

FFI RAPPORT

METODER FOR EMITTERIDENTIFISERING I EN ESM-DEMONSTRATOR

SUNDGOT Rune, ØYEHAUG Leiv

FFI/RAPPORT-2002/00555

FFIE/805/113

Godkjent
Kjeller 31 januar 2002

Torleiv Maseng
Forskningsjef

**METODER FOR EMITTERIDENTIFISERING I EN
ESM-DEMONSTRATOR**

SUNDGOT Rune, ØYEHAUG Leiv

FFI/RAPPORT-2002/00555

FORSVARETS FORSKNINGSINSTITUTT
Norwegian Defence Research Establishment
Postboks 25, 2027 Kjeller, Norge

1) PUBL/REPORT NUMBER FFI/RAPPORT-2002/00555	2) SECURITY CLASSIFICATION UNCLASSIFIED	3) NUMBER OF PAGES 41
1a) PROJECT REFERENCE FFIE/805/113	2a) DECLASSIFICATION/DOWNGRADING SCHEDULE -	
4) TITLE METODER FOR EMITTERIDENTIFISERING I EN ESM-DEMONSTRATOR EMITTER IDENTIFICATION METHODS IN AN ESM-DEMONSTRATOR		
5) NAMES OF AUTHOR(S) IN FULL (surname first) SUNDGOT Rune, ØYEHAUG Leiv		
6) DISTRIBUTION STATEMENT Approved for public release. Distribution unlimited. (Offentlig tilgjengelig)		
7) INDEXING TERMS IN ENGLISH: IN NORWEGIAN:		
a) <u>Electronic Support Measures</u>	a) <u>Elektroniske støttetiltak</u>	
b) <u>Specific Emitter Identification</u>	b) <u>Spesifikk emitteridentifikasjon</u>	
c) _____	c) _____	
d) _____	d) _____	
e) _____	e) _____	
THESAURUS REFERENCE:		
8) ABSTRACT This report describes methods for emitter identification that will be used in an Electronic Support Measures (ESM) – demonstrator. The demonstrator will be used by FFI to demonstrate modern passive identification techniques for the Norwegian Air Force. The demonstrator is divided in two modules. The sensor module consists of antennas, tuners, sampling devices, and a pulse sorting device that cluster pulses from the same pulse train into an emission description word (EDW). The output of the sensor module is a continuous flow of pulse sample files and EDWs. The second module is the identification module. This module consists of methods for classification of emitter type, specific emitter identification and visualisation of emitter activity. The different identification methods are based both on interpulse (EDW) and intrapulse (pulse shape) parameters. This report gives an overview of the identification module, with emphasis on pulse representation and emitter identification. It describes different ways to represent and compress pulse data (wavelets, splines, etc), and explains different ways of obtaining specific emitter identification by correlation of pulse parameters against an emitter library (statistical decision theory).		
9) DATE 31 January 2002	AUTHORIZED BY This page only Torleiv Maseng	POSITION Director of Research

INNHOOLD

	Side	
1	INNLEDNING	7
2	HOVEDKOMPONENTENE I DEMONSTRATOREN	8
2.1	Måleoppstilling	9
2.2	Input fra måleoppstilling	10
2.3	PIA struktur og hovedideer	10
2.3.1	UMOP	11
2.3.2	Databaser	11
3	EDW-PROSESSERING	12
3.1	Tracking	12
3.2	Klassifisering av radarmodell	13
3.2.1	PRF-mønsterkjennning	14
3.2.2	Korrelering mot modelldatabase	14
3.2.3	Oppdatering av modelldatabase	14
3.3	Identifikasjon vha krystalloscillatorfrekvens	14
3.4	Styring av pulsanalysen	16
4	PULSDETEKSJON OG PREPROSESSERING	17
4.1	Pulsdeteksjon	17
4.2	Beregning av envelope og IFV	19
5	PULSREPRESENTASJON	21
5.1	Desimering og splines.	21
5.2	Komprimering ved tilpasning	22
5.2.1	Tilpasning ved polynomer	22
5.2.2	Tilpasning ved wavelets	23
5.3	Egenskapsuttrekking	25
5.4	Eksempler på komprimert pulsrepresentasjon	25
5.4.1	Desimering	26
5.4.2	Tilpasning ved polynomer	27
5.4.3	Tilpasning ved wavelets	28
6	IDENTIFIKASJON VED PULSANALYSE	29
6.1	Statistisk desisjonsteori	29
6.1.1	Ikke-parametriske statistiske metoder	31
6.1.2	Parametriske statistiske metoder	32
6.1.3	Fordelingsfrie metoder	33
6.2	Valg av metode	33
6.2.1	Korrelering mot emitterdatabase	34

6.2.2	Oppdatering av emitterdatabase	36
7	FORELØPIGE ERFARINGER MED PIA	37
7.1	Multipath	37
8	OPPSUMMERING	38
	Litteratur	39
	Fordelingsliste	41

METODER FOR EMITTERIDENTIFISERING I EN ESM-DEMONSTRATOR

1 INNLEDNING

Prosjekt 805 PROFIL ved FFI skal bygge opp en ESM-demonstrator (Elektroniske støttetiltak) som skal brukes til demonstrasjonsformål for Luftforsvaret. Demonstratoren skal kunne brukes til å teste ut nye algoritmer for emitteridentifikasjon. Systemet består av to enheter:

- Selve ESM-sensoren med antenne, mottakere, digitalisering og pulssortering.
- Identifikasjonsapplikasjon (PIA) med klassifisering og identifikasjon basert på analyse av interpulsparametere (PRI, osv) og intrapulsparametere (pulsform, IFV, osv).

Prosjektet har så langt laget en del programvare for analyse og klassifisering av radarpulser. Programmet FIA (Fingerprinting applikasjon) omfatter innlesing og dekoding av datafiler, analyse av pulstog (pulssortering) og enkeltpulser, klassifisering, og oppbygging av emitterbibliotek. Denne programvaren er dokumentert i (1). Et annet program IDA (Identifikasjonsapplikasjon) gjør tracking basert på emisjonsbeskrivelsesord (EDWer), modellklassifisering basert på frekvens, pulsrepetisjonsintervall (PRI) og pulslengde, og emitteridentifikasjon basert på krystallosillatorfrekvens. Deler av IDA er dokumentert i (2).

Spesifikasjon av ESM-demonstratoren finnes i (3). Den viktigste endringen i forhold til spesifikasjonen er at FIA og IDA ikke lenger vil være separate programmer, men vil bli integrert i et felles program PIA (PROFILs identifikasjonsapplikasjon). Igangsetting av pulsanalysen (FIA) styres av de behov trackingalgoritmen (IDA) har for å identifisere emittertrack, og resultatene presenteres i et felles display.

Denne rapporten beskriver de sentrale funksjonene og algoritmene i PIA, med vekt på representasjon av radarpulser, komprimering og identifikasjon. Noen av algoritmene/metodene som beskrives er allerede implementert, andre er tatt med fordi det kan være aktuelt å implementere dem i PIA på et senere tidspunkt. Hensikten med rapporten er å beskrive selve metodene, ikke programvaren som implementerer dem.

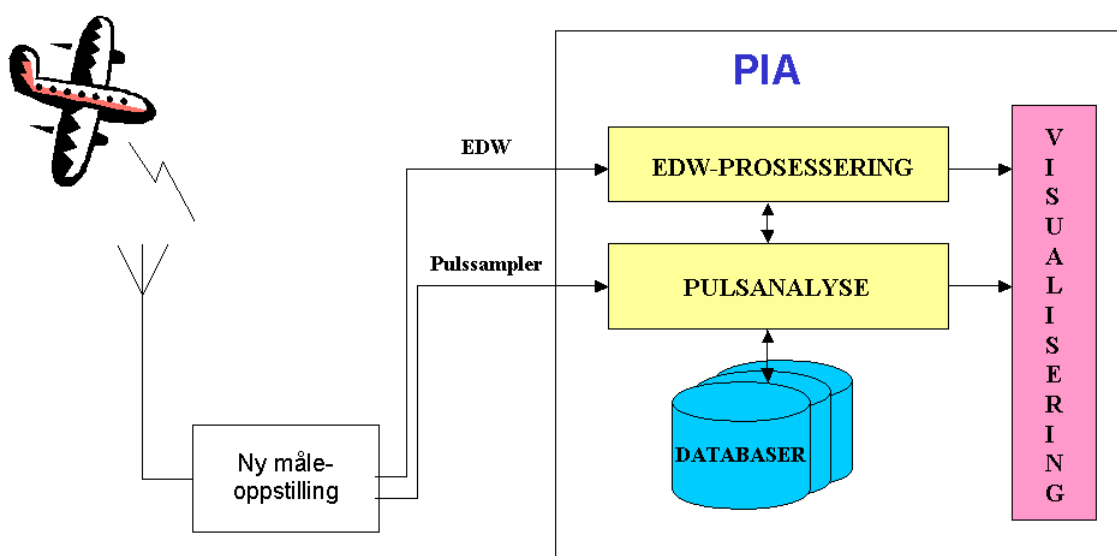
Test og evaluering av identifikasjonsmetodene i PIA vil bli dokumentert i en egen rapport.

2 HOVEDKOMPONENTENE I DEMONSTRATOREN

Hovedoppgaven til demonstratoren er å identifisere hvilke radaremittere vi mottar signaler fra. Basert på mottatte EDWer fra måleoppstillingen vil PIA estimere peiling til emitteren (ut fra antennens pekeretning), bestemme radarmodell (klassifisering) og gjøre emitteridentifikasjon basert på krystallosillatorfrekvens (utledet fra PRF-mønster). For utvalgte EDWer vil en også gå videre med emitteridentifikasjon basert på pulsanalyse (intrapuls egenskaper). Denne analysen baserer seg på pulldata for de gitte EDWer, som hentes fra harddisk.

Aktuelle radarmodeller kan f.eks. være AN/APG-66V2 (kampflyradar) eller WRT-701X (værradar), spesifikk emitter kan f.eks. være kampflyradaren til F16-304 eller værradaren til LN/BRN "Håkon Herdebrei". Bestemmelse av radarmodell brukes i trackingalgoritmen for å assosiere en EDW til riktig emittertrack, og det brukes i pulsanalysen for å avgrense hvilke emitterdatabaser en vil korrelere en mottatt puls mot. Dersom vi ikke lykkes med emitteridentifikasjonen kan angivelsen av radarmodell bli den mest nøyaktige beskrivelsen av emitteren vi klarer å finne.

Hovedresultatet fra PIA er en liste med emittere, hvor en for hver emitter har angitt radarmodell, radarmode, spesifikk emitteridentitet og evt annen informasjon om emitteren. Hvis peilingsinformasjon er tilgjengelig så vil emitterne også vises som peilinger i et kartvindu. I tillegg skal det også være mulig å få informasjon om de parameterverdier som klassifiseringen og identifiseringen bygger på, f.eks. for å vurdere hvor sikker man er på at resultatet er korrekt.



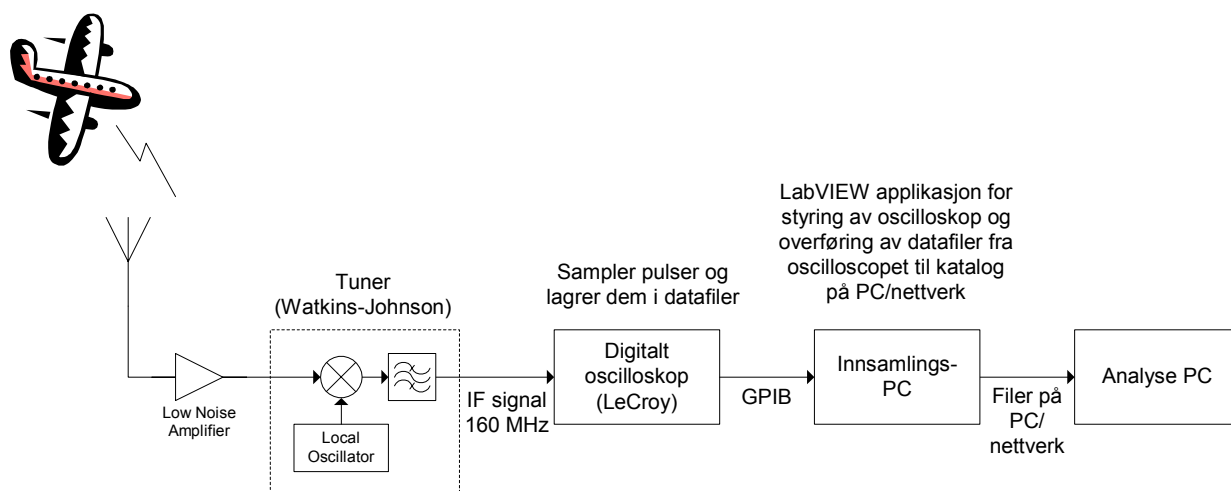
Figur 2.1 Hovedkomponentene i demonstratoren.

Figur 2.1 viser hovedkomponentene i demonstratoren. EDW-prosessering omfatter både tracking, klassifisering av radarmodell og emitteridentifikasjon basert på krystalloscillator-frekvens. Felles for disse funksjonene er at de baserer seg på informasjon i EDWene. Pulsanalyse-komponenten omfatter emitteridentifikasjon basert på pulsform, dvs pulsenvelope og instantan frekvensvariasjon (IFV).

Resten av dette kapittelet beskriver (gammel) måleoppstilling (avsnitt 2.1), input fra måleoppstilling (avsnitt 2.2) og PIA struktur og hovedideer (avsnitt 2.3). I neste kapittel beskrives innholdet i EDW-prosessering komponenten. Kapitlene 4, 5 og 6 beskriver pulsanalysen og omfatter henholdsvis preprosessering, komprimering/egenskapsuttrekking og identifikasjon. Kapittel 7 beskriver foreløpige erfaringer med PIA, og til slutt kommer kapittel 8 med en kort oppsummering.

2.1 Måleoppstilling

Figur 2.2 viser den måleoppstillingen som er brukt under opptak av de pulsfilene vi i dag bruker for uttesting av algoritmer. Samplingsraten i denne måleoppstillingen er typisk 500 MHz, og antall pulser per fil er omtrent 100-200. ”Analyse PC”-boksen i figuren tilsvarer PIA.



Figur 2.2 Måleoppstilling tidligere brukt for opptak av pulsdata.

Input til PIA er filer med pulsdata som ligger lagret på harddisk. Denne måten å overføre data på er brukt i tidligere måleoppstilling (se (1)) og vil også være mulig med ny måleoppstilling. Om andre raskere overføringsmåter (TCP/IP, Corba,...) er hensiktsmessig kan vurderes for den nye måleoppstillingen. Den nye måleoppstillingen vil bli gi bedre frekvensinformasjon enn tidligere, noe bedre nøyaktighet for amplitude (10 bit istedenfor 8 bit) og mer nøyaktig (absolutt) tidsangivelse.

2.2 Input fra måleoppstilling

I Figur 2.1 ser vi at PIA mottar to typer data, EDW-er og filer med pulssamplere. En EDW består av et relativt lite antall parametere som beskriver et pulstog. Typiske parametere er PRI, antall pulser, tidsangivelse for hver puls, antennens pekeretning, middelveier for frekvens, pulslengde og amplitude. En EDW representerer kun ett pulstog. Måleoppstillingen som produserer EDW-ene inneholder en pulssortering som skiller pulstogene fra hverandre.

Slik pulsanalysen vil bli beskrevet senere, så mottar PIA uprosesserte pulssamplere og gjør all preprocessing selv. For PIA sin del hadde det vært ønskelig om noe av denne preprocessingen ble gjort av måleoppstillingen. Måleoppstillingen har f.eks. allerede gjort pulsdeteksjon for å lage EDW-ene, og informasjon fra denne preprocessingen kunne legges ved pulsfilene. Men inntil videre antar vi altså at PIA baserer pulsanalysen kun på uprosesserte pulldata.

2.3 PIA struktur og hovedideer

PIA vil motta og prosessere alle EDW-ene som måleoppstillingen produserer. Vi kan grovt dele opp EDW-prosesseringen i tracking, modellklassifisering og identifikasjon. Foreløpig er trackingalgoritmen basert på at vi får en roterende antenne, hvor antennens pekeretning er grunnlaget for estimatet av peiling til emitteren. Vi har ingen informasjon om elevasjon eller avstand til emitteren, så den eneste informasjonen i tracket som sier noe om hvor emitteren befinner seg er peilingen.

Trackingalgoritmen oppretter nye track, oppdaterer peilingsestimatene for eksisterende track, og sletter track etter visse kriterier. For eksisterende track vil vi først foreta modellklassifisering. Det gjøres ut fra parametere i EDW-ene som PRI, pulsbredde og frekvens. For track hvor vi har foretatt modellklassifisering og i tillegg har observert at tracket skiftet mellom flere forskjellige PR-er, så kan vi forsøke å beregne emitterens krystallosillatorfrekvens. Denne sammenlignes med en database over kjente krystallosillatorfrekvenser for å angi spesifikk emitteridentifikasjon.

I tillegg til informasjon som vi utleder fra EDW-ene i EDW-prosessering modulen, så har vi muligheten for å gjøre pulsanalyse av enkeltpulser eller midlede pulser fra et pulstog. Pulssamplene vil utgjøre så store datamengder at PIA kun vil analysere en liten andel av alle pulsene. Det er naturlig at filer med pulldata derfor lagres til harddisk og at PIA leser disse pulsfilene etter behov. For å finne frem til pulsene og pulssamplene som danner grunnlaget for en gitt EDW, så kan man f.eks. bruke tidsangivelsene for pulsene. Resultatet fra pulsanalysen returneres til EDW-prosessering og assosieres til det tracket pulsen tilhører.

I de tilfellene hvor vi finner trackidentitet basert både på krystalloscillatorfrekvens og på pulsanalyse, så trenger vi en måte å kombinere resultatene fra de to metodene på. Vi vil først undersøke hvor godt de forskjellige metodene fungerer i praksis, før vi vurderer mulige datafusjonsalgoritmer.

2.3.1 UMOP

Begrepet intrapuls refererer til formen på pulsenvelopen (eller pulsmoduleringsfunksjonen) og til frekvens- eller fasevariasjoner innen pulsen. ”Unintentional modulation on pulse” (UMOP) betyr utilsiktede endringer i pulsenvelope og IFV. Hele ideen bak å gjøre identifikasjon basert på UMOP er at disse endringene er forskjellige fra emitter til emitter, og konstante over tid for en enkelt emitter.

UMOP er et resultat av en radars transmitter. Alle transmittere består av en modulator, høyspennings strømtilførsel (HVPS) og RF forsterkerenhet. De største UMOP-effekter er

- 1) Pushing effekter
- 2) Pulling effekter
- 3) Andre effekter (temperatur, gammelt utstyr, dårlig vedlikehold)

Pushing-effekter skyldes variasjoner på utgangen av modulator og HVPS. Disse variasjonene kommer til uttrykk som frekvens-, fase-, eller amplitudemodulasjon. Som eksempel vil ripple i HVPS i en magnetrontransmitter komme til uttrykk som frekvensmodulasjon i pulsen fordi frekvensen i en magnetron varierer med strømpådraget. Pulling er normalt et problem for pulsete oscillatortransmittere (magnetroner). En dårlig kobling i en roterende bølgeleder vil kunne gi frekvens- og fasevariasjoner gjennom pulsen. Dårlig vedlikehold og temperaturdrift i RF forsterkerenhet er to av flere elementer som vil påvirke pulsmodulasjonen. En radar som ikke vedlikeholdes vil kunne få drift i frekvens over tid som følge av drift i HVPS. UMOP-effekter er beskrevet i bl a (4) og (5).

2.3.2 Databaser

PIA inneholder flere typer databaser. Vi har en modelldatabase med informasjon om forskjellige radarmodeller, og for hver radarmodell i denne databasen har vi en emitterdatabase som inneholder informasjon om individuelle emittere av denne radartypen. I tillegg har vi også en kandidatdatabase. Den blir fortløpende oppdatert med pulsdata som ikke matcher noen av emitterdatabasene, dvs potensielle nye emittere. Overflytting av emittere fra kandidatdatabasen til en emitterdatabase gjøres foreløpig manuelt. Vi har også en tabell (database) med krystalloscillatorfrekvenser, som peker videre til riktig emitterdatabase.

3 EDW-PROSESSERING

Etter hvert som antennen roterer vil måleoppstillingen med jevne mellomrom oversende en samling av EDWer til PIA. Disse EDWene koder for de pulstog som er funnet siden forrige oversendelse. Det vil bli mange slike oversendelser i løpet av en antennerotasjon og EDWer fra en gitt emitter vil kunne opptre i flere påfølgende oversendelser.

3.1 Tracking

Ved å tracke emisjoner basert på type og retning vil de aller fleste emittere kunne skilles fra hverandre. Tracking er en løpende prosess der alle innkommende EDWer assosieres til eksisterende track og brukes til å oppdatere disse, eller tas vare på for senere assosiering. Track vil kontinuerlig opprettes, oppdateres, avsluttes og settes i ulike ventemodi.

Korrelering med track-liste

I dette trinnet undersøkes om den nye EDWen tilhører et av de eksisterende trackene. Dette gjøres ved først å finne det tracket som best matcher EDWen. Her sammenligner en både retningen og eventuell modell/identitet. Når en EDW matcher et av trackene i track-listen tilstrekkelig godt, så oppdateres dette trackets parametere basert på den nye EDWen.

Assosiering basert på retning

For EDWer som ikke ble assosiert til noe track blir det undersøkt om det er grunnlag for å opprette et nytt track. Først assosieres slike EDWer som kommer fra omtrent samme retning med hverandre. Kriteriet for å opprette et nytt track vil være at det kommer mange nok emisjoner fra en gitt retning (f eks 3 EDWer av høy kvalitet fra samme retning).

Dersom kriteriet for å opprette et nytt track ikke er oppfylt, så legges EDWene i et korttids-liste. Kanskje neste oversendelse av EDWer fra måleoppstillingen vil inneholde nye EDWer fra omtrent samme retning som sammen med EDWer i korttids-listen gir grunnlag for å opprette et nytt track.

Korttids-listen vil til enhver tid kun inneholde EDWer som kommer fra retninger som ligger et visst antall grader etter antennens pekeretning. Når denne terskelen er passert, så flyttes ”gamle” EDWer fra korttids-listen over i en langtids-liste. Hvor lenge en EDW kan ligge i korttids-listen før den flyttes over til langtids-listen vil avhenge av antennens rotasjonshastighet. EDWer i langtids-listen vil sammenlignes med EDWer i neste scan.

Assosiering fra scan til scan

Dersom tre EDWer kommer fra samme retning og i tillegg har andre sammenfallende karakteristikk kan disse assosieres, selv om de kommer fra to forskjellige scan.

Initiering av track

Initiering av track krever minst 3 EDW-er som kommer fra samme retning. Det første som gjøres nå er å korrelere EDW-ene mot modelldatabasen. Dersom alle EDW-ene kommer fra samme radartype/modell kan vi gå videre med identifikasjon. Identifikasjon basert på krystalloscillatorfrekvens eller pulsanalyse kan gi trackidentitet.

Selv om vi ikke finner radarmodeller i databasen som matcher PRI, pulsbredde og frekvens i EDW-ene, så kan vi opprette track som har ukjent type. Assosiering av nye EDW-er til dette tracket kan f.eks. basere seg på konstant PRI (fiskebåt-radar), PRF-mønster, eller krystalloscillatorfrekvens.

Oppdatering av track

For oppdatering av peilingsestimatet for et track brukes et Kalmanfilter. Nytt estimat av peiling beregnes på bakgrunn av de nye EDW-ene som assosieres til tracket, og trackets tidligere peilingsestimat. Hvis vi plottes amplituden til de nye EDW-ene som funksjon av antennens pekeretning, så vil vi typisk få en kurve som kan ligne på en gaussfordeling, med et maksimumspunkt og jevnt avtagende verdier til begge sider. Vi approksimerer denne kurven med et annengrads polynom, og bruker polynomets maksimumspunkt som input til Kalmanfilteret.

Trackets identitet vil også kunne kontrollsjekkes med jevne mellomrom, ved å gjøre pulsanalyse på utvalgte pulser tilhørende tracket.

3.2 Klassifisering av radarmodell

Klassifisering av radarmodell er basert på parametere som

- bærefrekvens
- PRI
- pulsbredde (-middelverdi)

Disse er parametere som karakteriserer pulstog fra en gitt radarmodell. Vi kaller disse parametere for modellparametere. Disse parametere hentes fra de mottatte EDW-ene.

Klassifiseringen gjøres ved korrelasjon mot en modelldatabase. I modelldatabasen lagres modellparametere for de forskjellige radarmodellene. I tillegg lagres også navn på radarmodell og eventuelt mode. En radarmodell med flere forskjellige moder vil representeres i modelldatabasen med ett element for hver mode.

I operativ sammenheng kan det også være aktuelt å sette sammen en modelldatabase (med tilhørende emitterdatabaser) avhengig av den forventede trusselen i området. Det vil også bidra til å gjøre korrelasjon/identifikasjon mindre regnekrevende.

3.2.1 PRF-mønstergjenkjenning

I mange radarer er det vanlig å benytte et sett av ulike PRF-er for å løse opp mål i avstand og hastighet. En typisk måte å gjøre dette på er ved å anvende 5-10 ulike PRF-er som dekker ca en oktav (typisk 10-20 kHz). Det er i tillegg vanlig at PRF-ene gjennomløper et fast mønster. Ved å finne dette mønsteret er det mulig å entydig bestemme radartypen. I tillegg til dette vil det være mulig å tracke slike PRF-mønster når de først er detektert. To radarer av samme type kan da lett skilles i og med at de tilhørende PRF-mønstre ikke vil være synkroniserte i tid. Kompleksiteten i å kunne gjøre automatisk mønstergjenkjenning av slike MPRF-mønster øker drastisk når vi har flere emittere, eller flere ulike MPRF-mønster. I tillegg vil mønstrene kunne forsvinne når radarens sidelover kommer under deteksjonsterskelen til ESM-mottakeren. Metoder for å gjøre slik automatisk mønstergjenkjenning er beskrevet i større detalj i (6).

3.2.2 Korrelering mot modelldatabase

Korrelasjon mellom modellparametere gjøres på en svært enkel måte. For hver av parameterne har vi en terskel som angir hvor stort avvik vi kan tolerere fra de lagrede verdier i modelldatabasen

FC_THR	terskel for avvik mellom bærefrekvenser (f eks 1 MHz)
PW_THR	terskel for avvik mellom pulsbredden (f eks 10 ns)
PRF_THR	terskel for avvik mellom PRF-er (f eks 1%)

Hvis alle avvikene er innenfor de gitte tersklene, så velges radarmodellen ut som en mulig kilde til pulstoget. Geometrisk kan de lagrede modellparametere for en gitt radarmodell betraktes som et punkt i et tre-dimensjonalt rom. De tre tersklene definerer en rektangulær boks sentrert i dette punktet. De målte modellparametere må ligge innenfor denne boksen for at radarmodellen skal velges. Merk at avvik i PRF måles relativt (i %), så størrelsen på boksene i PRF-dimensjonen øker med økende PRF.

I første omgang vil vi bruke de samme terskelverdiene for alle radarmodellene. Om nødvendig kan dette lett endres til at vi har forskjellige terskelverdier for hver radarmodell. For raskest mulig å finne de elementene i modelldatabasen som gir best korrelasjonen kan det også være aktuelt å sortere modelldatabasen på en av modellparametere. På denne måten vil vi kunne redusere antall databaseelementer vi må korrelere mot.

3.2.3 Oppdatering av modelldatabase

Oppdatering av modelldatabasen gjøres manuelt. Modellparametere fra EDWer som vi ikke klarer å finne radarmodell til, tas vare på for senere manuell analyse.

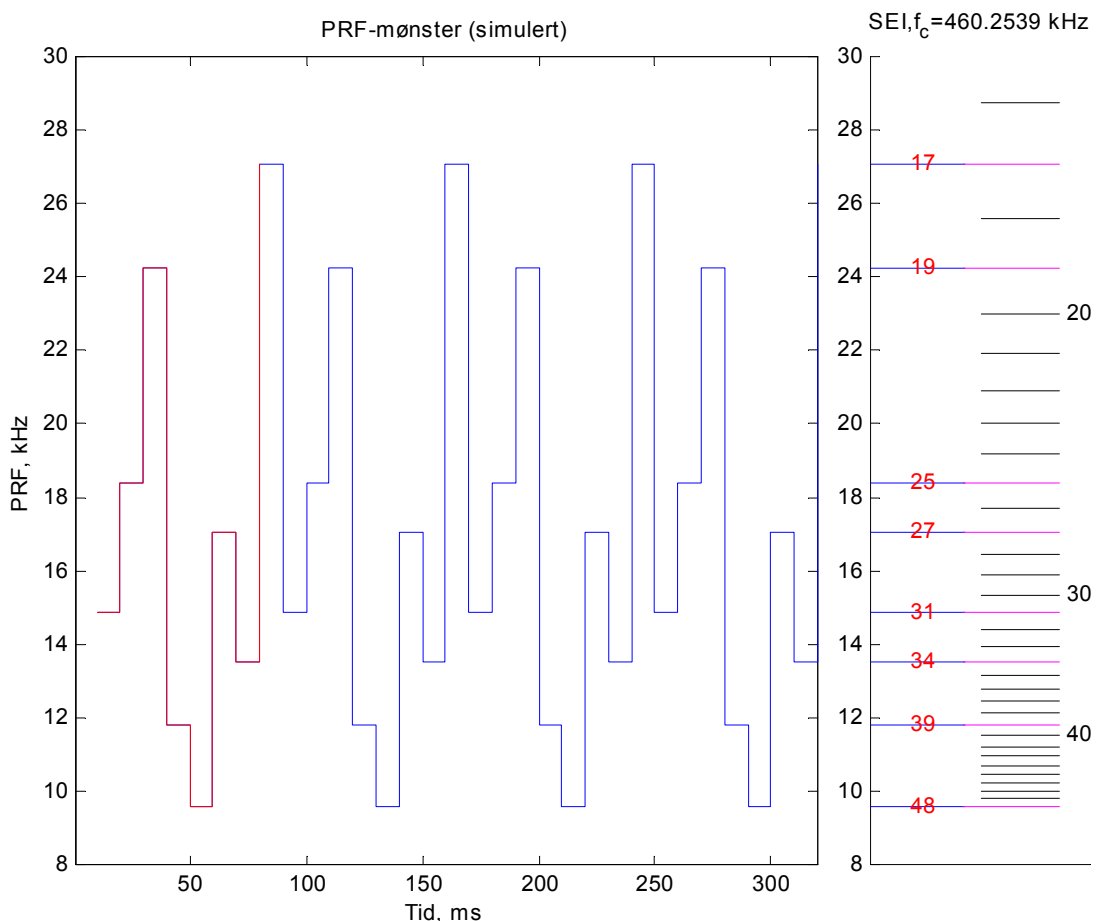
3.3 Identifikasjon vha krystalloscillatorfrekvens

En radar vil alltid avlede sine PRF-er fra en gitt krystall-oscillatorfrekvens f_c . Gitt at radaren har N PRF-er ($PRF_1, PRF_2, \dots, PRF_N$) så vil alle disse være gitt ved

$$PRF_n = \frac{f_c}{k_n} \text{ der } k_n \in \mathbb{N} \quad (3.1)$$

Ved en ESM-måling av kun en PRF vil det derfor ikke være mulig å utlede krystaloscillatorfrekvensen f_c siden divisoren k_n fortsatt er ukjent. Dersom vi har to eller flere PRFer som vi mistenker er avledet fra samme krystaloscillatorfrekvens så kan dette påvises ved å prøve å finne minste felles multiplum av disse. Nøyaktigheten vi kan måle krystaloscillatorfrekvensen med ved denne fremgangsmåten er tilstrekkelig til ikke bare å bestemme ulike radartyper; den er også god nok til å kunne skille krystall fra radarer av samme type. I Figur 3.1 vises et eksempel på sammenhengen mellom et PRF-mønster og hvordan dette mønsteret er avledet av krystaloscillatorfrekvensen.

Når en krystaloscillatorfrekvens først er funnet kan denne trackes over tid. Dette kalles SET eller spesifikk emitter tracking. En ny EDW kan raskt korreleres med tracket av krystaloscillatorfrekvensen. Betingelsen for at EDWen er en del av tracket blir da at dets PRF er en underharmonisk av f_c , som i ligning (3.1).



Figur 3.1 Spesifikk emitter identifikasjon av et simulert PRF-mønster. Kurven til venstre viser fire gjennomløp av mønsteret. Til høyre er hver målt PRF markert med blått. Til høyre for disse er alle mulige PRF-er som kan avledes av oscillatorfrekvensen $f_c = 460.2539$ kHz markert. De målte PRF-ene og de teoretiske samsvarer eksakt.

3.4 Styring av pulsanalysen

Hvis man etter modellklassifiseringen velger å gå videre med identifikasjon basert på pulsanalyse, så brukes de utvalgte radarmodeller for å velge ut hvilke emitterdatabaser som skal brukes. For hver radarmodell i modelldatabasen har vi en tilhørende emitterdatabase som inneholder pulsparametere for forskjellige emittere av akkurat denne radarmodellen. Vi har altså kun en modelldatabase, men mange emitterdatabaser.

Hovedhensikten med en oppdeling i modell- og emitterdatabaser er å redusere antall emitterdatabaser man korrelerer mot. Antall emitterdatabaser kan bli svært stort, så det er viktig å gjøre korreleringen mot færrest mulig emitterdatabaser, og på en rask og effektiv måte.

Under identifikasjonen vil pulsparametere korreleres mot alle elementene i de utvalgte emitterdatabasene. Ideelt sett vil vi da få god korrelasjon mot kun ett av elementene i en av emitterdatabasene, og dårlig korrelasjon mot de andre.

Korrelasjon av pulsparametere mot emitterdatabasen er såpass tidkrevende at det ikke vil gjøres for alle pulser i alle mottatte pulstog. Pulsanalysen vil kun bli foretatt for en liten andel av pulsene. Dette trenger ikke være noen vesentlig begrensning – det burde være unødvendig å sjekke alle pulsene fra et track som vi allerede kjenner identiteten til, det burde holde å gjøre kontrollsjekker av og til. Det er viktigere å bruke ressursene på å gi rask identifikasjon av eventuelle nye track. Kriteriene for å gjøre pulsanalyse på utvalgte pulser vil være en funksjon av tiden siden siste sjekk av track identitet og usikkerheten knyttet til identifikasjonen.

4 PULSDETEKSJON OG PREPROSESSERING

Innlesing av pulsfil, pulsdeteksjon og preprosessering gjøres av programmodulen PEPP (Pulse extraction and preprocessing). PEPP kan lese filer fra LeCroy digitale oscilloskop, Condor CS-D-250 digitizer og FFIs Synnøve BAD filer. PEPP vil bli utvidet slik at den også kan lese pulsfilene fra den nye måleoppstillingen.

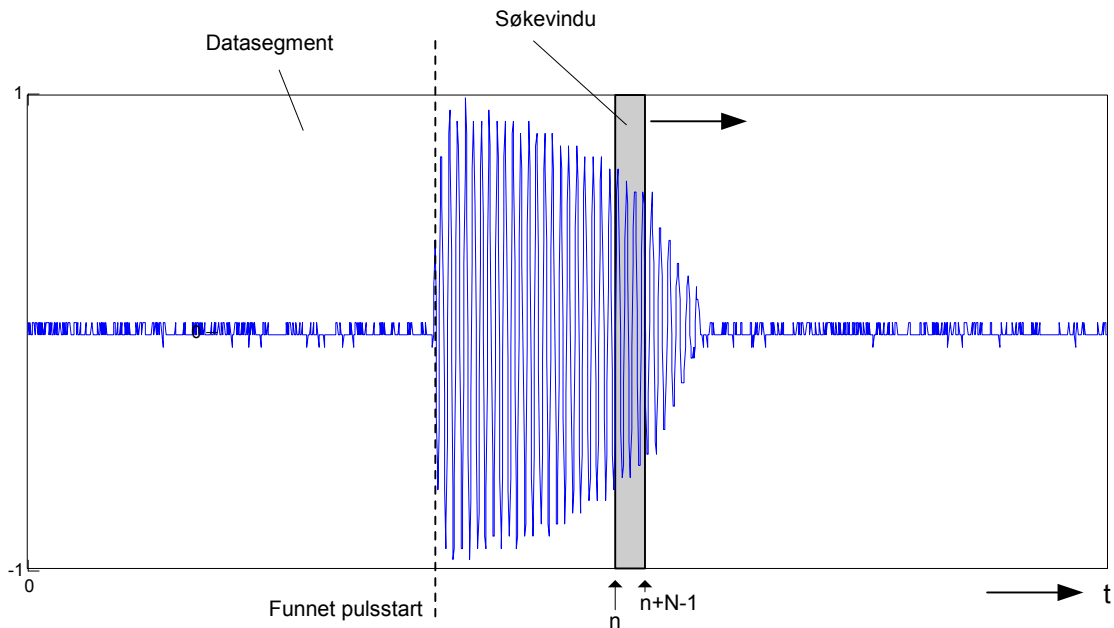
En pulsfil inneholder bl a:

- Samplingsfrekvens
- Antall pulser i filen
- Estimert SNR for hver puls
- Antall sample for hver puls
- Selve pulssamplene for hver puls
- Tidspunkter for alle pulssamplene

Pulsfilene inneholder også informasjon om innstillinger på mottakerutstyret, dato og tidspunkt for når filen ble lagret, samt en rekke andre parametere. Vi nøyer oss med å beskrive data som inngår direkte i identifikasjonsalgoritmene og som er uavhengig av måleoppstilling.

4.1 Pulsdeteksjon

Deteksjon og uttrekking av en puls i et datasegment x blir gjort med energiterskling som prinsipp. Signalet i datasegmentet blir først normalisert og middelveiden fjernet. PEPP søker gjennom dette segmentet etter en pulsstart og pulsstopp. Dette blir gjort ved å måle gjennomsnittsenergien $E(n)$ i et søkevindu. Dette søkevinduet blir flyttet i diskrete steg gjennom datasegmentet, se Figur 4.1.



Figur 4.1 Deteksjon av puls i et datasegment.

Deteksjonen styres av tre parametere:

<code>PULSE_START_THR</code>	Energiterkel for å finne pulsstart ($\in [0,1]$)
<code>PULSE_STOP_THR</code>	Energiterkel for å finne slutten på pulsen ($\in [0,1]$)
<code>WINDOW_SIZE</code>	Størrelsen på energivinduet (antall sampler)

Gjennomsnittsenergien $E(n)$ er definert som

$$E(n) = \frac{1}{N} \sum_{i=n}^{n+N-1} |x(i)|^2 \quad (4.1)$$

hvor n er samplenummer og N er antall sampler i søkevinduet, dvs $N = \text{WINDOW_SIZE}$. Når denne energien overstiger terskelen `PULSE_START_THR` blir samplet i midten av søkevinduet merket som starten av pulsen. Når energien faller under terskelen `PULSE_STOP_THR` blir samplet i midten av søkevinduet merket som slutten av pulsen.

Pulsdeteksjon er implementert som et to-pass søk. Første passet gjør et grovt søk og bruker store steg i flyttingen av søkevinduet. Det andre passet finner start og stopp mer nøyaktig ved å søke med mindre forflytninger i området hvor start og stopp ble funnet i det første passet. Denne todelingen er gjort for å øke hastigheten på søket.

UTVELGELSE AV "GODE" PULSER

PEPP sjekker også pulsens "godhet" ut fra følgende seks parametere:

<code>MINIMUM_LENGTH</code>	Minimum pulslengde (i antall sampler)
<code>MINIMUM_SNR</code>	Minimum SNR (dB)

AMPLITUDE_CLIP_THR	Maksimum antall klippede sampler (pr. puls)
LOWER_LIMIT	Laveste sample-verdi (avhenger av filformat, f eks -128)
UPPER_LIMIT	Høyeste sample-verdi (avhenger av filformat, f eks 127)
END_CLIP	Forkast pulser med mulig endeklipping

Med endeklipping menes det at ikke hele pulsens start eller stopp har kommet med i datasegmentet. Pulsene anses for å være tilstrekkelig ”gode” hvis ingen av grensene definert ovenfor blir brutt.

PEPP estimerer pulsens SNR på følgende måte:

$$SNR = 10 \log_{10} \left(\frac{e_{signal} - e_{noise}}{e_{noise}} \right) \quad (4.2)$$

hvor e_{signal} og e_{noise} beregnes på bakgrunn av henholdsvis pulssampler og støyssampler som angitt i (4.1). Som støysampler brukes området i datasegmentet før pulsens start (evt etter pulsens slutt dersom det er for få sampler før pulsens start).

4.2 Beregning av envelope og IFV

Vi ønsker å finne ut pulsenvelope og IFV for hver enkelt puls. Det vil vi gjøre ved å bestemme amplituden $A(t)$ og fasen $\varphi(t)$ fra samplene i pulsen. Signalet (pulsens) er representert ved funksjonen $f = f(t)$ hvor f er på formen

$$f(t) = A(t) \cos(2\pi \varphi(t)). \quad (4.3)$$

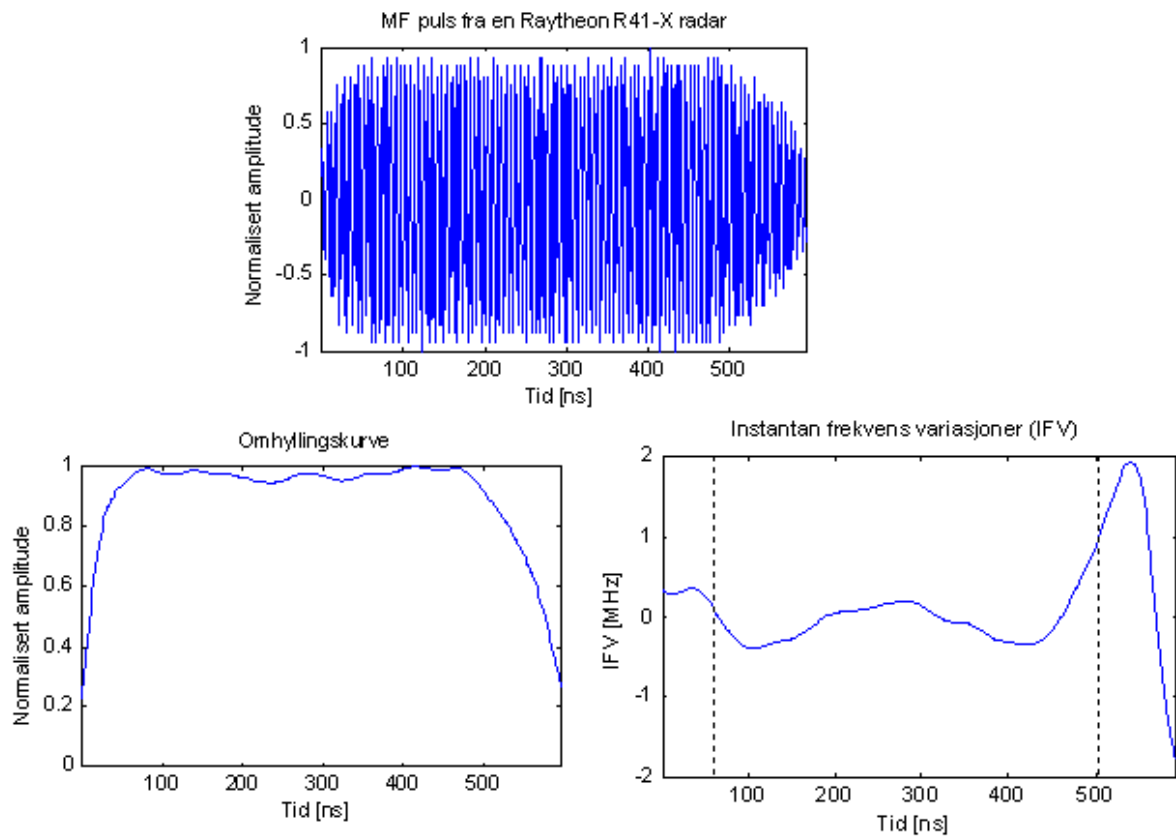
Gitt signalet $f(t)$ bruker vi Hilberttransformen H til å lage en kompleks versjon av signalet på formen

$$f_c(t) = f(t) + jH(f(t)). \quad (4.4)$$

Nå kan amplituden $A(t)$ og fasen $\varphi(t)$ bestemmes ved hjelp av formlene

$$A(t) = \sqrt{f(t)^2 + H(f(t))^2}, \quad \varphi(t) = \arctan \frac{H(f(t))}{f(t)}. \quad (4.5)$$

Amplituden $A(t)$ representerer envelope (omhyllingskurven) til pulsen og den deriverte av fasen $\varphi(t)$ gir oss IFV. For å fjerne støy i både envelope og IFV filtreres signalene med et lavpassfilter. For å minimere endeeffekter reverseres signalet etter filtrering og sendes tilbake gjennom filteret.



Figur 4.2 En puls og dens envelope (omhyllingskurve) og IFV.

5 PULSREPRESENTASJON

Etter at vi har lest inn pulldata, gjort pulsdeteksjon og preprosessering, så har vi representert en puls i form av pulsenvelope og IFV. Disse er begge to representert ved vektorer med like mange datapunkter som det er sampler. Antall sampler per puls kan f.eks. være omtrent 1000 (samplingsfrekvens 500 MHz og pulslengde ca $2 \mu s$). Å lagre pulser representert på denne måten vil ta mye plass og sammenligning av pulser vil være unødig regnekrevende. En reduksjon i antall datapunkter er helt nødvendig.

Datareduksjon/komprimering må gjøres på en måte som fjerner redundant informasjon, men som bevarer egenskaper som er egnet for identifikasjon, dvs. egenskaper som er like for pulser fra samme emitter, men forskjellig for pulser fra forskjellige emittere. Vi vil se på følgende tre metoder:

- 1) Desimering; redusering av datamengden ved å fjerne datapunkter ved hjelp av mer eller mindre velfunderte utvelgelsesmetoder.
- 2) Tilpasning; redusering av datamengden ved bestemmelse av koeffisientene til et sett av på forhånd valgte basisfunksjoner.
- 3) Egenskapsuttrekking; representasjon ved parametere som beskriver bestemte karakteristiske egenskaper.

Vi ser nærmere på hvert enkelt av disse områdene, ulike metoder som hører inn under disse og fordeler og ulemper ved disse metodene. Senere vil vi vise hvordan de tre metodefamiliene kan anvendes til å representere radarpulser ved relativt små mengder data.

5.1 Desimering og splines.

Dataopptak med moderne utstyr har en svært høy samplingsfrekvens. I mange tilfeller er samplingsfrekvensen langt høyere enn frekvensen i det innkommende signalet. Hvis dette er tilfelle, kan en del av dataene fjernes for å redusere datamengden uten at kvaliteten på det gjenværende signalet er nevneverdig forringet sammenlignet med det opprinnelige. Dette er desimering. Det er flere måter å gjøre dette på, for eksempel kan vi bestemme at en viss andel, for eksempel halvparten, av alle data skal fjernes, uavhengig av hvordan signalet ser ut. Vi kan også legge andre kriterier til grunn for å fjerne data som vi anser som mindre viktige. Vi vil også se på bruk av mer avanserte metoder for å rekonstruere det opprinnelige signalet, metoder som ikke krever at mye data lagres, dvs. desimering basert på visse kriterier som maksimerer evnen til signalrekonstruksjon.

Kombinasjonen av desimering og splines kan forbedre ytelsen til metoder som bruker desimering. Splines er en effektiv metode til å representere mer eller mindre kompliserte funksjonsformer ved få datapunkter. Generelt er en spline-approksimasjon av orden n en funksjon som er et n 'te grads polynom i intervallet mellom datapunktene og som er deriverbar

$n-1$ ganger i datapunktene. Disse kalles skjøtepunkter siden ulike polynomer skjøtes sammen her. Interpolanter er dermed spline-approksimasjoner av grad 1. Spline-approksimasjoner av grad 2 er andregradspolynomer som er limt sammen i datapunktene og som er kontinuerlige og deriverbare i disse punktene. Kubiske splines (spline-approksimasjoner av grad 3) er to ganger deriverbare i skjøtepunktene.

5.2 Komprimering ved tilpasning

En forsker som har utført eksperimenter ønsker ofte å bestemme en funksjonell relasjon som resultatene fra eksperimentene passer til. En vanlig fremgangsmåte er å trekke den rette linjen som best passer til dataene, basert på minimering av summen av kvadratene til avvikene. Denne linjen som utgjør en modell av dataenes oppførsel og kan derfor ses på som en approksimasjon eller representasjon av dataene.

Bestemmelse av en representasjon av en puls utgjør et tilsvarende problem: Vi har et sett av datapunkter og ønsker å bestemme en funksjonell relasjon som er en god modell for disse dataene. I et matematisk språk vil dette si å bestemme den beste approksimasjon i et gitt rom av funksjoner. Sett at vi har dataene (x_k, y_k) , $k = 1, \dots, N$, og vi ønsker en representasjon med en funksjon fra funksjonsrommet Φ . Da får vi et optimaliseringsproblem (her: et minimaliseringsproblem) av denne typen;

$$\min_{f \in \Phi} \sum_{k=1}^N (y_k - f(x_k))^2, \quad (5.1)$$

hvor funksjonen f som oftest vil være en lineær kombinasjon av basisfunksjoner f_k fra det endeligdimensjonale funksjonsrommet Φ , dvs. $f = \sum_{k=1}^N a_k f_k$. Da finner vi funksjonen f ved at koeffisientene a_k bestemmes i optimeringsproblemet (5.1).

5.2.1 Tilpasning ved polynomer

En analytisk funksjon f kan approksimeres vilkårlig godt i en omegn av et punkt x_0 med Taylor-polynomet i dette punktet. Hvor god approksimasjonen er, avhenger av hvor mange ledd som tas med i Taylor-rekken. Generelt kan vi, innenfor et endelig intervall, approksimere uniformt kontinuerlige funksjoner med polynomer, hvor approksimasjonens nøyaktighet avhenger av antall ledd.

I praksis vil vi ofte ikke ønske å ha tilnærminger til funksjoner, men representasjoner av et endelig antall data, observasjoner eller sampler. Dette gir oss følgende problemstilling: Gitt dataene y_k i punktene x_k , $k = 1, \dots, N$ ønsker vi å bestemme koeffisientene til et polynom $p = p(x)$ av grad m slik at summen av kvadratene til avvikene blir minimalisert;

$$\min_{p \in P_m} \sum_{k=1}^N (y_k - p(x_k))^2. \quad (5.2)$$

Her er P_m funksjonsrommet av alle polynomer av grad m . I ligning (5.2) brukes minste kvadraters metode, men andre kriterier for valg av koeffisienter kunne også ha vært brukt, for

eksempel minimering av maksimalt avvik eller summen av tallverdien til avvikene i datapunktene.

5.2.2 Tilpasning ved wavelets

Innen tradisjonell signalteori er Fourier-transformen blitt brukt til å generere matematiske representasjoner av kompliserte signaler. Wavelet-transformen gir, i motsetning til Fourier-transformen, oppløsning i både tid og frekvens, se f eks (7). Vi skal se hvordan wavelets representerer en elegant metode for effektiv representasjon og komprimering av signaler.

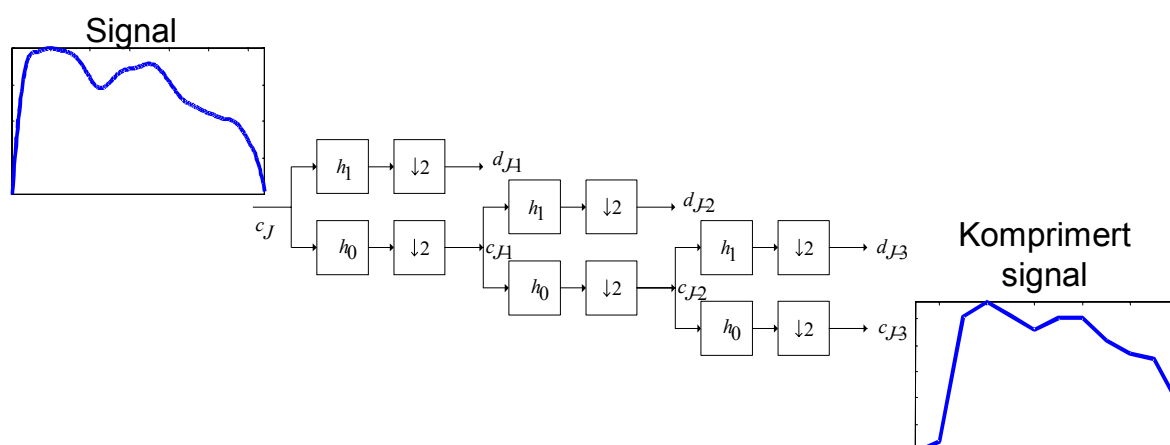
Vi tenker oss at vi har et signal (eller en funksjon) $f(t)$. Vi ønsker en representasjon av signalet i et lite antall koeffisienter fremfor et stort antall sampler. Dette kan gjøres på følgende måte. Vi antar at signalet kan representeres som en sum av de lavfrekvente (approximasjonen) og de høyfrekvente (detaljene) delene av signalet, på følgende måte;

$$f(t) = \underbrace{\sum_k a_{J_{\max}}(k) \varphi_{J_{\max},k}(t)}_{\text{Approximasjon}} + \underbrace{\sum_{j=1}^{J_{\max}} \sum_k d_j(k) \psi_{j,k}(t)}_{\text{Detalj}}. \quad (5.3)$$

Her er funksjonene $\varphi_{j,k}$ og $\psi_{j,k}$ definert ved

$$\begin{aligned} \varphi_{j,k}(t) &= 2^{-j/2} \varphi(2^{-j}t - k) \\ \psi_{j,k}(t) &= 2^{-j/2} \psi(2^{-j}t - k) \end{aligned} \quad (5.4)$$

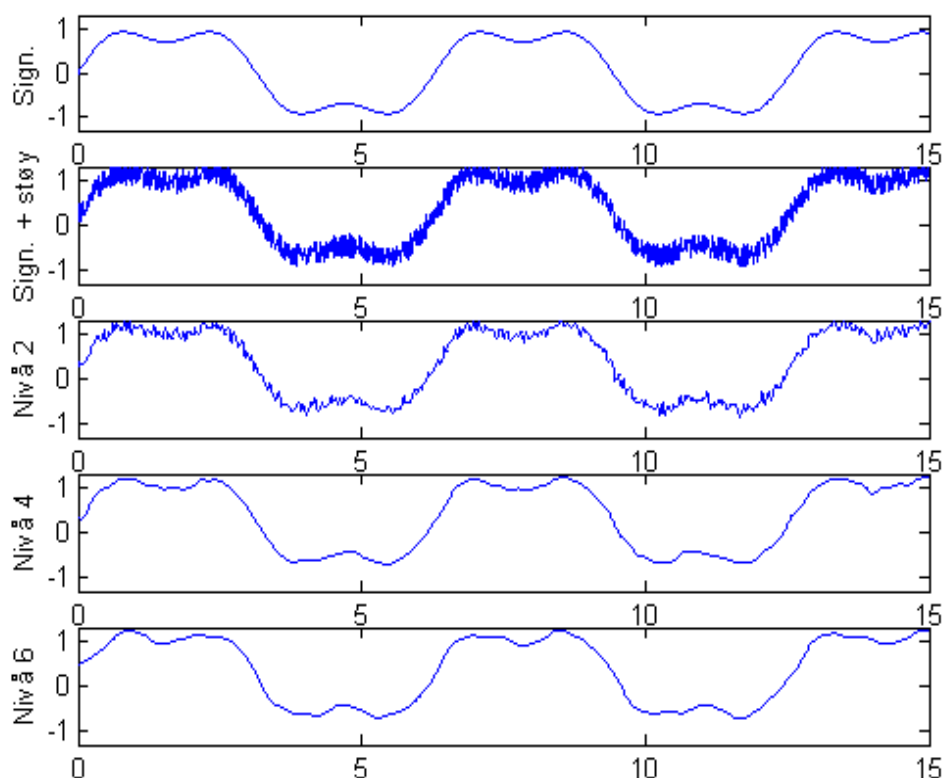
Funksjonene ϕ og ψ vil bli forklart nedenfor. Parametriseringen av tiden ved indeksen k og av skaleringen ved indeksen j , har vist seg å være svært effektiv. Varierende verdier av k fører til at wavelet-representasjonen flytter seg langs tidsaksen, mens j styrer bredden og høyden til wavelet-representasjonen. Dette gjør det mulig å analysere detaljene i signalet og representere (approximere) signalet ved et relativt lite antall approximasjonskoeffisienter.



Figur 5.1 Wavelet-dekomponering med 3 nivåer.

Ved representasjon av signalet er vi interessert i koeffisientene $a_{J_{\max}}$ og d_j fra ligning (5.3). Disse bestemmes ved følgende algoritme. Signalet sendes gjennom filtrene h_0 (høypass) og h_1 (lavpass) og desimeres så ved at annenhver sample fjernes, dvs de filtrerte signalene sendes

gjennom desimeringsfilteret $\downarrow 2$. Dermed har vi to signaler, hvert med halvparten så mange sampler som det opprinnelige, det ene inneholder den høyfrekvente halvdelen av signalet (detaljene), det andre den lavfrekvente halvdelen (approximasjonen). Approximasjonen tas vare på, sendes så gjennom de samme filtrene en gang til, den neste approximasjonen tas vare på og prosedyren gjentas osv. Figur 5.1 viser komprimering med 3 nivåer. Det komprimerte signalet består nå av en vektor inneholdende et antall koeffisienter som er mye lavere enn det opprinnelige antallet sampler. I tilfellet med tre nivåer er antallet koeffisienter ca. $1/2^3 = 1/8$ av antallet sampler i signalet. Dersom antallet nivåer er n , så er antall wavelet-koeffisienter tilnærmet lik opprinnelig antall sampler dividert med 2^n . Figur 5.2 viser approximasjon på tre ulike nivåer bestemt ved bruk av wavelets.



Figur 5.2 Eksempel på approximasjon og støyreduksjon av signal ved bruk av wavelets. De to øverste panelene viser henholdsvis opprinnelig signal ($\sin(t) + 0.3\sin(3t)$) og det samme signalet med pålagt støy. De tre neste panelene viser wavelet-approximasjonene ved nivå 2, 4 og 6, respektive. Antall koeffisienter er henholdsvis 380 (nivå 2), 100 (nivå 4) og 30 (nivå 6). Antall sampler i det opprinnelige signalet er 1500. Her er det brukt db4-wavelets.

Den matematiske teorien som ligger til grunn for wavelets bygger på multiresolusjon, dvs oppløsning av signaler i flere skalakomponenter. Hovedideen ved denne teorien er å dele opp funksjonarommet $L^2(\square)$ i en følge av underrom som er generert av en skaleringsfunksjon φ hvor følgende betingelser er oppfylt.

1. $\{\phi(t-n)\}_{n=-\infty}^{\infty}$ danner en ortonormal basis for V_0 .
2. (i) $\dots \subset V_k \subset V_{k+1} \subset \dots$, (ii) $\bigcap_{k=-\infty}^{\infty} V_k = \{0\}$ og (iii) $\overline{\bigcup_{k=-\infty}^{\infty} V_k} = L^2(\square)$.
3. (i) $f(t) \in V_k \Leftrightarrow f(2^{-k}t) \in V_0$ og (ii) $f(2^k t) \in V_k \Leftrightarrow f(t) \in V_0$.

Hva slags funksjoner er ϕ og ψ ? Siden ϕ er i V_0 og $V_0 \subset V_1$ så kan vi uttrykke ϕ som en lineær kombinasjon av basisfunksjonene i V_1 . Disse basisfunksjonene er $\sqrt{2}\phi(2t-k)$. Dermed får vi skaleringsligningen

$$\phi(t) = \sum_k h_0(k) \sqrt{2} \phi(2t-k). \quad (5.5)$$

Et lignende, men mer komplisert, argument gir waveletligningen

$$\psi(t) = \sum_k h_1(k) \sqrt{2} \phi(2t-k). \quad (5.6)$$

Når man er interessert i approksimasjonen må man kjenne filteret h_0 . Hvordan ser så komponentene i filteret h_0 ut? Dette kan vi finne ut av ved å multiplisere skaleringsligningen (5.5) med $\sqrt{2}\phi(2t-k)$, integrere og så bruke at $\sqrt{2}\{\phi(2t-k)\}_{k=1}^N$ utgjør en ortonormal basis for V_1 :

$$\sqrt{2} \int_{-\infty}^{\infty} \phi(t) \phi(2t-k) dt = 2 \sum_{l=1}^N h_0(l) \int_{-\infty}^{\infty} \phi(2t-l) \phi(2t-k) dt = h_0(k). \quad (5.7)$$

Komponentene i filteret kan altså beregnes ved å løse integralet i ligning (5.7).

5.3 Egenskapsuttrekking

Den metoden som har størst potensial med hensyn til datareduksjon er utvilsomt egenskapsuttrekking. Kort fortalt vil representasjon ved egenskapsuttrekking si at vi representerer et sett av data ved noen få størrelser som vi antar er karakteristiske for hvert datasett og som dermed kan brukes som en identifikasjonsnøkkel for dette datasettet.

Så langt har vi stort sett tatt for oss egenskapen ”form” ved en puls. Denne egenskapen kan ikke karakteriseres ved få parametere, generelt må et stort antall parametere lagres for at vi skal kunne beskrive pulsformen. Kunne vi ha nøyd oss med å la et mye mindre antall karakteristiske egenskaper ved pulsen representere den uten at dette går utover muligheten til gjenkjenning av spesifikke emittere? Og i så fall, hvilke egenskaper kan regnes som karakteristiske? De mest påfallende karakteristika ved amplitude- eller envelope-formen er stigetid og falltid som er mål på hvor raskt amplituden stiger fra puls-start til amplituden er på sitt terrassenivå og tilsvarende, tiden fra dette nivået til puls-stopp. Andre karakteristiske egenskaper kan være antall lokale maksima og minima i envelope og frekvens.

5.4 Eksempler på komprimert pulsrepresentasjon

I dette avsnittet vil vi vise eksempler på komprimering av pulsenvelope og IFV vha de metoder som nettopp er beskrevet. I eksamplene har vi brukt en (tilfeldig valgt) pulsform fra værradaren

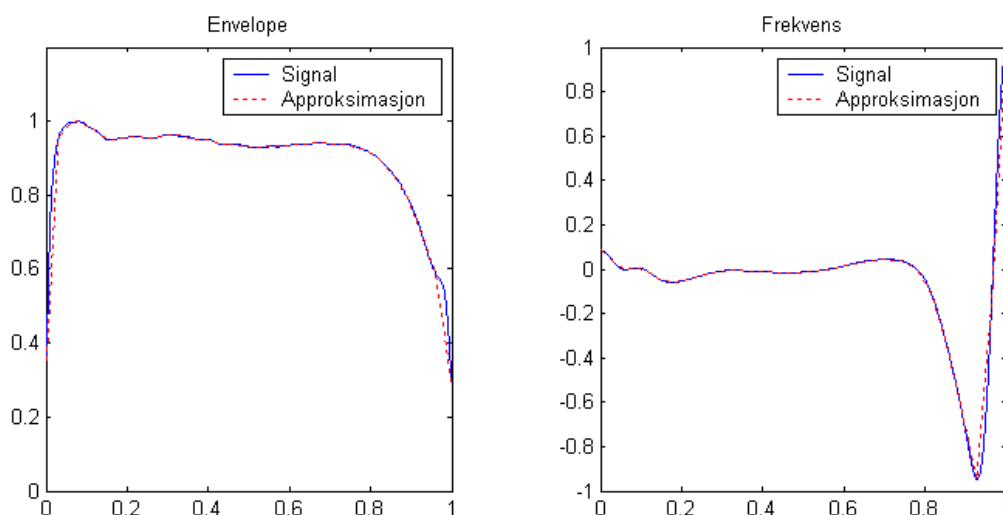
til et sivil passasjerfly på vei til eller fra Oslo Lufthavn, Gardermoen. Opptaket er gjort av en passiv sensor (Figur 2.2) ved FFI, Kjeller. Pulsenvelope og IFV er beregnet for et gjennomsnitt av 100 pulser i et pulstog, og preprosesseringen er som beskrevet tidligere (glattingsfilteret har 10 filterkoeffisienter). Antall datapunkter i både pulsenvelope og IFV er 1043.

5.4.1 Desimering

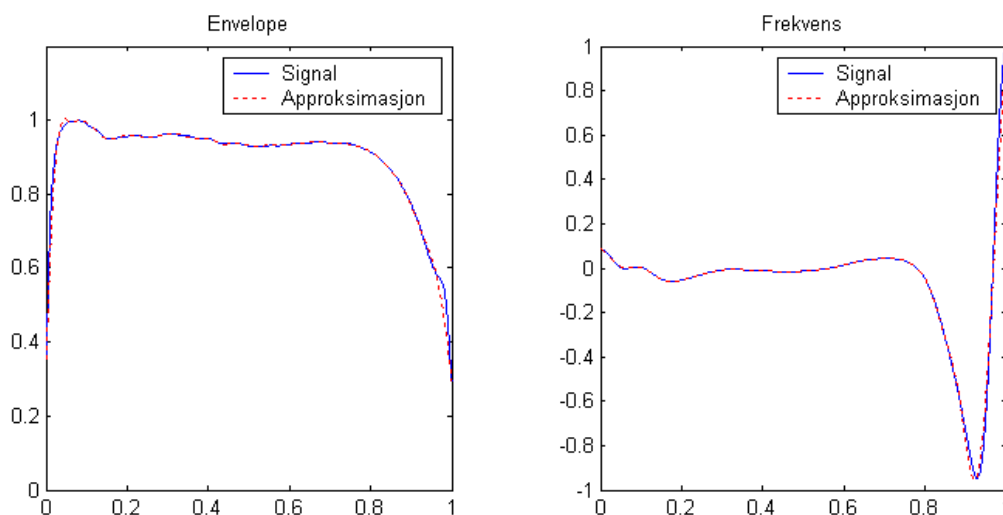
Vi ser først på hvor gode approksimasjoner som kunne genereres ved desimering. En viktig problemstilling ved desimering er hvilke datapunkter som skal beholdes og hvilke som skal fjernes. For at de mest sentrale egenskapene ved pulsformen skal bli best mulig bevart, velger vi i utgangspunktet å beholde endepunktene samt alle lokale maksimums- og minimumspunkter. Dette kan gi uheldige utslag dersom avstanden mellom lokale maksima og minima er svært ujevn. Vi vil derfor fjerne og legge til punkter etter følgende prosedyre:

1. Dersom avstanden mellom to nabopunkter blir mindre enn en på forhånd valgt verdi ε_1 , så blir ett av disse punktene fjernet
2. Dersom avstanden mellom to nabopunkter er større enn en på forhånd valgt verdi ε_2 , så legges ett eller flere nye punkter til mellom nabopunktene.

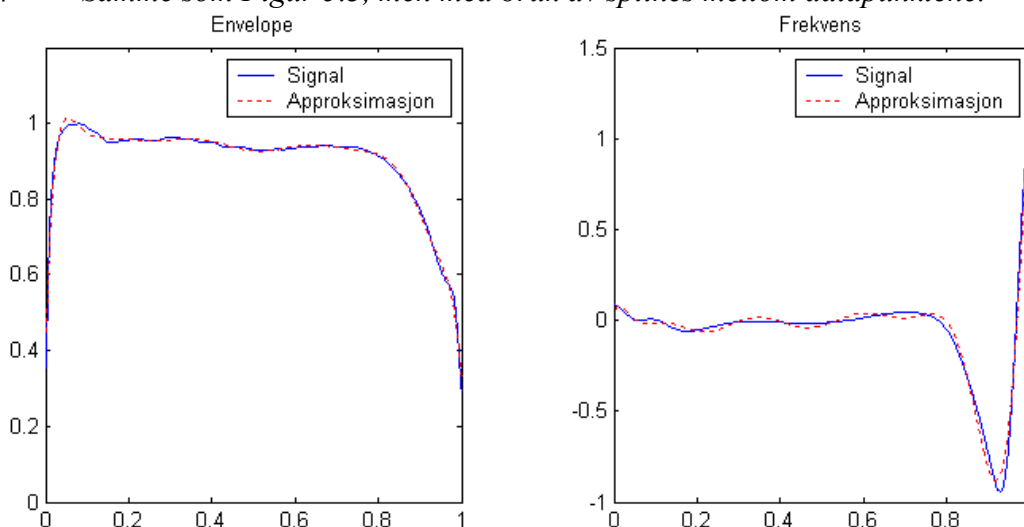
Metoden ovenfor ble utprøvd for pulsrepresentasjon. Desimeringer av envelope og IFV er vist i Figur 5.3 og Figur 5.4. I Figur 5.3 er den desimerte representasjonen stykkevis lineær, dvs rette linjer er strukket mellom datapunktene. I Figur 5.4 har vi benyttet kubiske splines mellom datapunktene for å bedre approksimasjonen i forhold til pulsformen. I begge tilfeller er profilen representert ved 32 datapunkter, en betydelig reduksjon i forhold til mengden data i den opprinnelige pulsformen. Vi observerer at bruk av splines forbedrer tilnærmingen betraktelig uten å øke datamengden i representasjonen. I både Figur 5.3 og Figur 5.4 valgte vi verdiene $\varepsilon_1 = 0.02$ og $\varepsilon_2 = 0.04$.



Figur 5.3 Envelope og frekvens (blått) og approksimasjon (rødt) bestemt ved desimering. Approksimasjonen er representert ved 32 punkter for hver profil.



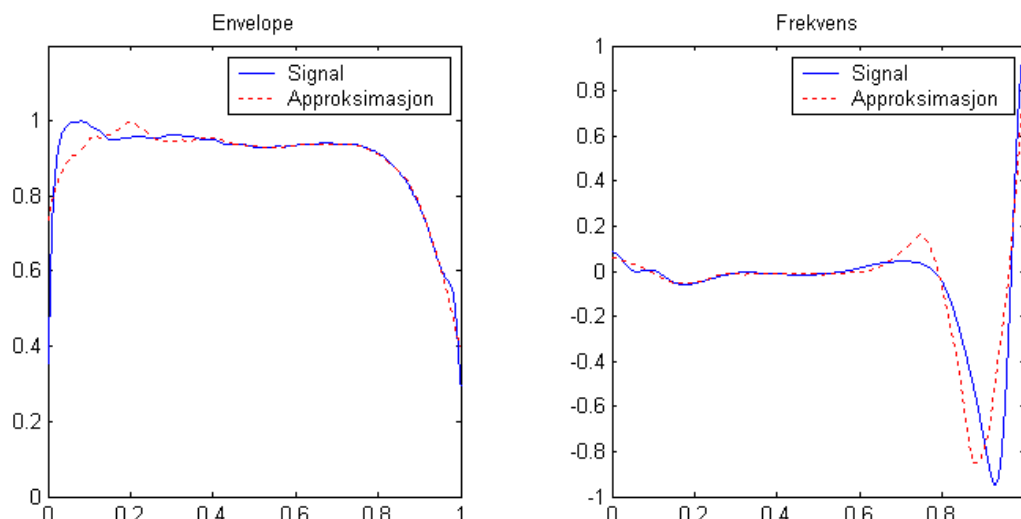
Figur 5.4 Samme som Figur 5.3, men med bruk av splines mellom datapunktene.



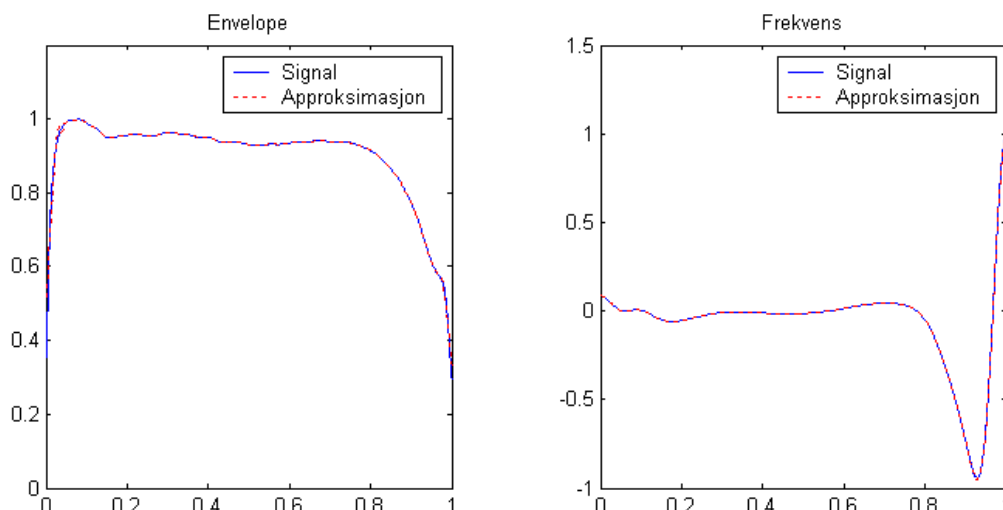
Figur 5.5 Envelope og frekvens til et gjennomsnitt av pulser (blå heltrukket linje) og det best tilpassede polynomet av grad 12 (rød stiplede linje).

5.4.2 Tilpasning ved polynomer

Envelopen og IFV til pulsen ble approksimert ved et polynom av 12. grad, resultatet kan ses i (5.5). Polynomet approksimerer godt både envelopen og IFV og bør kunne brukes som en representasjon av pulsen. Polynomer av høyere grad approksimerer enda bedre, men foreløpige undersøkelser tyder på at det ikke er noe å vinne på å bruke polynomer med grad over 25. Noen pulser er også langt mer irregulære enn den vi har sett på her og da viser det seg at polynomtilpasning er vanskelig. For ”pene” pulser er tilpasning ved polynomer en robust og effektiv måte å representere pulser på.



Figur 5.6 Samme puls som i Figur 5.3, men med approksimasjon beregnet ved db4-wavelets på nivå 7. Approksimasjonen gjør bruk av 15 koeffisienter for hver av profilene. Legg merke til den svake approksimasjonen.



Figur 5.7 Samme som Fig. 6, men denne gang er approksimasjonen på nivå 4. Dette tilsvarer 71 koeffisienter for hver profil. Legg merke til den gode approksimasjonen.

5.4.3 Tilpasning ved wavelets

Approksimasjoner til envelope og IFV ble beregnet ved bruk av Daubechies4 (db4) wavelets for to ulike nivåer. Approksimasjonen på nivå 7 (Figur 5.6) er dårlig, mens approksimasjonen på nivå 4 (Figur 5.7) er svært god. Dette viser hvor viktig det er å velge passende nivå ved bruk av wavelets. Valget vil alltid være en avveining mellom grad av komprimering og ytelse.

Approksimasjonen på nivå 7 er representert ved 15 koeffisienter for både envelope og IFV, dvs en svært høy kompresjonsfaktor. Ytelsen, derimot, er dårlig. På nivå 4 er approksimasjonen representert ved 71 koeffisienter, men ytelsen er fremragende. Vi har her kun brukt db4-wavelets, men foreløpige undersøkelser viser at biortogonale wavelets viser tilsvarende ytelse. Hvilken type wavelets som bør velges vil i fremtidige undersøkelser være gjenstand for testing.

6 IDENTIFIKASJON VED PULSANALYSE

Hensikten med pulsanalysen er å finne forskjeller i pulsform som skiller individuelle emittere fra hverandre. Disse forskjellene skyldes utilsiktet pulsmodulasjon (se avsnitt 2.3.1) og gir utslag på pulsenvelope og IFV. I forrige kapittel beskrev vi forskjellige måter å representere slike pulsdata på, og i dette kapittelet vil vi beskrive metoder som bruker disse pulsdata til å identifisere individuelle emittere.

Uavhengig av hvordan vi velger å komprimere/desimere og trekke ut egenskaper fra pulsdata, så er sluttresultatet en vektor med tallverdier. Denne vektoren kan i enkleste form være bare sammensetningen av de ukomprimerte vektorene med pulsenvelope og IFV-verdier, eller den kan bestå av mer kompakte representasjoner. Uansett kaller vi dette sluttresultatet for en *vektor med pulsparametere*, eller bare en *pulsvektor*. Dette er vektorer i et N -dimensjonalt parameterrom, og det er disse vektorene som er utgangspunktet for identifikasjonsmetodene.

For hver radarmodell vil vi ha en emitterdatabase som inneholder informasjon som kan brukes til å avgjøre hvilken individuell emitter en gitt vektor av pulsparametere kommer fra. Hva slags informasjon som lagres i databasen skal vi komme tilbake til senere. For å bygge opp slike emitterdatabaser er vi avhengig av å ha pulsdata fra emittere vi kjenner identiteten til. La oss anta at vi har f.eks. 10 kjente emittere og fra hver av disse emitterne har vi pulsdata fra 100 pulser. Fra hver puls beregnes en N -dimensjonal vektor med pulsparametere. For hver av de 10 emitterne får vi altså 100 slike vektorer. Geometrisk er hver emitter (så langt) representert med en klynge av 100 punkter i det N -dimensjonale parameterrommet. For en ny ukjent vektor med pulsparametere, dvs. et nytt punkt i dette parameterrommet, skal emitterdatabasen gi svar på hvilken av emitterne målingen kommer fra.

6.1 Statistisk desisjonsteori

Statistisk kan vi beskrive problemet på følgende måte: Vi har forskjellige klasser (emittere) som vi kan benevne $\omega_1, \omega_2, \dots, \omega_M$. For hver klasse ω_i har vi klassebetingede sannsynlighetstettheter $p(x | \omega_i)$ som angir sannsynlighetsfordeling for sample (vektorer med pulsparametere) fra klasse ω_i . I tillegg har vi for hver klasse en a priori sannsynlighet $P(\omega_i)$ som angir hvor ofte sampler fra denne klassen forekommer. Sannsynligheten for at et sample x kommer fra klasse ω_i er gitt ved a posteriori sannsynlighet

$$P(\omega_i | x) = \frac{p(x | \omega_i)P(\omega_i)}{p(x)} \quad (6.1)$$

hvor $p(x) = p(x | \omega_1)P(\omega_1) + \dots + p(x | \omega_M)P(\omega_M)$. Bayes beslutningsregel kan nå formuleres som

$$\text{Bayes_1: Hvis } P(\omega_i | x) > P(\omega_j | x) \text{ for alle } j \neq i \text{ så hører } x \text{ til klasse } \omega_i \quad (6.2)$$

mao, vi tilordner sample x til klasse ω_i hvis denne klassen har størst a posteriori sannsynlighet. Merk også at (6.2) er ekvivalent med regelen

$$\text{Bayes_2: Hvis } p(x | \omega_i)P(\omega_i) > p(x | \omega_j)P(\omega_j) \text{ for alle } j \neq i \text{ så hører } x \text{ til klasse } \omega_i \quad (6.3)$$

Hvis vi vet at alle klassene er like sannsynlige, så reduseres (6.3) til

$$\text{Bayes_3: Hvis } p(x | \omega_i) > p(x | \omega_j) \text{ for alle } j \neq i \text{ så hører } x \text{ til klasse } \omega_i \quad (6.4)$$

Det kan vises at Bayes regel gir en *minimum feilrate klassifikator*, dvs det finnes ingen andre klassifiseringsregler som gjøre færre feilklassifiseringer enn denne.

Vi kan dele opp det N -dimensjonale parameterrommet i M disjunkte områder R_1, R_2, \dots, R_M , hvor R_i er det området som tilordnes til klasse ω_i . Bayes beslutningsregel gjør feilklassifisering når vi har et sample x som tilhører klasse ω_i , men som ligger utenfor området R_i . Minimum feilrate er dermed gitt av formelen

$$P_{\text{Bayes}} = \sum_i P(x \notin R_i | \omega_i)P(\omega_i) \quad (6.5)$$

På grenseflaten mellom to områder R_i og R_j så er $P(\omega_i | x) = P(\omega_j | x)$. Her har det ingen betydning for feilraten hvilken av de to klassene som velges. Slike overganger mellom to områder kalles *desisjongsgrenser*.

Ideen er nå å lagre informasjon om hver emitter i en database, og bruke Bayes regel (6.2) for emitterklassifikasjon. Problemet er imidlertid at vi i utgangspunktet verken kjenner sannsynlighetstetthetene $p(x | \omega_i)$ eller a priori sannsynlighetene $P(\omega_i)$. Det vi har å forholde oss til er sampler fra de nevnte sannsynlighetstetthetene. Vi vil i de neste avsnittene se på tre grupper av metoder som forsøker å løse dette problemet på forskjellige måter:

- a) Ikke-parametriske statistiske metoder
- b) Parametriske statistiske metoder
- c) Fordelingsfrie metoder

Informasjonen som representerer hver emitter i emitterdatabasen er for disse metodene henholdsvis a) en samling av sampler fra klassen, b) parametere som beskriver sannsynlighetstettheten for klassen og c) desisjongsgrensene som skiller klassen fra andre klasser. Beskrivelsene av metodene er delvis hentet fra (8) og (9).

6.1.1 Ikke-parametriske statistiske metoder

Emitterdatabasen vil i dette tilfellet bestå av en stor samling av pulsvektorer fra alle klassene, og for hver pulsvektor har vi også informasjon om hvilken klasse den tilhører. Fordelen med ikke-parametriske metoder er at man ikke trenger å gjøre antagelser om formen på tetthetsfordelingene, de vil være indirekte representert gjennom samlingen av pulsvektorer. Vi skal se på to metoder, Parzen-vinduer og k -nærmeste-nabo (k -NN) metoden.

PARZEN-VINDUER

Parzen-vinduer er en metode for å estimere tetthetsfunksjonen $p(x|\omega_i)$ basert på sampler fra klasse ω_i . Hvis vi har k sampler $x_1, \dots, x_k \in R^N$ fra klasse ω_i , så kan vi estimere $p(x|\omega_i)$ ved

$$p_k(x|\omega_i) = \frac{1}{k} \sum_{j=1}^k \frac{1}{h^N} \varphi\left(\frac{x_j - x}{h}\right) \quad (6.6)$$

hvor vindufunksjonen φ tilfredsstillers $\varphi(x) \geq 0$ og $\int \varphi(x) dx = 1$, og h er en skaleringsfaktor. Hvis φ og h tilfredsstillers visse krav så kan det vises at $p_k(x|\omega_i)$ konvergerer mot $p(x|\omega_i)$ når $k \rightarrow \infty$. Vindufunksjonen φ kan i tilfellet $N=1$ f.eks være

$$\varphi(x) = \begin{cases} 1 & |u| \leq \frac{1}{2} \\ 0 & \text{ellers} \end{cases}, \quad \text{eller} \quad \varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (6.7)$$

Hvis vi antar at antall pulssamplers fra de forskjellige klassene også representerer godt forholdene mellom de a priori sannsynlighetene $P(\omega_i)$, så har vi det vi trenger for å bruke Bayes beslutningsregel i (6.3). Hvis vi ikke kjenner de a priori sannsynlighetene, så er det naturlig å anta at alle klassene er like sannsynlige og dermed bruke Bayes beslutningsregel i (6.4).

k-NÆRMESTE-NABO METODEN (k-NN)

Klassifisering av en ny pulsvektor x gjøres i tre trinn:

1. Finn de k vektorene i databasen som ligger nærmest x .
2. Blant disse k vektorene, tell opp antall sampler k_i som tilhører klasse ω_i , $i = 1, 2, \dots, M$.
3. Tilordn x til den klassen ω_i med det maksimale antall k_i av sampler.

Begrunnelsen for dette er at

$$p(x|\omega_i)P(\omega_i) \propto \frac{k_i/k}{V} \quad (6.8)$$

hvor V er volumet til den sfæren rundt vektoren x som er akkurat så stor at den inneholder de k nærmeste vektorene i databasen. Fra (6.8) har vi det vi trenger for å bruke Bayes beslutningsregel i (6.3).

Tilnærmingen i (6.8) blir generelt bedre for større k -verdier og mindre volum V . Med mange sample i databasen så kan vi velge stor k -verdi og likevel finne de k -nærmeste naboer i liten

avstand fra x (dvs lite volum V). I spesialtilfellet $k=1$ så kan det vises at feilraten for denne metoden, P_{1-NN} , er asymptotisk begrenset av 2 ganger den optimale (Bayes-) feilraten (6.5), dvs

$$P_{Bayes} \leq P_{1-NN} \leq 2P_{Bayes} \quad (6.9)$$

1-NN metoden kan derfor brukes for å få en ide om størrelsesorden for den optimale feilraten.

6.1.2 Parametriske statistiske metoder

Parametriske statistiske metoder går ut på å representere de forskjellige klassene med parametere som beskriver fordelingen til samplene, dvs for klasse ω_i lagres parametere som beskriver fordelingen $p(x|\omega_i)$. Hvilken form denne fordelingen har er ukjent, men en vanlig approksimasjon er å bruke en multivariat normalfordeling (Gaussfordeling)

$$p(x|\omega_i) = \frac{1}{(2\pi)^{N/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right) \quad (6.10)$$

hvor vektoren $\mu_i = E[x]$ er middelverdi for samplene fra klasse ω_i , $|\Sigma_i|$ betegner determinanten til Σ_i , og Σ_i er $N \times N$ kovarians-matrisen definert ved

$$\Sigma_i = E\left[(x-\mu_i)(x-\mu_i)^T\right] \quad (6.11)$$

Utrekningen av $p(x|\omega_i)$ i (6.10) kan være regnekrevende pga inverteringen av Σ_i . Hvis derimot alle komponentene i pulsvektoren x er statistisk uavhengige variable, så reduseres Σ_i til en diagonalmatrise med standardavvikene σ_i^2 langs diagonalen. I så fall har vi

$$p(x|\omega_i) = \frac{1}{(2\pi)^{N/2} \sigma_i(1)\dots\sigma_i(N)} \exp\left(-\sum_{j=1}^N \frac{(x(j)-\mu_i(j))^2}{2\sigma_i(j)^2}\right) \quad (6.12)$$

Hvis de a priori sannsynlighetene er like for alle klassene, så vil (6.12) i kombinasjon med (6.4) gi oss vår beslutningsregel.

Om vi erstatter $p(x|\omega_i)$ i (6.3) (eller (6.4)) med $g_i(x) = \varphi(p(x|\omega_i))$ for en monotont voksende funksjon φ , så endrer ikke det klassifiseringsresultatet. Den klassen ω_i med størst verdi for $p(x|\omega_i)$ vil også ha størst verdi for $g_i(x)$. Hvis i tillegg alle klasser har samme kovariansmatrise Σ , og vi velger $\varphi = \exp$, så er beslutningsregelen redusert til å finne klasse ω_i som *minimerer* Mahalanobis-avstanden d mellom x og μ_i , gitt ved

$$d^2 = (x-u_i)^T \Sigma^{-1}(x-u_i) = \sum_{j=1}^N \frac{(x(j)-u_i(j))^2}{\sigma_i(j)}, \quad (6.13)$$

der likheten * holder når kovariansmatrisen er diagonal.

6.1.3 Fordelingsfrie metoder

Fordelingsfrie metoder har ingen representasjon av sannsynlighetstetthetene til emitterklassene, isteden har de representert desisjongsgrensene mellom klassene. Plasseringen av disse desisjongsgrensene gjøres gjennom en "læringsalgoritme". Denne algoritmen justerer iterativt parameterne som (direkte eller indirekte) definerer desisjongsgrensene. Dette gjøres på en måte som minimerer en gitt feilfunksjon. Vi skal kort se på diskriminantfunksjoner og en type nevrale nettverk kalt flerlags perseptroner.

DISKRIMINANTFUNKSJONER

Lineære diskriminantfunksjoner representerer desisjongsgrensen mellom to klasser med et hyperplan. Generaliserte lineære diskriminantfunksjoner bruker hyperkvadratiske flater (hypersfære, hyperellipsoide,...) for å representere desisjongsgrensene. Parameterne som definerer desisjongsgrensene finnes ved å minimere en gitt feilfunksjon. Denne funksjonen kan f eks være kvadratet av avstanden til desisjongsgrensen, summert over for alle feilklassifiserte sampler.

FLERLAGS PERSEPTRONER

Flerlags perseptroner (Multi Layer Perceptrons) er en type nevrale nett som er mye brukt til klassifisering. Her representeres også (indirekte) desisjongsgrensene mellom klassene, men med mer generelle flater enn lineære og kvadratiske. Denne og enkelte andre typer nevrale nett (f eks radielle basis-funksjoner) gjør det ofte bra i sammenligning med andre klassifiseringsmetoder. For en grundigere sammenligning se f eks (10), og for en beskrivelse av nevrale nett fra en statistisk synsvinkel se (11).

6.2 Valg av metode

Hvilket utvalg av pulsparametere som viser seg å være best egnet for å skille ulike emittere fra hverandre er ikke fastlagt ennå, men vi regner med at antall pulsparametere vil være i størrelsesorden 10-100. Det totale antall emittere i alle emitterdatabasene vil kunne være fra flere hundre til flere tusen, avhengig av hvor god tilgang vi får til dataopptak. Pulsparameterne vil representere pulsenvelope og IFV og forskjellige måter å gjøre dette på ble diskutert i forrige kapittel.

Basert på eksempler av pulsparametere fra de forskjellige (emitter-) klassene skal vi lage emitterdatabaser som kan gi svar på hvilken klasse vi skal tilordne nye eksempler. I forrige avsnitt så vi på tre hovedgrupper av metoder for å representere (emitter-)klassene. Det var ikke-parametriske statistiske metoder, parametriske statistiske metoder, og fordelingsfrie metoder.

Ulempen med ikke-parametriske metoder (k -NN, Parzen-vinduer) er først og fremst at man må lagre store mengder med data, og at det er regnekrevende å søke gjennom disse dataene. Det finnes riktignok algoritmer for å tynne ut datamaterialet uten at det går ut over

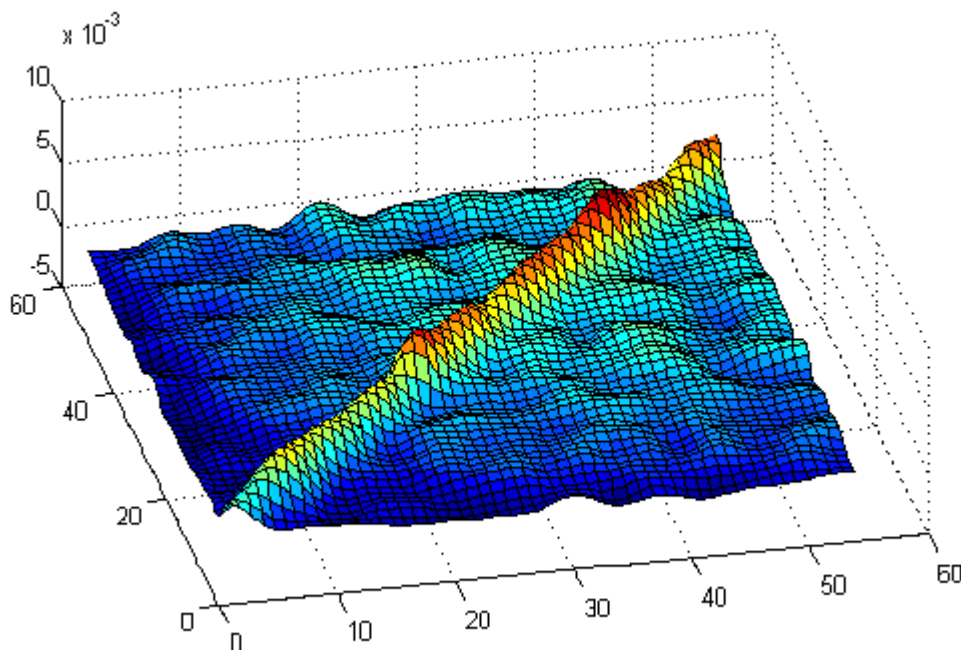
klassifiseringsevnen, men generelt er dette et problem. Spesielt for oss som etter hvert kan få svært mange emittere, så vil dette bli problematisk. Den foreløpige konklusjonen er at slike metoder *ikke* er blant dem vi vil prioritere å implementere. Det eneste unntaket er muligens 1-NN som kan brukes for å angi en øvre begrensning for den optimale feilraten (se (6.9)).

Når det gjelder fordelingsfrie metoder er det ofte en omstendelig prosess å utvide med nye klasser. I verste fall må alle parameterne finnes på nytt, og spesielt for nevrale nettverk kan det være en regnekrevende prosess. Likevel kan det være interessant å sammenligne med nevrale nett når vi etter hvert har bygd opp en rimelig stor emitterdatabase.

Foreløpig regner vi parametriske metoder som enklest og best egnet til å representere emitterklasser. Vi vil i første omgang representere tetthetsfunksjonen til emitterklassen med en Gaussfordeling (6.10). I mangel av a priori sannsynligheter vil vi også anta at alle emittere er like sannsynlige, så vi vil bruke Bayes beslutningsregel i (6.4). Neste avsnitt viser hvordan identifikasjonsmetoden blir i dette tilfellet. Om dette fungerer bra avhenger langt på vei av om Gaussfordelingen er en god approksimasjon til fordelingen av pulsvektorer for emitterne. Vi har tro på at det er en god approksimasjon, men det vil bli undersøkt nøyere senere.

6.2.1 Korrelering mot emitterdatabase

Med metoden vi foreløpig har valgt som klassifikasjonsmetode, så vil hvert element i emitterdatabasen bestå av en vektor $\mu = (\mu(1), \mu(2), \dots, \mu(N))$ av midlede pulsparametere og en kovariansmatrise Σ , som reduseres til en diagonalmatrise i tilfellet hvor pulsparametere er statistisk uavhengige variable. En ny pulsvektor $x = (x(1), x(2), \dots, x(N))$ klassifiseres i henhold til Bayes beslutningsregel i (6.4), og klassenes sannsynlighetstetthet finnes fra (6.10) eller (6.12). I Figur 6.1, hentet fra (12), ser vi plot av kovariansmatrisen for en emitter. De ca 50 parameterne er ordnet etter hvilken del av pulsen de koder for.



Figur 6.1 Eksempel på kovariansmatrise for pulsvektorer fra en emitter.

Vi ser at korrelasjonen tilnærmet lik null for parametere som koder for helt forskjellige deler av pulsen. For parametere som ligger nær hverandre ser vi imidlertid at korrelasjonen er stor ("bred avrundet diagonal"), noe som skulle tilsi at det fremdeles er rom for ytterligere kompresjon/desimering.

Det er fristende å bruke (6.12) fremfor den mer regnekrevende (6.10), men det må undersøkes nærmere hvor mye vi taper på å gjøre antagelsen om at pulsparametrene er statistisk uavhengige variable.

En ting vi har unnlatt å nevne så langt er at pulsene i et pulstog ikke har nøyaktig samme lengde, det er små variasjoner i antall sample fra puls til puls. Vektorene med pulsparametre vil dermed også kunne variere litt i lengde, og vi blir nødt til å bestemme hvordan vi skal gå frem når vi sammenligner pulsvektorer med litt forskjellig lengde. To nærliggende måter å gjøre dette på er enten å ignorere de komponentene som er "til overs" i den lengste vektoren, eller å utvide den korteste vektoren med 0'er slik at den får samme lengde som den lengste. Det virker riktigst å bruke sistnevnte metode siden det vil føre til at forskjellige lengde "straffes" i form av lavere sannsynlighet for at den ene vektoren kommer fra emitteren representert ved den andre.

En annen ting vi også må ta hensyn til er at vi vil motta pulser fra emittere som ikke finnes i emitterdatabasen. Vi kan ikke bare tilordne nye pulsvektorer til den emitterklassen som er mest sannsynlig, vi må også sette en terskel for hvor sannsynlig denne klassen må være. Pulser som ikke blir assosiert til noen klasse på denne måten blir lagret i en egen kandidatdatabase.

6.2.2 Oppdatering av emitterdatabase

Pulsvektorer i kandidatdatabase kan komme fra potensielle nye emittere. Sammenslåing av slike vektorer til en emitter og overflytting til emitterdatabase gjøres manuelt.

Etter at emittere er lagt inn i emitterdatabase, så vil også emitterparameterne oppdateres på bakgrunn av nye pulser som assosierer til denne emitteren.

7 FORELØPIGE ERFARINGER MED PIA

PIA, eller rettere sagt FIA, har blitt prøvd ut med data (flyradarer) som er tatt opp med måleoppstillingen i Figur 2.2. Antall radarmodeller har vært begrenset, 3-4 stykker, men med flere ulike emittere for hver modell. Alle målingene har vært av ”targets of opportunity” som gjør at det er vanskelig å vite eksakt hva vi ser på. De foreløpige erfaringene viser at klassifiseringen og databaseoppbyggingen fungerer bra for enkelte radarmodeller og mindre bra for andre (1).

Oscilloskopet i måleoppstillingen tar opp typisk 200 pulser fortløpende før disse lagres til disk. Etter lagringen, som kan ta noen sekunder, tas nye 200 pulser opp osv. Tilsynelatende forandrer enkelte radarmodeller egenskaper (envelope eller IFV) fra fil til fil. Innad i en fil er egenskapene like, men er ulike fra en fil tatt opp noen sekunder senere. Hva denne forandringen i envelope eller IFV skyldes er uvisst. Mulige årsaker kan være:

- Ustabiliteter i måleoppstillingen (i første rekke tunereren eller oscilloskopet).
- Multipath problemer. Pulsrefleksjoner fra bakke og bygninger adderes til direktesignalet og kan forvrengte karakteristikker i pulsen som f.eks. envelope og IFV. Endringer i emitterens posisjon fra ett opptak til neste kan gi variasjoner i eventuelle multipath refleksjoner og dermed gi utslag i pulsens karakteristikker.
- Pulsene blir normalisert ved at alle samplene deles på $\max(\text{abs}(\text{amplitude}))$. Dette kan være for enkelt dersom opptaket har ”støy” eller artefakter med amplitude over det som kan antas å være høyeste pulsamplitude. På den måten vil pulsene normaliseres feil og det kan gi utslag i spesielt amplituden. NB. Dette ser ikke ut til å være noe stort problem i praksis.
- Radaren er ikke stabil over tid (eller fra pulstog til pulstog). Dette kan være tilsiktet eller ikke. Dersom dette er tilfelle blir individgjenkjenning av slike radarmodeller en svært vanskelig oppgave

7.1 Multipath

Problemet med forandring av egenskapene fra fil til fil gjelder bare for noen radarmodeller. For andre modeller vil emittere kjennes igjen over flere filer, og også over tid. For å redusere problemet med multipath vil vi forsøke å sett opp den nye måleoppstillingen slik at kun direktesignalet fanges opp.

8 OPPSUMMERING

Denne rapporten beskriver hovedfunksjonene i PIA og gir en oversikt over aktuelle metoder på de forskjellige funksjonstrinnene. Spesielt er det lagt vekt på pulsanalysen, som omfatter preprosessering, komprimering/egenskapsuttrekking og identifikasjon.

For å klare å identifisere emittere vha pulsanalyse er det tre ting som må oppfylles. For det første så må pulsene fra forskjellige emittere av samme radartype faktisk være forskjellige. For det andre så må vår preprosessering og påfølgende komprimering/egenskapsuttrekking fange opp disse forskjellene, og til sist så må identifikasjonsmetoden klare å utnytte disse forskjellene til å skille emitterne fra hverandre. Hvis ikke pulsvektorene som identifikasjonsmetoden tar utgangspunkt i er rimelig godt separert fra hverandre i parameterrommet, så har identifikasjonsmetoden et dårlig utgangspunkt. Det er derfor helt avgjørende å finne riktig måte å representere pulldata på, og denne rapporten beskriver flere aktuelle alternativer.

Hovedvekten i rapporten er på hvordan pulldata representeres, og hvordan identifikasjonsmetoden fungerer. Test og evaluering av metoder beskrevet i denne rapporten vil bli presentert i en senere rapport.

Litteratur

- (1) Eilivstjønn J (2001): FIA - programvare for klassifisering og identifikasjon av radarpulser, FFI/NOTAT-2001/02261, Ugradert
- (2) Malnes E (2000): Pulssortering og emitteridentifikasjon, FFI/RAPPORT-2000/02428, Ugradert
- (3) Malnes E, Eilivstjønn J, Kulbotten G, Kvernsveen K, Øhra H (2000): Spesifikasjon av ESM-demonstrator, FFI/RAPPORT-2000/06388, Ugradert
- (4) Mjanger M (1999): Deinterleaving og pulsanalyse, hovedfagsoppgave, Universitetet i Oslo, Oslo.
- (5) Wiley R G (1993): Electronic Intelligence: The Analysis of Radar Signals, Artech House.
- (6) Pedersen S I, Malnes E (2000): To metoder for automatisk gjenkjenning av PRF-mønstre, FFI/RAPPORT-2000/02480, Ugradert
- (7) Strang G, Nguyen T (1997): Wavelets and Filter Banks, Wellesley - Cambridge Press.
- (8) Theodoridis S, Koutroumbas K (1999): Pattern Recognition, Academic Press, UK.
- (9) Duda R O, Hart P E, Stork D G (2001): Pattern Classification, Wiley-Interscience, USA.
- (10) Michie D, Spiegelhalter D J, Taylor C C (1994): Machine learning, neural and statistical classification, Ellis Horwood.
- (11) Bishop C M (1995): Neural Networks for Pattern Recognition, Oxford University Press.
- (12) Høydal K (2000): Individgjenkjenning av radarpulser ved bruk av Wavelettransform, diplomoppgave, NTNU, Trondheim.

FORDELINGSLISTE

FFIE **Dato:** 31 januar 2002

RAPPORTTYPE (KRYSS AV) <input checked="" type="checkbox"/> RAPP <input type="checkbox"/> NOTAT <input type="checkbox"/> RR	RAPPORT NR. 2002/00555	REFERANSE FFIE/805/113	RAPPORTENS DATO 31 januar 2002
RAPPORTENS BESKYTTELSESGRAD UGRADERT		ANTALL EKS UTSTEDT 52	ANTALL SIDER 41
RAPPORTENS TITTEL METODER FOR EMITTERIDENTIFISERING I EN ESM-DEMONSTRATOR		FORFATTER(E) SUNDGOT Rune, ØYEHAUG Leiv	
FORDELING GODKJENT AV FORSKNINGSSJEF Torleiv Maseng		FORDELING GODKJENT AV AVDELINGSSJEF: Johnny Bardal	

EKSTERN FORDELING

INTERN FORDELING

ANTALL	EKS NR	TIL	ANTALL	EKS NR	TIL
			14		FFI-Bibl
1		LKI	1		Adm direktør/stabssjef
1		v/Ob Jon R Krogstad	1		FFIE
1		v/Oblt Morten Stokke	1		FFISYS
			1		FFIBM
1		LTI	1		FFIN
1		v/Oblt Erik Pande-Rolfsen	1		Svein-Erik Hamran, FFIE
			1		Atle Knapkog, FFIE
1		FO/FST	1		Rune Sundgot, FFIE
1		v/Oblt Gaute Dyrdal	1		Leiv Øyehaug, FFIE
			1		Tore Smestad, FFIE
1		FO/LST	1		Reidar Haugen, FFIE
1		v/Maj Eivind Sørdal	1		Hans Øhra, FFIE
			1		Kirsten Kvernsveen, FFIE
1		FO/E	1		Morten Mjanger, FFIE
1		v/Rådgiver Tore Belsnes	1		Jan Hammerstad, FFIE
			1		Arne Petter Bartholsen, FFIE
1		LFK	3		Arkiv, FFIE
1		v/Oblt Harald Guntvedt			FFI-veven
1		FKN			
1		v/Oblt Bror Myrvang			
1		FKS			
1		v/Oblt Finn A Jørgensen			
1		SSLV			
1		v/Maj Ove Jørn Luktasslimo			

FFI-K1

Retningslinjer for fordeling og forsendelse er gitt i Oraklet, Bind I, Bestemmelser om publikasjoner for Forsvarets forskningsinstitutt, pkt 2 og 5. Benytt ny side om nødvendig.