



FFI-rapport 2014/00696

Undersøkelse av sonarsimulatoralgoritme i PROTEUS



Karl Thomas Hjelmervik



Undersøkelse av sonarsimulatoralgoritme i PROTEUS

Karl Thomas Hjelmervik

Forsvarets forskningsinstitutt (FFI)

9. mai 2014

FFI-rapport 2014/00696

1288

P: ISBN 978-82-464-2376-0

E: ISBN 978-82-464-2377-7

Emneord

Simulator

Sonar

Modellering

Sonarprosessering

Godkjent av

Håkon S Andersen

Prosjektleder

Elling Tveit

Avdelingsjef

Sammendrag

KDA har utviklet en sonarsimulator som er basert på en metode utviklet av FFI [1].

Sonarsimulatoren brukes i dag i den KDA-utviklede treneren, PROTEUS. Sjøforsvaret har rapportert manglende realisme i simuleringene og hevdet at dette er en delvis årsak til at AU-lag som tjenestegjør om bord på fregattene ikke bruker simulator til landbasert trening i dag.

Som en del av et pågående arbeid i FFI prosjekt 1288 – “Kampsystemet Fridtjof Nansen – Taktikk, operasjonalisering og trening”, har FFI implementert sonarsimulatoren i MATLAB for å gjennomføre en følsomhetsstudie for ulike integrerte parametere i simulatoren. Målet er å finne årsaken til den manglende realismen. Parameterne som undersøkes er modellopløsning, deteksjonsterskel og normaliseringsvindu.

Studien er gjennomført på to ulike datasett som KDA hentet ut fra PROTEUS. Det første datasettet er hentet fra et scenario i Bjørnafjorden der kort sonarrekkevidde (12 km) benyttes. Det andre datasettet er hentet fra et scenario utenfor Marsteinen. Her benyttes lang sonarrekkevidde (48 km). PROTEUS benytter et maksimalt celleantall på 50, noe som betyr at modellopløsningen er avhengig av sonarrekkevidden. Dette medfører at det brukes relativt høy rekkeviddeoppløsning (100 m til 240 m) i det første scenarioet og lav oppløsning (212 m til 960 m) i det andre scenarioet.

Resultatene fra denne studien viser at rekkeviddeoppløsningen som brukes i LYBIN alltid bør være 200 m eller bedre. Videre bør størrelsen på normaliseringsvinduet som brukes være invariant i antall meter. Her foreslås et normaliseringsvindu på 1000 m og et tilhørende vaktbånd på 170 m. Deteksjonsterskelen bør styres av den som har ansvaret for treneren. Dermed kan den ansvarlige både velge vanskelighetsgrad (lav terskel fører til flere falske alarmer og dermed høyere vanskelighetsgrad) og dessuten også tilpasse terskelen til miljøet scenarioet kjøres i. For tilfellene studert her anbefales det 100m - 200 m oppløsning og en deteksjonsterskel på 7 - 10 dB for å oppnå en realistisk sky av falske alarmer.

Merk at denne studien ikke inkluderer trackeren som KDA bruker i treneren sin. Det planlegges å kjøre et scenario i PROTEUS med de foreslåtte parameterne og deretter å vurdere realismen i trackene som simulatoren kommer frem til.

English summary

KDA has implemented a sonar simulator based on an algorithm developed by FFI [1]. The sonar simulator is used in the KDA sonar trainer PROTEUS. The Norwegian navy reports that the sonar simulator does not have sufficient fidelity and claims that this is partially the reason why operational ASW-teams do not use the simulator for land-based training.

We have implemented the sonar simulator in MATLAB in order to assess the sensitivity of the algorithm to a handful of parameters. The goal is to discover why the simulator lacks realism. The assessed parameters are model resolution, detection threshold, and the size of the normalization window.

Two different data sets taken from PROTEUS are used in the study. The first data set is extracted from a scenario in Bjørnafjorden. In this scenario a short sonar range setting was used (12 km). The second data set employed a long sonar range (48 km) and was extracted from a scenario just outside Marsteinen in the Norwegian Trench. The model range resolution used in PROTEUS depends on the sonar range. High resolutions are used for short sonar ranges (100 m to 240 m in the Bjørnafjorden case), and low resolutions for long ranges (212 m to 960 m in the Marsteinen case).

The result from the study indicates that the model range resolution should always be better than 200 m. The size of the normalization window should be 1000 m, and the corresponding guard band should be 170 m. The detection threshold should be available for adjustment by the supervisor of the training. That way, the difficulty of the training can be determined by the supervisor (low threshold results in many false alarms and thereby high difficulty). Some modeled environments are also more prone to false alarms than other, and may therefore need adjustments in the detection threshold to get a realistic sonar picture. For the cases studied in this report a range resolution between 100 m and 200 m and a detection threshold of 7 dB to 10 dB are advised.

Note that this study does not include the tracking algorithm in PROTEUS. The parameters suggested above are planned tested in PROTEUS to see if a realistic sonar picture is achieved.

Innhold

1	Innledning	7
2	Algoritmebeskrivelse	7
3	Data fra KDA	8
3.1	Topografi	8
3.2	Lydhastighetsprofil	9
4	Simulerte sonarekko - Bjørnafjorden	11
4.1	Modellopløsning	13
4.2	Normaliseringsvindu	15
4.3	Deteksjonsterskel	17
4.4	Eksempel	17
5	Simulerte sonarekko – Marsteinen	20
5.1	Modellopløsning	22
5.2	Eksempel	23
6	Konklusjon	26
	Referanser	26

1 Innledning

KDA har utviklet en sonarsimulator som er basert på en metode utviklet av FFI [1]. Sonarsimulatoren brukes i dag i den KDA-utviklede treneren, PROTEUS. Sjøforsvaret har rapportert manglende realisme i simuleringene og hevdet at dette er en delvis årsak til at operative AU-lag ikke bruker simulator til landbasert trening i dag.

Vi har implementert sonarsimulatoren i MATLAB for å gjennomføre en følsomhetsstudie for ulike integrerte parametre i simulatoren er gjennomført. Målet er å finne årsaken til den manglende realismen.

En kort beskrivelse av sonarsimulatoralgoritmen er gjengitt i kapittel 2. Kapittel 4-5 inneholder en følsomhetsanalyse basert på data vi har fått av KDA. Til slutt konkluderes det i kapittel 6 med noen konkrete forslag til endringer av sonarsimulatoren som kan øke realismen.

2 Algoritmebeskrivelse

Metoden i PROTEUS som brukes for å simulere sonarekko er hentet fra [1]. Forløpet til metoden kan kort oppsummeres som følger:

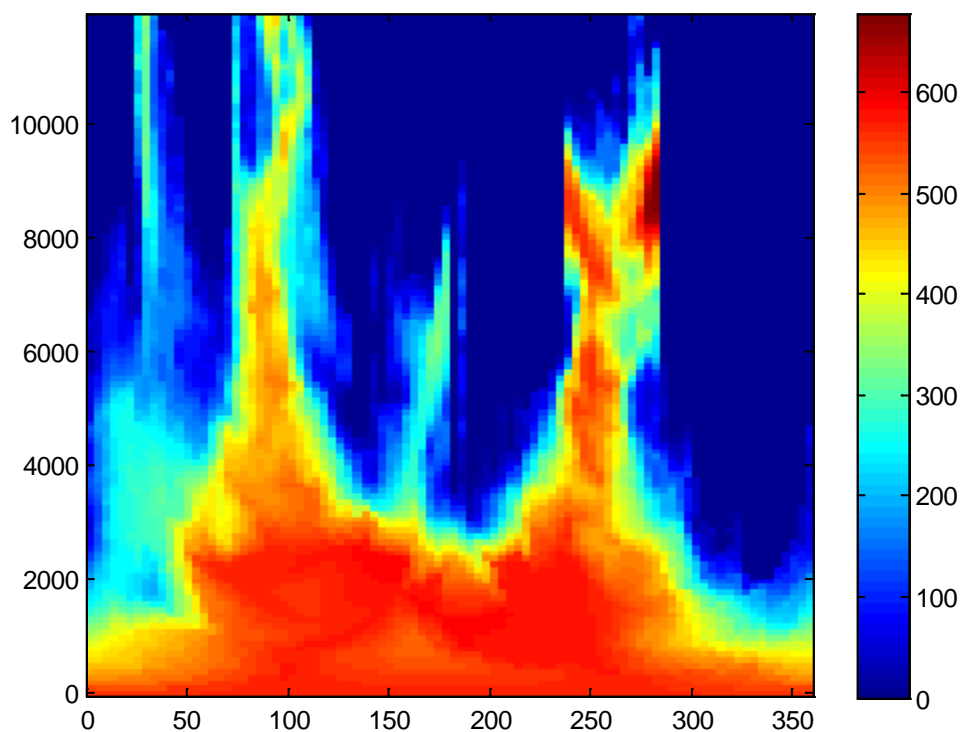
1. Hent informasjon om sonarplattform.
 - a. Posisjon.
 - b. Sonarparametre.
2. Hent miljøinformasjon.
 - a. Lydhastighetsprofil.
 - b. Topografi (N bunnprofiler jevnt spredt over 360°).
 - c. Bunntype.
 - d. Vindhastighet.
3. Kjør LYBIN i $N*2d$ med sonar- og miljøinformasjon hentet i stegene ovenfor.
 - a. Modellopløsning.
 - i. 2000 stråler.
 - ii. 50 dybdeceller.
 - iii. 50 rekkeviddeceller.
 - b. Hent gjenklangsverdiene som funksjon av avstand for hver retning.
 - c. Hent estimert støynivå.
4. Normaliser summen av støy og gjenklang vha en CA CFAR metode [2].
 - a. Et normaliseringsvindu på 10 samples blir brukt.
 - b. Vaktvinduet (guardband) er på 1 sample (en sjettedel av normaliseringsvinduet størrelse rundet ned).
5. Terskle de normaliserte gjenklangsverdiene for å finne sonarekko.
 - a. En terskel på 10 dB er brukt.
6. Legg til gaussisk støy på posisjonene til de genererte sonarekkoene for å finne de endelige posisjonene.

3 Data fra KDA

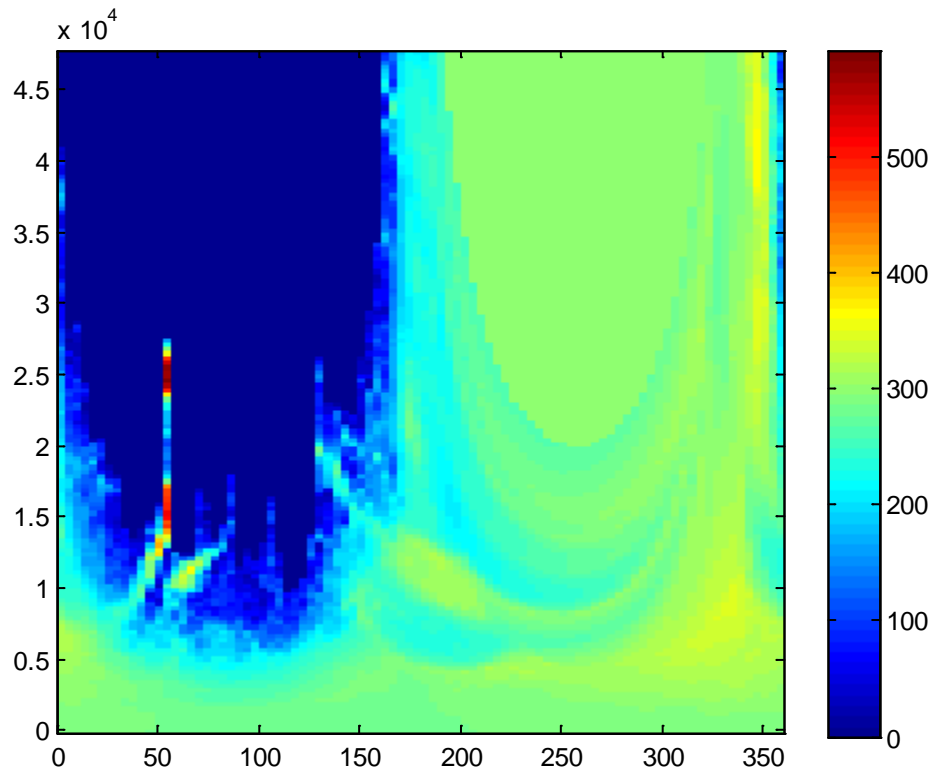
De følgende delkapitlene viser miljøinformasjonen som ble hentet ut av dataene vi fikk av KDA i forbindelse med dette arbeidet. Vi fikk både data fra Bjørnafjorden og Marsteinen. Datasettet fra Bjørnafjorden benyttet en kort rekkeviddeinnstilling (12km), men datasettet fra utenfor Marsteinen benyttet en lang rekkeviddeinnstilling (48 km).

3.1 Topografi

Topografien fra Bjørnafjorden som ble hentet ut fra PROTEUS hadde ca 90m oppløsning (DTED1 bunn), se Figur 3.1. Tilsvarende har datasettet fra utenfor Marsteinen kun 320 m oppløsning, se Figur 3.2.



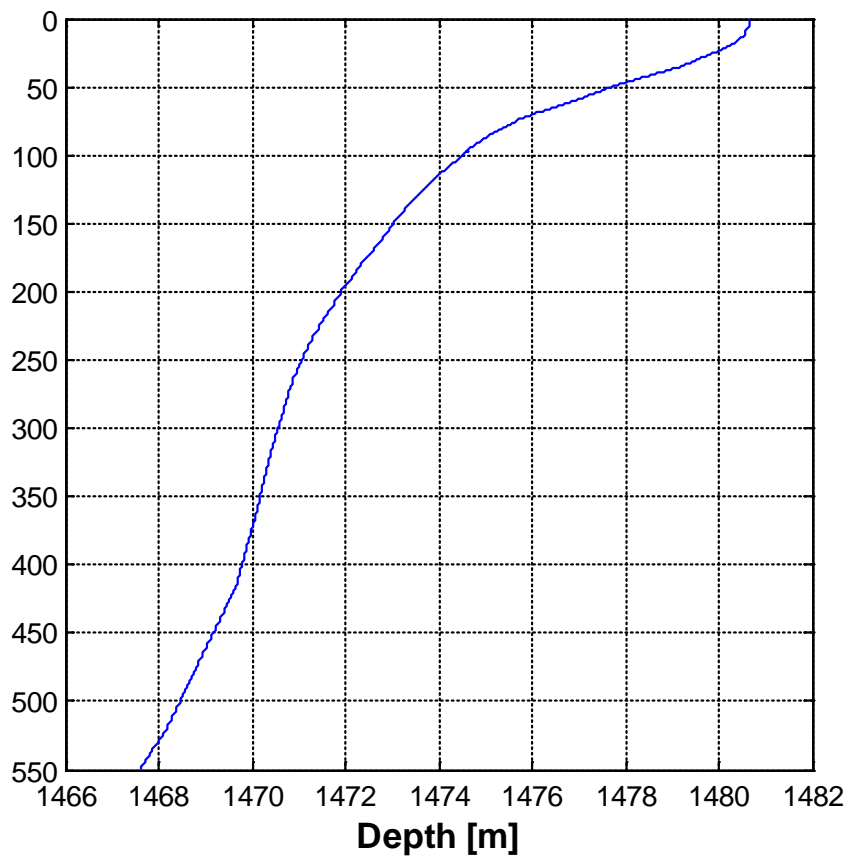
Figur 3.1 Bunntopografi fra Bjørnafjorden plottet som funksjon av avstand og retning (relativt nord) i forhold til sonarfartøyets posisjon.



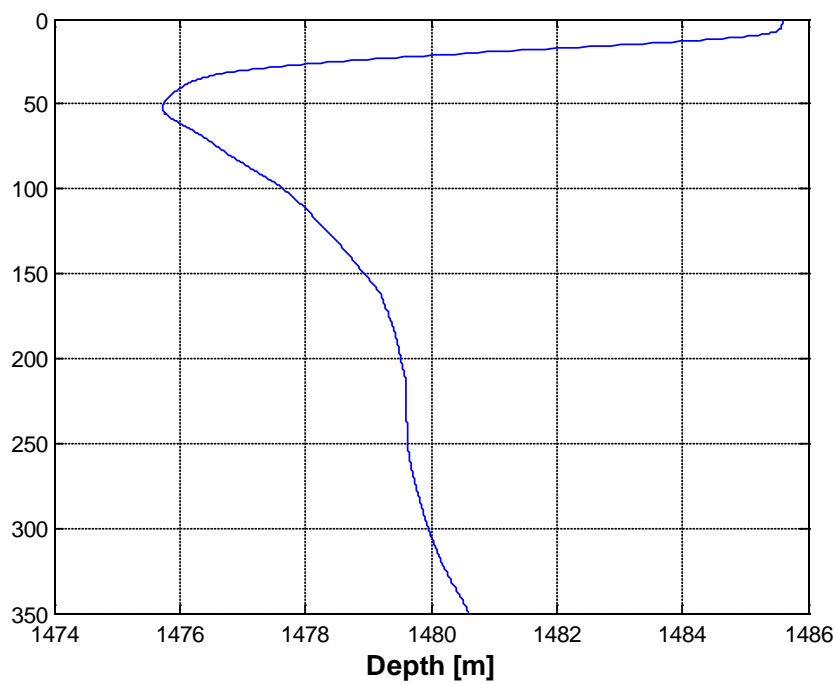
Figur 3.2 *Bunntopografi fra utenfor Marsteinen plottet som funksjon av avstand og retning (relativt nord) i forhold til sonarfartøyets posisjon.*

3.2 Lydhastighetsprofil

Figur 3.3 viser lydhastighetsprofilen som fulgte med eksempeldatasettet fra Bjørnafjorden. Dette er en nedoverrefrakterende profil som tvinger lyden tidlig ned i bunn. En kan derfor ikke forvente lange rekkevidder verken på sanne eller falske alarmer. Figur 3.4 viser profilen fra datasettet utenfor Marsteinen. Dette er nok en vårprofil med soloppvarmet vann i det øverste laget og ellers en oppover-refrakterende profil.



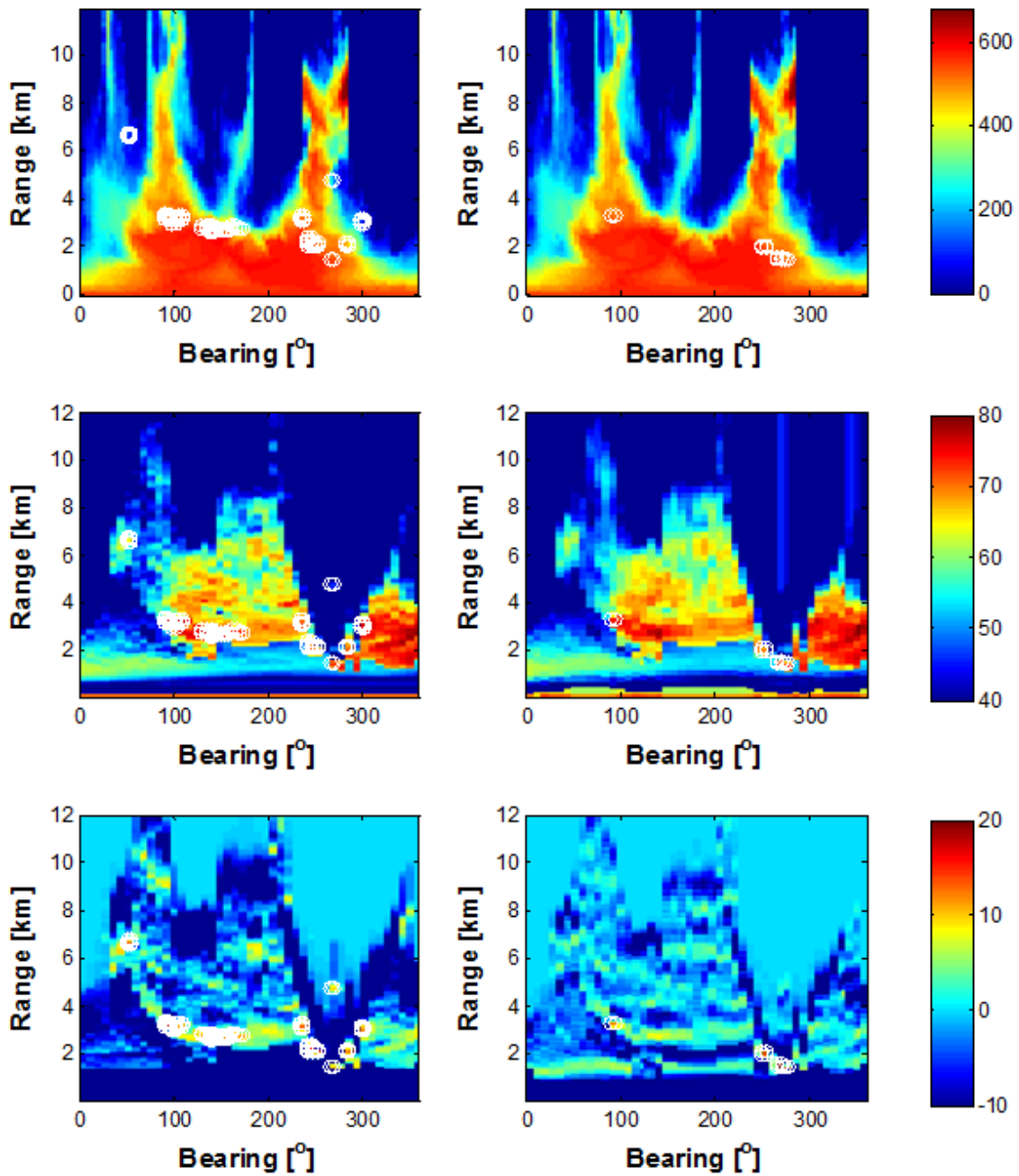
Figur 3.3 Lydhastighetsprofilen i Bjørnafjorden hentet ut fra PROTEUS.



Figur 3.4 Lydhastighetsprofilen utenfor Marsteinen hentet ut fra PROTEUS.

4 Simulerte sonarekko - Bjørnafjorden

For å teste sonarsimulatoren i PROTEUS har vi implementert en tilsvarende metode i MATLAB som gjennomfører alle stegene fra LYBIN-simuleringene til og med tersklingen (altså uten gaussisk støy på sonarekkoposisjonene). Miljøinformasjonen som er brukt er hentet fra datasettet vi fikk fra KDA. Figur 4.1 viser topografien og gjenklngen (unormalisert og normalisert) sammen med de simulerte sonarekkoene for to ulike modellopløsninger for tilfellet i Bjørnafjorden. Legg merke til at sonarekkoene dukker opp på ulike posisjoner for ulike modellopløsninger og at antallet sonarekko er vesentlig lavere for de lavoppløselige enn de høyoppløselige dataene. Samme oppløsningen på topografien (90 m) er brukt i begge tilfeller. Den høye modellopløsningen er på 100 m i rekkevidde. Den lave oppløsningen benytter maks 50 celler. Siden sonarrekkevidden er satt til 12 km, så er oppløsningen 240 m, men i de tilfeller der land er innenfor de 12 km, så blir oppløsningen bedre (men aldri bedre enn 100 m).



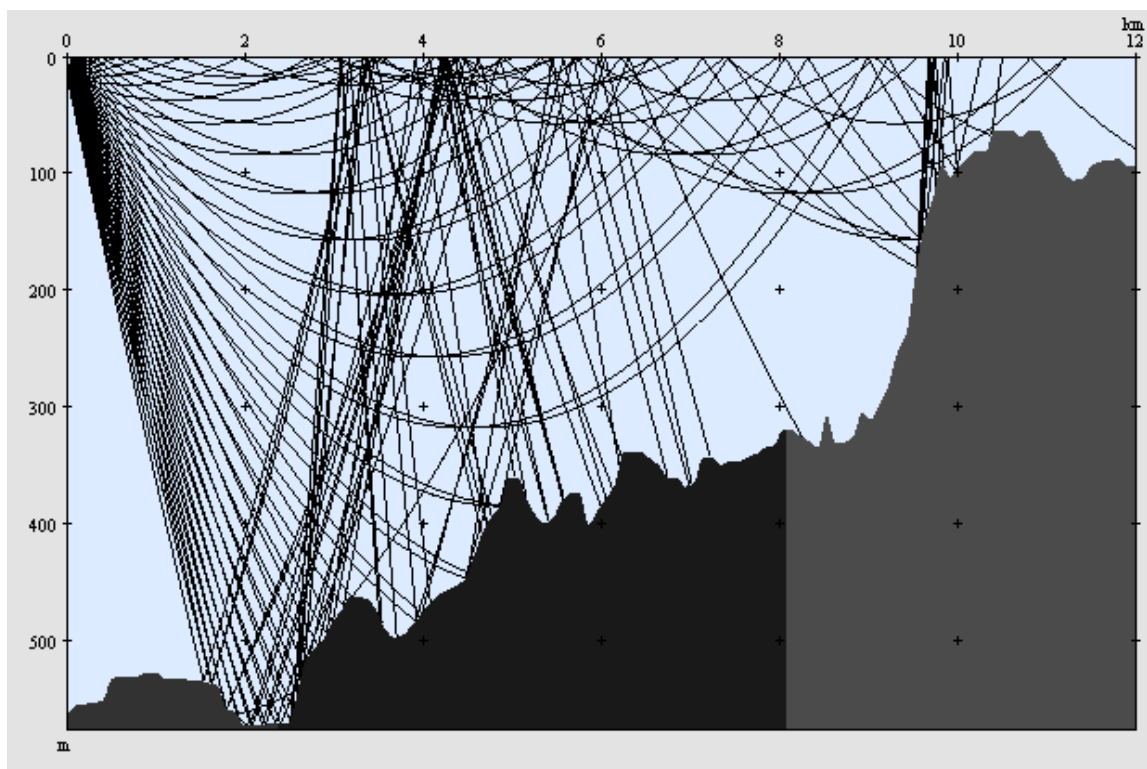
Figur 4.1 Simulerte sonarekko for Bjørnafjorden-tilfellet plottet på topografi (toppen), unormalisert gjenklang (midten) og normalisert gjenklang (bunnen) for høy (venstre) og lav (høyre) modellopløsning.

4.1 Modellopløsning

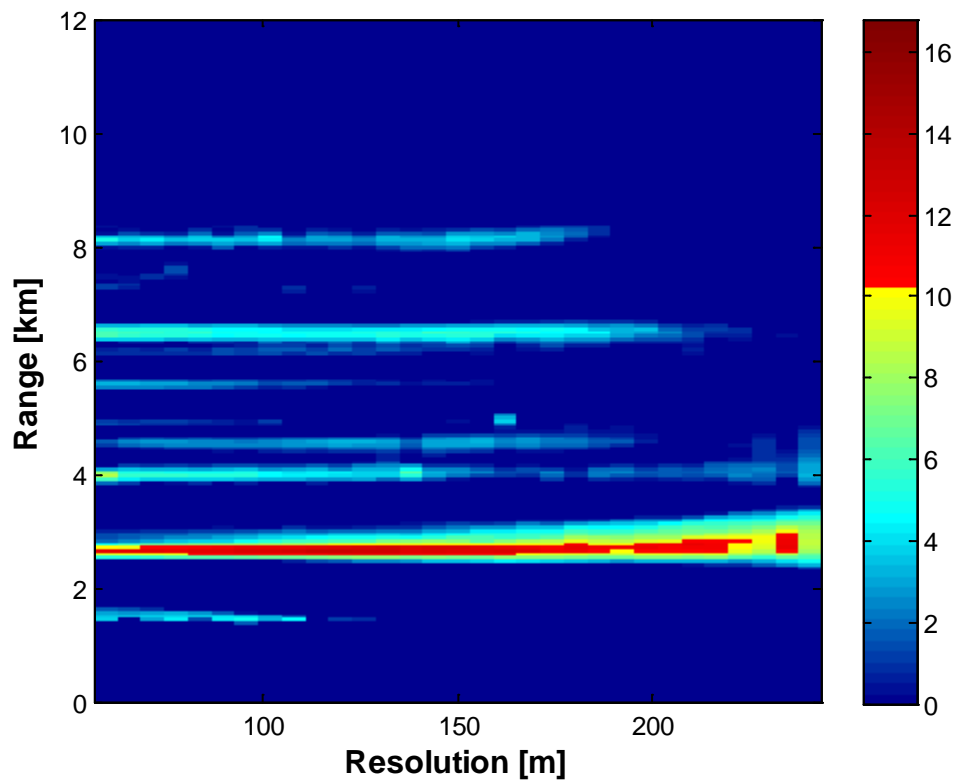
I denne analysen har vi valgt ut en enkelt retning, 78° , i første transmisjon. Figur 4.2 viser et stråleplott for dette tilfellet. Figur 4.3 viser normalisert gjenklang. Noen avstander skiller seg ut med høye normaliserte gjenklingsverdier på enkelte avstander. Disse rekkeviddene er potensielle kandidater for simulerte sonarekko.

Legg merke til at økende oppløsning fører til redusert tykkelse på gjenklingsintensive områder, dette er spesielt tydelig ved avstanden 6.5 km. Et tverrsnitt av konturplottet er vist i Figur 4.4 for denne avstanden. Legg merke til at nivået øker med oppløsningen, dvs at energien blir fokusert og øker sannsynligheten for terskeloverskridelser. Dette er som forventet og problemstillingen er adressert i større detalj i [1]. Siden topp-punktene i normalisert gjenklang varierer med oppløsningen, så bør også terskelen som brukes for å finne ekkoene avhenge av oppløsning.

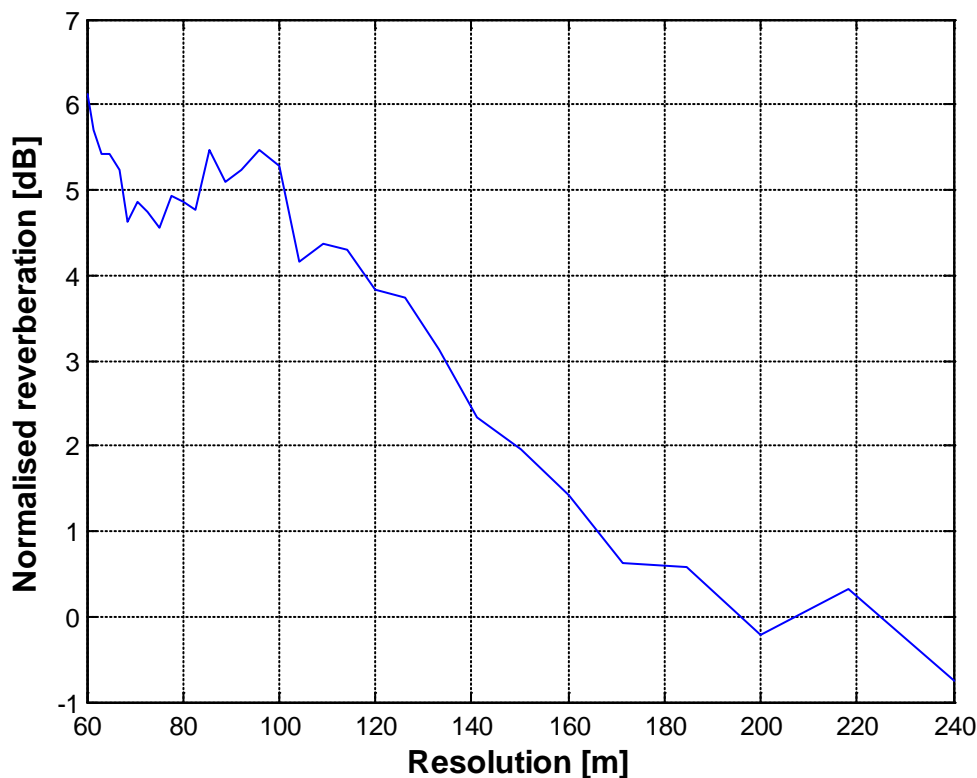
Et annet viktig poeng er at mange av de gjenklingsintensive områdene ikke kommer frem i resultatene fra simuleringer med lav oppløsning. Det gjenklingsintensive området ved ca 6.5 km blir overskygget av støy når oppløsningen blir svakere enn 220 m. Andre områder forsvinner også for simuleringene med enda bedre oppløsning. Dette viser at dersom en skal ha en mest mulig realistisk fordeling av sonarekko så er det også et krav til minimumsoppløsning i kjøringen. For dette tilfellet kan en oppløsning på 100 m virke som en fornuftig grense.



Figur 4.2 Stråleplott for Bjørnafjorden-tilfellet.



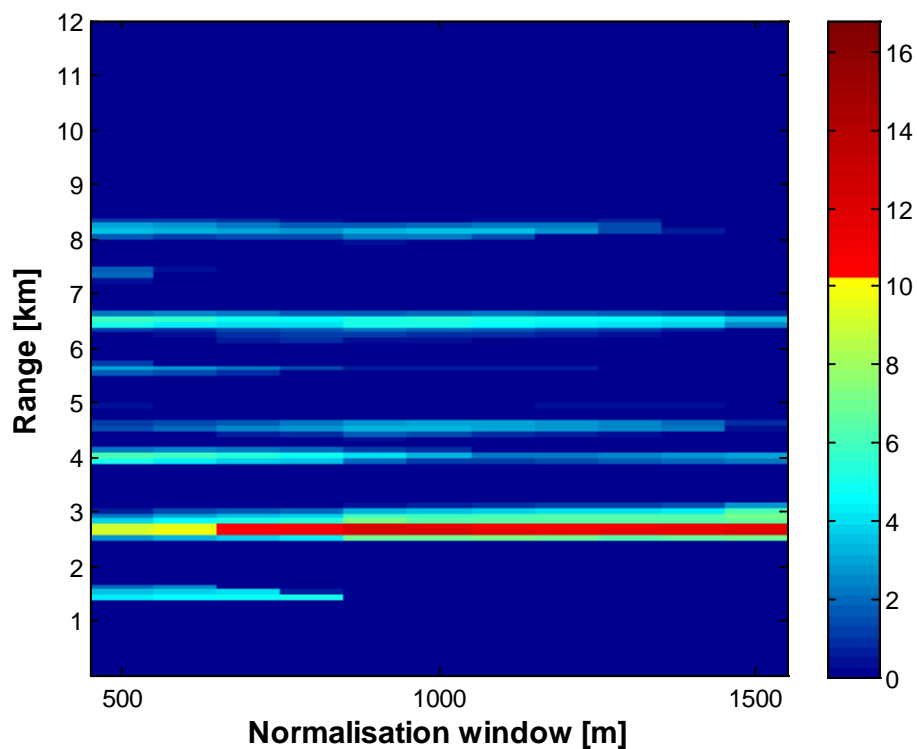
Figur 4.3 Bjørnafjorden. Normaliserte gjenklingsnivåer som funksjon av avstand og oppløsning. Overgangen mellom gult og rødt indikerer deteksjonsterskelen som brukes i PROTEUS.



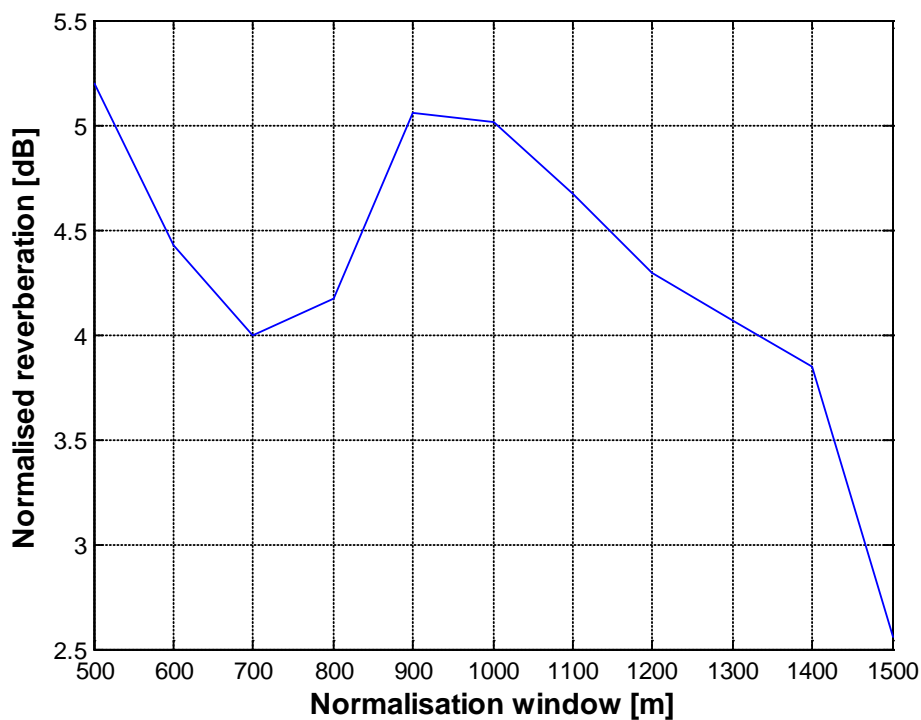
Figur 4.4 Bjørnafjorden. Normaliserte gjenklingsnivåer som funksjon av oppløsning for en rekkevidde på 6.5 km.

4.2 Normaliseringsvindu

I likhet med modellopløsning så har også størrelsen på normaliseringsvinduet innflytelse på den normaliserte gjenklagen, se Figur 4.5. Merk at vaktbåndet holdes på en sjettedel av lengden av normaliseringsvinduet (rundet ned) i denne analysen. Ved for store normaliseringsvindu så kan gjenklagsintensive områder forsvinne siden normaliseringsvinduet da også omfatter celler for korte avstander der gjenklagen er svært høy. På den andre siden, viser Figur 4.6 at det ikke er en en-entydig sammenheng mellom vindustørrelsen og det normaliserte gjenklagsnivået. En vindustørrelse på 900 m maksimerer (lokalt maksimum) den normaliserte gjenklagen ved 6.5 km rekkevidde, men til gjengjeld så mistes flere andre gjenklagsintensive områder. I dette tilfellet kan det virke som at bruk av et kort normaliseringsvindu (500 m) sikrer at flest mulig gjenklagsintensive områder er bevart, men at en til gjengjeld får et lavere nivå enn det en kunne fått med et større vindu.



Figur 4.5 Bjørnafjorden. Normaliserte gjenklagnivåer som funksjon av avstand og størrelse på normaliseringsvinduene. Overgangen mellom gult og rødt indikerer deteksjonsterskelen som brukes i PROTEUS.



Figur 4.6 Bjørnafjorden. Normaliserte gjenklagnivåer som funksjon av oppløsning for en rekkevidde på 6.5 km.

4.3 Deteksjonsterskel

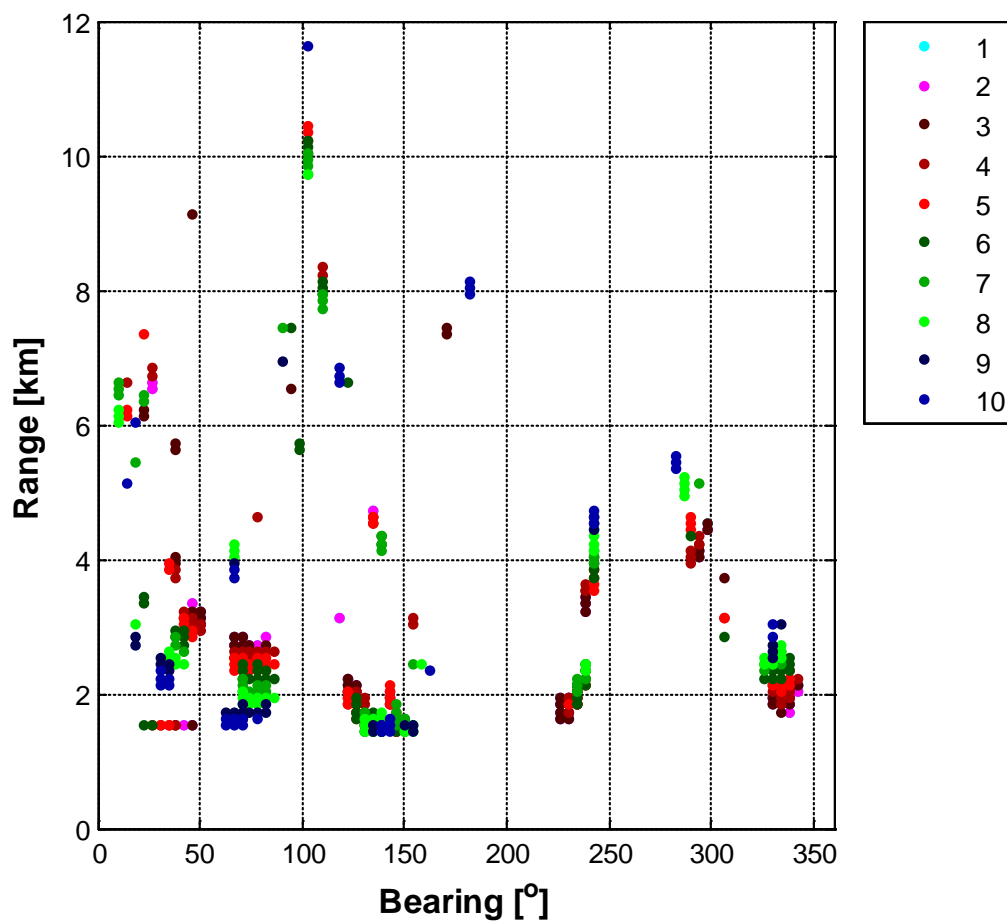
Deteksjonsterskelen er svært avgjørende for hvor mange sonarekko som genereres. KDA, ved Helge Haaland, har foreslått at brukeren selv bør få stille på hvor mange falske alarmer som skal genereres. En brukerbestemt deteksjonsterskel er et naturlig valg for en slik funksjonalitet.

4.4 Eksempel

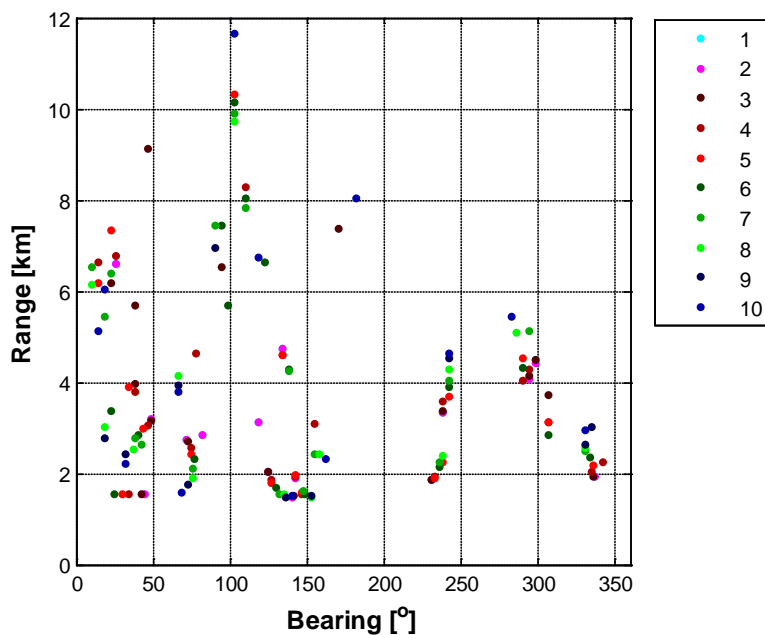
Figur 4.7 viser et eksempel på distribusjon av sonarekko. Legg merke til at en enkel transmisjon kan få sonarekko i naboceller både i rekkevidde og vinkel. Moderne sonarsystemer løser dette ved hjelp av klyngealgoritmer [3] som føyer sammen slike naboer til en enkelt klynge. Det anbefales å gjøre det også for dette tilfellet, slik det er vist i Figur 4.8. For dette eksempelet ble det generert gjennomsnittlig 15 klynger per transmisjon (60 sonarekko per transmisjon). En tracker vil redusere antallet falske alarmer ytterligere, men det er tydelig i figuren at et vesentlig antall falske tracks vil bli generert. Det er dog ikke forsøkt å kjøre noen tracker på dette datasettet.

Figur 4.9 viser sonarekkoklynger for en lavoppløselig modellkjøring. Det er nå noe færre klynger, gjennomsnittlig 12 per transmisjon (50 sonarekko per transmisjon). Det er hovedsakelig klyngene på lang avstand som fjernes, dvs de som forventes å ha lavest normalisert gjenklang. En reduksjon av deteksjonsterskelen fører til en stor økning i antall falske alarmer (mer enn en dobling for en 3 dB reduksjon), men fører ikke til noen økning av antall klynger (fortsatt 12 ved 3 dB reduksjon av deteksjonsterskelen). Årsaken er at de aller fleste sonarekkoene genereres nær hverandre og havner derfor i samme klynge.

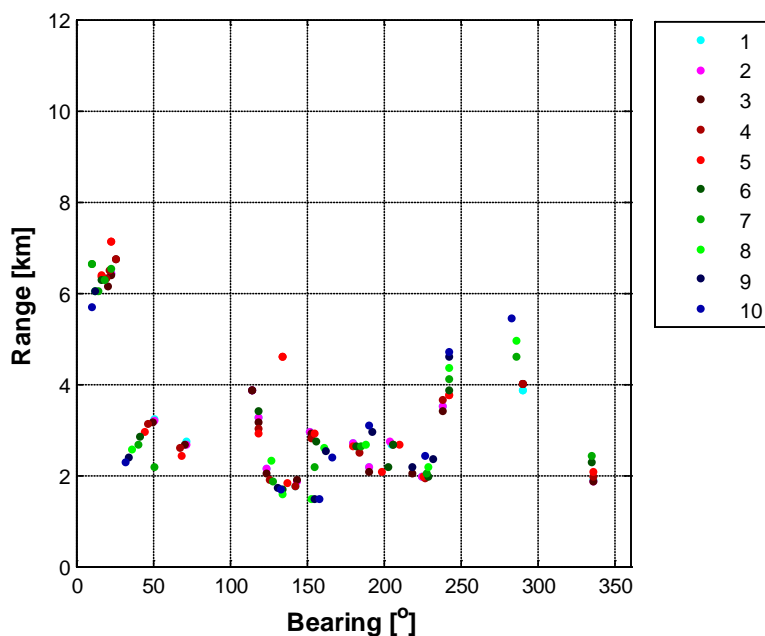
PROTEUS bruker samme størrelse i antall celler på normaliseringsvinduet uansett oppløsning. Dette er uheldig siden ulike oppløsninger fører til ulik lengde på vinduet. Figur 4.10 viser klynger når et smalere normaliseringsvindu benyttes. Denne gangen genereres 11 klynger i snitt (331 sonarekko), men de er nå spredt over større avstander. Ved bruk av for langt normaliseringsvindu så vil de høye gjenklangsverdiene i tidlige celler ta livet av eventuelle gjenklangsintensive områder på lange avstander.



Figur 4.7 Eksempel på distribusjon av sonarekko relativt sonarfartøyets posisjon for alle ti transmisjonene. 10 dB deteksjonsterskel. 100 m oppløsning i rekkevidde. 1000 m normaliseringsvindu.

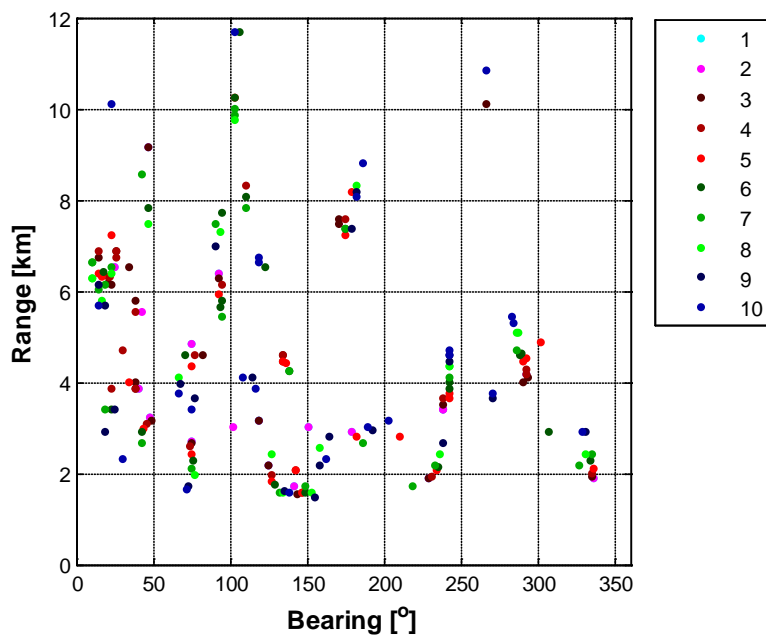


Figur 4.8 Bjørnafjorden. Eksempel på distribusjon av sonarekkoklynger relativt sonarfartøyets posisjon for alle ti transmisjonene. 10 dB deteksjonsterskel. 100 m oppløsning i rekkevidde (120 celler på 12 km avstand). 1000 m normaliseringsvindu (10 celler). MATLAB-funksjonen imdilate.m er brukt¹.



Figur 4.9 Eksempel på distribusjon av sonarekkoklynger relativt sonarfartøyets posisjon for alle ti transmisjonene når en lav rekkeviddeoppløsning er brukt. 10 dB deteksjonsterskel. 240 m oppløsning i rekkevidde (50 celler på 12 km avstand). 2400 m normaliseringsvindu (10 celler). MATLAB-funksjonen imdilate.m er brukt.

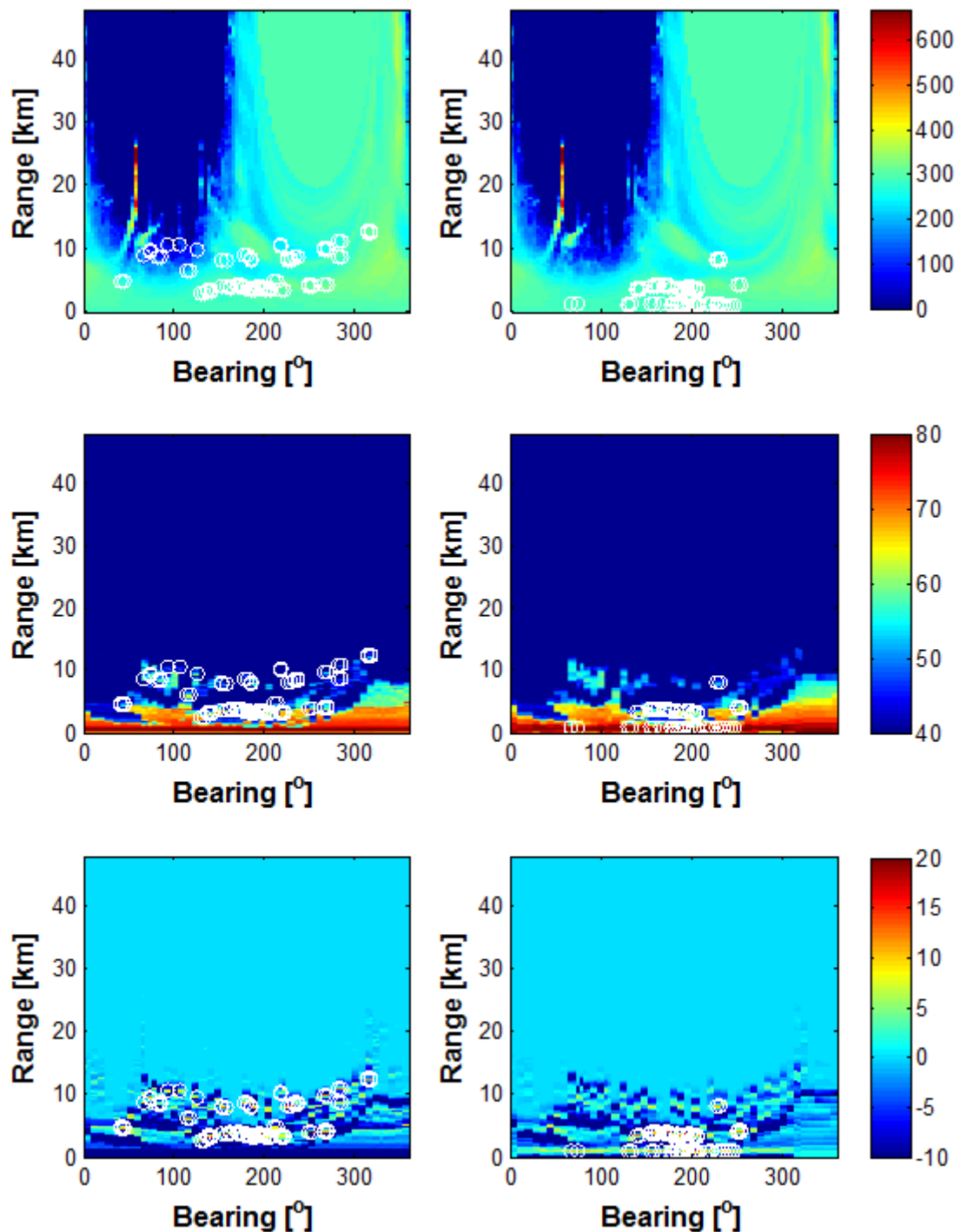
¹ Et rektangel med tre celler i bredden i retning og fem celler i bredden i avstand er brukt.



Figur 4.10 Eksempel på distribusjon av sonarekkoklynger relativt sonarfartøyets posisjon for alle ti transmisjonene når en lav rekkeviddeoppløsning og lite normaliseringsvindu er brukt. 10 dB deteksjonsterskel. 240 m oppløsning i rekkevidde (50 celler på 12 km avstand). 960 m normaliseringsvindu (4 celler). MATLAB-funksjonen `imdilate.m` er brukt.

5 Simulerte sonarekko – Marsteinen

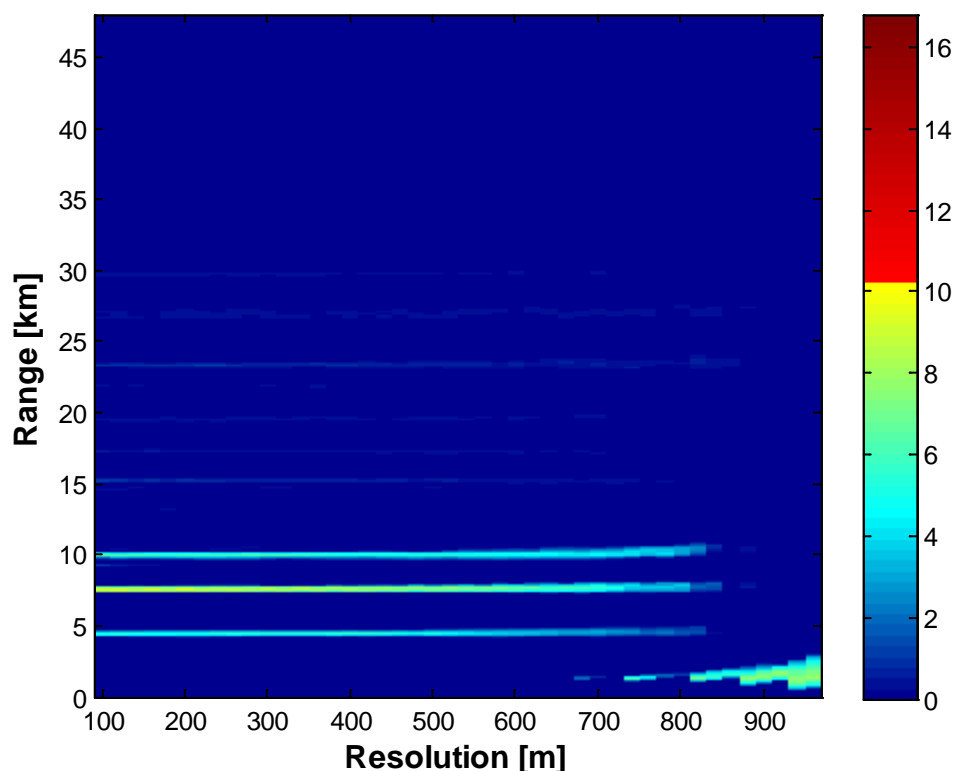
Figur 5.1 viser topografien, gjenklangen og de simulerte ekkene for både høy og lav modellopløsning for Marsteinen-tilfellet. Store deler av området er rimelig flatt (ca 300 m dypt), men den lange rekkeviddeinnstillingen som brukes i dette datasettet betyr at sonaren rekker helt inn til land, der terrenget er svært varierende og ofte bratt (bæring mellom 0° og 150°).



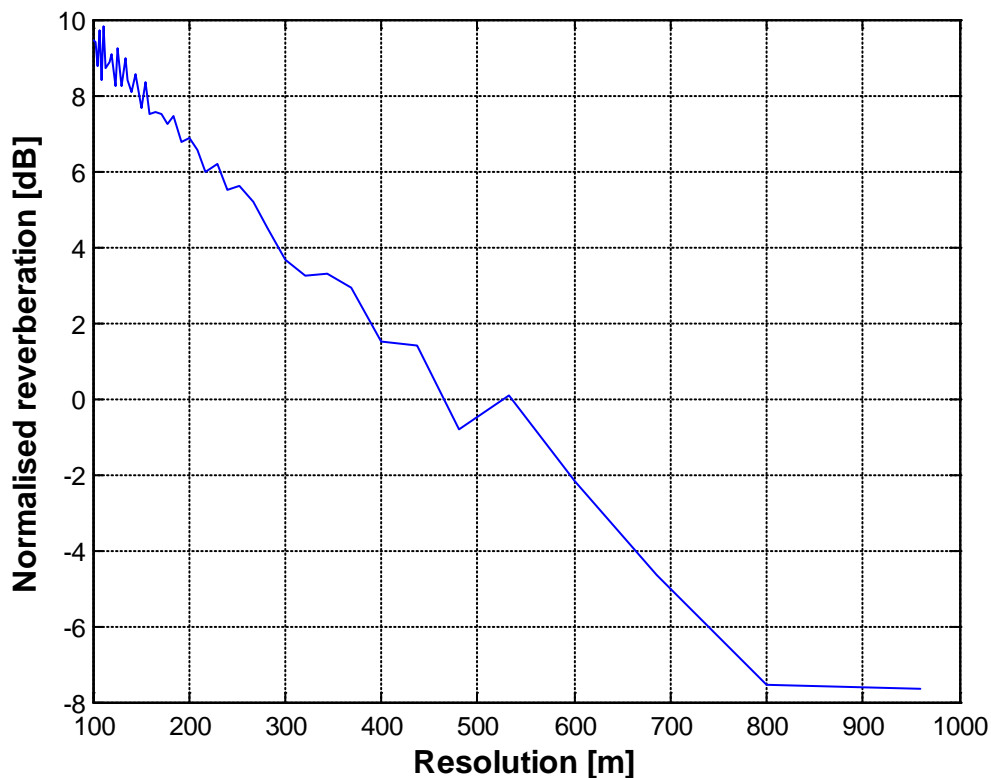
Figur 5.1 Simulerte sonarekko for Marsteinen-tilfellet plottet på topografi (toppen), unormalisert gjenklang (midten) og normalisert gjenklang (bunnen) for høy (venstre) og lav (høyre) modellopløsning.

5.1 Modellopløsning

I likhet med analysen av Bjørnafjorden-tilfellet, varieres også her modellopløsningen. Til forskjell, har vi her fulgt anbefalingen fra forrige kapittel om å la normaliseringsvinduet ha en fast størrelse på 1000 m (dvs varierende antall celler). Figur 5.2 viser normaliserte gjenklangsverdier i bæring (relativt nord) 218°. Som for Bjørnafjorden-tilfellet, så faller nivået på de gjenklangsintensive områdene når modellopløsningen blir svakere. Unntaket er et område på svært kort avstand (ca 2 km), der vi ser økt gjenklang for svake oppløsninger. Figur 5.3 viser hvordan den normaliserte gjenklangen endrer seg som funksjon av oppløsning. Ved å redusere deteksjonsterskelen vil en fortsatt kunne generere et ekko i denne posisjonen, men ikke for alle oppløsninger. Et minstekrav må være at deteksjonsterskelen er positiv, og den bør være et stykke over 0 dB for å unngå at tilfeldige utfall i ekkostyrken (for eksempel på grunn av hvordan LYBIN tilfeldig perturberer stråler som treffer overflaten) leder til for mange genererte ekko. For dette tilfellet vil kanskje en deteksjonsterskel på 6 dB og en minimumsoppløsning på 200 m være passende, men i Bjørnafjorden-tilfellet er en deteksjonsterskel på 10 dB tilstrekkelig til å få en grei fordeling av ekko. Valget av deteksjonsterskel bør derfor varieres fra scenario til scenario, og kanskje også varieres med hvilken type trening en ønsker. Dette underbygger forslaget fra KDA om at brukeren selv bør få velge deteksjonsterskelen.



Figur 5.2 Marsteinen. Normaliserte gjenklangs nivåer som funksjon av avstand og oppløsning. Overgangen mellom gult og rødt indikerer deteksjonsterskelen som brukes i PROTEUS.



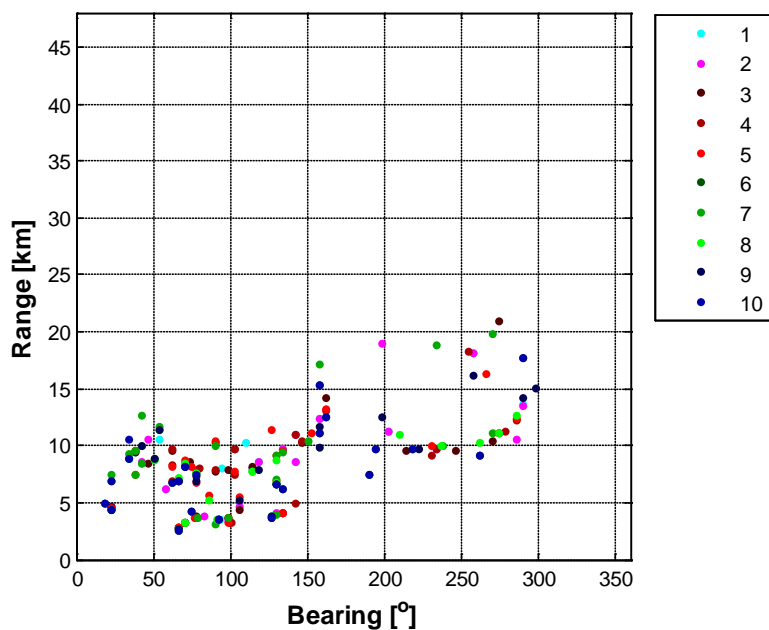
Figur 5.3 Marsteinen. Normaliserte gjenklangs nivåer som funksjon av oppløsning for en rekkevidde på 7.6 km.

5.2 Eksempel

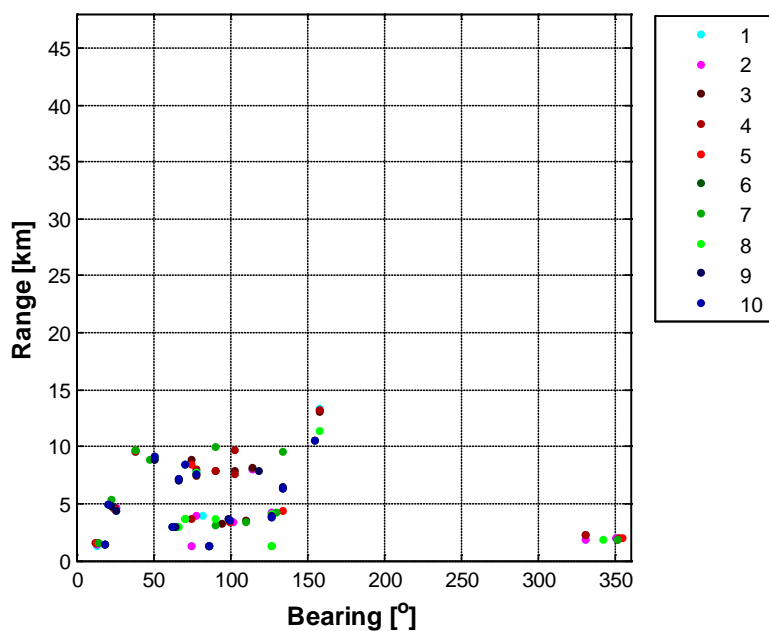
Figur 5.4 og Figur 5.5 viser henholdsvis ekkofordelingen (etter klynging) for det høyoppløselige tilfellet (100 m oppløsning) og når samme oppløsning som i PROTEUS brukes. Legg merke til at svært mange ekko går tapt i det lavoppløselige tilfellet (PROTEUS-oppløsningen) sammenlignet med det høyoppløselige tilfellet. På det svakest er PROTEUS-oppløsningen på 960 m.

Oppløsningen er noe bedre i de retningene der modellrekkevidden settes ned grunnet avstanden til land (0° til 150°), og det er også disse retningene det blir generert ekko for. Resultatene her viser at en oppløsning på 960 m ikke er godt nok for dette tilfellet. Et tilfelle med modellopløsning på 200 m er også inkludert (Figur 5.6). Legg merke til at sammenlignet med tilfellet med 100 m oppløsning så får en her vesentlig færre sonarekko. Det kan dog løses enkelt ved å redusere deteksjonsterskelen, se Figur 5.7. Dessverre kan dette dog ikke løses på samme måten for den lave oppløsningen som PROTEUS bruker.

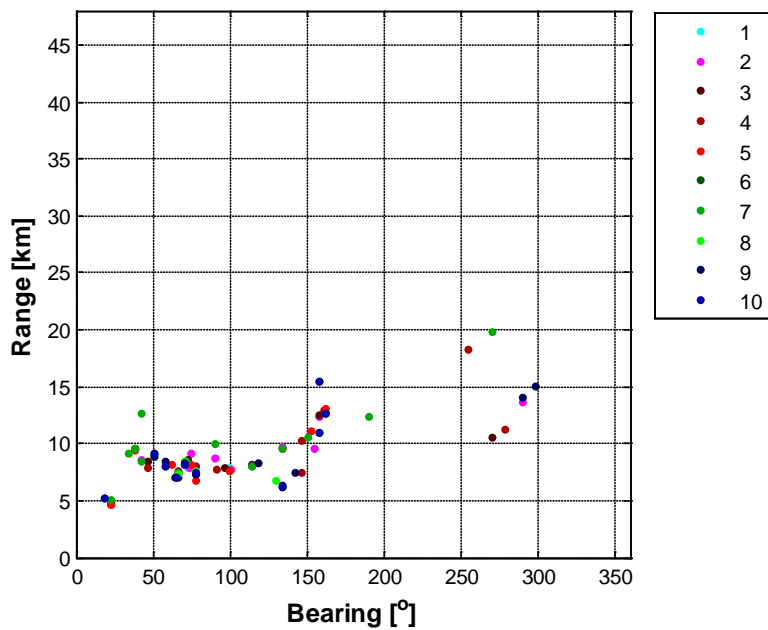
Ved lange rekkeviddeinnstillinger så er det også mer tid mellom hver transmisjon, noe som betyr at PROTEUS kan bruke lengre tid per LYBIN-kjøring og dermed bruke høyere oppløsning.



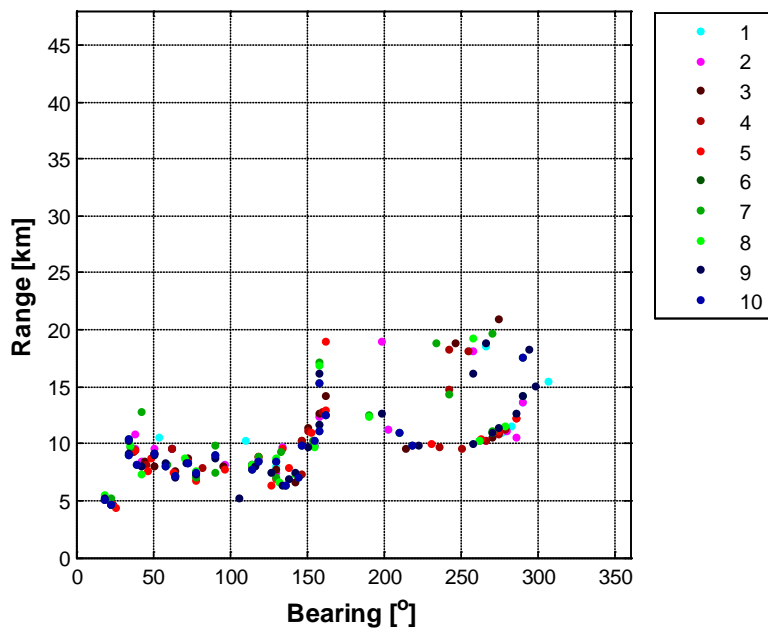
Figur 5.4 Marsteinen. Eksempel på distribusjon av sonarekkoklynger relativt sonarfartøyets posisjon for alle ti transmisjonene. 10 dB deteksjonsterskel. 100 m oppløsning i rekkevidde (120 celler på 12 km avstand). 1000 m normaliseringsvindu (10 celler).



Figur 5.5 Marsteinen. Eksempel på distribusjon av sonarekkoklynger relativt sonarfartøyets posisjon for alle ti transmisjonene. 10 dB deteksjonsterskel. Oppløsningen PROTEUS bruker er også brukt her. 1000 m normaliseringsvindu (10 celler).



Figur 5.6 Marsteinen. Eksempel på distribusjon av sonarekkoklynger relativt sonarfartøyets posisjon for alle ti transmisjonene. 10 dB deteksjonsterskel. Modellopløsningen er her 200 m i rekkevidde. 1000 m normaliseringsvindu (10 celler).



Figur 5.7 Marsteinen. Eksempel på distribusjon av sonarekkoklynger relativt sonarfartøyets posisjon for alle ti transmisjonene. 7 dB deteksjonsterskel. Modellopløsningen er her 200 m i rekkevidde. 1000 m normaliseringsvindu (10 celler).

6 Konklusjon

Ut i fra analysene som er gjort foreslår vi følgende tiltak for å forbedre realismen til sonarsimulatoren:

1. La rekkeviddeoppløsningen alltid være 200 m eller bedre i rekkevidde. PROTEUS bruker svakere oppløsning på lengre rekkevidder, men i disse tilfellene er det også lengre tid mellom transmisjoner, noe som betyr mer tid til LYBIN-beregninger. Derfor burde modellopløsningen kunne økes noe uten at det går utover kjøretiden.
2. Normaliseringsvinduet må alltid har samme størrelse i meter. Et normaliseringsvindu på 1000 m og et vaktbånd på 170 m ser ut til å være fornuftig for datasettene studert her.
3. Bruk av klynging for å samle ekko i naboskap i en enkelt klynge og deretter levere klyngene til trackeren kan redusere belastningen på trackeren.
4. Vi støtter KDA sitt forslag om å la deteksjonsterskelen være brukerstyrt slik at den som setter opp scenarioet også kan velge hvor mange falske alarmer AU-laget skal måtte håndtere. Det er tilstrekkelig at deteksjonsterskelen settes i forkant av avspillingen av scenarioet, for eksempel i en kofigurasjonsfil.

Referanser

- [1] K T Hjelmervik, A L Gjersøe og S Alsterberg: "Sonar Simulation for the Action Speed Tactical Trainer at KNM Tordenskjold", FFI rapport 2007/02348, ugradert.
- [2] B R Mahafza: "Radar Systems Analysis and Design Using MATLAB", Chapman & Hall/CRC, 2000.
- [3] S P Beerens og W Boek: "A robust algorithm for LFAS target classification", UDT 2007 proceedings, 2007.