja ett flertal källor som finns inom dess totala synfält. En annan anmärkningsvärd egenskap hos rosettmönstret är att dess centrum genomkorsas av detektorelementet en gång för varje ögla, medan varje ögla bara korsas en gång per bild, d.v.s. mönstret har en inneboende egenskap att framförallt skaffa information från bildens mitt. Det faktum att det rosettskannande systemet utnyttjar ett litet detektorelement för att söka av ett större område bidrar även till att systemet är svårt att störa ut med laser och andra direktriiktade störkällor (enligt [12]), eftersom den tid som detektorn uppehåller sig på målet är relativt kort.

En rosettmålsökaralgoritm utan störskydd kan inte hantera mer än ett objekt i taget inom synfältet. I de fall då det förekommer flera objekt i synfältet, kommer rosettmålsökaren att tolka det som ett mål och den valda målpositionen kommer att hamna mellan dessa objekt. Ett flertal störskydd, såsom användandet av två tröskelvärden eller sektordämpning, syftar till att utesluta objekt så att det bara blir ett intressant objekt kvar i signalen eller synfältet. Dessa störskydd är därmed ganska känsliga. I fall att delar av de ointressanta objekten fortfarande finns kvar inom synfältet eller i signalen kommer resultatet i princip bli likadant som om inget störskydd fanns implementerat överhuvudtaget - den valda målpositionen kommer att hamna någonstans mellan objekten. I detta hänseende fungerar andra störskydd bättre, t.ex. klusteruppdelning av signalen eller någon form av multispektral teknik. Klusteruppdelning är dock mer beräkningsintensivt och multispektral teknik kräver mer komplicerad sensorhårdvara. Klusteruppdelningsmetoden klarar att särskilja ett godtyckligt antal objekt, men när falskmålen uppträder på ett sätt som liknar det riktiga målet blir det slutgiltiga målvalet svårt. I en sådan situation underlättar det om rosettmålsökaren utnyttjar två eller flera våglängdsområden. Det finns oftast många skillnader i fördelningen av strålningsenergi hos mål och facklor om ett större våglängdsband, från IR till UV inklusive VIS, utnyttjas.

Validering och verifiering av modellerna mot ett befintligt system har ej kunnat genomföras, då tillgång till ett sådant system saknas. Dock har rimligheten i simuleringresultaten bedömts som god, utgående från teorin. Ytterligare utvärdering av modellerna är önskvärd när/om ett verkligt system blir tillgängligt.

Endast ett fåtal simuleringar har utförts med de olika modellerna och i framtiden skulle det vara av intresse med en simuleringstudie för att ytterligare beskriva och bekräfta ett rosettskannade systems funktion under olika omständigheter.

6 Referenser

- [1] Tajima T, Wakabayashi S, Kondo M, Takei T: *Rosette scan infrared sensor*, Electro-Optical Technology for Autonomous Vehicles, Proc. SPIE, Vol. 219, pp. 51-57, 1980.
- [2] S.G. Jahng, H.K. Hong, S.H. Han, J.S. Choi: *Design and analysis of improved instantaneous field of view of rosette scanning infrared seeker*, Electronics Letters, IEE, Vol. 33, No. 23, pp. 1964-1965, November 1997.
- [3] S.G. Jahng, H.K. Hong, S.H. Han, J.S. Choi: *Design of instantaneous field of view of rosette scanning infrared seeker and dynamic simulation*, Proc. SPIE, Vol. 3365, pp. 158-168. April 1998.
- [4] S.G. Jahng, H.K. Hong, S.H. Han, J.S. Choi: *Dynamic simulation of the rosette scanning infrareds seeker and an infrared counter-countermeasure using the moment technique*, Opt. Eng. , Vol. 38, No. 5, pp. 921-928, May 1999.
- [5] S.G. Jahng, H.K. Hong, J.S. Choi: *Simulation of rosette scanning infrared seeker and counter-countermeasure using K-means algorithm*, IEICE Trans. Fundamentals of Electronics, Communication and Computer Sciences, Vol. E82-A, No. 6, pp. 987-993, June 1999.
- [6] S.G. Jahng, H.K. Hong, D.S. Seo, J.S. Choi: *New infrared counter-countermeasure technique using an iterative self-organizing data analysis algorithm for the rosette scanning infrared seeker*, Opt. Eng., Vol. 39, No. 9, pp. 2397-2404, September 2000.
- [7] W. Haifeng, L. Zhi, Z. Qing, S. Xinzhi: *A double band infrared image processing system using rosette scanning*, Detectors, Focal Plane Arrays and Applications, Proc. SPIE, Vol. 2894, pp. 2-10, 1996.
- [8] K.S Doo, J.S Oh, S.G. Jahng, H.K Hong, J.S Choi, D.S. Seo: *Simulation of target detection in ultraviolet and infrared bands*, Opt. Eng. , Vol. 40, No. 11, pp. 2646-2654, November 2001.
- [9] J.J. May, M.E. Van Zee: *Electro-Optic and Infrared Sensors*, Microwave Journal, pp. 121-131, September 1983.
- [10] Maj Craig: *Advanced Infrared Missile Counter-countermeasures*, Journal of Electronic Defense, pp. 47-50, 67, 70, January 1994.
- [11] G.J. Zissis, W.L. Wolfe, eds.: *The Infrared Handbook*, Environmental Research Institute of Michigan, Ann Arbor, MI, Ch. 10, 22, 1978.
- [12] J.S. Accetta, D.L. Shumaker, eds.: *The infrared and Electro-Optical Systems Handbook*, Environmental Research Institute of Michigan, Ann Arbor, MI, Vol. 7, Ch. 3, 268, 1993.
- [13] C. Hedberg, L. Tydén, C. Wigren: *Generell metod för simulering av elektro-optiska telekrigdueller*, FOA report, FOA-R--99-01160-616--SE, Juni 1999.
- [14] C. Hedberg, L. Tydén, C. Wigren: *EwSim – Electronic Warfare Real Time Simulation in the Visual an Infrared Wavelength Range*, FOI report, FOI-R--0709--SE, December 2000.