

Detektering, målföljning och identifiering med robotsystem 90

Simon Lindgren



LUNDS
UNIVERSITET

Reglerteknik



Luftvärnsregementet

Examensarbete
ISRN LUTFD2/TFRT--5973--SE
ISSN 0280-5316

Institutionen för Reglerteknik
Lunds universitet
Box 118
221 00 LUND

© 2015 by Simon Lindgren. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2015

Förord

Detta examensarbete gjordes vid luftvärnsregementet med handledning av LTH. Syftet med projektet var att undersöka funktionerna automatisk detektering, målföljning och identifiering med en EO/IR (elektrooptisk/infraröd) sensor. Dessutom skulle siktet till robotsystem 90 undersökas i syfte att klargöra möjligheterna att använda siktet som en sensor.

Jag vill rikta ett särskilt tack till följande personer:

Dr. Rolf Johansson, handledare, Inst. Reglerteknik, LTH, Lunds Universitet

Lt Håkan Larsson, handledare, Lv6, Försvarmakten

Mj Per Gerdle, Lv6, Försvarmakten

Kn Fredrik Svanström, Lv6, Försvarmakten

Halmstad, juni 2015

Simon Lindgren

Sammanfattning

Projektets syfte var att klargöra om och i så fall hur det är möjligt att detektera, mälffölja och identifiera med en EO/IR (elektrooptisk/infraröd) sensor. Dessutom skulle siktet till robotsystem 90 testas i syfte att klargöra möjligheterna att använda siktet som en sensor. Siktet har ett synfält på endast $3 \times 4^\circ$ med TV-kameran och $4 \times 6^\circ$ med IRV-kameran. Därför har projektet fokuserat på en rörlig sensor i syfte att kunna avsökta en större yta.

Genom bakgrundsubtrahering är förändringar, d.v.s. rörelser möjliga att detektera. Bakgrunden måste röra sig med siktet och därför behöver mätvärdena på siktets riktning vara noggranna. Med ett Kalmanfilter brusreducerades mätvärdena och detektering möjliggjordes. Den maximala avsökningshastigheten med siktet blev $10.8^\circ/s$. Siktet bedöms inte kunna användas som en sensor p.g.a. den låga avsökningshastigheten.

Kalmanfilter kan även utnyttjas för att filtrera detektioner och under en kortare tid prediktera målets bana. När målet inte manövrerade kunde felet på hastighetsvektorn minskas med en skalningsmatris och därav kunde målet predikteras under en längre tid. Det ledde dock till fördröjningar när målet manövrerade vilket resulterade i ett större fel för målets position.

Bilderna i videon från siktet är i gråskala vilket innebär att färgen inte kunde användas för att identifiera flygplan. Med TV-kameran är flygplanen mörka och med IRV-kameran är de ljusa. Det som återstod därefter var att identifiera genom att analysera formen.

En befintlig metod för identifiering genom formanlys är HOG (*Histogram of Oriented Gradients*). En stor mängd träningsdata krävdes till stödvektormaskinen för att hantera skillnaderna på formen beroende på sensorns betraktningvinkel, flygplanets rotation samt de relativt små bilderna. Tillförlitligheten med metoden blev inte tillräckligt hög.

Två egna metoder testades därför i syfte att försöka identifiera målen genom formanlys. I den första metoden byggdes en 3D-modell av ett flygplan. Identifieringen skulle ske genom att rotera och projicera modellen till en tvådimensionell bild vilken jämfördes med bilden på målet. Att hitta bästa tredimensionella rotationen är ett problem, som endast går att lösa genom att testa samtliga lösningar. Därför tar det för lång tid att identifiera med metoden.

Den andra metoden syftar till att analysera formen genom att rotera bilden tvådimensionellt så nosen pekar längs x-koordinaten i ett tvådimensionellt koordinatsystem. Därefter analyserades vingarna genom att analysera värdena på y-axeln relaterat till x-axeln. Maxima, minima samt gradienterna analyserades i syfte att placera vingarna, uppskatta deras relativa storlek mot varandra och dess form.

Metod två kan utnyttjas för att öka tillförlitligheten på identifieringen av flygplan. För att ett autonomt system ska kunna bekämpa mål så krävs en mycket tillförlitlig identifiering. Därför krävs användning av optisk zoom för förstoring av bilderna. Dessutom bör fler parametrar som t.ex. position, hastighet och storlek analyseras under identifieringen.

Abstract

The objective was to clarify whether and how it is possible to detect, track and identify with an EO/IR (electro optical/infrared) sensor. Moreover, the sight of robotic system 90 was tested in order to clarify the possibilities of using it as a sensor. The sight has a field of view of $3 \times 4^\circ$ with the TV camera and $4 \times 6^\circ$ with the IRV camera. Therefore, the project focused on a moving sensor, in order to be able to scan a larger area.

By background subtraction it's possible to detect changes, i.e., movements. The background must move with the sight and therefore requires measurements of the sight direction to be accurate. Detection was made possible by noise reduction of the measured values using a Kalman filter. The maximum scan rate of the sight was $10.8^\circ/\text{s}$. The sight can thus not be used as a sensor because the scanning speed is judged to be too low.

Kalman filters may also be used to filter the detections and for a short time predict the target path. With a scaling matrix, error in the velocity vector was reduced when the target was not manoeuvring and the target movement could be predicted over time. However, it led to delays when the target manoeuvred, resulting in poor accuracy of the target position.

The video color from the sight is in grayscale, which means that the color couldn't be used to identify aircraft. The aircraft is dark with the TV camera and bright with the IRV camera. What was left then was to identify the type of aircraft by analysing the shape.

An existing method for identification by shape analysis is HOG (Histogram of Oriented Gradients). A large amount of training data was required for the support vector machine to handle the differences in shape depending on the sensor viewing angle, the aircraft's rotation and the relatively small pictures. However, the reliability of the method was still not high enough.

Two custom methods were therefore tested in order to try to identify targets by shape analysis. In the first method a 3D model of an airplane was built. The identification would be done by rotating and projecting the model to a two-dimensional image which was compared with the image of the target. Finding the best three-dimensional rotation is a problem which can be solved only by testing all solutions. Therefore, it takes too long to identify with this method.

The second method aims to analyse the shape by rotating the two-dimensional image so that the nose is pointing along the x-axis of a two-dimensional coordinate system. Wings were then analysed by analysing the values on the y-coordinates relative to the x-coordinates. Maxima, minima and the gradients were analysed in order to place the wings, estimate their relative size to each other and the shape of the aircraft.

Method two may be used to increase the reliability of the identification of the aircraft. An autonomous combat system requires the identification to be very reliable, thus requiring the use of an optical zoom to enlarge the images. More parameters such as position, speed and size should also be analysed when identifying the target.

Innehåll

1	Bakgrund	1
1.1	Huvudstudie luftvärn	1
1.2	Robotsystem 90	2
2	Problemformulering	3
2.1	Projektets mål, syfte och krav	3
2.2	Begränsningar	3
2.3	Problemfrågeställning	4
2.4	Kravspecifikation	4
3	Teori	5
3.1	En generisk radars signal- och databehandling	5
3.1.1	Blockschema	5
3.1.2	AESA	7
3.1.3	MTI - Moving target indication	7
3.1.4	KFA - Konstant falsk alarmering	8
3.1.5	Klotterkarta	9
3.1.6	Kalmanfilter	9
3.2	Bildanalys	11
3.2.1	Johnsons kriterium	11
3.2.2	Bakgrundssubtrahering	11
3.2.3	HOG - Histogram of Oriented Gradients	12
4	Utvecklingsbeskrivning	13
4.1	Utvecklingsverktyg	13

4.2	Utvecklingsprocessen	13
4.3	Verifiering	14
5	Metoder	15
5.1	Systemuppbyggnad	15
5.1.1	Kommunikationsprotokoll	15
5.1.2	Intern och extern server	16
5.1.3	Bildbehandling	17
5.1.4	Kontrollenhet sensor	17
5.1.5	Övervakningsenhet sensor	17
5.2	Simulatorn	18
5.3	Detektering	19
5.3.1	Bakgrundssubtrahering	19
5.3.2	Tröskelvärde för signaldetektering	21
5.3.3	Stilla detektering	21
5.3.4	Detektionsanalys	22
5.4	Målföljning	22
5.5	Identifieringen	23
5.5.1	3D formanalys	24
5.5.2	2D grafanalys	25
6	Experiment	26
6.1	Målföljning under manövrering	26
6.2	Identifiering	28
6.2.1	HOG på inspelad video från siktet	28
6.2.2	3D formanalys på positiv och negativ referensbild	29

6.2.3	2D grafanalys på testbilder	29
7	Resultat	30
7.1	Avsökningshastighet	30
7.2	Målföljning	30
7.3	Identifiering	35
7.3.1	HOG	35
7.3.2	3D formanalys	39
7.3.3	2D grafanalys	41
8	Diskussion	42
8.1	Jämförelse med befintliga system	42
8.2	Avsökningshastighetens betydelse för en sensor	42
8.3	Positions- eller hastighetsprecision vid målföljning	43
8.4	Det optimala identifieringssystemet för flygplan	44
8.5	Samhälleliga och etiska aspekter	45
9	Slutsats	46
	Referenser	47
	Figur- och tabellförteckning	51

1 Bakgrund

1.1 Huvudstudie luftvärn

Huvudstudie luftvärn är framtagen av Försvarsmakten i syfte att långsiktigt inrikta luftvärnsfunktionen efter 2020 [1]. Det konstateras i studien att endast radar används som sensor inom Sveriges luftvärn. Radar är en aktiv sensor, vilket betyder att den sänder ut elektromagnetisk strålning och lyssnar efter ekot. Att själv sända ut signalen ökar kontrollen på signalen, t.ex. energi, riktning och karaktär. Radarn får en god avståndsupplösning genom att mäta tiden som det tar för signalen att nå målet och komma tillbaka. Motståndaren kan dock upptäcka radarns signal vilket möjliggör både pejling av radars position samt återgivning av signalen i syfte att störa radarn.

Passiva sensorer kan istället utnyttjas för att undgå upptäckt. De lyssnar efter målets egna signaler. Om signalerna är kända möjliggörs även igenkänning och identifiering av målet. Vid dagsljus reflekterar motståndaren synligt ljus. Infraröd strålning genereras alltid p.g.a. dels friktionen mot luften samt den egengenererade värmen från t.ex. motorerna. Mikrovågor genereras av bl.a. kommunikationsutrustningen och sensorerna.

Synligt och infrarött ljus har kortare våglängder än mikrovågor. Nordström konstaterar att väderkänsligheten generellt ökar med minskande våglängd [2]. Han skriver att "Radarn påverkas mer av atmosfärens vertikala förändringar i brytningsindex än optiska system medan optiska system påverkas mer av moln och dimma". Fördelen med mikrovågor blir därav att de generellt fungerar bättre oberoende vädret. Nackdelen med passiva system som lyssnar på mikrovågor är att motståndaren enkelt kan undgå upptäckt genom att stänga av sina mikrovågsgenererande system. Det är betydligt svårare för motståndaren att sluta generera en avvikande signal i spektrumet för synligt och infrarött ljus.

I huvudstudie luftvärn konstateras därför att det vore bra att komplettera radar med passiva sensorer i syfte att utnyttja system med olika fördelar. Ett exempel på en passiv sensor är EO/IR (elektrooptisk/infraröd) sensor. Försvarsmakten ville därför undersöka hur detektering, målföljning och identifiering fungerar med denna sensortyp.

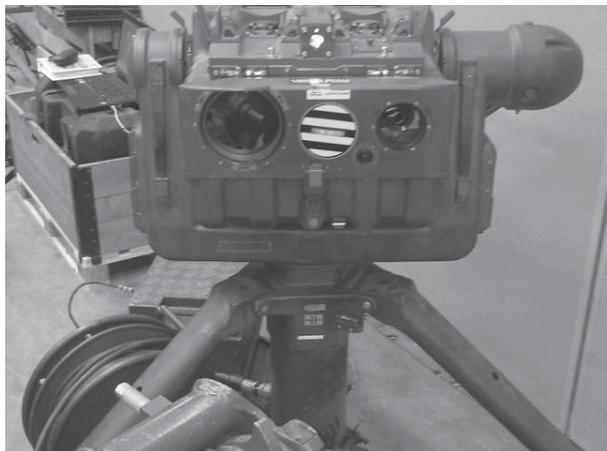
1.2 Robotsystem 90

RBS 90 (robotssystem 90) består av ett EO/IR sikte, se figur 1, på vilket två robottuber kan monteras. Siktet styrs ifrån den tillhörande radarn Ps91. I Ps91 sitter två operatörer där den ena är ansvarig för siktet och den andra för radarn. Att Ps91 är en 3D-radar innebär att den mäter målets avstånd, bäring och elevation, d.v.s. målets position i samtliga dimensioner [3]. Se tabell 1 för mer information angående RBS 90.

Tillverkare	Bofors och Ericsson Microwave Systems
Utvecklingsstart	1983
I tjänst	1993-2004
Radar typ	3D pulsdopplerradar
Radar frekvens	X-band 8.8-9.3 GHz
Radar räckvidd	20km
Robot navigation	Ledstrålestyrt
Robot räckvidd	5-7km
Antal robotar	2st
Synfält TV-kamera	3x4°
Synfält IRV-kamera	4x6°
Skärpedjup	från 1000m till ∞

Tabell 1: Information RBS 90 [3]

I LvBat2017 (Luftvärnsbataljonen 2017) förbandssätts Ps91 igen. Huvudargumentet till det är införandet av EldE 98, med roboten IRIS-T som kräver 3D-invisning. Därav blir Ps91 ett stöd till IRIS-T med syfte att förbättra den lokala luftlägesbilden. Med anledning av det har Forsvarsmakten tillgång till siktet som skulle kunna användas som EO/IR sensorer.



Figur 1: Siktet till RBS 90

2 Problemformulering

2.1 Projektets mål, syfte och krav

Projektets slutmål var att ta reda på om och hur detektering, målföljning och identifiering fungerar med RBS 90. För att dela upp processen i examensarbetet sattes 5 delmål. Sista delmålet skulle ingå om det bedömdes finnas tid över. Då det inte gjorde det, ströks slutligen delmålet. För fullständighets skull redovisas det dock nedan.

Delmål

1. Styrning av siktet.
2. Upptäcka ett potentiellt mål.
3. Följa ett potentiellt mål.
4. Identifiera ett potentiellt mål.
5. Skicka mäldata till en server vilken mottager mäldata från andra sikten och krysspejlar målets position.

Det fanns två syften med examensarbetet, dels att undersöka tillgängliga metoder i syfte att hitta lämpligaste metoderna för den specifika uppgiften, dels att utveckla en prototyp i syfte att kunna förevisa teknikens möjligheter och begränsningar.

Försvarsmakten ställde som krav att RBS 90 siktet skulle användas och att befintlig utrustning inte fick förändras. Dessutom skulle ingen ny hårdvara utvecklas utan projektet skulle bara omfatta nyutvecklad programvara för att lösa uppgiften.

2.2 Begränsningar

Den största begränsningsfaktorn var de tillgängliga resurserna. Försvarsmaktens krav berodde framförallt på en begränsad budget för projektet. Förändringar i befintlig utrustning och utveckling av hårdvara hade kostat för mycket. Dessutom sattes tiden för att slutföra projektet till 800h vilket motsvarar 40 veckors halvtidsarbete. Min bedömning är att med mer resurser hade det varit lämpligt att utveckla delar av bildbehandlingen som hårdvara i syfte att förbättra prestandan.

Siktet blev tillgängligt den 13 januari 2015. All utveckling under projektets första 4 månader fick därför testas genom simuleringar. Siktet är dessutom en del

av ett vapensystem vilket innebar att siktet behövde förvaras enligt särskilda bestämmelser. Det begränsade tillgängligheten vilket bl.a. innebar att utvecklingen och testningen inte kunde bedrivas på samma plats.

Utvecklingen av RBS 90 skedde under 80-talet, vilket innebär att systemet idag är runt 30 år gammalt. Jag bedömer att det idag finns bättre kameror med bättre kontrast, högre pixeltäthet och större synfält. Synfältet på $3 \times 4^\circ$ respektive $4 \times 6^\circ$ på TV- och IRV-kameran innebar att detekteringen bedömdes kunna liknas med att leta efter flugor genom ett sugrör. Dessutom var dokumentation för siktet bristfällig vilket innebar att mycket av informationen om hur siktet fungerade fick testas fram.

2.3 Problemfrågeställning

Projektets mål och begränsningar ledde fram till följande problemfrågeställning.

1. Hur stort område kan siktet avsöka och hur snabbt?
2. Vilken metod kan användas för att följa målen?
3. Vilken metod kan användas för att se skillnad på olika föremål?

2.4 Kravspecifikation

Följande krav på prestandan specificerades för de olika funktionerna.

1. Ytan $90 \times 60^\circ$ ska kunna avsökas på 10 sekunder.
2. Vid målföljning ska standardavvikelsen på positionsfelet vara $< 0.1^\circ$ och standardavvikelse på hastighetsfelet $< 0.4^\circ/s$. Dessutom ska målföljningen kunna följa mål som manövrerar med en acceleration på minst 9G, d.v.s. $9 \cdot 9.82 = 88.38 \text{ m/s}^2$, på avståndet 5000m. Det motsvaras av en vinkelaccelerationen på $1.01^\circ/s^2$.
3. Metoden för identifiering ska se skillnad på följande:
 - (a) Flygplan, robotar, helikoptrar, fåglar och moln.
 - (b) Jet- och propellermotorer och antalet motorer.
 - (c) Attack- och transportflygplan.

3 Teori

3.1 En generisk radars signal- och databehandling

3.1.1 Blockschema

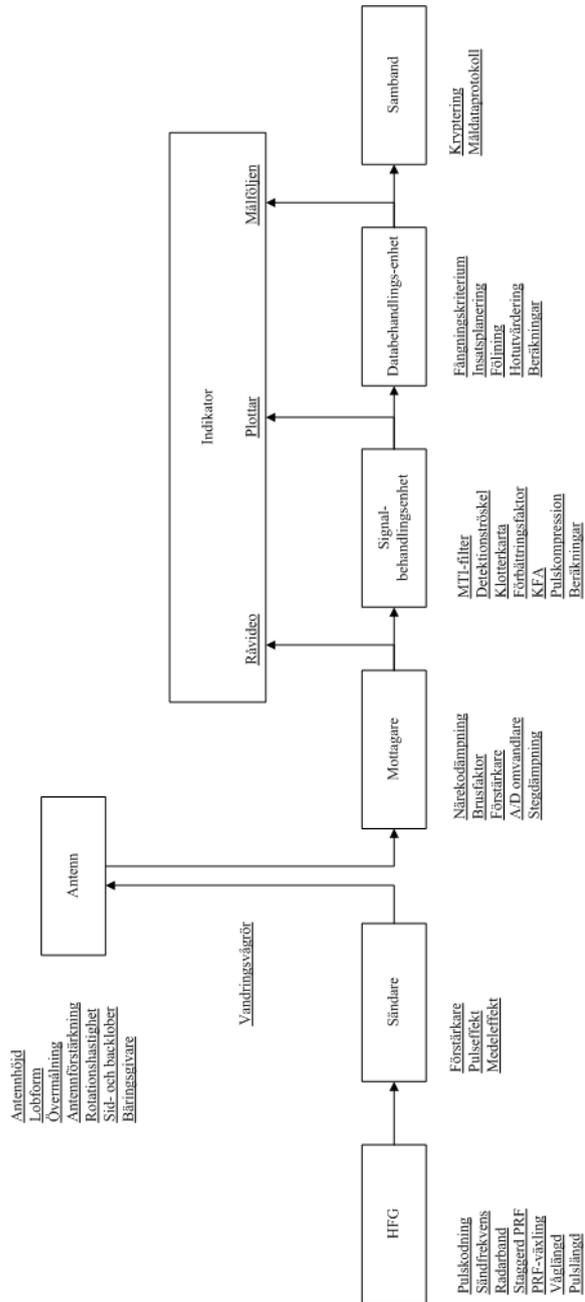
Radarn kan schematiskt indelas i ett antal block enligt figur 2 [4]. I HFG (högfrekvensgeneratoren) genereras och karakteriseras den elektromagnetiska signalen. Sändaren förstärker signalen från HFG innan vägen leds genom vandringsvägrören till antennen. Av antennens utformning och riktning bestäms signalens lobe och riktning. Efter att hela signalen skickats genom antennen växlar SM-omkopplaren (sändnings-/mottagningsomkopplaren) till mottagningsläge. Det gör att radarn börjar lyssna efter ekot från signalen.

I mottagaren justeras signalen för att göra ekon från olika avstånd jämförbara, d.v.s. ekon som tidigt kommer tillbaka dämpas mer än ekon som färdats/dämpats längre i atomsfären. Signalen som genereras av mottagaren kallas för råvideon.

Signalen kommer därefter till signalbehandlingsenheten vars uppgift är att detektera målen och generera plottar på målens positioner. MTI-filtret (*Moving Target Indicator*) tar bort signaler som inte rör sig. KFA (Konstant Falsk Alarmering) bestämmer signal/brus-förhållandet som krävs för att detektera målet. Klotterkartan sparar signalerna i en karta för att dämpa signaler som ständigt uppkommer på samma ställe.

Plottarna skickas till databehandlingsenheten som skapar och följer målen utefter plottarnas position. Målföljerna utvärderas utifrån vilka som utgör störst hot och motåtgärder föreslås av insatsplaneraren. Avslutningsvis krypteras måldatan innan den skickas till övriga enheter som tar del av luftläget.

Blockschema radar



Figur 2: Blockschema Radar [4]

3.1.2 AESA

I flera länder pågår idag utveckling av AESA (aktivt elektroniskt styrd antenn) vilket ger radarn fler möjligheter än vad den haft tidigare [4]. AESA innebär att radarn aktivt kan styra loberna d.v.s. vart den riktar energin från antennen, vilket möjliggör flera olika sändar- och mottagarlobar [5]. Utvecklingen av radarn har därav gått från att radarn endast har lyssnat på en punkt vid varje tidsenhet till flera punkter. Punkterna måste dock vara på samma avstånd från radarn av två anledningar. Första anledningen är att energin från antennen färdas med samma hastighet i samtliga lobar. Den andra anledningen är att radarn inte kan sända i en del av loberna samtidigt som den lyssnar i andra, eftersom det skulle innebära att energin från sändaren läcker in i mottagaren. Det hade i sin tur riskerat att förstöra mottagaren som lyssnar efter mycket mindre energier än det från sändaren.

Generellt vet en EO/IR sensor inget om avståndet till målet, utan det krävs minst två sensorer för att krysspejla ett mål och på det sättet beräkna avståndet. Om målet är identifierat och dess storlek känt kan dock avståndet uppskattas genom att beräkna vinkelstorleken i bilden. EO/IR sensorn riktas i en riktning och beroende på storleken på synfältet detekterar sensorn flera punkter runt den riktningen samtidigt. Det innebär att förutsättningarna för AESA radarns och EO/IR sensorns analys av punkterna ungefär är densamma. Därför har grunden till metoderna för signal- och databehandlingen utgått ifrån radarns metoder enligt figur 2.

3.1.3 MTI - Moving target indication

MTI-filtret i radarn fungerar som en rörelsedetektor [6]. Radarn kan ha två kanaler, en där signalen går genom MTI-filtret och en där den inte gör det. För att kunna detektera en luftballong som långsamt rör sig framåt kan dock inte MTI-filtret användas. Syftet med MTI-filtret är främst att ta bort markklotter, men det kan även användas för att t.ex. ta bort de remsor som flygplan ibland faller då det blir belysta.

Det finns två metoder som filtreringen kan bygga på. Den första metoden bygger på att dopplerfrekvensen på ekot mäts. Om frekvensen är större än ett valt tröskelvärde, så bedöms objektet röra sig i förhållande till radarn och MTI-filtret släpper genom signalen.

Det andra metoden bygger på mätning av fasen på ekot. Genom att jämföra fasen på två olika ekon kan radarn bedöma om objektet rör sig i förhållande till radarn.

Det finns både för- och nackdelar med båda metoderna. Den främsta fördelen med att mäta dopplerfrekvensen är att ett mätvärde fås efter första ekot. Genom att mäta fasskillnaden krävs minst två ekon med samma utsända frekvens för att kunna detektera ifall objektet är rörligt. Det är dock enklare att jämföra

fasskillnaden eftersom den är relativt stor i förhållande till skillnaden mellan utsänd frekvens och dopplerfrekvensen, härlett ur ekvation 1 [7].

$$f' = f \frac{c + v}{\sqrt{c^2 + v^2}} \quad (1)$$

Den upplevda frekvensen hos mottagaren f' beror på utsänd frekvens f , ljusets hastighet c och målets hastighet v .

$$f_d = 2(f' - f) = 2\left(f \frac{c + v}{\sqrt{c^2 + v^2}} - f\right) \approx 2\left(f \frac{c + v}{c} - f\right) = \frac{2vf}{c} = \frac{2v}{\lambda} \quad (2)$$

Dopplerfrekvensen f_d kan approximeras enligt ekvation 2 om $v \ll c$ och $f_d \ll f$. Dopplerfrekvensen beror då på målets hastighet v och våglängden λ . Sändaren mottar reflektionen av f' , vilket innebär att dopplereffekten uppstår två gånger.

Om utsänd frekvens på radarn är 5.5GHz och objektets hastighet relativt sändaren är 300m/s blir $f_d \approx 11\text{kHz}$ enligt ekvation 3.

$$f_d = \frac{2v}{\lambda} = \frac{2fv}{c} \approx \frac{2 \cdot 5.5 \cdot 10^9 \cdot 300}{3 \cdot 10^8} = 11\text{kHz} \quad (3)$$

Fasskillnaden ger upphov till en amplitudskillnad på den mottagna signalen. Amplitudskillnaden beror på dopplerfrekvensen f_d och tiden mellan pulserna T_p enligt ekvation 4 [6].

$$H(f) = 2\sin(\pi f_d T_p) \approx 2\sin\left(\frac{\pi f_d r_{max}}{c}\right) \quad (4)$$

T_p approximeras som tiden det tar för pulsen att nå radarns borte räckvidd och tillbaka. Med räckvidden r_{max} 50km kan $|H(f)|$ approximeras som 1.73 enligt ekvation 4.

I ekvation 3 är kvoten mellan dopplerfrekvensen och utsänd frekvens 2ppm. Amplitudskillnaden som fasskillnaden ger upphov till är 173%, vilket i jämförelse är det en relativt stor skillnad.

3.1.4 KFA - Konstant falsk alarmering

KFA är det signal/brus-förhållandet som krävs för att en signal ska detekteras [6]. Om KFA-nivån höjs krävs det en högre signal för en detektion. Det innebär att med en högre nivå minskar risken att bruset ger detektioner i radarn. Samtidigt krävs en högre signal från målet innan radarn får en detektion vilket innebär senare detektioner. Med en låg nivå ökar radarns känslighet och

små mål upptäcks tidigare. Det kommer dock innebära att bruset oftare ger s.k. falska mål.

Det finns flera olika metoder på hur KFA-nivån kan implementeras. Det enklaste alternativet är att endast sätta ett fast värde. Den metoden bedöms dock som olämplig ifall bruset varierar eftersom det då riskerar bildas falska mål. Därför bör KFA-nivån bero på brusnivån.

3.1.5 Klotterkarta

Klotterkartans syfte är att ta bort plottar som ständigt uppkommer på samma ställe [6]. Därför sparas signalerna ner i en 3D-karta. Rutor där det ständigt råder höga signalnivåer stängs för att förhindra generering av falskmål. Det innebär att om det står t.ex. ett vindkraftverk i en ruta, så kan den rutan stängas ifall signalen reflekteras tillbaka till radarn.

Om man byter inställningar i radarn kommer signalerna i mottagaren se anorlunda ut. Varje gång inställningarna byts kommer därför klotterkartan behöva uppdateras. Beroende på rotationshastigheten kan det ta upp till flera minuter för klotterkartan att optimeras och därför kan dess funktion nedgå ifall operatören i radarn ständigt behöver byta inställningar som svar på motståndarens åtgärder.

3.1.6 Kalmanfilter

Kalmanfiltret använder en rekursiv algoritm som utifrån ett antal brusiga mätvärden uppskattar tillståndet hos ett system [8]. För att följa och prediktera ett objekts bana kan Kalmanfilter användas [9]. Det kan även användas i andra applikationer som t.ex. positionsbestämning med GPS [10]. Tabell 2 beskriver algoritmen.

Algoritmen förutsätter att bruset på både processen och mätvärdena är normalfördelade med medelvärde 0. Process- och mätbrusets kovarians måste därför mätas innan användning för optimal funktion. Den uppskattade kovariansen P beror på fördelningen mellan prediktioner och korrigeringar. Filtret tar inte hänsyn till förändringar av process- och mätbrusets kovarians utan förutsätter att de är konstanta över tiden. Om brusets varians inte är konstant bör det mätas så att modellerna för process- och mätbrusets varians, Q och R , kan uppdateras.

Syftet med ett Kalmanfilter är därav inte att prediktera ett objekts bana utan att minska bruset från mätvärdena, vilket filtret gör m.h.a. prediktering. Därför fungerar predikteringen bättre då frekvensen på korrigeringarna är hög.

Om filtret används för att prediktera en mänsklig handling kan taktiska variabler förbättra predikteringen då frekvensen på korrigeringarna sänks. Exempelvis om ett mål flyger ner bakom en kulle, då predikterar Kalmanfiltret att målet

fortsätter ner tills det träffar marken. Bedömt kommer flygplanet att försöka lyfta innan det slår i marken och därför borde ett filter med taktiska variabler prediktera att flygplanet dyker upp ovanför kullen.

Prediktion

Predikterar systemets tillståndet	$\hat{x}_{k k-1} = A_k \hat{x}_{k-1 k-1} + B_k u_k$
Predikterar uppskattad kovarians	$P_{k k-1} = A_k P_{k-1 k-1} A_k^T + Q_k$

Korrektion

Innovation eller uppmätt rest	$\tilde{y}_k = z_k - H_k \hat{x}_{k k-1}$
Innovationens (eller felets) kovarians	$S_k = H_k P_{k k-1} H_k^T + R_k$
Optimal Kalmanförstärkning	$K_k = P_{k k-1} H_k^T S_k^{-1}$
Korrigerar systemets tillstånd	$\hat{x}_{k k} = \hat{x}_{k k-1} + K_k \tilde{y}_k$
Korrigerar uppskattad kovarians	$P_{k k} = (I - K_k H_k) P_{k k-1}$

- k - tidsindex i systemet.
- \hat{x} - systemets skattade tillstånd.
- A - systemets övergångsmodell.
- u - styrsignal.
- B - modell för styrsignalen.
- P - uppskattad kovarians.
- Q - kovariansen av processens brus, systemets osäkerhet.
- z - mätvärde.
- H - observationsmodell, beskriver skillnaden mellan systemets tillstånd och mätvärdet.
- \tilde{y} - uppmätt förändring av systemets tillstånd.
- R - kovariansen av mätbruset.
- S - uppskattad kovarians av systemets förändring.
- K - optimal Kalmanförstärkning.

Tabell 2: Kalmanfiltrets algoritim

3.2 Bildanalys

3.2.1 Johnsons kriterium

Johnsons kriterium beskriver vad som är möjligt att detektera, känna igen och identifiera [11]. Kriteriet påvisades genom experiment, där ett antal personer sattes på en stol långt ifrån ett militärt objekt och kördes fram till de kunde detektera objektet. När de kunde detektera att det fanns ett objekt ersattes objektet med en rektangel med ljus- och mörkgrå linjepar med samma kontrast som mellan bakgrunden och objektet. Linjerna orienterades så att de löpte längs långsidorna på rektangeln. Antalet linjepar som personerna kunde urskilja blev mätetalet. Därefter flyttades personerna fram tills de kunde känna igen och identifiera objektet för att få dess mätetal. När experimentet var klart valdes mätetalet då 50% av personerna hade klarat uppgiften. Resultaten visas i tabell 3.

<i>Objekt</i>	<i>Detektion</i>	<i>Igenkänning</i>	<i>Identifiering</i>
Lastbil	0.90	4.5	8.0
M-48 Stridsvagn	0.75	3.5	7.0
Stalin Stridsvagn	0.75	3.3	6.0
Jeep	1.2	4.5	5.5
105-mm haubits	1.0	4.8	6.0
Soldat (stående)	1.5	3.8	8.0
Medel	1 ± 0.25	4.0 ± 0.8	6.4 ± 1.5

Tabell 3: Resultat från Johnsons försök [11]

Det innebär att Johnsons kriterium ger ett mätvärde på när det är 50% sannolikhet att en människa kan identifiera ett objekt. Talet är oberoende av avståndet och kontrasten mellan objektet och bakgrunden. Om ett program kan konstrueras, vilket är lika bra på att identifiera objekt som en människa, så kan objektets teoretiskt minsta storlek beräknas. Det görs genom att bestämma då programmet kan urskilja linjeparen, ju lägre skärpa desto fler pixlar krävs. Sedan kan antalet pixlar multipliceras med mätetalet i Johnsons kriterium för att få fram minsta storlek på objektet. Möjligheten att identifiera beror därför teoretiskt endast på skärpan, kontraståtergivningen och pixeltätheten i förhållande till synfältets storlek.

3.2.2 Bakgrundsubtrahering

För att upptäcka rörelser så krävs en lång observationstid. Ju långsammare en rörelse är desto längre måste observationstiden vara. En lång exponeringstid kan användas för att upptäcka rörelser i en bild. Då blir objekten som rör sig i bilden suddiga. För att upptäcka rörelser i en film så kan istället bilder från olika tidpunkter jämföras mot varandra.

Bakgrundsubtrahering innebär att varje ny bild jämförs mot en referensbild av

bakgrunden. Området som jämförs kan vara en eller flera pixlar och om skillnaden är större än ett satt tröskelvärde märks området. Det innebär att förändringar i bilden, d.v.s. rörelser detekteras. Algoritmen har dock vissa svårigheter i hanteringen av följande förändringar [12].

1. Snabba ljusförändringar, t.ex. när solen går i moln eller en lampa som tänds.
2. Regelbundna förändringar, t.ex. vinden som tar tag i löven i träden eller flaggan som svajar i flaggstången.
3. Långvariga scenförändringar, t.ex. ett fordon som parkerats.

3.2.3 HOG - Histogram of Oriented Gradients

Identifiering sker genom att analysera olika egenskaper hos ett visst objekt. Det kan vara hastighet, färg, storlek etc. HOG innebär att formen på objektet analyseras. Enligt Dalal och Triggs karakteriseras ofta olika objekt ganska väl av kanternas riktningar [13]. HOG kan beskrivas i sex steg enligt figur 3.



Figur 3: Implementering av HOG [13]

I många implementationer är första steget att bildens färger och gammavärde (ljusstyrkans gradient) normaliseras. Dalal och Triggs påpekar dock att det under vissa förhållanden kan uteläutas, eftersom blocken som byggs upp under processens fjärde steget normaliseras.

Kanternas gradienter beräknas i det andra steget genom att filtrera bilden i X-respektive Y-led med $[-1, 0, 1]$ och $[-1, 0, 1]^T$. Andra möjliga filter är t.ex. 3×3 Sobel eller Roberts kors. Filtret bör anpassas för objektets form.

I tredje steget grupperas pixlarna till celler vilket ökar vinkelupplösningen på gradienterna. Om cellerna innehåller fler pixlar kommer vinkelupplösningen hos cellen öka. Det innebär samtidigt att snabba förändringar inne i cellen kan filtreras bort. Därav måste storleken på cellerna relateras till målets storlek och form. Efter tredje steget är gradienterna i bilden kända.

I det fjärde steget filtreras gradienterna som uppkommer av kontrast- eller ljusförändringar bort. Det görs genom att cellerna grupperas ihop till block. Blocken byggs upp så att de överlappar varandra och därefter normaliseras varje block.

Därefter är bilden behandlad och histogrammet kan analyseras. För att analysera histogrammet behövs ett tränat igenkänningssystem. Om en SVM [14] (stödvectormaskin) tränas upp med ett visst objekt kan den analysera histogrammet för att avgöra om och var liknande objekt finns.

4 Utvecklingsbeskrivning

4.1 Utvecklingsverktyg

Under projektet användes framförallt två verktyg. För utveckling i *C#* användes *Visual Studio Express 2013 for Desktop* [15]. Anledningen till att välja *C#* var att projektets deltagare tidigare haft en god erfarenhet av språket med dess verktyg. Förhoppningen var att det skulle minska tiden för uppstarten av projektet. Dessutom var tillgången till dokumentation och exempel god vilket underlättade när utvecklingen väl kom igång.

Matlab [16] valdes för att hantera och analysera videon. *Computer Vision System Toolbox* [17] i *Matlab* kan användas för både målföljning och identifiering. Anledningen till att *Matlab* användes berodde också på att projektets deltagare tidigare haft god erfarenhet av verktyget. Andra tänkbara alternativ var *EMGU CV* [18]. Det alternativet analyserades dock inte eftersom *Matlab* ansågs vara tillräckligt kapabelt för att klara av uppgiften.

Andra program som också användes var *Tera Term* [19] och *VLC media player* [20]. De användes för testning av seriell datakommunikation respektive inspelning av videoströmmen från siktet.

4.2 Utvecklingsprocessen

Inledningsvis lades en grund för systemets uppbyggnad. Syftet var att få en helhetsbild av hela systemets funktionalitet och delarnas interaktionsflöde mellan varandra. Efterhand fick systemuppbyggnaden modifieras i processen för att uppfylla de fastställda kraven. Grunden kunde dock bibehållas.

Därefter inhämtades information om vilka metoder som fanns för att klara av detektering, målföljning och identifiering. För att analysera metoderna implementerades en simulator i *Matlab*.

Siktet blev tillgängligt 4 månader efter projektets start. Utvecklingen kunde därefter starta och styrprogrammet till siktet implementerades först. Efter vissa problem med att styra siktet från *Matlab* fick styrprogrammet skrivas i *C#*. Eftersom bildhanteringen och styrprogrammet till siktet inte kunde integreras fick ett protokoll för kommunikationen utvecklas.

När kommunikationsprotokollet var implementerat fortsatte utvecklingen av bildhanteringen. Delarna som testats i simulatoren anpassades till siktet och användarfunktionerna lades till. Slutligen testades och optimerades bildhanteringen.

4.3 Verifiering

Då dokumentation till siktet var bristfällig blev behovet av testning under utvecklingen stort. Ingen funktion bedömdes vara verifierad innan den testats mot siktet. Radarn och siktet är skyddsvärd materiel vilket innebär att de behöver förvaras enligt särskilda bestämmelser. Då det inte var praktiskt möjligt att flytta radarn och siktet varje dag kunde testningen och utvecklingen inte bedrivas på samma plats.

Det var därför extra viktigt att dokumentera planen för utvecklingen och testningen då det ibland tog flera veckor från att en funktion implementerats till att den testats. Dokumentationen bidrog också till en bättre uppfattning för hur projektet fortskred. När materielen väl var tillgänglig kunde testningen bedrivas på en kontrollerat vis, vilket ledde till att eventuella åtgärder kunde genomföras, då materielen inte längre var tillgänglig.

5 Metoder

5.1 Systemuppbyggnad

Inledningsvis fastställdes systemuppbyggnaden i projektet. Syftet var att få en stomme varifrån systemets olika delarna kunde byggas. För att skicka och sammanställa informationen från flera olika sikten krävs en server dit siktena kan ansluta sig som klienter. Förutom att ta emot och sammanställa informationen var även tanken att servern skulle kopplas till en presentationsenhet som kunde visa den sammanställda informationen från klienterna. Det bestämdes därför att två program behövde implementeras. Det första programmet fick ansvaret för att styra och bearbeta informationen från siktet, medan det andra fick ansvaret för att sammanställa och presentera informationen från olika sikten till användaren.

Initialt var tanken att utveckla både servern och klienten i *C#*. Bildhanteringen skulle ske i *Matlab* och funktionerna skulle anropas ifrån *C#*. Efter försök med att hämta videon i *C#* och skicka över bilderna till *Matlab* insågs att det tog för lång tid att konvertera bilderna mellan *C#* och *Matlab*. Därför fick de försöken avbrytas och bilden hämtas direkt ifrån *Matlab*.

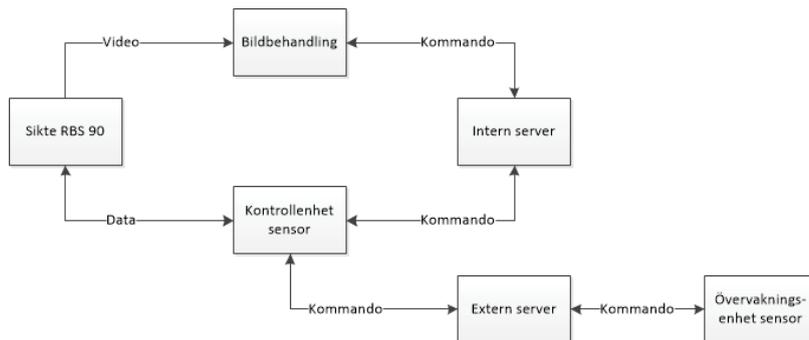
Tanken var därefter att utveckla hela klienten i *Matlab*. Anledningen var att slippa skicka data mellan *C#* och *Matlab*. Det ledde också till att felsökningen underlättades när båda programmen kunde felsökas i realtid. Att hela klienten skrevs i *Matlab* innebar dessutom att siktet behövde kontrolleras från *Matlab*. Vid testerna mot siktet insågs att klienten i *Matlab* inte kunde kontrollera siktet. Efter felsökning och försök att styra siktet från *C#* löstes problemet. Enligt min bedömning berodde felet på ett tidsfel på bekräftelserna. Vissa av de meddelanden som siktet skickar måste bekräftas [21]. Slutsatsen är dock osäker då ingen dokumentation angående problemet fanns tillgänglig.

Istället för att använda *Real-Time Workshop* [22] i *Matlab* delades klienten upp i syfte att spara tid. I detta läge var styrningen redan implementerad och verifierad i *C#*. *Matlab* ansvarade för att hämta och analysera videon och *C#* för att styra siktet. För att skicka data mellan programmen utvecklades ytterligare en server som båda programmen anslöt till och utbytte information via, se figur 4.

5.1.1 Kommunikationsprotokoll

IP valdes som protokoll för nätverksskiktet eftersom det stöds av både *C#* och *Matlab*. För transportskiktet stod valet mellan UDP (*User Datagram Protocol*) och TCP (*Transmission Control Protocol*). I valet mellan protokollen prioriterades trafiksäkerheten och därför valdes TCP [23].

Mellan den externa servern och dess klienter hade förmodligen UDP varit lämpligt. I strid är det viktigt att undgå upptäckt från motståndaren. Med TCP



Figur 4: Blockschema Systemuppbyggnad

skickas det bekräftelser på varje meddelande vilket innebär att båda enheter röjer sin position då information sänds från en enhet till en annan via radio.

Fast UDP ansågs mest lämpligt valdes TCP även för den externa servern. Projektet leder fram till en prototyp och skall inte anses som en färdig produkt. Valet av TCP grundades i att spara tid genom att inte behöva implementera två olika typer av servrar och klienter. Fokuset för projektet låg på bildhanteringen och inte på kommunikationsprotokollet.

Grunden till klienterna och servrarna är hämtad från Allexy [24]. I Allexys applikation är varje meddelande som skickas ett kommando. För att köra flera klienter från samma IP behövde kommandona modifieras. Servern behöver veta till vilken port meddelandet skall sändas vidare och därför behövde mottagarens portnummer läggas till i kommandot. Informationen i kommandona beskrivs i tabell 4.

- Sändarens IP
- Sändarens namn
- Mottagarens IP
- Mottagarens portnummer
- Kommandotyp
- Data

Tabell 4: Kommandoinformation

5.1.2 Intern och extern server

Den interna och externa servern är två instanser av samma implementation vilket innebär att de fungerar på samma sätt. Servrarnas uppgift är främst att

förmedla trafiken mellan dess klienter. Att inte ha en direktförbindelse mellan klienterna gör programmet mer agilt vilket innebär att det i framtiden blir lättare om ytterligare funktioner skall läggas till. Nya klienter som ansluts mot servern får tillgång till all information som sprids. Möjlighet för fjärrstyrning finns eftersom de befintliga klienterna agerar på alla kommandon oavsett avsändare.

När en klient ansluts mot servern måste klienten skicka ett unikt namn. Namnet talar om för servern vilken uppgift klienten har. När klienten ansluts mot servern ska klienten fråga om det finns en annan klient med samma namn (uppgift). Om så är fallet ska klienten stänga ner förbindelsen. I annat fall svarar servern att klienten anslutits korrekt. Slutligen har klienten möjlighet att fråga servern vilka klienter som redan är anslutna.

5.1.3 Bildbehandling

Bildbehandlingen är implementerad i *Matlab*. Klienten ansvarar för hela bildbehandlingen, d.v.s. detekteringen, målföljningen och identifieringen. Siktet styrs via kontrollenhet sensor, vilket innebär att klienten inte själv har möjlighet att skicka data direkt till siktet.

Simulatorn är inbyggd i bildbehandlingen. Syftet med det är att endast använda en kodbas för bildbehandlingen och att säkerställa att samma kod används för simuleringarna som de verkliga testerna.

5.1.4 Kontrollenhet sensor

Kontrollenhet sensor ansvarar för att upprätthålla kommunikationen med siktet samt dess styrning. Styrningen sker med kommandona beskrivna i RMU RBOK 3 (Robotmarkutrustning reparationsbok 3) [21].

Siktet kan sätta sig själv i personsäkert läge [21]. Det innebär att siktet fysiskt måste startas om innan kommunikationen kan återupprättas. Det fastslogs under testerna att det händer då bekräftelserna på meddelandena dröjer mer än ett par millisekunder. Tidsgränsen för bekräftelserna anges inte i dokumentation och kunde inte heller fastställas genom testerna. När kontrollenhet sensor verifierades styrdes siktet i 3 timmar utan att det sattes i personsäkert läge.

5.1.5 Övervakningsenhet sensor

Övervakningsenhet sensor övervakar samtliga kontrollenheter (se paragraf 5.1.4) som ansluts till extern server. Syftet med programmet är att sammanställa och presentera luftläget. Endast delar av programmet implementerades och verifierades under projektet då ingen prioritet lades på krysspejlingen.

5.2 Simulatoren

Inledningsvis studerades tillgängliga metoder för att uppfylla kraven på bildbehandlingen. Först gjordes en teoretisk analys av metoderna i syfte att sälla ut de metoder som ansågs mest lämpliga för att uppfylla kraven. Därefter byggdes simulatoren i syfte att göra en fortsatt analys av de utvalda metoderna.

Simulatoren användes för att verifiera detekteringen och målföljning. Ett flygplans projektion i 2D har en betydande variation beroende på betraktarens position och flygplanets läge. Därför användes inte simulatoren för att verifiera de olika teknikerna för identifiering. För att ytterligare testa identifieringen bör simuleringar av 3D-modeller i olika flygbanor implementeras i simulatoren. Det bedömdes dock kräva för mycket tid för att ingå i projektet.

När simulatoren initieras läses en bakgrundsbild in. Bilder från siktet användes som bakgrund i syfte att göra simulatorns miljö mer realistisk. Det är inte möjligt att använda både TV- och IRV-kamera i simulatoren och på så sätt förändra dels målen och bakgrunden. Istället är det möjligt att generera mål med olika färger och intensitet. Målen flyger i olika banor som är konstruerade med olika variabler a , k och l enligt ekvation 5 och 6.

$$x = \min(\sin(a_x t + k_x), l_x) \quad (5)$$

$$y = \min(\cos(a_y t + k_y), l_y) \quad (6)$$

Simulatoren märker när målen flyger bakom mask, d.v.s. ett skymmande objekt som befinner sig i siktlinjen mellan målet och siktet, genom att analysera bakgrunden. När målen befinner sig i mask slutar de att genereras på bilden. Därav kunde predikteringen studeras då målföljen tappas.

För att efterlikna det riktiga siktet är det viktigt att simulera brusets i mätvärdena på bäringarna och i bilderna. Därför har brusets analyserats och lagts in i simulatoren. För att testa brusets påverkan är det även möjligt att justera brusnivåerna.

En komplett analys av siktets rörelsemönster utifrån styrsignalen har inte gjorts. Det bedöms inte påverka verifieringen av metoderna för detekteringen och målföljningen eftersom varken bäringarna eller bildens noggrannhet påverkas. Tiden det tar innan ett mål är i centrum av bilden påverkas dock eftersom den maximala hastigheten sänks. Efter mätningar där hastigheten på siktet jämfördes utifrån samma styrsignal som till simulatoren kunde hastighetssänkningen bestämmas till ca 20%.

5.3 Detektering

Det finns två möjliga metoder för detektering; rörelsedetektering och stilla detektering. Rörelsedetektering upptäcker förändringar i bilden. För att upptäcka förändringar krävs en lång observationstid vilket i fallet med siktet innebär att flera bilder jämförs mot varandra. Stilla detektering innebär att objektet urskiljs från bakgrunden genom t.ex. identifiering.

I fallet med en blå himmel med vita moln kan bakgrunden filtreras bort genom t.ex. ekvation 7.

$$i_{gs} = 2 \cdot i_b - i_g \quad (7)$$

Gråskalebilden i_{gs} beror på de blå värdena i_b och de gröna värdena i_g av färgbilden, enligt ekvation 7.

Därav kan stilla detektering tyckas enkel, men följande utmaningar identifierades.

1. Bilderna genereras i gråskala.
2. Ljusstyrkan på kamerorna är adaptiv vilket innebär att ljusstyrkan i varje bild normaliseras.
3. Himmelens naturliga variation i ljusstyrka.
4. Skillnaden mellan TV- och IRV-kamerans färgsättning.

Eftersom himlens ljusstyrka inte kunde predikteras behövde detekteringen utnyttja de skarpa kontrastförändringar som objekt ger upphov till. Det finns olika metoder för att detektera kontrastförändringar, även kallat kantdetektering. Utmaningen med kantdetektering är att allt upptäcks med metoden, även de objekt som inte eftersöks t.ex. fåglar, moln och träd. För att urskilja de intressanta målen krävs därför identifiering. Om inte objekten identifieras kommer detekteringen hela tiden varna för falska mål, d.v.s. mål som inte är intressanta.

Utan tillförlitlig identifiering fungerar därför endast rörelsedetektering. För att inte tappa mål som stannar alternativt flyger rakt mot eller från siktet krävs en stilla detektering. Därför behövde mål som fångats identifieras och återkopplas till en stilla detektering.

5.3.1 Bakgrundssubtrahering

Bakgrundssubtrahering valdes som metod för rörelsedetekteringen. Utmaningen låg i att balansera svårigheter som nämns i paragrafen 3.2.2. För att kompensera för både snabba ljusförändringar och långvariga scenförändringar behöver

bakgrunden konstant uppdateras. Ett alternativ var då att spara ner hela bakgrunden i minnet och uppdatera bakgrunden, där kameran rör sig.

TV-kameran har upplösningen $180 \times 160 \text{px}/^\circ$ och IRV-kameran har upplösningen $120 \times 120 \text{px}/^\circ$. Färgen för en pixel beskrivs med en byte. Det innebär 32.96MB ifall en yta av $60 \times 20^\circ$ avsöks med TV-kameran. Med den bedömda hastigheten $v_s = 2^\circ/\text{s}$ kan upp till $(6 \cdot 4)/v_s = 12^\circ^2/\text{s}$ avsökas med IRV-kameran. Att söka av hela området bedömdes därför ta $(60 \cdot 20)/12 = 100\text{s}$.

Snabba ljusförändringar bedömdes ske under 100 sekunder och därför räckte det inte att spara ner endast bilden utan varje pixel behövde tidsstämplas. Kommer siktet tillbaka till en bild som inte uppdaterats på länge måste tröskelvärdet för detektion påverkas. Därför behövde ytterligare 4 bytes sparas för varje pixel. Det hade inneburit att bakgrunden krävt $5 \cdot 32.96 = 164.8 \text{MB}$ av minnet för en yta av $60 \times 20^\circ$. Det maximala värdet för bakgrunden bedöms till $360 \times 60^\circ$ vilket inneburit $164.8 \cdot 18 = 2966.3 \text{MB}$.

Med hänsyn till minnesanvändningen och att bakgrunden, p.g.a. snabba ljusförändringar, oftast bedömdes vara för gammal, fastställdes att det var olämpligt att spara ner hela bakgrunden. Istället sparades endast nuvarande område i bakgrunden. Minnesanvändningen för bakgrunden blir därför densamma som för en bild.

Eftersom metoden innebär att ett nytt område i bakgrunden behandlas likt ett okänt område kommer långvariga scenförändringar inte bli ett problem då siktet rör sig. Problemet med snabba ljusförändringar löstes genom att bakgrunden uppdateras enligt ekvation 8 för varje ny bild från siktet.

$$b_{k+1} = a \cdot b_k + (1 - a) \cdot i_{k+1} \quad (8)$$

Bakgrunden b_{k+1} beror på dämpningen a där $0 < a < 1$, föregående bakgrund b_k och bilden i_{k+1} enligt ekvation 8.

Med ett lågt värde på a blir det svårare att upptäcka skillnader mellan bakgrunden och bilden. Dessutom riskerar stora skillnader sparas i bakgrunden vilket kan leda till falska detektioner på målets spår i bakgrunden. Med ett högt värde på a kommer objekt som rör sig fastna i bakgrunden för länge och detekteras på den position där de kommit in i bakgrunden.

Eftersom siktet ständigt rör sig då den söker måste bakgrunden skiftas beroende på siktets rörelser. För att minska bruset i mätningarna av siktets rörelser filtreras därför mätvärdena med Kalmanfiltrering. Ett litet brus på mätvärdena leder till att bakgrundssubtraheringen uppfattar förändringar. Det är viktigt att hastigheten på siktet regleras för att dämpa snabba hastighetsförändringar då det leder till ett större fel på Kalmanfiltreringen.

Regelbundna förändringar bedöms inte vara ett problem då siktet studerar himlen och därav har ingen metod implementeras för att lösa problemet. Ifall sektorn där siktet avsöker ska kunna sättas så att träd är innanför finns det bedömt två

metoder att lösa problemet. Den ena är genom att använda en klotterkarta. Den andra är genom att identifiera träd och andra objekt i förgrunden och i förväg stänga av detekteringen i områdena.

5.3.2 Tröskelvärde för signaldetektering

Tröskelvärdet för signaldetekteringen är värdet som skillnaden mellan bilden och bakgrunden måste överstiga för att ge en detektion. Tröskelvärdet sätts till ett tal mellan 0 och 1. Värdet 1 är den maximala skillnaden mellan bilden och bakgrunden, det innebär att inget kan detekteras. Om värdet är 0 detekteras alla skillnaderna.

När kameran rör sig måste tröskelvärdet för signaldetekteringen, precis som bakgrunden, skiftas. I de nya områdena skiftas värdet 1. Det innebär att längs kanten dit kameran rör sig kan ingenting fångas. För varje ny bild sänks värdet, dock måste en konstant adderas beroende på bruset mellan bilderna för att värdet inte ska nå 0. Tröskelvärdet kan liknas vid KFA-nivån i en radar, se paragraf 3.1.4.

$$t_{k+1} = a \cdot t_k + k \quad (9)$$

Tröskelvärdet t_{k+1} beror på dämpningen a , föregående tröskelvärde t_k och konstanten k enligt ekvation 9.

5.3.3 Stilla detektering

Den stilla detekteringen tar endast hänsyn till färgen och positionen som identifierats. Formen varierar beroende på flygplanets lutning och bedömdes därför som olämplig att använda.

Färgen identifieras när målet upptäckts och målföljning påbörjats. Först avsöks en yta av ca $0.01^{\circ 2}$ runt målets predikterade position. Därefter väljs färgen med störst skillnad mot bakgrunden. Den identifierade färgen uppdateras var femte sekund då målet skiljer sig mer från bakgrunden än tröskelvärdet för signaldetektering.

När färgen på målet identifierats kan målet detekteras då det är stillastående. Färgen på pixlarna och målet jämförs och om skillnaden är mindre än ett valt gränsvärde markeras detektion. När färgen inte uppdaterats på 15 sekunder stoppas detekteringen av det stillastående målet i syfte att förhindra mållåsning på falskmål.

Metoden kräver att rörelsedetektorn först upptäcker målen för att position och färg ska identifieras. När målen har identifierats förhindrar metoder att målen tappas under 15 sekunder om de antingen stannar, alternativt flyger i rak bana mot eller från siktet.

5.3.4 Detektionsanalys

Rörelsedetekteringen och den stilla detekteringen generar varsin bild där varje detektion är markerad. Avslutningsvis slås bilderna samma och områdena i bilderna där detekteringar markerats kopplas samman. För att lösa uppgiften användes ett inbyggt verktyg i *Matlab* kallat *BlobAnalysis* [25].

Detektioner som är mindre än 8px från varandra slås ihop till ett objekt. Objekt som är större än 3000px filtreras bort då de bedöms som för stora för att vara flygplan. Objekt som är mindre än 30px filtreras bort i syfte att minska antalet falskmål.

5.4 Målföljning

När ett mål väl detekterats är det viktigt att målföljningen optimeras för att öka noggrannheten. På alla mätvärden, från bäringarna av siktet till detektionerna på bilderna, finns ett visst brus. Syftet med målföljningen var att minska bruset samt prediktera målet då det tappats.

Målföljningen beskrivs i fem steg enligt nedanstående lista.

1. Befintliga målföljen predikteras.
2. Detektioner kopplas till befintliga målföljen.
3. Målföljen korrigeras.
4. Gamla målföljen tas bort.
5. Oanvända detektioner bildar nya målföljen.

Om siktet tappar målföljet på ett flygplan som flyger bakom en kulle bör det leta på andra sidan efter flygplanet istället för att tappa målföljet. Av den anledningen krävs att predikteringen dels utgår från mätvärdena, men också att den värderar en taktisk aspekt då målets positionssäkerhet ökar.

Flygplanets manövrering begränsas av att piloten inte klarar obegränsat antal g-krafter samt att det endast går att flyga i luften. Vill piloten dessutom undgå upptäckt är det oftast bäst att flyga lågt. Bedömningen bygger på att radar tidigare upptäcker mål på högre höjder p.g.a. mask och radarhorisont.

I detta projektet har inte utformningen av en optimal taktisk prediktering analyserats. En metod har dock utvecklats i syfte att påvisa möjligheterna till taktiskt prediktering. Predikteringen av målföljerna modifierades därför från Kalmanfiltret beskrivet i paragraf 3.1.6. Modifieringen innebar att målföljets hastighet sänktes med 5%/s i X-led och 10%/s i Y-led då ingen korrigering gjorts på över två sekunder. Syftet var att söka närmre den position där målet försvunnit och minska sannolikheten för att siktet riktas under horisonten, d.v.s mot marken.

För att koppla detektionerna mot befintliga målföljen används *the Hungarian Method* [26]. Beräkningen av kostnaden för metoden sker genom att en ellips ritas runt hastighetsvektorn. Eftersom predikteringen av positionen var bättre än predikteringen av hastigheten lades storaxeln längs hastighetsvektorn. Storaxeln längd sattes till det dubbla av lillaxeln. Kostnaden $c = 1$ vid ellipsens kant och minskas linjärt med avståndet till mitten där $c = 0$. Det innebär att osäkerheten ortogonalt mot hastighetsvektorn är mindre än osäkerheten längs hastighetsvektorn.

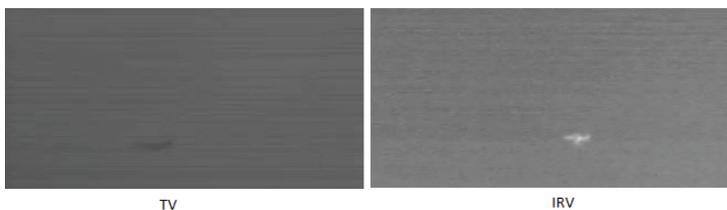
Även korrigeringen av målföljningen modifierades från Kalmanfiltret beskrivet i paragraf 3.1.6. Istället för att beräkna Kalmanförstärkningen enligt tabell 2, så skalades hastighetsförändringarna med skalningsmatrisen M_s enligt ekvation 10. Syftet med modifieringen var framförallt att kunna dämpa hastighetsförändringarna i Kalmanfiltret då målet inte manövrerar.

$$M_s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & s_x & 0 \\ 0 & 0 & 0 & s_y \end{pmatrix}$$

$$K_k = M_s^{-1} P_{k|k-1} H_k^T S_k^{-1} \quad (10)$$

5.5 Identifieringen

I figur 5 syns samma mål från TV- och IRV-kameran. Syftet med identifieringen var att analysera bilder som dessa och klassificera flygplanen. Som synes finns det en del begränsningar.



Figur 5: Bilder från TV- och IRV-kameran

Om flygplanet är 10m stort blir vinkeln på 5km avstånd $\arcsin(10/5000) = 0.11^\circ$. Med en pixelupplösning på 120 eller 180 pixlar/ $^\circ$ blir flygplanet antingen 14 eller 21 pixlar.

Färgen kan inte användas för identifieringen eftersom alla flygplan antingen är mörka med TV-kameran eller ljusa med IRV-kameran. Position, hastighet och storlek är okända variabler eftersom mätningen endast sker i två dimensioner. Formen bedömdes därför vara det enda särdrag som kunde analyseras av identifieringen.

För att undersöka möjligheterna till identifieringen undersöktes totalt tre metoder. Först testades HOG beskrivet i avsnitt 3.2.3, därefter testades två egna metoder. Den första egna metoden benämndes 3D formanalys och den andra 2D grafanalys.

5.5.1 3D formanalys

3D formanalys bygger på att jämföra bilden med en projicerad bild av en 3D-modell. Se figur 6 för exempel på en 3D-modell. Modellerna ritas upp i *Matlab* då programmet startar och varje koordinat av modellen sparas så att medelpunkten för modellen ligger i origo på ett 3D koordinatsystem. Eftersom koordinaterna sparas istället för pixlarna kan rotationen göras direkt utan någon omvandling. Metoden beskrivs stegvis i nedanstående lista.

1. Modellen roteras 2° runt samtliga axlar, varje rotation fortsätter till steg 2.
2. En 2D bild projiceras från modellen.
3. Storleken på modellen anpassas till objektet på 2D bilden.
4. Kanten på modellen och objektet detekteras.
5. Korrelationen sätts till skillnaden mellan kanten på modellen och objektet enligt ekvation 11.
6. Börjar om på steg 1 ifall korrelationen är bättre än föregående värde.
7. När modellen roterats i samtliga axlar utan att korrelationen sänkts returneras korrelationen.

$$K = \left(\sum_{i=1}^n \left(\frac{(b_x(i) - m_x(i))^2 + (b_y(i) - m_y(i))^2}{n} \right) \right)^{-1} \quad (11)$$

För att beräkna korrelationen K testades två metoder. Den första byggde på avståndet mellan de närmsta punkterna på kanten i modellen och bilden, se ekvation 11. Avståndet bestäms enligt Pythagoras sats. Korrelationen K bestämdes därefter som det inverterade medelvärde av summan på avståndet i kvadrat. I ekvation 11 är $[m_x(i), m_y(i)]$ den punkt på kanten i modellen som är närmst till $[b_x(i), b_y(i)]$ kanten i bilden.

Den andra metoden byggde på att beräkna hur stor andel av objekten som täcker varandra då bilderna överlappas. Ifall rotationen inte optimeras kan t.ex. hela vingarna missa varandra. Det påverkar korrelationen negativ eftersom vingarna kan utgöra en relativt stor del av ytan på ett flygplan, se figur 6. Därför bedömdes den första metoden som mest lämplig.

5.5.2 2D grafanalys

2D grafanalys bygger på att analysera flygplanets vingar. Utifrån vingarnas form, storlek och placering kan det vara möjligt att klassificera flygplanstypen. Metoden beskrivs stegvis i nedanstående lista.

1. Analys av hastighetsvektorn bestämmer nosens position.
2. Bilden roteras tills nosen pekar åt höger $x_{nos} = \max(x)$ och flygplanet ligger horisontellt, $y_{medel} = 0$.
3. Nedre delen av flygplanet identifieras, alla värden med $y > 0$ tas bort.
4. Värdena på y vänds, $y = -y$.
5. Lokala maximum och minimum beräknas.
6. Maximumen normaliseras.
7. Vingarnas storleksförhållande är y -värdet på maximumen och deras position x -värdet.
8. Vingarnas form bestäms av derivatan dy/dx .

Eftersom avståndet är okänt kan endast storleken och positionen beräknas relativt flygplanets storlek. Flygplanets faktiska storlek kan inte beräknas utan kompletterande avståndsmätning. Det innebär en begränsning i metodens möjlighet att klassificera flygplanstypen.

6 Experiment

6.1 Målföljning under manövrering

Experimenten för målföljningen gjordes i simulatorn. Simulatorns begränsningar omöjliggjorde simulerade mål utan kursändringar, eftersom målen då hade flugit ut ur bilden. Därför flög målen i en cirkulär bana under experimenten.

Följande värde för systemets övergångsmodell A , observationsmodellen H och styrsignalsmodellen B användes under samtliga tester. De två övre raderna beskriver målets position i bäring och elevation, de två undre beskriver hastigheten. T_s är tiden mellan två prediktioner av målföljet.

$$A = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Den uppskattade kovariansen P initierades med följande värde.

$$P_0 = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{pmatrix}$$

Under test 1-3 användes följande värde för kovariansen på processbruset Q , kovariansen på mätbruset R och skalningsmatrisen M_s .

$$Q = \begin{pmatrix} 0.3^2 & 0 & 0 & 0 \\ 0 & 0.3^2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, R = \begin{pmatrix} 0.03^2 & 0 \\ 0 & 0.03^2 \end{pmatrix}, M_s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Under test 4-6 användes följande värde för kovariansen på processbruset Q , kovariansen på mätbruset R och skalningsmatrisen M_s .

$$Q = \begin{pmatrix} 0.1^2 & 0 & 0 & 0 \\ 0 & 0.1^2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, R = \begin{pmatrix} 0.3^2 & 0 \\ 0 & 0.3^2 \end{pmatrix}, M_s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Under test 7-9 användes följande värde för kovariansen på processbruset Q , kovariansen på mätbruset R och skalningsmatrisen M_s .

$$Q = \begin{pmatrix} 0.1^2 & 0 & 0 & 0 \\ 0 & 0.1^2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, R = \begin{pmatrix} 0.3^2 & 0 \\ 0 & 0.3^2 \end{pmatrix}, M_s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$$

Under test 10-12 användes följande värde för kovariansen på processbruset Q , kovariansen på mätbruset R och skalningsmatrisen M_s .

$$Q = \begin{pmatrix} 0.1^2 & 0 & 0 & 0 \\ 0 & 0.1^2 & 0 & 0 \\ 0 & 0 & 0.5^2 & 0 \\ 0 & 0 & 0 & 0.5^2 \end{pmatrix}, R = \begin{pmatrix} 0.3^2 & 0 \\ 0 & 0.3^2 \end{pmatrix}, M_s = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Med lägre uppskattning av processbruset för hastigheten blev följningen instabil, 0.5^2 var det lägsta värdet som testades då följningen inte tappades.

6.2 Identifiering

6.2.1 HOG på inspelad video från siktet

HOG utnyttjar formen för att identifiera ett objektet. Det fungerar bättre om formen alltid är densamma oavsett betraktningvinkeln, som t.ex. en fotboll. Det blir svårare om formen förändras beroende på betraktningvinkeln. Ett flygplan kan t.ex. se ut som figur 6.



Figur 6: Olika betraktningvinklar på samma 3D-modell

En SVM [14] kan tränas att hantera vissa förändringar av betraktningvinkeln. Hur stora förändringarna kan vara bedömdes dock som osäkert. Därför användes den största möjliga mängden träningsdata för att maximera sannolikheten att klara hanteringen av förändringarna.

SVM till HOG tränades upp av *trainCascadeObjectDetector* [27] i *Computer Vision System Toolbox* [17]. 1808 positiva och 5085 negativa bilder användes, samtliga tagna av siktet. Fler bilder fanns inte att tillgå. Ungefär hälften av bilderna föreställde en Panavia Tornado och hälften en Learjet 35A. Total tränades 24 steg i SVM upp mot en sann positiv faktor av 0.995 och en falsk alarmering på 0.5. En sann positiv faktor på 0.995 innebär att 99.5% av målen skall detekteras.

Syftet med experimentet var att analysera ifall HOG kunde användas för att detektera små flygplan. Ifall inte detektion var möjligt bedömdes inte heller identifiering vara möjligt. Då gradienterna beräknas från celler av 4x4 pixlar bedömdes det finnas en risk att detaljerna på formerna kunde filtreras bort. Därav bedömdes det som osäkert ifall HOG överhuvudtaget kunde användas för detektion av små flygplan.

6.2.2 3D formanlys på positiv och negativ referensbild

För att testa 3D formanlys byggdes en 200x200x200 pixlar stor 3D-modell av en Learjet 35A [28], se figur 6. Därefter jämfördes modellen med olika rotationsvinklar mot den positiva referensbilden figur 7.



Figur 7: Positiv referensbild

Även negativa bilder jämfördes för att jämföra skillnaden på korrelationen. Som referensbild användes t.ex. figur 8.



Figur 8: Negativ referensbild

Referensbilderna var minst 200x200 pixlar. Syftet med experimenten var att testa om metoden kunde användas på större objekt.

6.2.3 2D grafanalys på testbilder

I testerna med 2D grafanalys var flygplanets bedömda hastighetsvektor given. Bilderna i figur 9 testades. Återigen var syftet med experimentet att testa om metoden kunde användas på större objekt, därför var den minsta bilden 276x183 pixlar.



Figur 9: Flygplan

7 Resultat

7.1 Avsökningshastighet

Avsökningshastigheten beror på bakgrundssubtraheringen och tröskelnivån för signaldetektering. En snabbare uppdatering ger möjlighet till högre hastighet. Vid en för snabb uppdatering av bakgrunden detekteras målspåret. Efter försök sattes det minsta tillåtna tröskelvärde för signaldetektering till 0.1. Därav blev det största tillåtna målspåret i bakgrunden 0.1 och maximalt 10% av bilden kunde användas till bakgrunden. Dämpningen a på bakgrundssubtraheringen blev 90%.

Tröskelnivån bör inte dämpas snabbare än bakgrundnivån eftersom målet då kan fångas där det först kommer in i bakgrunden. Därför sattes även dämpningen på tröskelnivån till 90%. Konstanten varierar efter brusets på videon, dock med lägsta värde på 0.01.

Vid en horisontell hastighet på $0.2^\circ/\text{bild}$ görs 20 respektive 30 detekteringar med TV- och IRV-kameran för varje punkt. Med dämpningen $a = 90\%$ och konstanten $k = 0.02$ sänks tröskelnivån, enligt ekvation 9, till 0.2973 med TV-kameran samt 0.2339 med IRV-kameran. För att inte missa mål sattes den maximala hastigheten vid sökning till $0.2^\circ/\text{bild}$.

$$v_a = v_b \cdot f_{bps} \cdot s_{max} \quad (12)$$

v_a är avsökningshastigheten ($^\circ/s$), v_b är hastigheten per bild ($^\circ/\text{bild}$), f_{bps} bilder per sekund (fps) och s_{max} maximala bildbredden ($^\circ$).

Enligt ekvation 12 blev maximala avsökningshastigheten $10.68^\circ/s$ med $v_b = 0.2^\circ/$, $f_{bps} = 8.9\text{fps}$ och $s_{max} = 6^\circ$.

7.2 Målföljning

I tabell 5 uppskattas mätbruset som lägre än processbruset, d.v.s. filtret litrar mer på mätvärdena än predikteringen. I tabell 6 uppskattas positionens processbrus som lägre än mätbruset, det innebär att filtret litrar mer på predikteringen av positionen. I tabell 7 dämpades dessutom hastighetsförändringarna m.h.a. skalningsmatrisen. I tabell 8 sänktes uppskattningen av hastighetens processbrus i syfte att värdera predikteringen så högt som möjligt.

Det angivna mätbruset på bäringarna är normalfördelat med angiven standardavvikelse. Samtliga avvikelser anges som standardavvikelsen. Målets hastighet anges utifrån betraktarens position och accelerationen är beräknad utifrån att målet flyger 5km från betraktaren.

Syftet med tabellerna är att påvisa vilka inställningar Kalmanfiltret bör ha vid olika manövreringar. Test 3 har det bästa positionsfelet vid en hög acceleration. Hastighetsavvikelsen visar dock ingen signifikant skillnad mot inställningarna vid test 12.

Då accelerationen sjunker minskar också skillnaderna mellan testerna. Hastighetsavvikelsen är som störst för test 7 och positionsavvikelsen för test 1. Anledningen till att hastighetsavvikelsen är störst i test 7 beror på att skalningsmatrisen fördröjer förändringen vilket medför ett större fel under manövrering. I de andra metoderna är det konstanta felet mindre eftersom hastigheten uppdateras snabbare. Anledningen till att positionsfelet är störst i test 1 beror på att Kalmanfiltret nästan uteslutande använder mätvärdet då positionen korrigeras. Det innebär att mät- och positionsfelet blir ungefär densamma.

Om figur 11 jämförs med figur 10 och 12 syns det tydligt att bruset i hastighetsvektorerna är större enligt inställningarna för test 3 och 12. Plottarna är jämnare i figur 11. Det innebär att skalningsmatrisen kan dämpa felet på hastighetsvektorn till nära 0 då målet inte manövrerar. Det hade i sin tur avsevärt förbättrat predikteringen då målet inte manövrerar.

<i>Test</i>	1	2	3
Målets hastighet ($^{\circ}/s$)	0.900	1.80	2.70
Målets acceleration, avstånd 5000m (m/s^2)	11.8	47.2	106
Mätbrus ($^{\circ}$)	0.0200	0.0200	0.0200
Filtrerad mätfelsavvikelse bäring ($^{\circ}$)	0.0161	0.0159	0.0159
Filtrerad mätfelsavvikelse elevation ($^{\circ}$)	0.0157	0.0160	0.0159
Antal testvärden	2115	2155	2744
Positionsavvikelse bäring ($^{\circ}$)	0.0159	0.0159	0.0158
Positionsavvikelse elevation ($^{\circ}$)	0.0156	0.0160	0.0159
Hastighetsavvikelse bäring ($^{\circ}/s$)	0.0989	0.223	0.404
Hastighetsavvikelse elevation ($^{\circ}/s$)	0.1079	0.2409	0.451
Antal testvärden	1846	1991	2548

Tabell 5: Test Kalmanfilter, test 1-3

<i>Test</i>	4	5	6
Målets hastighet ($^{\circ}/s$)	0.900	1.80	2.70
Målets acceleration, avstånd 5000m (m/s^2)	11.8	47.2	106
Mätbrus ($^{\circ}$)	0.0200	0.0200	0.0200
Filtrerad mätfelsavvikelse bäring ($^{\circ}$)	0.0161	0.0154	0.0165
Filtrerad mätfelsavvikelse elevation ($^{\circ}$)	0.0163	0.0154	0.0167
Antal testvärden	3989	2473	2492
Positionsavvikelse bäring ($^{\circ}$)	0.0144	0.0140	0.0177
Positionsavvikelse elevation ($^{\circ}$)	0.0144	0.0142	0.0188
Hastighetsavvikelse bäring ($^{\circ}/s$)	0.132	0.288	0.567
Hastighetsavvikelse elevation ($^{\circ}/s$)	0.123	0.316	0.630
Antal testvärden	3723	2311	2292

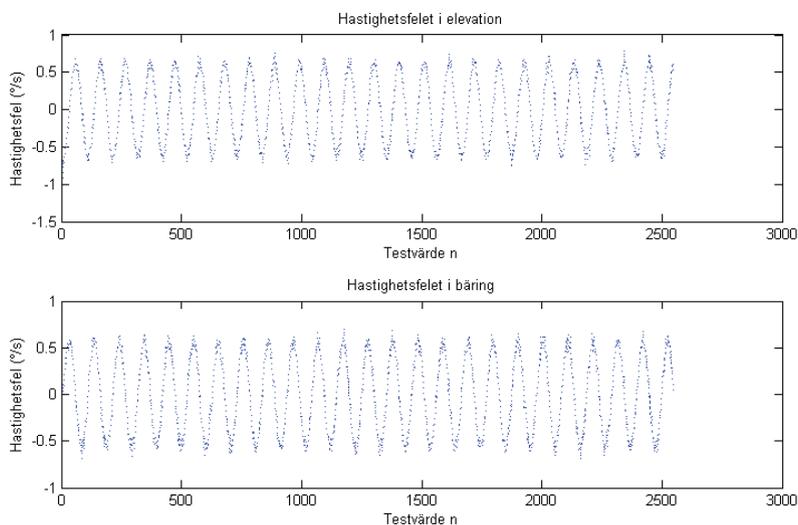
Tabell 6: Test Kalmanfilter, test 4-6

<i>Test</i>	7	8	9
Målets hastighet ($^{\circ}/s$)	0.900	1.80	2.70
Målets acceleration, avstånd 5000m (m/s^2)	11.8	47.2	106
Mätbrus ($^{\circ}$)	0.0200	0.0200	0.0200
Filtrerad mätfelsavvikelse bäring($^{\circ}$)	0.0157	0.0164	0.0163
Filtrerad mätfelsavvikelse elevation($^{\circ}$)	0.0159	0.0163	0.0162
Antal testvärden	1881	1652	3632
Positionsavvikelse bäring($^{\circ}$)	0.0145	0.0290	0.0495
Positionsavvikelse elevation($^{\circ}$)	0.0157	0.0306	0.0549
Hastighetsavvikelse bäring ($^{\circ}/s$)	0.185	0.565	0.985
Hastighetsavvikelse elevation ($^{\circ}/s$)	0.207	0.619	1.08
Antal testvärden	1631	1463	3585

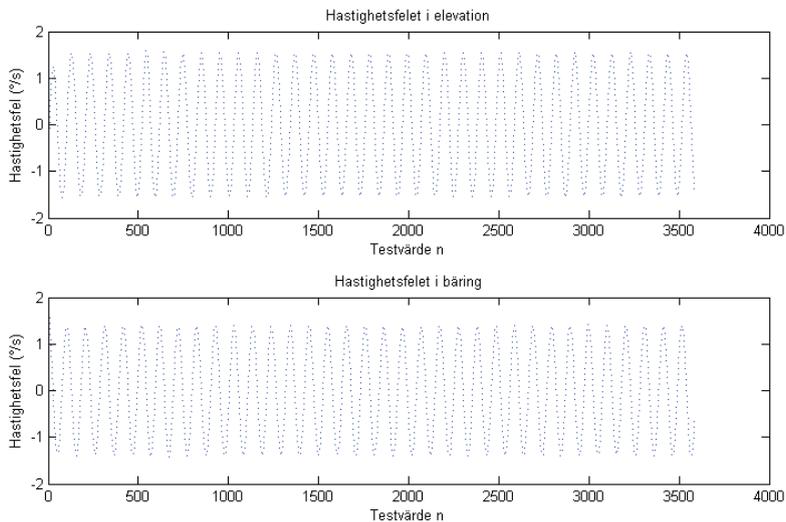
Tabell 7: Test Kalmanfilter, test 7-9

<i>Test</i>	10	11	12
Målets hastighet ($^{\circ}/s$)	0.900	1.80	2.70
Målets acceleration, avstånd 5000m (m/s^2)	11.8	47.2	106
Mätbrus ($^{\circ}$)	0.0200	0.0200	0.0200
Filtrerad mätfelsavvikelse bäring($^{\circ}$)	0.0166	0.0163	0.0161
Filtrerad mätfelsavvikelse elevation($^{\circ}$)	0.0164	0.0167	0.0167
Antal testvärden	2512	2098	2026
Positionsavvikelse bäring($^{\circ}$)	0.0131	0.0184	0.0314
Positionsavvikelse elevation($^{\circ}$)	0.0132	0.0192	0.0345
Hastighetsavvikelse bäring ($^{\circ}/s$)	0.0927	0.217	0.399
Hastighetsavvikelse elevation ($^{\circ}/s$)	0.103	0.237	0.441
Antal testvärden	2093	1949	1968

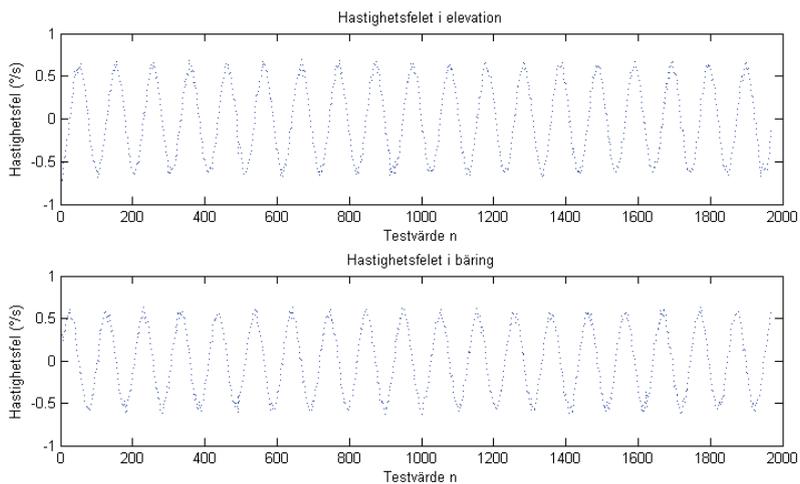
Tabell 8: Test Kalmanfilter, test 10-12



Figur 10: Test3 hastighetsavvikelse



Figur 11: Test9 hastighetsavvikelse



Figur 12: Test12 hastighetsavvikelse

7.3 Identifiering

7.3.1 HOG

Resultatet av figur 13-17 visar tydligt att det går att detektera flygplan med HOG, även relativt små flygplan. Det går dock inte att identifiera flygplanen och inte heller att särskilja flygplanen mot t.ex. fåglar, se figur 18-23. Även om flygplanen generellt detekteras oftare än fåglarna blir tillförlitligheten ett problem. För att förbättra prestandan hade en större mängd träningsdata krävts.

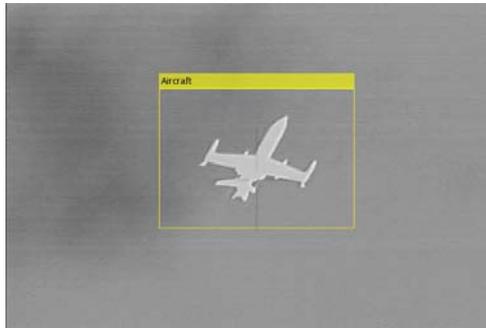
Inga försök att klassificera olika flygplanstyper gjordes då resultatet inte påvisade en tillräckligt hög tillförlitlighet då fåglar och flygplan jämfördes.



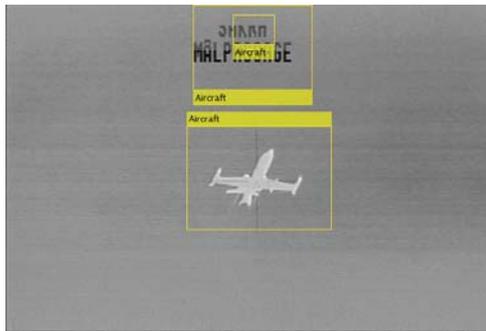
Figur 13: Sann positiv identifiering



Figur 14: Sann positiv identifiering



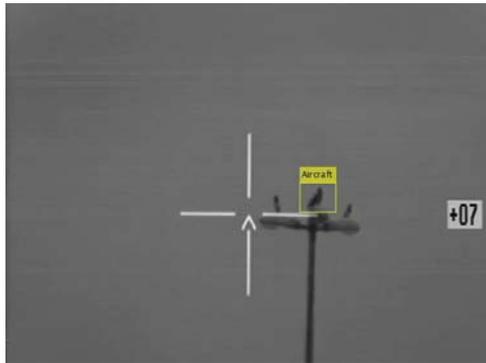
Figur 15: Sann positiv identifiering



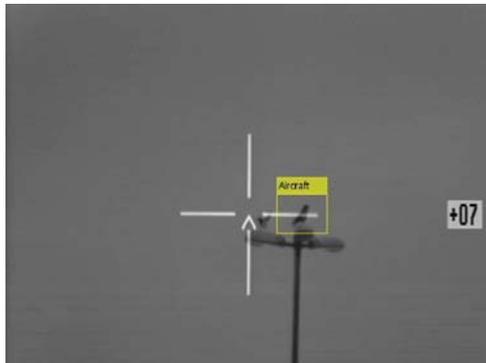
Figur 16: Sann positiv och falsk positiv identifiering



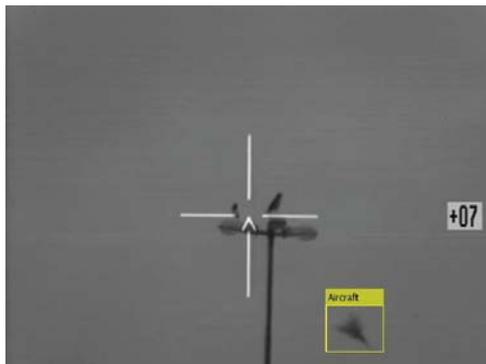
Figur 17: Molnbild gav ingen identifiering



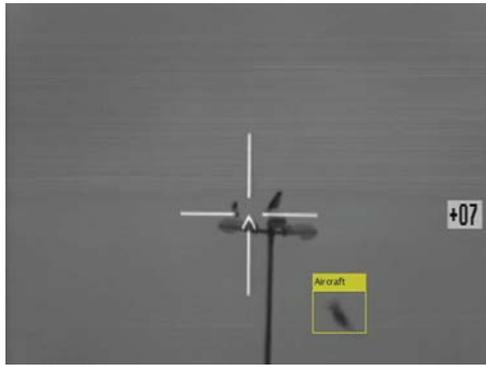
Figur 18: Falsk positiv identifiering



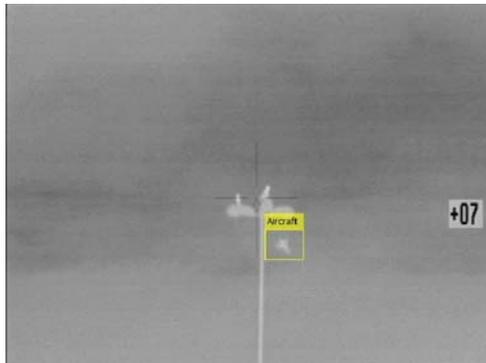
Figur 19: Falsk positiv identifiering



Figur 20: Falsk positiv identifiering



Figur 21: Falsk positiv identifisering



Figur 22: Falsk positiv identifisering



Figur 23: Falsk positiv identifisering

7.3.2 3D formanalys

Skillnaderna mellan de positiva och negativa bilderna var för små för att kunna särskiljas. Det betyder att metoden ibland gissar rätt, ibland fel. Efter försök med metoden bedömdes att den inte skulle klara av att med tillräckligt hög tillförlitlighet identifiera målen och därför avbröts försöken.

Optimeringen av rotationen är flaskhalsen i metoden. Eftersom inte korrelationen förbättras hela vägen till maximal korrelation finns det ett okänt antal lokala maxima. Optimeringen fastnar i de lokala maximumen. För att lösa problemet måste samtliga vinklar analyseras. Det är dock inte möjligt eftersom varje iteration tog 0.200s. Att testa 360 vinklar i 3 dimensioner hade tagit $0.2 \cdot 360^3 / (60 \cdot 60 \cdot 24)$ sekunder = 108 dagar.



Figur 24: Modell med rotation $[-110^\circ, -4^\circ, 17^\circ]$

Figur 24 jämfördes med den positiva figuren 7. Resultatet av korrelation blev 0.1864. Rotationens ursprungsvärde var $[-100^\circ, -8^\circ, 15^\circ]$.



Figur 25: Modell med rotation $[-131^\circ, 3^\circ, 8^\circ]$

Figur 25 jämfördes med den positiva figuren 7. Resultatet av korrelation blev 0.2998. Rotationens ursprungsvärde var $[-105^\circ, 15^\circ, 12^\circ]$.



Figur 26: Modell med rotation $[-123^\circ, 7^\circ, 0^\circ]$

Figur 26 jämfördes med den positiva figuren 8. Resultatet av korrelation blev 0.1288. Rotationens ursprungsvärde var $[-125^\circ, 15^\circ, 0^\circ]$.



Figur 27: Modell med rotation $[-153^\circ, 27^\circ, 90^\circ]$

Figur 27 jämfördes med den positiva figuren 8. Resultatet av korrelation blev 0.1163. Rotationens ursprungsvärde var $[-125^\circ, 15^\circ, 90^\circ]$.

7.3.3 2D grafanalys

Tabell 9 beskriver resultaten från testerna med figur 9. Den relativa position utgår från nosen, den relativa bredden utgår från flygplanets mittpunkt.

<i>Bild</i>	fighter1	fighter2	fighter3	fighter4
Relativ position vinge 1	0.763	0.581	0.662	0.573
Relativ bredd vinge 1	1	1	1	0.394
Form vinge 1	Delta	Delta	Delta	Victor
Relativ position vinge 2	0.980	0.932	0.911	0.855
Relativ bredd vinge 2	0.524	0.651	0.526	1
Form vinge 2	Delta	Delta	Delta	Delta

<i>Bild</i>	fighter5	fighter6	fighter7	bomber1
Relativ position vinge 1	0.843	0.656	0.823	0.654
Relativ bredd vinge 1	1	1	1	1
Form vinge 1	Delta	Delta	Victor	Delta
Relativ position vinge 2	0.994	0.823	1	-
Relativ bredd vinge 2	0.544	0.500	0.261	-
Form vinge 2	Victor	Delta	Victor	-

Tabell 9: Test 2D grafanalys

Begränsningen med metoden är att flygplanets vingarna måste synas. När experimenten gjordes visade det sig att resultatet dels berodde på betraktningvinkeln och dels rotationen. Det kvarvarande felet när flygplanet vrids får algoritmen att beräkna gradienterna på vingarna fel. Det gör att ett gränsvärde för formen kan passeras.

Vingarnas form sätts enligt programkoden nedan. Gradienterna *forward* respektive *backward* beskriver framsidan respektive baksidan av vingen. *India* representerar raka vingar; *Yankee*, framåtsvepta vingar; *Victor*, bakåtsvepta vingar; och *Delta*, deltavingar.

```

if abs(forward) > 1.5 && abs(backward) > 1.5
    forms(end+1) = WingForm.India;
elseif abs(forward) > 1.5 && abs(backward) < 0.5
    forms(end+1) = WingForm.Yankee;
elseif backward >= 0
    forms(end+1) = WingForm.Delta;
elseif backward < 0
    forms(end+1) = WingForm.Victor;
end

```

Sannolikheten att dels formen klassificeras fel, samt att vingar inte detekteras ökar med mindre bilder. Mindre bilder leder nämligen till en sämre upplösning då gradienterna beräknas. Dessutom kan vingarna försvinna i brusreduceringen om det inte är tillräckligt stor skillnad mellan maxima och minima.

8 Diskussion

8.1 Jämförelse med befintliga system

Systemet i tabell 10 detekterar mål med optroniska sensorer.

<i>System</i>	<i>Tillverkare</i>
FIRST - Fast InfraRed Search and Track [29]	Rheinmetall
MSP 500 - Multi Sensor Platform [30]	Rheinmetall
MSP 600 - Optronic Modular Sensor Platform [30]	Rheinmetall
SkyWard [31]	Selex ES
AN/AAQ-37 Distributed Aperture System (DAS) [32]	Northrop Grumman
MTS-B Multi-Spectral Targeting System [33]	Raytheon

Tabell 10: Potentiella system för jämförelse

Den öppna informationen om systemen bedömdes varken som tillräckligt tillförlitlig eller tillräckligt utförlig för att en vetenskaplig jämförelse av resultaten skulle vara möjlig. Att systemen finns och används bevisar dock att det är möjligt med detektering och målföljning m.h.a. optroniska sensorer.

Resultaten på målföljningen skulle kunna jämföras med t.ex. radarstationers målföljning. Inga jämförelser har dock kunnat göras då prestandan på samtliga system som undersökts varit hemligstämplad.

Det har inte kunnat bekräftas att system som identifierar flygplan med optroniska sensorer finns. Därför har ingen jämförelse av resultaten kunnat göras på identifieringen.

8.2 Avsökningshastighetens betydelse för en sensor

Avsökningshastigheten $10.68^{\circ}/s$ bedöms inte vara tillräckligt hög för att hinna upptäcka ett inkommande flygplan. Det tar ett flygplan 20 sekunder att flyga 6km med en hastighet av 300m/s. Det innebär att siktet maximalt hinner söka av 211.6° . Siktet kan då täcka 10° i bredd om osäkerheten på inkommande mål bedöms till 20° i höjd.

Detta är dock inte hela sanningen. För att lösa eventuell uppgift krävs en tidig detektering av målet, eftersom beslut om motåtgärder tar tid. Vid en sen upptäckt finns det ingen tid att besluta om motåtgärder innan målet passerat sensorn. Om en bekämpning av målet tar 15s inklusive beslut har siktet endast 5 sekunder till att upptäcka målet. En mer realistisk täckningsyta mot ett inkommande mål på 300m/s blir då 52.9° . Om osäkerheten i höjd på det inkommande målet fortfarande bedöms som 20° kommer siktet maximalt hinna söka av 2.65° i sida, d.v.s. mindre än dess synfält. Det innebär att siktet inte ens hinner täcka den bedömda osäkerheten i höjd.

Därför bedöms utformningen av siktet inte medge att det används som en sensor. Alternativen kräver en tidig målupptäckt. Med det kravet finns det minst två sätt att lösa uppgiften. Antingen krävs en sensor med ett stort synfält som stirrar mot samma område, eller en sensor som rotera med en hastighet på ett par varv/sekund.

Med en roterande sensor behöver synfältet inte vara lika stort. För att inte behöva ändra elevationen på kameran behöver synfältet i höjd ändå vara minst lika stort som det område där flygen bedöms komma. Det innebär att synfältet fortfarande behöver vara relativt stort. Synfältet $32 \times 18^\circ$ ger exempelvis en höjdtäckning på 1545m vid ett avstånd av 5000m. Mål över 1500m upptäcks sannolikt av radarstationer då de har bättre sikt på högre höjder. Ifall sensorn roterar med en hastighet av 4 varv/sekund med en bredd på 32° så krävs bilduppdateringshastighet på 45 fps. Panorambilden ökar täckningsytan, dock innebär det samtidigt att en zoomfunktion inte är möjlig, vilket försvårar automatisk identifiering. Framställningen av panorambilden kräver en bra lappning mellan bilderna vilket förenklas med ett litet vinkelfel. För det krävs en noggrann bäringangivare som skickar bäringarna samtidigt som bilderna skickas.

Fördelen med den roterande sensorn är att den täcker en större yta. Fördelen med den stirrande sensorn är en högre uppdateringstakt, samt möjlighet att med ett zoomobjektiv kunna generera högupplösta bilder på okända objekt för att hjälpa till med identifieringen.

8.3 Positions- eller hastighetsprecision vid målföljning

I dagens luftvärnssystem låser skytten på målet innan roboten avfyras. Därefter leds roboten av skytten till träff. Eftersom fördröjningen från radarn till skytten är relativt låg blir precisionen på målets position viktigast.

I framtiden kommer RBS 98 införas. Roboten kan skjutas iväg mot en beräknad framförpunkt där den letar efter målet. Det ger fördelen att robotsystemet kan grupperas utan att robotarna behöver en direkt siktlinje mellan sig och målet. Utmaningen blir istället tidsfördröjningen mellan målinmätningen och att roboten själv upptäcker målet. Samma utmaning bedöms även gälla ersättaren till dagens RBS 97. Det systemet kommer förmodligen också skjuta mot en framförpunkt, även om den längre räckvidden förmodligen leder till att roboten förses med en länkförbindelse där den kan motta uppdateringar av framförpunkten.

Därför blir precisionen av hastigheten av större betydelse än positionen för luftvärnet. Felet på framförpunkten kan uppskattas enligt ekvationen 13 då målet inte manövrerar.

$$f_f = v_f \cdot t + p_f \quad (13)$$

f_f framförpunktsfelet, v_f hastighetsfelet, t tiden och p_f positionsfelet.

Avvikelsen på hastighetsvektorn var större än avvikelsen på positionen enligt samtliga tester i detta projekt. Risken att missa ökar därför väsentligt ifall endast Kalmanfiltret utformat enligt tabell 2 utnyttjas för predikteringen. Det krävs att bruset på hastighetsvektorn minskas och skalningsmatrisen kan vara ett sätt att lösa det då målet inte manövrerar.

Det bästa alternativet blir därför att låta parametrarna på filtret utgå från en uppskattning av målets manövrering. När målet inte manövrerar används skalningsmatrisen för att dämpa förändringarna på hastigheten i syfte att minimera hastighetsfelet. När målet manövrerar justeras processbruset upp i syfte att inte tappa målet och för att predikteringen skall ske i rätt riktning.

Avslutningsvis kan det tilläggas att det blir svårare för motståndaren att upptäcka och bekämpa sina mål ifall han tvingas till konstant manövrering för att inte bli bekämpad. Dessutom kommer pilotens uppdrag att fördröjas då han måste flyga en längre sträcka för att slutföra uppdraget. Det innebär att luftvärnet verkar som bäst då motståndaren är som bäst, d.v.s. när han inte behöver manövrera.

8.4 Det optimala identifieringssystemet för flygplan

Detta projektet har fokuserat på identifiering av flygplan genom formanalys på bilder. Resultaten visar att bilderna behöver vara relativt högupplösta. Bättre än vad som beskrivs i Johnsons kriterium. Ett flygplan med en kortsida på 8 pixlar har inte kunnat identifieras.

Med en större mängd träningsdata hade förmodligen tillförlitligheten förbättrats. Det är dock tveksamt ifall tillförlitligheten klarat kraven för ett autonomt militärt system. För att öka tillförlitligheten och kunna särskilja olika objekt bör fler parametrar analyseras då flygplanen identifieras.

En människa använder ett antal olika parametrar för igenkänning och identifiering av objekt. Dels används formen och färgen, men om t.ex. ett lysande föremål på himlen rör sig kan det oftast klassificeras genom att hastigheten analyseras. En stjärnas vinkelhastighet på himlen är oftast mycket lägre än ett flygplans.

Människan lär sig genom hela livet att känna igen olika objekt. Språket används som hjälpmedel för att kategorisera och dela in dem i grupper. En person kan identifiera ett objekt som en bil, t.ex. kan det vara en röd Volvo V70. Ser personen inte direkt att det är en V70 kan han åtminstone säga att det är en röd bil.

En SVM är inte konstruerad på samma sätt. Den delar inte in objektet i olika kategorier. Antingen är den en röd Volvo V70 eller inte. Det innebär att det krävs en SVM för varje val. En som bestämmer ifall det är en röd bil, en som bestämmer ifall det är en röd Volvo V70 o.s.v. Det är en begränsning som användandet av SVM innebär.

När flygplan identifieras bör det göras på relativt långa avstånd. För att snabbt kunna uppfatta ifall det är värt att identifiera bör först snabba analyser göras. Exempelvis kan positionen, hastigheten och storleken analyseras. På så sätt är det relativt enkelt att klassificera objektet som ett stort flygande objekt med hög hastighet, vilket förmodligen kan vara ett flygplan. När det gjorts krävs det ett zoomobjektiv för högupplösta bilder till identifiering. Utan högupplösta bilder kommer tillförlitligheten inte vara tillräckligt hög.

I ett mänskligt öga finns två olika sorters detektorer, stavar och tappar [7]. Stavarna är ljuskänsliga medan tapparna ger skärpa under god ljusförhållanden. Det finns tre olika sorters tappar som reagerar på olika färger, därav färgseende. I gula fläcken finns en hög koncentration av tapparna. Det ger en god förmåga att kunna identifiera objekt i en relativt smal lob. Samtidigt kan de mer ljuskänsliga detektorerna varna för förändringar i periferin.

En kamera har samma sorts detektorer över hela synfältet med samma detektortäthet. Det innebär att endast en uppgift löses optimalt åt gången, antingen identifiering eller detektering. Därför behöver en prioritering mellan detektering och identifiering göras.

8.5 Samhälleliga och etiska aspekter

Sveriges försvarsmakt har som uppgift att skydda Sverige och försvara landets frihet. Det inkluderar att upptäcka och avvisa kränkningar av det svenska territoriet [34].

Om Försvarsmakten inte har förmågan lösa sin uppgift kan landet falla och invånarnas frihet begränsas. Det skulle kunna innebära att de etiska värderingar som präglar landet kan forceras bort och ersättas av främmande makt.

För att säkerställa att Försvarsmakten effektivitet kan försvara landet krävs modern utrustning. Det innefattar bl.a. sensorer för luftrumsövervakning så att t.ex. kränkningar kan upptäckas och avvisas. Detta arbete har bidragit med kunskap i syfte att hjälpa Försvarsmakten att ta fram rätt sorts utrustning.

9 Slutsats

Metoderna som är beskrivna i rapporten för detektering och målföljning fungerar. I simuleringen uppfylls kravet för positionsnoggrannheten. Hastighetsnoggrannheten för manövrerande mål ligger på gränsen till kravet. Två delar testades inte under projektet. Det var dels automatisk justering av parametrarna till Kalmanfiltret för målföljningen. Dels var det implementeringen av en klotterkarta, vilket krävs för att filtrera bort regelbundna förändringar när sensorn ser ner på marken.

Siktet fungerade inte i praktiken eftersom kamerornas synfält var för små. Det innebär att en för liten yta hann avsökas och om målen upptäcktes var det förmodligen redan försent. När siktet byggdes var det tänkt att en radar skulle invisa målen och slutsatsen är att det fortfarande behövs. Därav ströks målet att utveckla en fullständig prototyp med siktet. För att lösa uppgiften krävs det en annan kamera. Antingen krävs det en kamera med ett synfält som täcker hela området och som har en hög pixeltäthet, alternativt en roterande kamera med en rotationshastighet på minst ett par varv/sekund för bra uppdateringshastighet.

Automatisk identifiering skulle kunna fungera. Med enbart metoderna som används i projektet blir tillförlitligheten en utmaning. Objekt som förändras mycket stort beroende på betraktningvinkeln är svårare att identifiera än objekt som alltid förblir densamma oavsett projektion. För att kraven skall uppfyllas behöver därför även parametrar som position, hastighet och storlek analyseras. Att försöka analysera formen och framförallt vingarna på flygplanet är en framgångsfaktor eftersom de karakteriseras av flygplanstypen. Högupplösta bilder ökar tillförlitligheten på en sådan analys.

För identifieringen har tre metoder testats. HOG är den enda som testats på mindre objekt. För att HOG ska klara identifieringen krävs stora mängder träningsdata till SVM. Det krävs dessutom att en SVM tränas upp för varje objekt som skall identifieras. 3D formanalys är inte praktiskt möjligt att använda då rotationen i 3D inte löstes. 2D grafanalys kan ge ytterligare parametrar som kan användas under klassificeringen. Det krävs dock relativt stora bilder på objektet för att metoden skall vara tillförlitlig. Som enskilt verktyg klarar inte 2D grafanalys att identifiera.

Referenser

- [1] Mattias Elfström et al.
Huvudstudie luftvärn - MARK 101104S Slutrapport 2011
Försvarsmakten, 2011.
21 120:20663
- [2] Mikael Nordström
Atmosfärens påverkan på det taktiska utnyttjandet av elektromagnetisk vågutbredning för radar och optiska sensorer.
Försvarshögskolan, 2003.
Beteckning 19 100:2056
<http://www.diva-portal.org/smash/get/diva2:428795/FULLTEXT01.pdf>
Hämtad, 2015-04-16.
- [3] Försvarsmaktens Materielverk
IBOK RBS 90 ELDENHET
Försvarets bok- och blankettförråd, 2001.
M7786-009060
- [4] Totalförsvarets Forskningsinstitut
FOI orienterar om: Sensorer
FOI, Stockholm, 2004.
ISBN 91-7056-119-2
- [5] Kristian Artman och Anders Westman
Lärobok i Militärteknik, vol. 2: Sensorteknik
Elanders, Vällingby, 2007.
ISBN 978-91-85401-73-4
- [6] Merrill I. Skolnik
Introduction to Radar Systems, Third Edition
McGraw-Hill Higher Education, New York, 2001.
ISBN 007-118189-X
- [7] Jonas Persson
Vågrörelselära, akustik och optik
Studentlitteratur, Lund, 2007.
ISBN 978-91-44-01997-0
- [8] R. E. Kalman
A New Approach to Linear Filtering and Prediction Problems
Transaction of the ASME-Journal of Basic Engineering, March 1960, s.
35-45.
- [9] Greg Welch och Gary Bishop
An Introduction to the Kalman Filter
TR 95-041.
http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf
Hämtad, 2015-03-18.

- [10] Brian W. Tolman
GPS Precise Absolute Positioning via Kalman Filtering
ION GNSS 21st. International Technical Meeting of the Satellite Division,
sep 2008, s. 1864-1874
[http://www.gpstk.org/pub/Documentation/GPSTkPublications/
ION2008TolmanPrecisePositioningKalman.pdf](http://www.gpstk.org/pub/Documentation/GPSTkPublications/ION2008TolmanPrecisePositioningKalman.pdf)
Hämtad, 2015-05-04.
- [11] Morris R. Driels.
Weaponneering: Conventional Weapon System Effectiveness
American Institute of Aeronautics and Astronautics Inc., Reston, Virginia
2004.
ISBN 1-56347-665-7
- [12] Birgi Tamersoy
Background Subtraction - Lecture Notes
The University of Texas, Austin, 2009.
[http://www.cs.utexas.edu/~grauman/courses/fall12009/slides/
lecture9_background.pdf](http://www.cs.utexas.edu/~grauman/courses/fall12009/slides/lecture9_background.pdf)
Hämtad, 2015-03-18.
- [13] Navneet Dalal och Bill Triggs
Histogram of Oriented Gradients for Human Detection
In CVPR, 2005, s. 886-893.
<http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>
Hämtad, 2015-03-18.
- [14] Corinna Cortes och Vladimir Vapnik
Support-Vector Networks
Machine Learning, sep 1995, volym 20, s. 273-297.
<http://link.springer.com/content/pdf/10.1007%2FBFB00994018.pdf>
Hämtad , 2015-05-08.
- [15] *Visual Studio Express 2013 for Desktop*
<https://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>
2015-04-16.
- [16] *Matlab*
<http://se.mathworks.com/products/matlab/>
2015-04-16.
- [17] *Computer Vision System Toolbox*
<http://se.mathworks.com/products/computer-vision/>
2015-04-16.
- [18] *EMGU CV*
http://www.emgu.com/wiki/index.php/Main_Page
2015-04-16.
- [19] *Tera Term*
<http://tssh2.sourceforge.jp/>
2015-04-16.

- [20] *VLC media player*
<http://www.videolan.org/vlc/>
2015-04-16.
- [21] Försvarmaktens Materielverk
RMU RBOK 3
Försvarets bok- och blankettförråd, 1994.
M7787-0007313
- [22] *Real-Time Workshop*
<http://se.mathworks.com/products/simulink-coder/?refresh=true>
2015-04-16.
- [23] Andrew S. Tanenbaum
Computer Networks, Fourth Edition
Prentice Hall, New Jersey, 2003
ISBN 01-3066-102-3
- [24] Allexy
TCP/IP Chat Application Using C#
<http://www.codeproject.com/Articles/12893/TCP-IP-Chat-Application-Using-C>
Hämtad, 2014-10-13.
- [25] *BlobAnalysis*
<http://se.mathworks.com/help/vision/ref/blobanalysis.html>
2015-04-16.
- [26] Derek Bruff
The Assignment Problem and the Hungarian Method
Harvard, 2005.
http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf
Hämtad, 2015-03-18.
- [27] *trainCascadeObjectDetector*
<http://se.mathworks.com/help/vision/ref/traincascadeobjectdetector.html>
2015-04-16.
- [28] *Learjet 35A*
<http://jetadvisors.com/learjet-35a/>
2015-04-18
- [29] *FIRST - Fast InfraRed Search and Track*
http://www.rheinmetall-defence.com/en/rheinmetall_defence/systems_and_products/c4i_systems/reconnaissance_and_sensor_systems/fast_infrared_search_and_track_reconnaissance_sensor/index.php
2015-04-17
- [30] *MSP - Multi Sensor Platform*
http://www.rheinmetall-defence.com/en/rheinmetall_defence/systems_and_products/c4i_systems/reconnaissance_and_sensor_systems/multi_sensor_plattform/index.php
2015-04-17

- [31] *SkyWard*
www.selex-es.com/documents/737448/30695920/body_mm08130_Skyward_IRST_LQ_.pdf
2015-04-18
- [32] *AN/AAQ-37 Distributed Aperture System (DAS)*
<http://www.northropgrumman.com/Capabilities/ANAAQ37F35/Pages/default.aspx>
2015-04-18
- [33] *MTS-B Multi-Spectral Targeting System*
http://www.raytheon.com/capabilities/products/mts_b/index.html
2015-04-18
- [34] *Förordning (2007:1266) med instruktion för Försvarsmakten*
http://rkrattsbaser.gov.se/cgi-bin/thw?%24{HTML}=sfst_1st&%24{OOHTML}=sfst_dok&%24{SNHTML}=sfst_err&%24{BASE}=SFST&%24{TRIPSHOW}=format%3DTHW&BET=2007%3A1266%24
2015-05-01

Figur- och tabellförteckning

Figurer

1	Siktet till RBS 90	2
2	Blockschema Radar [4]	6
3	Implementering av HOG [13]	12
4	Blockschema Systemuppbyggnad	16
5	Bilder från TV- och IRV-kameran	23
6	Olika betraktningvinklar på samma 3D-modell	28
7	Positiv referensbild	29
8	Negativ referensbild	29
9	Flygplan	29
10	Test3 hastighetsavvikelse	33
11	Test9 hastighetsavvikelse	34
12	Test12 hastighetsavvikelse	34
13	Sann positiv identifiering	35
14	Sann positiv identifiering	35
15	Sann positiv identifiering	36
16	Sann positiv och falsk positiv identifiering	36
17	Molnbild gav ingen identifiering	36
18	Falsk positiv identifiering	37
19	Falsk positiv identifiering	37
20	Falsk positiv identifiering	37
21	Falsk positiv identifiering	38
22	Falsk positiv identifiering	38

23	Falsk positiv identifiering	38
24	Modell med rotation $[-110^\circ, -4^\circ, 17^\circ]$	39
25	Modell med rotation $[-131^\circ, 3^\circ, 8^\circ]$	39
26	Modell med rotation $[-123^\circ, 7^\circ, 0^\circ]$	40
27	Modell med rotation $[-153^\circ, 27^\circ, 90^\circ]$	40

Tabeller

1	Information RBS 90 [3]	2
2	Kalmanfiltrets algoritm	10
3	Resultat från Johnsons försök [11]	11
4	Kommandoinformation	16
5	Test Kalmanfilter, test 1-3	31
6	Test Kalmanfilter, test 4-6	32
7	Test Kalmanfilter, test 7-9	32
8	Test Kalmanfilter, test 10-12	33
9	Test 2D grafanalys	41
10	Potentiella system för jämförelse	42

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden	<i>Document name</i> MASTER'S THESIS
	<i>Date of issue</i> June 2015
	<i>Document Number</i> ISRN LUTFD2/TFRT--5973--SE
<i>Author(s)</i> Simon Lindgren	<i>Supervisor</i> Håkan Larsson, The Swedish Armed Forces Rolf Johansson, Dept. of Automatic Control, Lund University, Sweden Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden (examiner)
	<i>Sponsoring organization</i>
<i>Title and subtitle</i> Detektering, målföljning och identifiering med robotsystem 90	
<i>Abstract</i> <p>The objective was to clarify whether and how it is possible to detect, track and identify with an EO/IR (electro optical/infrared) sensor. Moreover, the sight of robotic system 90 was tested in order to clarify the possibilities of using it as a sensor. The sight has a field of view of $3 \times 4^\circ$ with the TV camera and $4 \times 6^\circ$ with the IRV camera. Therefore, the project focused on a moving sensor, in order to be able to scan a larger area.</p> <p>By background subtraction it's possible to detect changes, i.e., movements. The background must move with the sight and therefore requires measurements of the sight direction to be accurate. Detection was made possible by noise reduction of the measured values using a Kalman filter. The maximum scan rate of the sight was 10.8^{02} /s. The sight can thus not be used as a sensor because the scanning speed is judged to be too low.</p> <p>Kalman filters may also be used to filter the detections and for a short time predict the target path. With a scaling matrix, error in the velocity vector was reduced when the target was not manoeuvring and the target movement could be predicted over time. However, it led to delays when the target manoeuvred, resulting in poor accuracy of the target position.</p> <p>The video color from the sight is in grayscale, which means that the color couldn't be used to identify aircraft. The aircraft is dark with the TV camera and bright with the IRV camera. What was left then was to identify the type of aircraft by analysing the shape.</p> <p>An existing method for identification by shape analysis is HOG (Histogram of Oriented Gradients). A large amount of training data was required for the support vector machine to handle the differences in shape depending on the sensor viewing angle, the aircraft's rotation and the relatively small pictures. However, the reliability of the method was still not high enough.</p> <p>Two custom methods were therefore tested in order to try to identify targets by shape analysis. In the first method a 3D model of an airplane was built. The identification would be done by rotating and projecting the model to a two-dimensional image which was compared with the image of the target. Finding the best three-dimensional rotation is a problem which can be solved only by testing all solutions. Therefore, it takes too long to identify with this method.</p> <p>The second method aims to analyse the shape by rotating the two-dimensional image so that the nose is pointing along the x-axis of a two-dimensional coordinate system. Wings were then analysed by analysing the values on the y-coordinates relative to the x-coordinates. Maxima, minima and the gradients were analysed in order to place the wings, estimate their relative size to each other and the shape of the aircraft.</p> <p>Method two may be used to increase the reliability of the identification of the aircraft. An autonomous combat system requires the identification to be very reliable, thus requiring the use of an optical zoom to enlarge the images. More parameters such as position, speed and size should also be analysed when identifying the target.</p>	
<i>Keywords</i>	

<i>Classification system and/or index terms (if any)</i>		
<i>Supplementary bibliographical information</i>		
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 1-52	<i>Recipient's notes</i>
<i>Security classification</i>		

<http://www.control.lth.se/publications/>