



FFI-RAPPORT

19/00564

Kvalitetsmetriker for målfølging

— en undersøkelse av et utvalg metoder

Ole Halvard Sætran

Kvalitetsmetriker for målfølgning

– en undersøkelse av et utvalg metoder

Ole Halvard Sætran

Emneord

Målfølging

FFI-rapport

19/00564

Prosjektnummer

1467

ISBN

E: 978-82-464-3208-3

Godkjennerne

Arvid Melkevik, *forskningsleder*

Håkon Storli Andersen, *forskningssjef*

Dokumentet er elektronisk godkjent og har derfor ikke håndskreven signatur.

Opphavsrett

© Forsvarets forskningsinstitutt (FFI). Publikasjonen kan siteres fritt med kildehenvisning.

(U) Sammendrag

Målfølging (tracking) er nødvendig i situasjoner hvor man mottar målinger av et system eller område og ønsker å beregne tilstanden. I militær sammenheng kan det for eksempel være snakk om å bruke sensorer som kan gi oversikt over et område.

Prosjektet *Robust og modulbasert målfølging* ved Forsvarets forskningsinstitutt (FFI) har som oppgave å utarbeide et rammeverk for utprøving av forskjellige målfølgingsalgoritmer (trackere). Trackerene bruker ulike assosiasjonsmetoder og/eller ulike estimatorer. I den sammenheng er det nødvendig å kunne sammenligne ytelsen til to eller flere trackere. Metoder for å måle kvaliteten kalles kvalitetsmetrikk, i dette tilfellet kvalitetsmetrikk for tracking.

Denne rapporten presenterer noen forskjellige kvalitetsmetrikker for tracking. Metrikkene kan deles i to hovedtyper som vi har valgt å kalle multimetrikker og enkeltmetrikker. Multimetrikker kjennetegnes ved at vi har mange forskjellige ytelsesmål for en tracker, og beregner alle sammen. Resultatet blir en tabell per tracker som beskriver ytelsen. Hvilke ytelsesmål vi ønsker å inkludere velger vi selv. Enkeltmetrikker gir ett tall for kvaliteten per tidssteg, og kan lett visualiseres som en graf over tidsrommet. I denne rapporten setter vi opp et eksempel på multimetrikk, og ser på tre forskjellige enkeltmetrikker: OSPA-T, GOSPA-T og OSPA-on-OSPA.

I rapporten blir de forskjellige metrikkene anvendt på et eksempelscenario. Eksempelscenarioet består av to mål. Vi har konstruert resultatet fra to fiktive trackere, og beregner kvaliteten til disse resultatene.

I denne rapporten kommer vi ikke med noen anbefaling om hvilke(n) kvalitetsmetrikk(er) man bør benytte seg av. Flere av metrikkene kommer til å bli prøvd ut i FFI-prosjektet, og vi vil forhåpentlig få nok erfaring med de forskjellige metrikkene til å komme med en slik anbefaling på et senere tidspunkt.

(U) Summary

Tracking is necessary in many situations where one receives measurements of a system or area and wishes to estimate the state of the system. A military example of this is using sensors (such as radars) to create an overview of an area.

The project *Robust and module based tracking* at the Norwegian Defence Research Establishment (FFI) is developing a framework with which to test different trackers. The trackers will differ from each other in what estimator they use, what association algorithm they use, or both. In order to be able to compare trackers we need some methods to measure the performance of the trackers. Such methods are called quality metrics for tracking.

This report presents some quality metrics for tracking. The metrics can be separated into two categories, which we call multi-metrics and single-metrics. In multi-metrics one has many performance measures, and calculates each one. The outcome is a table of performance measures per tracker. There is no hard rule for which performance measures to include in such a table. The single-metrics yield one number per time step, and can be visualized as a graph covering the time of interest, which is most likely the entire run of the scenario in question. In this report we detail one set of performance measures that can be used in a multi-metric approach, and three different single-metrics: OSPA-T, GOSPA-T and OSPA-on-OSPA.

The different metrics are applied to a test scenario. The scenario has two targets moving in a scene. We have created the output from two fictitious trackers, and calculate the quality of said output.

We do not make any recommendations regarding which quality metrics to use. In the FFI project we will use several of the metrics as we implement and test various trackers, and will hopefully gain enough experience in the use of quality metrics to be able to make such recommendations at a later time.

Innhold

(U) Sammendrag	3
(U) Summary	4
1 Introduksjon	7
2 Notasjon og eksempelscenario	8
2.1 Notasjon	8
2.2 Eksempelscenario	9
3 Multimetrikker	11
3.1 Drummond	11
3.1.1 Assosiasjon	12
3.1.2 Ytelsesmål beregnet for hvert mål	12
3.1.3 Ytelsesmål som beregnes for hele mengden mål	13
4 Enkeltmetrikker	14
4.1 OSPA	14
4.2 Assosiering	19
4.3 OSPA-T	20
4.3.1 Bakgrunn og motivasjon	20
4.3.2 Den globalt beste assosiasjonen	22
4.3.3 Den assosiasjonsavhengige avstanden $d_{c,\lambda,\alpha}$	23
4.3.4 Fremgangsmåte for å finne OSPA-T-avstanden	24
4.4 GOSPA og GOSPA-T	25
4.5 OSPA-on-OSPA (OSPA ²)	26
4.5.1 Den tidsvektavhengige avstanden $d_{\Delta,w}$	27
4.5.2 OSPA-on-OSPA-avstanden	29
5 Kvalitetsmetrikkene anvendt på eksempelscenarioet	30
5.1 Multimetrikk	31
5.2 Enkeltmetrikk	32
5.2.1 OSPA-T	33
5.2.2 GOSPA-T	35
5.2.3 OSPA-on-OSPA	36
5.2.4 Betraktninger rundt resultatene	38
6 Oppsummering	40
Referanser	41
Vedlegg	



1 Introduksjon

Målfølging (tracking) er nødvendig i mange situasjoner hvor man mottar målinger av et system eller område og ønsker å beregne tilstanden til systemet. I den militære verden kan det være snakk om å bruke sensorer til å skape et oversiktsbilde. Et eksempel er å få oversikt over luftbildet ved bruk av radarer. En eller flere radarer søker over et område, og rapporterer deteksjoner videre. Disse deteksjonene kan være ekte mål, eller falske mål grunnet støy eller clutter. Oppgaven til en tracker er å ta disse deteksjonene, beregne hvilke deteksjoner som stammer fra samme mål, og for hvert track samle disse. I denne prosessen bør falske deteksjoner bli forkastet, og man sitter igjen med et oversiktsbilde som forhåpentlig samsvarer godt med virkeligheten.

FFIs prosjekt 1467 – *Robust og modulbasert målfølging* har som oppgave å utarbeide et rammeverk for utprøving av forskjellige trackere. Trackerene vil bruke ulike assosiasjonsmetoder og/eller ulike estimatorer. I den sammenheng er det nødvendig å kunne sammenligne ytelsen til to eller flere trackere. Metoder for å måle kvaliteten kalles kvalitetsmetrikk, i dette tilfelle kvalitetsmetrikk for tracking.

I denne rapporten presenterer vi noen kvalitetsmetrikker som blir benyttet i litteraturen. Vi setter opp noen scenarier og viser hvordan de forskjellige metrikkene scorer resultatet.

2 Notasjon og eksempelscenario

2.1 Notasjon

Denne rapporten er skrevet med bakgrunn i arbeid gjort innenfor luftmilitær overvåking. Det typiske scenarioet innenfor luftmilitær overvåking er at man har en eller flere sensorer som overvåker et stort luftområde. Sensorene kan være av flere typer, for eksempel radar eller ESM-mottaker. Et mål vil være noe som beveger seg i luftrommet – typisk et fly, men det kan også være andre objekter, for eksempel et helikopter, en drone eller en fugl. Målene vil bli levert fra sensor til tracker som en punktdeteksjon. De vil med andre ord ikke ha noen utstrekning.

Kvalitetsmetrikkene beskrevet er ikke begrenset til luftmilitær bruk. De kan brukes på alle scenarier hvor man har sannhetsdata fra mål og track fra en tracker. Bakgrunnen nevnes kun for å tydeliggjøre forfatterens innfallsvinkel.

Under følger først definisjonen på track og mål, tilstandsvektorer og trackmengder, før et scenario beskrives for å eksemplifisere disse størrelsene.

X, Y Track og mål. Track er resultatet fra en tracker, mens mål er sannhetsdata fra et reelt eller simulert mål. Begge inneholder den samme informasjonen: Et tracknummer som identifiserer hvilket track eller mål det er snakk om, en serie tidsstempler som forteller hvilke tider vi har informasjon om tracket, og en serie tilstandsvektorer som beskriver tilstanden til tracket eller målet ved et gitt tidssteg. Merk at tracknummeret er ment til intern identifisering, for eksempel å skille mellom to track fra trackeren. Selv når et track følger et mål helt perfekt og med stor sikkerhet er et track av dette målet, vil det ikke sies å være koblet sammen med målet før en assosiering er gjort. Assosiering mellom track og mål beskrives senere i rapporten.

x, y Tilstandsvektorer. En tilstandsvektor inneholder informasjon om tracket eller målet ved ett gitt tidspunkt. Et vanlig eksempel på tilstandsvektor i horisontalplanet (uten høyde) er

$$\mathbf{x} = [pos_e \quad pos_n \quad v_e \quad v_n] \quad (2.1)$$

I dette tilfelle er koordinatsystemet *East–North*, og variablene i vektoren er henholdsvis posisjon i østlig retning, posisjon i nordlig retning, fart i østlig retning og fart i nordlig retning.

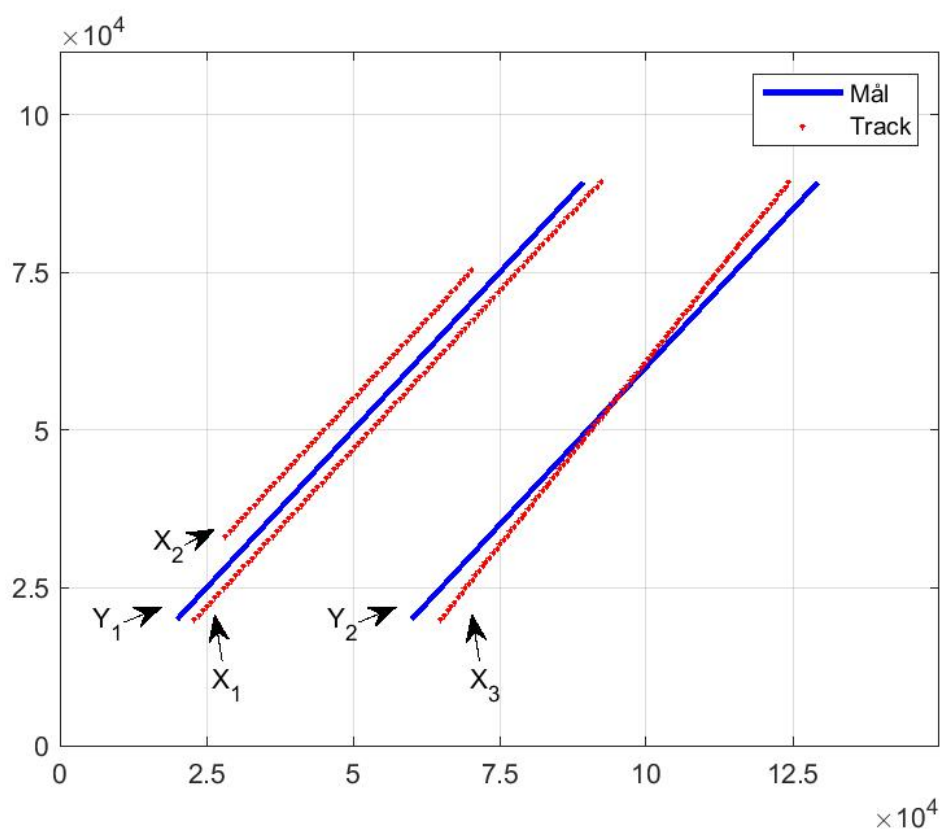
X, Y De to mengder med henholdsvis track og mål som vi ønsker å kvantisere forskjellen på.

Tidssteg blir i denne rapporten betegnet som $k \in (1, \dots, K)$, hvor K er det totale antall tidssteg i scenarioet man analyserer. Tidsindeksen er alltid den siste av indeksene. Under følger eksempler på tidsindeksering.

Track: $X_{m,k}$ Track m ved tidssteg k . Inneholder enten den tomme mengde \emptyset hvis tracket ikke har noen tilstandsvektor (altså ikke eksisterte) i dette tidssteget, eller tilstandsvektoren, $X_{m,k} = \mathbf{x}_{m,k}$.

Trackmengde: \mathbf{X}_k Dette er mengden med track ved tidssteg k , og består av trackenes tilstand ved dette tidssteget. Hvis det totalt eksisterer m track har vi $\mathbf{X}_k = \{X_{1,k}, \dots, X_{m,k}\}$. Noen av disse trackene kan være tomme. Hvis vi har fire track X_1, X_2, X_3 og X_4 , og $X_{3,k} = \emptyset$, skriver vi $\mathbf{X}_k = \{X_{1,k}, X_{2,k}, X_{3,k}, X_{4,k}\} = \{\mathbf{x}_{1,k}, \mathbf{x}_{2,k}, \mathbf{x}_{4,k}\}$.

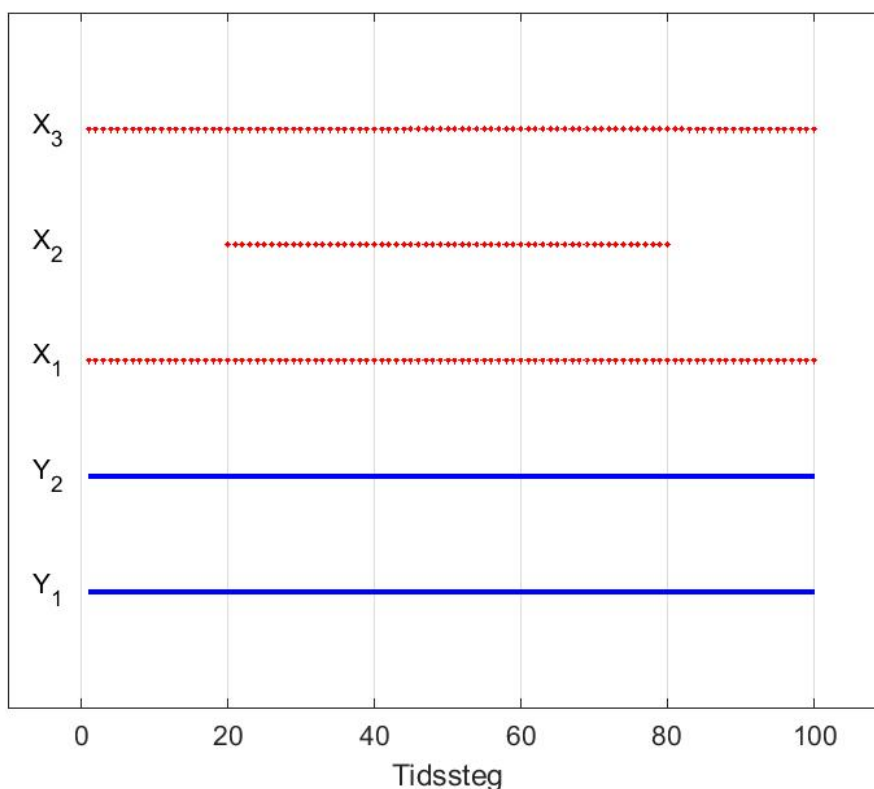
2.2 Eksempelscenario



Figur 2.1 Eksempelscenario.

Figur 2.1 viser et scenario med to ekte mål, vist ved de heltrukne blå strekene, og tre track som har kommet fra en tracker. Figuren viser følgende størrelser:

-
-
- Y_1 og Y_2 : Dette er de to blå strekene som viser to mål, og hver er en serie av 100 tidssteg, $Y_1 = (y_{1,1}, y_{1,2}, \dots, y_{1,100})$
- X_1, X_2 og X_3 : Dette er de tre røde trackene vist i figuren over. Merk at X_1 og X_3 dekker eksakt det samme tidsvinduet som Y_1 og Y_2 dekker, begge er 100 tidssteg lange. X_2 begynner 20 tidssteg senere og terminerer 20 tidssteg tidligere. Se figur 2.2.
- \mathbf{Y} Dette er mengden av måldata, $\mathbf{Y} = \{Y_1, Y_2\}$
- \mathbf{X} Dette er mengden av track fra trackeren, $\mathbf{X} = \{X_1, X_2, X_3\}$



Figur 2.2 Varigheten til de forskjellige trackene i eksempelscenarioet.

3 Multimetrikker

Kvalitetsmetrikk for tracking er delt inn i to hovedtyper: Metoder som ser på de mange forskjellige aspektene ved tracking og gir en score for hvert av disse, og metoder som prøver å inkludere alt dette ned i én (tidsavhengig) verdi. I denne rapporten kalles disse to hovedtypene henholdsvis multimetrikk og enkeltmetrikk. I dette kapitlet beskriver vi et sett av multimetrikk-metoder. I neste kapittel beskriver vi noen enkeltmetrikk-metoder.

Tanken bak multimetrikker er at det er mange forskjellige aspekter ved ytelsen til en tracker som er interessante å se på: Nøyaktighet i rom og tid, mengden falske track, hyppighet av brudd i tracking av et mål, og mange flere. I denne rapporten fokuserer vi på metrikker som kan benyttes på alle trackere, uavhengig av algoritmevalg. Det finnes metrikker utarbeidet for spesifikke trackingalgoritmer, men disse kan kun brukes mellom trackere som bruker samme algoritme, hvilket er en stor begrensning.

Enkeltmetrikkene vi skal se på i neste kapittel har spesifikke navn og formler. Multimetrikkene er ikke like lett å systematisere. En metrikk som de fleste vil benytte seg av er nøyaktigheten til track, men hvordan definerer vi denne? Er det den euklidske avstanden mellom posisjonen til tracket og posisjonen til målet som tracket assosieres med? Eller skal vi se på posisjonene og deres kovariansmatriser som sannsynlighetsfordelinger og benytte oss av Bhattacharyya-avstanden, som måler likheten til de to fordelingene? Multimetrikker gir deg muligheten til å se på mange aspekter ved ytelsen til en tracker, men det tvinger deg også til å tydelig definere hva du måler, og til å redegjøre for de valg du tar i utvelgesprosessen. Multimetrikker blir dermed et veldig bredt begrep som inneholder mye forskjellig, og det er ingen regler for hva du kan velge å benytte deg av.

Metodene beskrevet i dette kapitlet er hentet fra Drummond[1], og kan ses på som en parafrasering av den artikkelen.

3.1 Drummond

Drummond beskriver flere ytelsesmål som kan benyttes for å se på kvaliteten til en tracker. Ytelsesmålene er delt inn i to kategorier – ytelsesmål som beregnes for hvert track, og ytelsesmål som beregnes for hele mengden av track. Deretter tar man enten summen eller gjennomsnittet, avhengig av ytelsesmålet. Det er ofte ønskelig med et stort datagrunnlag for å sikre at resultatene er representative. En vanlig fremgangsmåte er å kjøre et antall Monte Carlo simuleringer av scenarioet, hvor målene er de samme i hver kjøring, men deteksjonene ikke er det. For hver mulig deteksjon blir målet enten detektert med en sannsynlighet P_D , eller ikke detektert med en sannsynlighet på $1 - P_D$. I tillegg kan og bør vi legge på tilfeldig støy som er representativ for sensoren som er tenkt brukt i scenarioet. Vi er dermed sikret forskjellig datagrunnlag for hver Monte Carlo simulering.

Før vi kan se på disse ytelsesmålene må vi assosiere track med mål. Under følger først en gjennomgang av denne assosieringen, før de forskjellige kategoriene ytelsesmål blir gjennomgått i større detalj.

3.1.1 Assosiasjon

For å kunne sammenlikne estimerte track fra en tracker med sannhetsdata fra målene må vi først assosiere track og mål. Drummond skisserer fire måter å assosiere på:

1. Assosiasjonen gjøres for hvert tidssteg, og gjøres på basis av avstand mellom track og mål i dette tidssteget. I hvert tidssteg kan hvert track kun assosieres med ett mål, og omvendt. Effekten av dette er at et mål kan assosieres med mange track over tidsrommet, og dette blir ikke straffet. I for eksempel militær luftromsovervåking er track-kontinuitet verdsatt, og denne metoden vil ikke plukke opp dårlig track-kontinuitet.
2. Assosiasjonen gjøres over hele tidsrommet. Hvert track kan bare assosieres med ett mål, og omvendt. Denne assosiasjonen tillater ikke track swap i det hele tatt, og vektlegger dermed track-kontinuitet veldig høyt. Ulempen er at hvis et mål har to gode track i nærheten av seg, hvor det ene dekker de første 60% av tidsrommet og det andre dekker de siste 40%, så vil kun det første tracket assosieres til målet, og trackeren vil bli straffet for å ikke tracke målet i resten av tidsrommet.
3. Assosiasjonen gjøres over hele tidsrommet. Hvert mål kan assosieres med flere track, men kun om det endelige resultatet gir en gyldig trackserie. Med gyldig trackserie menes at målet kun kan assosieres med et nytt track når det gamle har dødd ut, og at hvert track kun kan assosieres med ett mål. Det er mulig å tillate en liten periode med overlapp for å plukke opp forsinket track-død.
4. Assosiasjonen gjøres over hele tidsrommet. Hvert mål kan assosieres med flere track, og man følger en eller flere regler for å begrense mengden track swap. Drummond[1] oppgir en del slike regler, hvor man må se på hva som er relevant for det aktuelle scenarioet man ser på.

Hver av metodene kommer med fordeler og ulemper. Når man velger metode er det viktig å være klar over disse, og sørge for at svakhetene ved en metode ikke blir utslagsgivende for resultatet av kvalitetsmålingen.

3.1.2 Ytelsesmål beregnet for hvert mål

- **Tracknøyaktighet:** Beregn RMS-avviket (Root Mean Square) mellom mål og track. Dette kan gjøres for både posisjon og hastighet. Hvis flere Monte Carlo simuleringer har blitt kjørt, legg sammen gjennomsnittlig kvadrert avvik for hver kjøring, del på antall kjøring, og ta kvadratroten.
- **Normalisert konsistens til avviks-kovariansen:** Denne beregnes for hvert track i hvert tidssteg. Finn avviksvektoren til tilstandsvektoren, x_{err} , som er gitt ved avviket mellom tilstandsvektoren og målets reelle tilstand. Deretter beregnes den normaliserte konsistensen til avviks-kovariansen, NK , som følger:

$$NK = \frac{x_{err} P^{-1} x_{err}^T}{|x_{err}|} \quad (3.1)$$

hvor P^{-1} er den inverterte kovariansmatrisen til tracket, og x_{err}^T er den transponerte til x_{err} . Det beste resultatet er $NK = 1$. Jo høyere verdien er over 1, jo dårligere er trackeren. Hvis verdien er under 1 er kovariansmatrisen ukorrekt.

-
-
- **Bias til tilstandsvektoren:** Beregn biasen til tilstandsvektoren i hver retning. Dette gjøres ved optimeringsmetoder. Hvis flere Monte Carlo simuleringer har blitt kjørt, legg sammen kvadrert bias for hver kjøring, del på antall kjøring, og ta kvadratroten.
 - **Track Jitter:** Dette er et ytelsesmål som registrerer om et track har en glatt bane eller går i 'sikk-sakk'. I hver retning i koordinatsystemet, ta differansen mellom posisjonsavviket ved et tidssteg og posisjonsavviket ved forrige tidssteg. Posisjonsavviket er forskjellen mellom estimert posisjon og reell posisjon. Kvadrer dette og finn snittet over tidsrommet og ta kvadratroten. Hvis flere Monte Carlo simuleringer har blitt kjørt, ikke ta kvadratroten, men legg sammen dette kvadrerte jitteret for hver kjøring og del på antall kjøring. Ta så kvadratroten.
 - **Trackbytter:** For hvert mål (i hver Monte Carlo simulering), beregn hvor ofte det assosierte tracknummeret i et tidssteg er forskjellig fra det assosierte tracknummeret i forrige tidssteg.

3.1.3 Ytelsesmål som beregnes for hele mengden mål

- **Antall ikke-detekterte mål:** Summer opp antall mål som ikke har blitt assosiert med noe track. Hvis det har blitt kjørt flere Monte Carlo simuleringer, ta gjennomsnittet over kjøringene.
- **Antall falske track:** Summer opp antall track som ikke har blitt assosiert med noe mål. Hvis det har blitt kjørt flere Monte Carlo simuleringer, ta gjennomsnittet over kjøringene.
- **Gjennomsnittlig trackinitieringstid:** Gjennomsnittlig tid som trackeren bruker fra et mål er detektert til et modent track er etablert. Hvis trackeren mister målet for deretter å initiere et nytt track på samme mål litt senere brukes denne nye initieringstiden også i beregningene. Ta gjennomsnittet over alle track for hver Monte Carlo kjøring, og deretter gjennomsnittet over alle Monte Carlo kjøringene.
- **Gjennomsnittlig trackvarighet:** Gjennomsnittlig tid et track som er tilknyttet et mål lever. Hvis et mål blir fulgt av flere sekvensielle track vil dette trekke ned den gjennomsnittlige trackvarigheten kraftig. Ta gjennomsnittet over alle track for hver Monte Carlo kjøring, og deretter gjennomsnittet over alle Monte Carlo kjøringene.

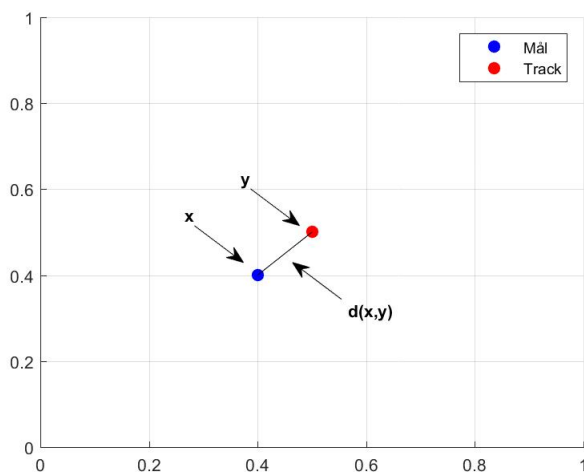
4 Enkeltmetriker

I dette kapittelet beskriver vi tre forskjellige enkeltmetriker for å måle kvaliteten til en tracker: OSPA-T, GOSPA-T og OSPA-on-OSPA. Før vi kommer til disse beskriver vi deres felles opphav, OSPA, som er en metode for å beregne kvaliteten til en estimator. Forkortelsene vil bli beskrevet i sine respektive kapitler.

4.1 OSPA

Enkeltmetrikkene som beskrives i denne rapporten har alle utspring i en metode for å måle avstanden mellom to mengder med tilstandsvektorer. Denne metoden kalles OSPA, som står for *Optimal SubPattern Assigment*. OSPA kan i seg selv ikke benyttes på track og mål, da den kun ser på ett tidssteg. Den er beskrevet her fordi de etterfølgende metodene er utvidelser av OSPA, og en god forståelse av denne enklere metoden er til god hjelp. Først følger en beskrivelse av metoden og hvordan den håndterer noen problemstillinger når man skal måle avstanden mellom to mengder tilstandsvektorer. Deretter beskriver vi formelen for å beregne OSPA-avstanden, og gir en trinn-for-trinn oppskrift på hvordan den kan implementeres.

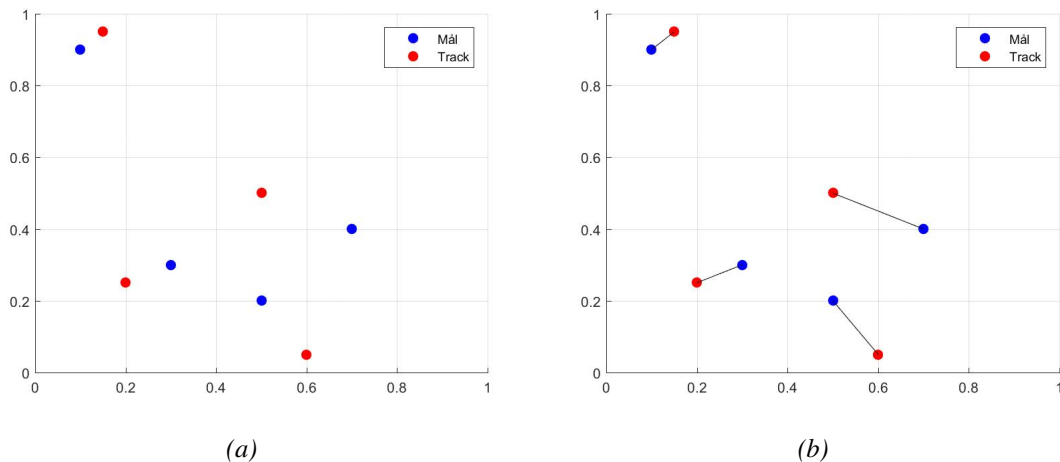
Med avstand menes vanligvis euklidsk avstand, som i denne rapporten vil betegnes som $d(\mathbf{x}, \mathbf{y})$. Denne avstanden tar kun hensyn til posisjonsdelen av tilstandsvektoren. Det er ikke nødvendig for å kunne bruke OSPA (eller de andre enkeltmetrikkene) at man benytter seg av denne avstanden. Det er fullt mulig å bruke en avstandsdefinisjon som også benytter seg av eventuelle hastighetskomponenter av tilstandsvektoren, eller akselerasjon. I denne rapporten og betraktningene rundt enkeltmetriker benytter vi oss av den euklidske avstanden.



Figur 4.1 Avstanden mellom et track og et mål er lengden på den rette linjen dem imellom.

Avstanden mellom to punkter er lengden på den rette linjen mellom de to punktene, se figur 4.1. Hvis vi har en enkeltmålstracker, er det å bruke avstanden som mål på trackkvalitet rett fram - bare finn

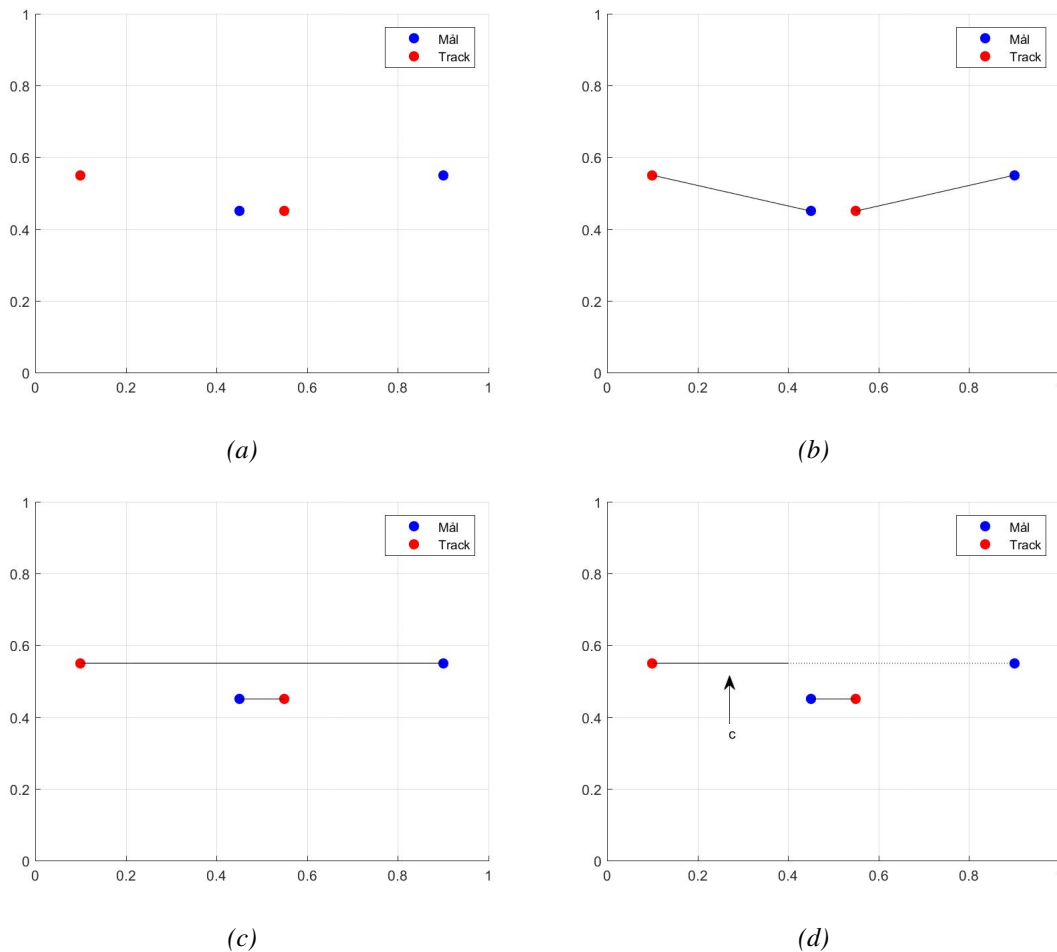
den euklidske avstanden mellom målet og tracket ved ethvert tidssteg. Dermed har man avstanden over tid, og kan beregne gjennomsnittsavstanden hvis man ønsker ett enkelt tall på kvaliteten. Men hvis du har flere mål, og dermed også flere track, hva er avstanden da ved et gitt tidssteg?



Figur 4.2 Vi har fire track og fire mål i (a). I (b) ser vi den beste assosiasjonen mellom disse fire trackene og de fire målene.

OSPA, som måler avstanden mellom to mengder tilstandsvektorer, beskriver en måte å gjøre det på. For det aktuelle tidssteget finner man den beste assosiasjonen mellom trackene og målene, og bruker den gjennomsnittlige avstanden mellom de assosierte track-mål-parene som et mål for avstanden mellom mengdene i dette tidssteget (se figur 4.2). Den beste assosiasjonen er definert som den assosiasjonen som gir den lavest mulige gjennomsnittlige avstanden. Det vil si at man må finne avstanden fra hvert track til hvert mål, og finne den gjennomsnittlige avstanden for alle mulige kombinasjoner hvor hvert track er assosiert med ett mål.

Det finnes noen problemer som må tas stilling til. Det kan tenkes at man for eksempel har ett mål som har et track veldig nærme, og begge disse ligger sentralt i det aktuelle området. Hvis man i tillegg har et track som ligger helt for seg selv, og på andre siden av området har et mål som også ligger helt isolert, kan dette føre til problemer (se figur 4.3). I dette hypotetiske eksempelet er de fysiske parameterne satt slik at det isolerte tracket og det isolerte målet ikke burde bli assosiert, men burde sees på som et falskt track og et ikke-detektert mål. For en assosiasjonsmetode som bare ser på de faktiske avstandene og ikke tar i betraktning andre parametere, kan det være at den beste assosiasjonen er at det isolerte målet assosieres med det 'opptatte' tracket, og at det isolerte tracket på lik linje vil assosieres med det 'opptatte' målet. Den beste assosiasjonen er som nevnt definert som den assosiasjonen som gir den laveste akkumulerte avstanden mellom track-mål-par.



Figur 4.3 Illustrasjon som viser nytten av å bruke en maksimal verdi i avstanden mellom track-mål-par. Se forklaring til figurene i egen boks under.

Forklaringstekst til figur 4.3:

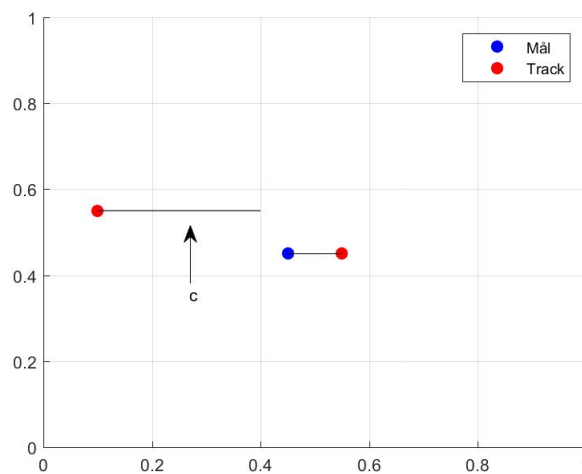
- Et tidssteg hvor vi har to mål og to track fra en tracker.
- Den globale assosiasjonen som gir kortest summert avstand mellom track og mål.
- Denne assosiasjonen kobler det sentralliggende målet til det sentralliggende tracket, hvilket virker intuitivt riktig. Dermed blir det isolertliggende tracket koblet til det isolertliggende målet, og den store avstanden mellom dem gjør at denne assosiasjonen gir større summert avstand.
- Ved å innføre en maksimal verdi for avstanden mellom to punkter vil målet og tracket sentralt i området bli assosiert med hverandre, og det isolertliggende tracket vil bli assosiert med det isolertliggende målet. Denne assosiasjonen er kun til bruk i beregning av OSPA-avstanden og betyr ikke at vi tror tracket burde kobles til målet.

Dette kan gi et feilaktig bilde på trackerens ytelse. Det finnes flere måter å unngå dette problemet på. En mulighet er å si at mål og track som ligger for langt fra hverandre ikke kan assosieres. I OSPA benytter man seg av en annen metode - det settes en maxsgrense på avstanden mellom et track og et

mål, c (se figur 4.3d). I scenarioet som vi akkurat har beskrevet, vil en slik maks grense føre til at det isolerte tracket assosieres med det isolerte målet, og avstanden mellom dem blir satt til c . Ved å inkludere denne maks grensen får vi et avstandsmål d_c , som er gitt ved

$$d_c(\mathbf{x}, \mathbf{y}) = \min(c, d(\mathbf{x}, \mathbf{y})) \quad (4.1)$$

Vi ser også at OSPA-avstanden D_{OSPA} , den gjennomsnittlige avstanden mellom track og mål, er avgrenset av $D_{OSPA} \in [0, c]$.



Figur 4.4 Samme scenario som i figur 4.3, med det unntak at det isolerte målet er fjernet. OSPA-avstanden blir den samme.

Denne maks grensen gjør at track og mål langt fra hverandre kan bli assosiert sammen, til en ikke altfor høy kostnad på OSPA-avstanden. Merk at denne assosiasjonen ikke er en reell kobling mellom målet og tracket. La oss se på eksempelet hvor vi har to track og ett mål. Se figur 4.4. Vi har her samme scenario som i figur 4.3, med den endringen at det ene målet er fjernet. Det isolerte tracket kan dermed ikke assosieres med noe mål. I beregningen av OSPA-avstanden bidrar dette enlige målet med en fastsatt avstand, som igjen er c . Vi ser dermed at OSPA-avstanden for scenarioet i figur 4.4 er lik OSPA-avstanden for scenarioet i figur 4.3. Når avstanden mellom et assosiert mål-track-par er større enn c er bidraget til OSPA-avstanden identisk til om målet eller tracket ikke hadde vært der. Det er det som menes med at assosiasjon mellom et slikt par ikke er en reell kobling mellom mål og track.

La oss nå se på formelen for å finne OSPA-avstanden mellom to mengder tilstandsvektorer. For å lettere koble OSPA til bruk på track og mål, velger vi som de to mengdene tilstandsvektorer \mathbf{X}_k og \mathbf{Y}_k , som er mengdene med henholdsvis track og mål ved tidssteg k . Formelen slik den er presentert i likning 4.2 forutsetter at $|\mathbf{X}_k| = n$, $|\mathbf{Y}_k| = m$ og $n \geq m$. Vi kan selvfølgelig beregne OSPA-avstanden også for tilfellet $|\mathbf{Y}_k| > |\mathbf{X}_k|$, men da må vi bytte om på \mathbf{x} og \mathbf{y} (men lar indekseringen stå, slik at de bytter indeksering), og i tillegg sette $|\mathbf{X}_k| = m$ og $|\mathbf{Y}_k| = n$. For å presisere, i formelen under er $|\mathbf{X}_k| = n$, $|\mathbf{Y}_k| = m$ og $n \geq m$.

$$D_{OSPA}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_{n,m}} \sum_{i=1}^m (d_c(\mathbf{x}_{i,k}, \mathbf{y}_{\pi(i),k}))^p + (n-m) \cdot c^p \right) \right]^{\frac{1}{p}} \quad (4.2)$$

Under deler vi opp formelen og forklarer hver bit for seg:

$$D_{OSPA}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_{n,m}} \sum_{i=1}^m (d_c(\mathbf{x}_{i,k}, \mathbf{y}_{\pi(i),k}))^p + (n-m) \cdot c^p \right) \right]^{\frac{1}{p}}$$

Dette er normaliseringsfaktoren. $n = |\mathbf{X}_k|$. Ved å bruke denne normaliseringsfaktoren får vi et gjennomsnitt av alle enkeltavstandene mellom track-mål-par (inkludert avstandene fra ikke-assosierte track/mål).

$$D_{OSPA}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_{n,m}} \sum_{i=1}^m (d_c(\mathbf{x}_{i,k}, \mathbf{y}_{\pi(i),k}))^p + (n-m) \cdot c^p \right) \right]^{\frac{1}{p}}$$

Dette er summen av avstandsmålet d_c (eller strengt tatt for d_c opphøyet i p , p vil bli forklart senere) for den beste assosiasjonen mellom \mathbf{X}_k og \mathbf{Y}_k . En assosiasjon er betegnet med π , mens rommet av alle mulige assosiasjoner er gitt ved $\Pi_{n,m}$. d_c er definert i likning 4.1. $\min_{\pi \in \Pi_{n,m}}$ betyr at vi skal finne den assosiasjonen π som gir lavest mulig verdi for $\sum_{i=1}^m (d_c(\mathbf{x}_{i,k}, \mathbf{y}_{\pi(i),k}))^p$. En forklaring av hvordan man finner den beste assosieringen følger i kapittel 4.2. Merk at summen går over $i = 1 : m$, hvor m er størrelsen til den minste av mengdene.

$$D_{OSPA}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_{n,m}} \sum_{i=1}^m (d_c(\mathbf{x}_{i,k}, \mathbf{y}_{\pi(i),k}))^p + (n-m) \cdot c^p \right) \right]^{\frac{1}{p}}$$

Dette er bidraget som kommer fra track eller mål som ikke har blitt assosiert. I formelen slik den er skrevet her vil det være track, gitt at vi har $|\mathbf{X}_k| = n \geq m = |\mathbf{Y}_k|$. Dette bidraget er lik forskjellen på antall i mengdene ganget med c^p . Forklaring på p følger under.

$$D_{OSPA}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_{n,m}} \sum_{i=1}^m (d_c(\mathbf{x}_{i,k}, \mathbf{y}_{\pi(i),k}))^p + (n-m) \cdot c^p \right) \right]^{\frac{1}{p}}$$

p er en parameter som påvirker OSPA-avstanden, og må være et heltall større enn null. Jo høyere verdi for p , jo mer vil store avstandsbidrag dominere den endelige OSPA-avstanden. Dette er delvis mitigert ved at vi setter en maksverdi c . Vi opphøyer alle avstandsbidrag i p før vi summerer, normaliserer og deretter tar p -roten. Den endelige OSPA-verdien vil dermed være mellom 0 og c . De typiske verdiene for p er de laveste positive heltallene. I [6] opererer de med $p = 2$ i eksemplene de viser.

For å finne OSPA-avstanden mellom de to mengdene \mathbf{X}_k og \mathbf{Y}_k gjøres følgende:

- Velg OSPA-parameterne c og p . c velges som en fornuftig verdi ut fra parameterne til scenarioet man har. Her vil nøyaktighet til sensor og bevegelsesdynamikk til mål være avgjørende. p velges ut fra hvor mye man ønsker å vektlegge store avstander; høyere p vektlegger disse mer.
- Finn den beste assosiasjonen π . Se kapittel 4.2 for en gjennomgang av hvordan man finner denne.

- For hvert track-mål-par i denne assosieringen, finn $d_c(\mathbf{x}, \mathbf{y})^p$.
- Summer opp alle disse bidragene.
- Finn absoluttverdien til differansen mellom antall track og antall mål, og legg dette antall ganger c^p til summen fra forrige steg.
- Divider denne summen på størrelsen på den største mengden, det vil si antall track hvis du har flere track enn mål og antall mål i motsatt tilfelle.
- Ta p -roten av resultatet fra forrige steg. Dette er OSPA-avstanden mellom de to mengdene \mathbf{X}_k og \mathbf{Y}_k .

4.2 Assosiering

I kapittel 4.1 nevnes flere ganger behovet for å finne den beste assosiasjonen mellom track og mål. I figur 4.2 ser vi en slik assosiasjon, som i dette tilfellet er gjort ved visuell inspeksjon. Dette var enkelt å gjøre siden scenarioet ble designet for at det skulle være lett å se hvilket mål som burde kobles til hvilket track. Hvis man skal implementere OSPA-algoritmen i et eller annet programmeringsspråk trenger man også en algoritme for å finne den beste assosiasjonen.

I dette kapittelet viser vi hvordan en slik assosiering kan gjøres. I OSPA-algoritmen brukes denne metoden til å finne den assosiasjonen mellom track og mål som gir den totalt lavest akkumulerte avstanden mellom track-mål-par, men metoden beskrevet her er generell for problemstillingen. Problemstillingen er av typen *assignment problem*, som finnes i mange fagområder. Felles er at man skal lage koblinger mellom elementene i to mengder, hvor hvert element bare kan kobles til ett element i den andre mengden, og hver kobling har en kostnad. Hvert element i den minste mengden må assosieres med et element i den største mengden, mens overflødige elementer i den største mengden ikke blir assosiert. Man er ute etter å finne hvilken assosiasjon som gir den laveste totale kostnaden. En vanlig måte å beskrive problemet på er at man har tre arbeidere og tre jobber, hvor hver arbeider tar forskjellig pris for de tre jobbene. Dette gir matrise 1 som vi ser under til venstre.

Matrise 1:	jobb 1	jobb 2	jobb 3
arbeider 1	1	2	3
arbeider 2	3	1	2
arbeider 3	2	2	2

Matrise 2:	jobb 1	jobb 2	jobb 3
arbeider 1	1	2	3
arbeider 2	3	1	2
arbeider 3	2	2	2
arbeider 4	5	5	0

Det er ikke nødvendig å ha like mange arbeidere som oppgaver for å kunne løse et *assignment problem*. Over til høyre i matrise 2 har vi en fjerde arbeider, som er så glad i en av jobbene at han gjør den gratis. Vi ser da at én arbeider ikke får noen jobb. Den beste assosiasjonen er i begge tilfellene markert i matrisene.

For å finne den beste assosiasjonen i vår tracking-verden lager vi en tilsvarende matrise for track og mål, hvor kostnaden ved å assosiere track i og mål j er gitt ved $d_c(\mathbf{x}_i, \mathbf{y}_j)^p$. La oss si at vi har valgt $p = 2$. Resultatet er matrise 3, vist under.

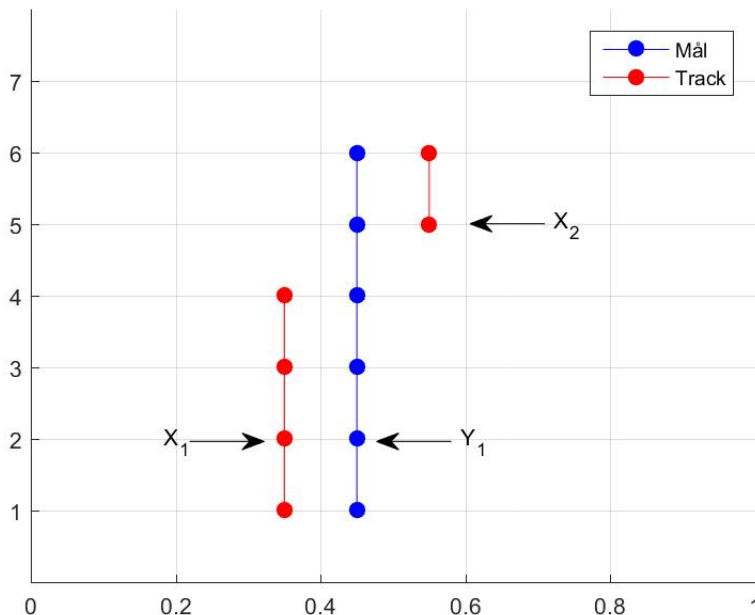
Matrise 3:			
	mål 1	mål 2	mål 3
track 1	$d_c(\mathbf{x}_1, \mathbf{y}_1)^2$	$d_c(\mathbf{x}_1, \mathbf{y}_2)^2$	$d_c(\mathbf{x}_1, \mathbf{y}_3)^2$
track 2	$d_c(\mathbf{x}_2, \mathbf{y}_1)^2$	$d_c(\mathbf{x}_2, \mathbf{y}_2)^2$	$d_c(\mathbf{x}_2, \mathbf{y}_3)^2$
track 3	$d_c(\mathbf{x}_3, \mathbf{y}_1)^2$	$d_c(\mathbf{x}_3, \mathbf{y}_2)^2$	$d_c(\mathbf{x}_3, \mathbf{y}_3)^2$
track 4	$d_c(\mathbf{x}_4, \mathbf{y}_1)^2$	$d_c(\mathbf{x}_4, \mathbf{y}_2)^2$	$d_c(\mathbf{x}_4, \mathbf{y}_3)^2$

Når vi har laget denne matrisen må vi finne den beste assosiasjonen. Det finnes allerede mange ferdiglagde programmer som kan løse et *assignment problem*, og som gir den beste assosiasjonen og kostnaden til denne. I Matlab-implementasjon av disse enkeltmetrikkene som er brukt for å lage resultatene vist i kapittel 5, er det benyttet eksisterende kode som er en implementasjon av en algoritme som kalles både *The Hungarian Algorithm* og *Munkres assignment algorithm*.

4.3 OSPA-T

4.3.1 Bakgrunn og motivasjon

OSPA-T står for *Optimal SubPattern Assignment – Tracking* og er en metrikk utarbeidet av Branko Ristic, Ba-Ngu Vo, Daniel Clark og Ba-Tuong Vo [1]. Metrikken er en videreutvikling av OSPA.



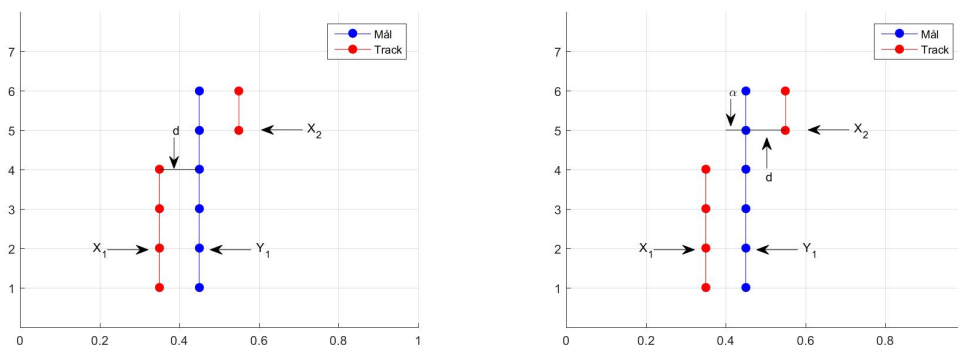
Figur 4.5 Et eksempel på trackbytte. Vi har et mål som beveger seg i én dimensjon (x-aksen). Y-aksen viser tidssteg.

Det er mulig å bruke OSPA til å måle kvaliteten til en tracker, men den har noen store svakheter. OSPA gjør en assosiasjon mellom track og mål ved hvert tidssteg for å finne den assosiasjonen som

gir den minste summerte avstanden mellom alle track-mål-par. Siden OSPA gjør denne assosiasjonen på nytt for hvert tidssteg kan det tenkes at et mål kan bli assosiert med et nytt track for hvert tidssteg.

Figur 4.5 viser et scenario hvor vi har ett mål og to track som resultat fra en tracker. Figuren viser seks tidssteg fra scenarioet. Posisjoner koblet sammen med en tynn strek tilhører altså samme mål eller track i etterfølgende tidssteg. Hvis vi benytter oss av OSPA-metrikken på hvert tidssteg av det viste scenarioet, får vi en konstant OSPA-avstand, siden de to trackene ligger like langt unna målet. Det at målet bytter assosiasjon fra et track til et annet kommer ikke frem i OSPA-avstanden.

OSPA-T er som navnet sier en tilpasning til OSPA for bruk innenfor tracking. OSPA-T ble utviklet for å kompensere for at OSPA ikke tar hensyn til trackbytting. Måten det gjøres på er at vi først gjør en global assosiering mellom track og mål. Med global menes her at assosiasjonen er gyldig for alle tidssteg, og vi definerer den som assosiasjonen som minimerer avstanden mellom alle track-mål-par, hvor avstanden mellom track-mål-par er gitt som avstanden mellom tracket og målet summert over alle tidssteg. Vi kaller denne assosiasjonen λ , og rommet av alle mulige globale assosiasjoner Λ . I scenarioet fra figur 4.5 vil den globalt beste assosiasjonen være at track X_1 assosieres med mål Y_1 , mens track X_2 ikke assosieres med noe mål. Merk at i assosieringen gjort for hvert tidssteg vil track X_2 assosieres med mål Y_1 ved de tidsstegene hvor X_1 ikke eksisterer. Vi kaller heretter assosieringen som gjøres for hvert tidssteg for den lokale assosieringen. Denne er separat fra den globale. Der den globale gjøres én gang og dekker hele tidsrommet gjøres den lokale for hvert tidssteg og dekker bare det øyeblikket.



(a) Lokal assosiasjon for tidssteg 4. Track X_1 assosieres med mål Y_1 . Siden disse er assosiert i den globale assosiasjonen vil bare den euklidiske avstanden d bidra til OSPA-T avstanden.
 (b) Lokal assosiasjon for tidssteg 5. Track X_2 assosieres med mål Y_1 . Disse er ikke assosiert i den globale assosiasjonen, og vil derfor få et straffebidrag til OSPA-T avstanden av størrelse α .

Figur 4.6

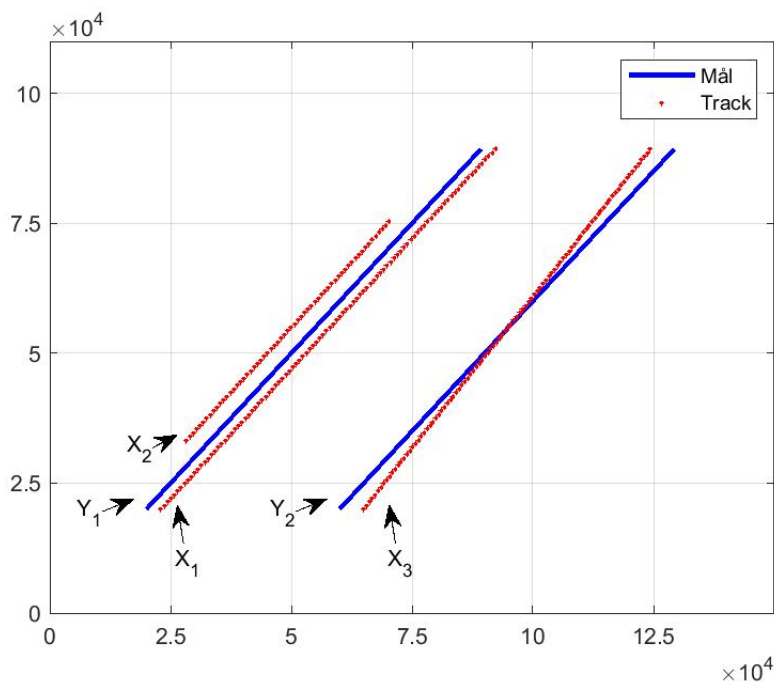
Hva brukes så denne globale assosiasjonen til? OSPA-T beregnes i stor grad på samme måte som OSPA, ved å finne beste lokale assosiasjon for hvert tidssteg og finne den gjennomsnittlige avstanden mellom track-mål-par, samt inkludere en avstand for hvert overflødig track eller ikke-trackede mål. Den globale assosiasjonen brukes i beregningen av avstanden mellom et track og et mål for et gitt tidssteg. Vi vet fra den globale assosiasjonen hvilket track som bør knyttes til et mål. Alle andre track får en straffeavstand α når vi beregner avstanden til dette målet. Se figur 4.6.

Under følger først en litt mer detaljert gjennomgang av hvordan man finner den globalt beste

assosiasjonen og hvordan parameteren α brukes i beregning av avstand mellom et track og et mål. Deretter kommer en oppsummering av hvordan man finner OSPA-T-avstanden.

4.3.2 Den globalt beste assosiasjonen

La oss se på eksempelscenariot fra kapittel 2.2, her vist igjen i figur 4.7. Vi har tre track og to mål. I dette enkle scenariot er det lett å se den globalt beste assosiasjonen: X_1 assosieres med Y_1 , X_3 assosieres med Y_2 , og X_2 blir ikke assosiert med et mål.



Figur 4.7 Eksempelscenario.

Det vi trenger er en algoritme for å finne den beste globale assosiasjonen mellom en mengde track og en mengde mål. Denne algoritmen er som følger:

- Finn den tidsakkumulerte avstanden mellom hvert mulig track-mål-par. Det gjøres ved at vi finner avstanden mellom tracket og målet for hvert tidssteg. Dette gjør vi uansett om målet og tracket eksisterer ved et tidssteg eller ikke. Vi innfører her avstandsmålet $d_{\Delta}(X_{i,k}, Y_{j,k})$ som angir avstanden mellom track i og mål j ved tidssteg k . Dette avstandsmålet er definert som

$$d_{\Delta}(X_{i,k}, Y_{j,k}) = \begin{cases} \min(\Delta, d(X_{i,k}, Y_{j,k})) & \text{hvis } X_{i,k} \neq \emptyset \ \& \ Y_{j,k} \neq \emptyset \\ 0 & \text{hvis } X_{i,k} = \emptyset \ \& \ Y_{j,k} = \emptyset \\ \Delta & \text{hvis bare én av } X_{i,k} \ \& \ Y_{j,k} \ \text{er lik } \emptyset. \end{cases} \quad (4.3)$$

Δ er en parameter hvis funksjon minner om parameteren c som vi allerede er kjent med. Δ er den avstanden som blir satt hvis bare én del av et track-mål-par eksisterer. I tillegg brukes den som en maksimal avstand, på lik linje med hva c gjør for den lokale assosiasjonen. Hvis vi

har K tidssteg i hele scenarioet vil den maksimale tidsakkumulerte avstanden mellom et track og et mål være $K \cdot \Delta$.

- Når vi har den tidsakkumulerte avstanden mellom alle mulige track-mål-par lager vi et *assignment problem* på lik linje som vi gjorde for et enkelt tidssteg i kapittel 4.2. For eksempelscenarioet vårt vil vi få følgende matrise:

	Y ₁	Y ₂
X ₁	300	1000
X ₂	695	1000
X ₃	1000	252.5

hvor vi har brukt $\Delta = 10$ og scenarioet som nevnt har $K = 100$. Deretter bruker vi en egnet algoritme (som beskrevet i kapittel 4.2) for å finne den globale assosiasjonen som gir den minste totale avstanden. I matrisen over er den beste globale assosiasjonen markert.

Vi har nå funnet den beste globale assosiasjonen mellom trackene i \mathbf{X} og målene i \mathbf{Y} . Vi kaller denne assosiasjonen λ . Formelen for λ i ren matematikk er

$$\lambda = \arg \min_{\lambda^* \in \Lambda_m} \sum_{l=1}^m \sum_{k=1}^K \min(\Delta, d(X_{\lambda^*(l),k}, Y_{l,k})) \quad (4.4)$$

med følgende

λ	den beste globale assosiasjonen
λ^*	en global assosiasjon, brukt som variabel i minimeringen for å finne λ
Λ_m	mengden av alle mulige globale assosiasjoner av lengde m med verdier tatt fra $\{1, 2, \dots, n\}$.
m	størrelsen på settet med mål, $ \mathbf{Y} = m$
$Y_{l,k}$	mål l ved tidssteg k
$X_{\lambda^*(l),k}$	tracket som er assosiert med mål l i assosiasjonen λ^* ved tidssteg k

Merk at formel 4.4 slik den er skrevet forutsetter at $|\mathbf{Y}| = m \leq n = |\mathbf{X}|$.

4.3.3 Den assosiasjonsavhengige avstanden $d_{c,\lambda,\alpha}$

Som beskrevet og vist i figur 4.6 innfører vi parameteren α , som legges til avstander mellom track-mål-par som ikke har blitt koblet sammen i den globalt beste assosiasjonen. Vi gjør dette ved å innføre et nytt avstandsmål mellom et mål og et track, $d_{c,\lambda,\alpha}(\mathbf{X}, \mathbf{Y})$, som er en funksjon som har både c , λ og α som parametere. Den er definert som følger:

$$d_{c,\lambda,\alpha}(X_{i,k}, Y_{j,k}) = \left(\min\left(c^s, (d(\mathbf{x}_{i,k}, \mathbf{y}_{j,k})^s + l_\alpha(\lambda, i, j)^s)\right) \right)^{\frac{1}{s}} \quad (4.5)$$

hvor

$$l_\alpha(\lambda, i, j) = \begin{cases} 0 & \text{hvis } \lambda(i) = j, \text{ det vil si at } X_i \text{ og } Y_j \text{ er assosiert i } \lambda \\ \alpha & \text{hvis } X_i \text{ og } Y_j \text{ ikke er assosiert i } \lambda \end{cases} \quad (4.6)$$

og s er en parameter som bestemmer hvor mye l_α skal påvirke $d_{c,\lambda,\alpha}$. For $s = 1$ er $d_{c,\lambda,\alpha}$ lik summen av faktisk avstand og straffeavstanden α , for $s \rightarrow \infty$ er $d_{c,\lambda,\alpha} = \max(d(\mathbf{x}_{i,k}, \mathbf{y}_{j,k}), l_\alpha)$. En vanlig verdi for s er de laveste positive heltallene, og i eksemplene i litteraturen blir den satt til samme verdi som p .

$d_{c,\lambda,\alpha}(\mathbf{X}_{i,k}, \mathbf{Y}_{j,k})$ kan med andre ord ha tre forskjellige verdier:

d	Den euklidske avstanden mellom punktene. Dette blir avstanden hvis \mathbf{X}_i og \mathbf{Y}_j er assosiert i λ , og d er mindre enn c .
$d + \alpha$	Den euklidske avstanden pluss tilleggsavstanden α . Dette blir avstanden hvis \mathbf{X}_i og \mathbf{Y}_j ikke er assosiert i λ , og $d + \alpha$ er mindre enn c .
c	Den maksimalt tillatte avstanden. Dette blir avstanden hvis $d \geq c$ eller $(d + \alpha) \geq c$ for \mathbf{X}_i og \mathbf{Y}_j henholdsvis assosiert og ikke assosiert i λ .

4.3.4 Fremgangsmåte for å finne OSPA-T-avstanden

OSPA-T-avstanden mellom trackene i \mathbf{X}_k og målene i \mathbf{Y}_k finner vi ved formelen

$$D_{OSPA-T}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m (d_{c,\lambda,\alpha}(\mathbf{X}_{i,k}, \mathbf{Y}_{\pi(i),k}))^p + (n-m) \cdot c^p \right) \right]^{\frac{1}{p}} \quad (4.7)$$

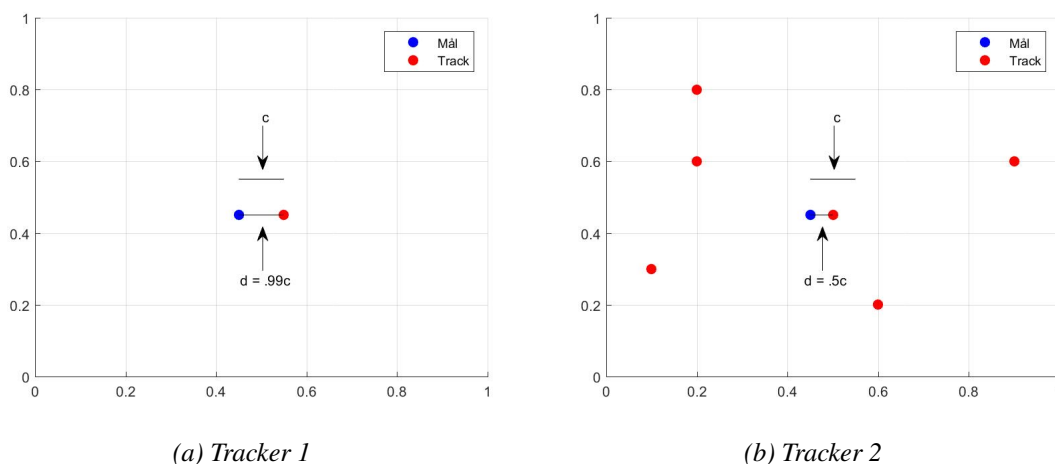
For en detaljert gjennomgang av denne formelen se gjennomgangen av tilsvarende formel av OSPA-avstanden i kapittel 4.1. Den eneste forskjellen er bruken av $d_{c,\lambda,\alpha}$ i stedet for d_c .

For å finne OSPA-T-avstanden mellom de to mengdene \mathbf{X}_k og \mathbf{Y}_k gjøres følgende:

- Finn den beste globale assosiasjonen λ ved å følge fremgangsmåten i kapittel 4.3.2.
- Velg OSPA-T-parameterne c, p, s , og α . c velges som en fornuftig verdi ut fra parameterne til scenarioet man har. Her vil nøyaktighet til sensor og bevegelsesdynamikk til mål være avgjørende. p velges ut fra hvor mye man ønsker å vektlegge store avstander; høyere p vektlegger disse mer. α er tilleggsavstanden til track-mål-par som ikke er assosiert i λ og settes høy hvis man ønsker å straffe trackbytter mye, lav hvis man vil straffe trackbytter mindre. Parameteren s påvirker avstanden $d_{c,\lambda,\alpha}$, og er et positivt heltall.
- Finn π , som er den beste lokale assosiasjonen ved tidssteg k . Se kapittel 4.2 for en gjennomgang av hvordan man finner denne.
- For hvert track-mål-par i denne assosiasjonen, finn $d_c(\mathbf{x}, \mathbf{y})^p$.
- Summer opp alle disse bidragene.
- Finn absoluttverdien til differansen mellom antall track og antall mål, og legg dette antall ganger c^p til summen fra forrige steg.
- Divider denne summen på størrelsen på den største mengden, det vil si antall track hvis du har flere track enn mål og antall mål i motsatt tilfelle.
- Ta p -roten av resultatet fra forrige steg. Dette er OSPA-T-avstanden mellom de to mengdene \mathbf{X}_k og \mathbf{Y}_k .

4.4 GOSPA og GOSPA-T

GOSPA står for Generalized Optimal Sub-Pattern Assignment, og er en videreutvikling av OSPA. GOSPA ble utviklet av Rahmathullah et al. [3]. Motivasjonen for utviklingen er at OSPA ikke straffer store mengder falske track eller manglende track i tilstrekkelig stor grad. GOSPA er som sagt en videreutvikling av OSPA, men vi kan lett lage en GOSPA-T ved å gjøre de samme forandringene som vi gjorde for å endre OSPA til OSPA-T: Gjør en global assosiering og legg til straffeavstand mellom track-mål-par som ikke har blitt assosiert i den globale assosieringen.

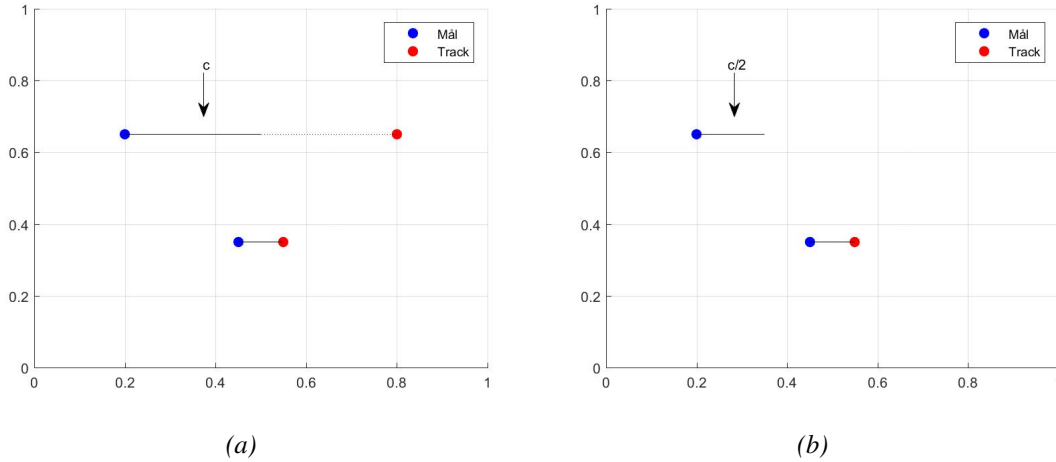


Figur 4.8 To trackere leverer track i et tidssteg hvor det kun er ett mål. Trackeren med mange overfløydige track får den laveste (beste) OSPA-T-avstanden. For å unngå dette fjerner GOSPA-T normaliseringsfaktoren i beregning av GOSPA-T-avstanden.

Hvis vi ser på likning 4.2 ser vi at den inneholder en faktor $\frac{1}{n}$, hvor n er $\max(|\mathbf{X}_k|, |\mathbf{Y}_k|)$. Denne faktoren gjør at OSPA scoren blir normalisert, og bare kan ta verdier i intervallet $[0, c]$. Vi kan tenke oss et scenario med ett ekte mål. Se figur 4.8. Tracker 1 leverer kun ett track som har en avstandsfeil på $0.99c$, vist i figur 4.8a. Tracker 2 leverer et track som har en avstandsfeil på $c/2$. Hvis tracker 2 har en håndfull falske track, som vist i figur 4.8b, vil den fremdeles ha en OSPA-T score $< .99c$, og fremstå som den bedre trackeren. I mange sammenhenger vil vi betrakte denne trackeren som dårligere grunnet de falske trackene.

GOSPA dropper faktoren $\frac{1}{n}$. Den normaliserer altså ikke, og en tracker med mange falske track vil bli straffet for dette i større grad enn for den normaliserte OSPA. I tillegg innføres en parameter β (i artikkelen[3] kalt α , her endret til β for å ikke blande med parameteren α i OSPA-T). Denne parameteren bestemmer i hvor stor grad forskjellen i kardinalitet mellom \mathbf{X}_k og \mathbf{Y}_k (forskjellen i antall track) skal straffes. Rahmathullah et. al. argumenterer for at denne parameteren bør settes til 2. Se figur 4.9 for forklaring. Under følger formelen for GOSPA-scoren.

$$D_{GOSPA}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\left(\min_{\pi \in \Pi_n} \sum_{i=1}^m (d_c(x_{i,k}, y_{\pi(i),k}))^p + (n-m) \cdot \frac{c^p}{\beta} \right)^{\frac{1}{p}} \right] \quad (4.8)$$



Figur 4.9 I (a) og (b) ser vi utfallet fra to trackere med input fra samme scenario. I (a) assosieres et track-mål-par som har avstand større enn c . Rahmatullah et. al. argumenterer at dette track-mål-paret ikke er reelt, men at dette er et ikke-assosiert track og et ikke-assosiert mål. Hvis straffeavstanden for dem tilsammen er c burde straffeavstanden for et enlig ikke-assosiert mål som i figur (b) være $c/2$.

Bortsett fra de nevnte endringene er 4.8 lik 4.2. For GOSPA-T får vi

$$D_{GOSPA-T}(\mathbf{X}_k, \mathbf{Y}_k) = \left[\left(\min_{\pi \in \Pi_n} \sum_{i=1}^m (d_{c,\lambda,\alpha}(\mathbf{X}_{i,k}, \mathbf{Y}_{\pi(i),k}))^p + (n-m) \cdot \frac{c^p}{\beta} \right)^{\frac{1}{p}} \right] \quad (4.9)$$

Fremgangsmåten for å finne GOSPA-T er den samme som for OSPA-T, med unntak av at man ikke normaliserer, samt husker å inkludere β .

4.5 OSPA-on-OSPA (OSPA²)

OSPA-T blir i [4] kritisert for ikke å oppfylle kravene til å være en metrikk. I [5] blir dette fulgt opp av Beard, Vo og Vo, hvor de to sistnevnte var med på å utvikle OSPA og OSPA-T. De presenterer OSPA-on-OSPA, også kalt OSPA².

OSPA-on-OSPA har følgende egenskaper som skiller den fra OSPA-T

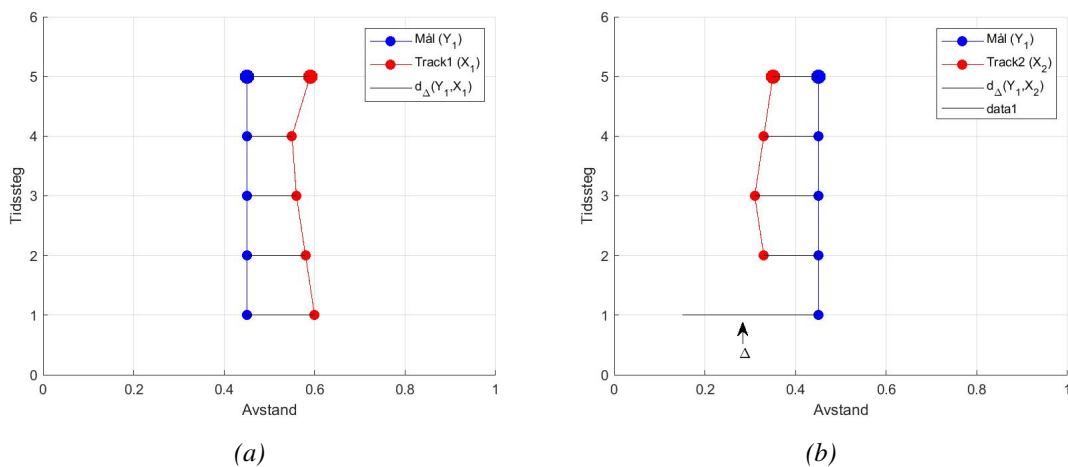
- **Ingen global assosiasjon** - OSPA-on-OSPA har ingen global assosiasjon. OSPA-T brukte den globale assosiasjonen til å kunne straffe trackbytter. OSPA-on-OSPA gjør dette på en annen måte.
- **Den lokale assosiasjonen ved tidssteg k gjøres over et tidsintervall** - i OSPA-T gjøres den lokale assosiasjonen kun basert på tilstandsvektorene i det tidssteget, med unntak av at den globale assosiasjonen kan bidra til en straffeavstand. I OSPA-on-OSPA ser man på et tidsvindu som i tillegg til tidssteg k ser på flere (potensielt alle) tidligere tidssteg.

OSPA-T hadde en eksplisitt straffeavstand ved trackbytte. I OSPA-on-OSPA fjernes behovet for en straffeavstand, siden intet trackbytte er mulig innenfor den utvidede lokale assosiasjonen. For å finne den lokale assosiasjonen i OSPA-on-OSPA trenger vi nok et nytt avstandsmål.

4.5.1 Den tidsvektavhengige avstanden $d_{\Delta, w}$

Figur 4.10 viser avstandene mellom to forskjellige track-mål-par over de siste 5 tidssteg. X-aksen viser posisjonen til tracket og til målet langs én akse i et kartesisk koordinatsystem, og y-aksen viser tidssteg. For enkelhet antar vi at det kun er en posisjonsforskjell mellom tracket og målet langs den viste aksene. Nåværende tidssteg er vist ved litt større markør. Vi ser at track 2 begynte å eksistere for 4 tidssteg siden. Hvis bare tracket eller målet eksisterer ved et tidssteg, settes avstanden dem imellom til Δ , se figur 4.10b.

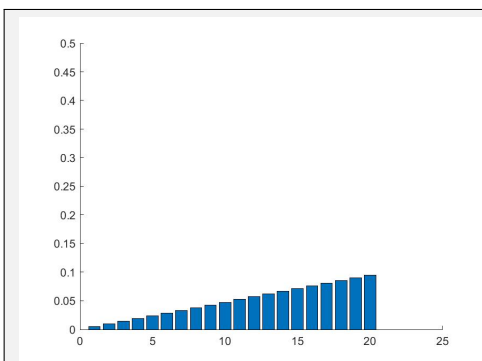
Vi ønsker nå å finne den tidsavhengige avstanden $d_{\Delta, w}$ mellom de to track-mål-parene vist i figur 4.10. Det enkleste er å ta gjennomsnittet av avstanden mellom track-mål-paret over det aktuelle tidsrommet. Vi får da en vektfunksjon som blir $w(g) = \frac{1}{N}$, hvor N er antall tidssteg i tidsrommet og g er et tidssteg. Ofte ønsker vi en vektfunksjon som vektlegger de siste tidsstegene mer enn tidligere tidssteg. I Beard et al. [5] foreslår de to forskjellige vektfunksjoner. Se innrammet område for detaljer.



Figur 4.10 To eksempler på track-mål-par og avstanden mellom track og mål for hvert enkelt tidssteg.

Følgende er verd å merke seg ved valget av vektfunksjon:

- **For at $d_{\Delta, w}$ skal være en gyldig metrikk må vektfunksjonen innfri $w_k(g) > 0$ for alle g , $k \in (0, g)$.** Det vil si at ingen av vektene kan være negative eller null. Vi ser at vektfunksjon 2 vist under bryter med dette, siden den har vektene som er lik null. Ifølge [5] blir $d_{\Delta, w}$ dermed en pseudometrikk. Det virker som om forfatterne mener at dette er tilstrekkelig, da det er denne vektfunksjonen de benytter seg av i sine eksempler.
- **Summen av vektene må være lik 1, $\sum_t w_k(g) = 1$.**

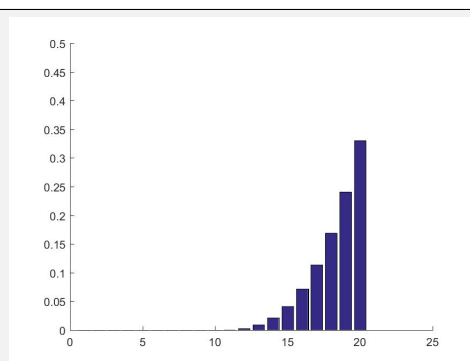


Vektfunksjon 1:

$$w_k(g) = \frac{g^r}{\sum_{g=1}^k g^r} \quad (4.10)$$

Vektfunksjonen er større enn null i hele tidsrommet fra starten av scenarioet og opp til tidssteg k .

Over ser vi vektfunksjonen ved tidssteg 20 for $r = 3$.



Vektfunksjon 2:

Antall tidssteg som vi ønsker å inkludere i utregningen av OSPA-on-OSPA-avstanden er gitt som N . Vi har dermed at for $g \in (g_0, \dots, k - N)$

$$w_k(g) = 0 \quad (4.11)$$

og for $g \in (k - N + 1, \dots, k)$ har vi

$$w_k(g) = \frac{(g + N - k)^r}{\sum_{g=k-N+1}^k (g + N - k)^r} \quad (4.12)$$

Over ser vi vektfunksjonen ved tidssteg 20 for $r = 3$ og $N = 10$.

- **Vektfunksjonen kan være forskjellig for hvert tidssteg k .** Som vi ser i de to punktene over, er vektfunksjonen skrevet med en indeksering på k , $w_k(g)$. Vi skjønner dette intuitivt når vi tenker på hva vektene skal være første og andre tidssteg, kontra det vi ser i det innrammete området med vektfunksjon 1 og vektfunksjon 2. Når vi bare har ett eller to tidssteg vi kan bruke til å finne $d_{\Delta, w}$ vil vektfunksjonen ha henholdsvis en og to verdier. Hvis vi bruker vektfunksjon 1 vil w_k ha forskjellig form for hver k , siden hvert tidssteg forlenger vektfunksjonen. For vektfunksjon 2 vil vi oppnå en stabil form på w_k siden denne bare ser på de siste N tidssteg. Så fort vi har minst N tidssteg vil funksjonen se lik ut hver gang. Det er like fullt ikke riktig å si at $w_k(g)$ er lik hver gang, siden verdiene vil flytte seg et tidssteg for hvert tidssteg scenarioet avanserer.
- **Eksempelene vektfunksjon 1 og vektfunksjon 2 benytter seg begge av parameteren r .** Denne parameteren bestemmer hvor mye vi skal vektlegge tidligere tidssteg i forhold til nyere tidssteg. Hvis vi setter $r = 0$ får vi en uniform vekt over tidsrommet vi ser på. Jo høyere verdi vi setter for r , jo mer vil vi vektlegge de nyeste tidsstegene.

Vi innfører nå den tidsvektavhengige avstanden $d_{\Delta, w}$. Den er definert som følger:

$$d_{\Delta, w}(X, Y, k) = \left(\sum_{g=k-N}^k [w_k(g) d_{\Delta}(X_g, Y_g)]^q \right)^{\frac{1}{q}} \quad (4.13)$$

hvor

k	er tidssteget vi ønsker å beregne avstanden for
N	er antall tidssteg vi ønsker å inkludere i beregningen
Δ	er den maksimale avstanden vi tillater mellom et track-mål-par ved tidssteg g , og også er den avstanden vi setter hvis tracket eller målet ikke eksisterer ved et tidssteg.
q	er en parameter som bestemmer hvor mye vi ønsker å vektlegge store utslag i forhold til små i beregningen av den tidsvektavhengige avstanden $d_{\Delta,w}$. I eksemplene i [5] bruker de $q = 2$.

En oppskrift for å finne denne avstanden er:

- Velg hvor mange tidssteg N du ønsker å inkludere i beregningen.
- Sett parameterne Δ og q .
- For hvert tidssteg $g \in (k - N, \dots, k)$, beregn $[w_k(g)d_{\Delta}(X_g, Y_g)]^q$
- Summer sammen bidragene fra steget over og ta q -roten.

4.5.2 OSPA-on-OSPA-avstanden

Nå som vi kan beregne avstanden mellom et track-mål-par for et tidssteg k kan vi for hvert tidssteg beregne OSPA-on-OSPA-avstanden mellom et sett med track \mathbf{X} og et sett med mål \mathbf{Y} ved tidssteget k . Formelen for dette er

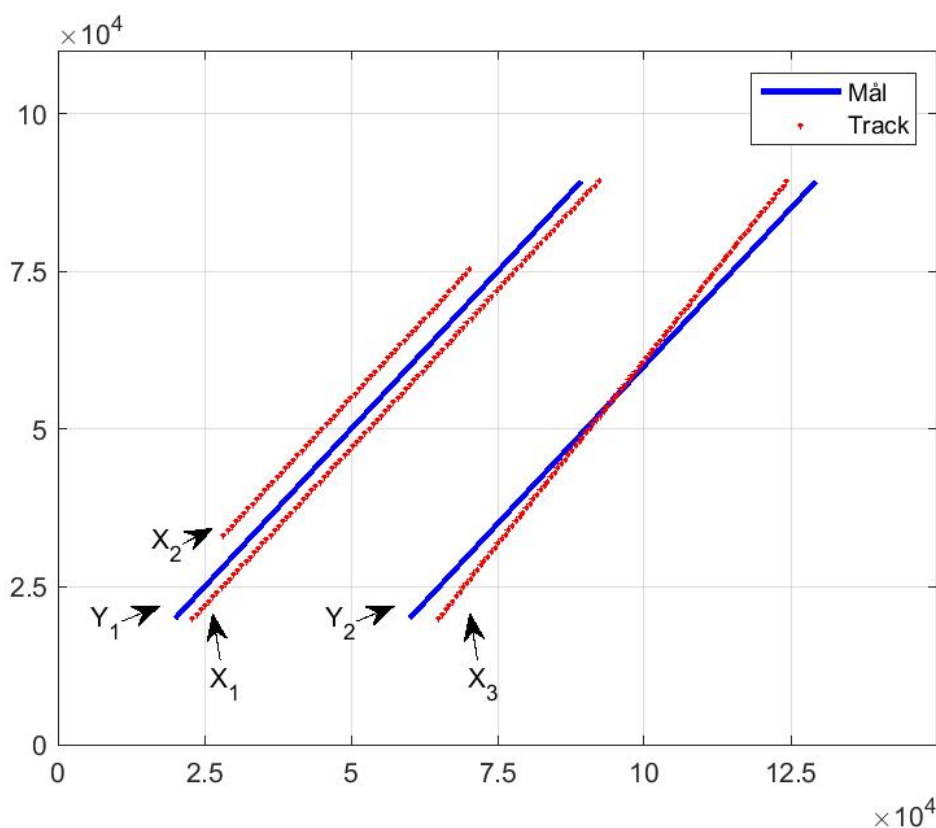
$$D_{OSPA-on-OSPA}(\mathbf{X}, \mathbf{Y}, k) = \left[\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^m (d_{\Delta,w}(X_{i,k}, Y_{\pi(i),k}))^p + (n - m) \cdot c^p \right) \right]^{\frac{1}{p}} \quad (4.14)$$

som vi løser på følgende måte:

- Velg OSPA-on-OSPA-parameterne Δ og p . Δ velges som en fornuftig verdi ut fra parameterne til scenarioet man har. Her vil nøyaktighet til sensor og bevegelsesdynamikk til mål være avgjørende. p velges ut fra hvor mye man ønsker å vektlegge store avstander; høyere p vektlegger disse mer.
- Velg en vektfunksjon $w_k(g)$.
- Finn den beste lokale assosiasjonen π ved tidssteg k . Det gjøres ved å lage en kostmatrise som beskrevet i kapittel 4.2. Elementene i assosiasjonsmatrisen er i dette tilfellet den tidsvektavhengige avstanden opphøyet i p : $d_{\Delta,w}(X_{i,k}, Y_{j,k})^p$.
- For hvert track-mål-par i denne assosiasjonen, finn $d_{\Delta,w}(X_{i,k}, Y_{\pi(i),k})^p$.
- Summer opp alle disse bidragene.
- Finn absoluttverdien til differansen mellom antall track og antall mål, og legg dette antall ganger c^p til summen fra forrige steg.
- Divider denne summen på størrelsen på den største mengden, det vil si antall track hvis du har flere track enn mål og antall mål i motsatt tilfelle.
- Ta p -roten av resultatet fra forrige steg. Dette er OSPA-on-OSPA-avstanden mellom de to mengdene \mathbf{X}_k og \mathbf{Y}_k .

5 Kvalitetsmetrikkene anvendt på eksempelscenarioet

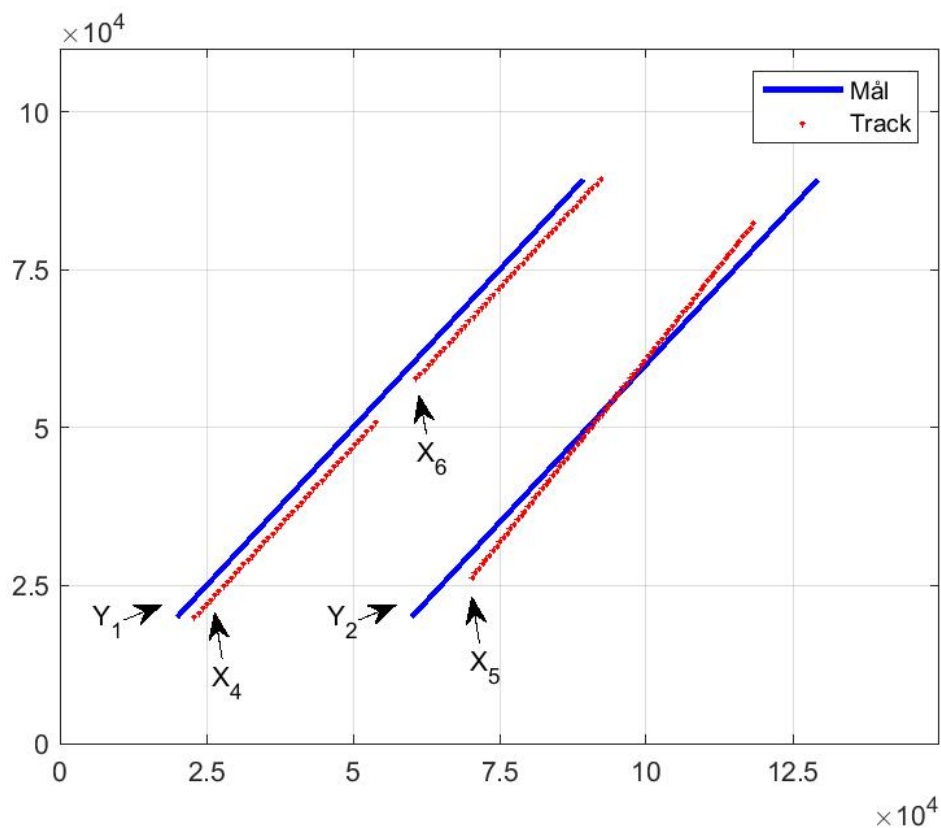
I kapittel 2 beskrives et eksempelscenario som er brukt til å illustrere notasjonene i rapporten. I dette kapitlet anvender vi de forskjellige kvalitetsmetrikkene på scenarioet.



Figur 5.1 Eksempelscenario med resultat fra tracker 1.

Figur 5.1 er samme figur som 2.1, og viser to mål og tre track. Trackene er konstruert, men for sammenlikningens skyld behandler vi dem som om de kommer fra en tracker, heretter kalt tracker 1. Figur 5.2 viser de to samme målene, men med andre track, som vi ser på som output fra tracker 2.

Tracker 1 tracker mål Y_1 kontinuerlig og med konstant avstandsfeil, men har et ekstra track som følger Y_1 deler av tiden. Tracker 1 følger også Y_2 i hele tidsrommet, og tracket krysser målet. Tracker 2 tracker Y_1 i starten og på slutten av tidsrommet, men har en periode på midten hvor det mister målet. Y_2 blir tracket i mesteparten av tidsrommet, men ikke i en kort periode i starten og en kort periode mot slutten.



Figur 5.2 Eksempelscenario med resultat fra tracker 2.

5.1 Multimetrikk

	Tracker 1	Tracker 2
Tracknøyaktighet	2.10	1.93
Bias til tilstandsvektoren	[3 0]	[3 0]
Track jitter	0.505	0.491
Trackbytter	0	0.5
Antall ikke-detekterte mål	0	0
Antall falske track	1	0
Gjennomsnittlig trackinitieringstid	6.67	6.67
Gjennomsnittlig trackvarighet	100	56.7

Tabellen over viser resultatene av å bruke multimetrikkene fra kapittel 3 på trackene fra tracker 1 og tracker 2.

Tracknøyaktigheten er gjennomsnittlig avvik mellom track og mål, beregnet for alle track som er assosiert med et mål. Biasen viser systematisk avvik i henholdsvis x- og y-retning. Trackinitieringstid og trackvarighet er angitt i antall tidssteg.

Når det gjelder nøyaktighet, bias og jitter er det lite som skiller de to trackerne. Forskjellen ligger hovedsaklig i trackbytter og falske track. At den gjennomsnittlige trackvarigheten er mye lavere for tracker 2 skyldes nettopp trackbyttet som skjer for mål 2. Det er vanskelig ut fra tabellen å konkludere hvilken tracker som er best. Fordelen med en slik tabell er at man kan se på det som er relevant for scenarioet og systemet man ser på. Hvis falske track er veldig uønsket vil man nok foretrekke tracker 2, selv om denne mister mål 1 i en periode. Omvendt er tracker 1 bedre hvis det viktigste er å ha sammenhengende track på alle mål, og man heller er villig til å leve med falske track.

5.2 Enkeltmetrikk

De forskjellige enkeltmetrikkene har alle parametere som input. Disse parameterne påvirker resultatet. For å illustrere dette har alle enkeltmetrikkene blitt kjørt to ganger med forskjellige parametere. Siden alle enkeltmetrikkene har sitt opphav i OSPA er noen av parameterne felles. Tabellen under viser parametersettene 1 og 2, samt hvilken enkeltmetrikk som benytter seg av de forskjellige parameterne. Vi har valgt bare å variere parameterne c og α , for lettere å kunne spore endringen i score til variasjonen i parametere. Hvis vi varierer for mange parametere på en gang vil det være vanskelig å vite hvilken variasjon som ga opphav til endringen i score.

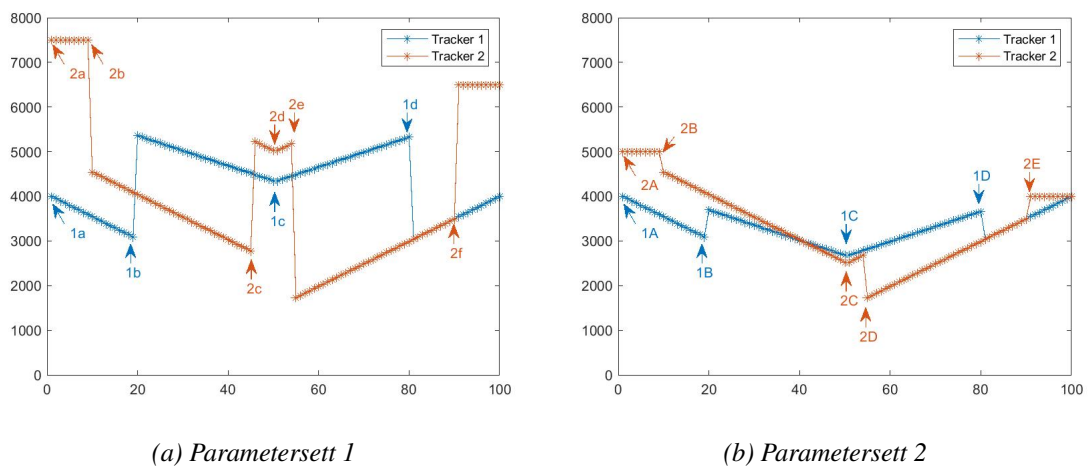
	Parametersett 1	Parametersett 2	OSPA-T	GOSPA-T	OSPA-on-OSPA
c	10000	5000	x	x	x
α	2000	5000	x	x	
β	2	2		x	
p	1	1	x	x	x
s	1	1	x	x	
q	2	2			x
N	10	10			x
r	3	3			x

Alle parameterne har blitt definert i kapittel 4. Under følger en liten repetisjon på hva de forskjellige parameterne er.

- c - den maksimale avstanden mellom et track-mål-par. Hvis avstanden er større enn c blir avstanden satt lik c for beregningen av enkeltmetrikken.
- α - straffeavstanden mellom et track-mål-par som ikke er i den globalt beste assosiasjonen.
- β - en faktor som bestemmer hvor mye ikke-assosierte track/mål skal bidra til resultatet. Brukes kun i GOSPA. Antallet ikke-assosierte track/mål deles på β og ganges med c .
- p - en parameter som bestemmer hvor mye vi ønsker å vektlegge store avstandsfeil. Jo høyere verdi vi velger for p , jo mer blir store avstandsfeil vektlagt.
- s - en parameter som bestemmer intern vektning av straffeavstand i den assosiasjonsavhengige avstanden $d_{c,\lambda,\alpha}$.
- q - en parameter som bestemmer hvor mye vi ønsker å vektlegge store utslag i forhold til små i beregningen av den tidsvektavhengige avstanden $d_{\Delta,w}$ når vi beregner OSPA-on-OSPA-avstanden.

- N - antall tidssteg vi ønsker å inkludere i beregningen av OSPA-on-OSPA-verdien for et enkelt tidssteg.
- r - en parameter som bestemmer hvor mye vi vektlegger nye tidssteg kontra eldre tidssteg i OSPA-on-OSPA. $r = 1$ gir lik vekt for alle N tidssteg, høyere r vektlegger de nyeste tidsstegene mer.

5.2.1 OSPA-T



Figur 5.3 OSPA-T score for tracker 1 og tracker 2. (a) viser resultatene for parametersett 1, (b) viser resultatene for parametersett 2.

Figurene 5.3a og 5.3b viser OSPA-T scoren for tracker 1 og tracker 2 for henholdsvis parametersett 1 og parametersett 2.

Tracker 1 - parametersett 1

- 1a** Tracker 1 begynner med en OSPA-T score på 4000, som er den gjennomsnittlige avstanden ved starten. Deretter faller OSPA-T scoren. Dette skyldes at track X_3 nærmer seg mål Y_2 .
- 1b** Ved dette tidspunktet oppstår track X_2 , som er et ikke-assosiert track. X_2 bidrar dermed med en avstand på $c = 10000$, og OSPA-T scoren, som er gjennomsnittet av avstandene, hopper kraftig.
- 1c** Track X_3 krysser mål Y_2 , og OSPA-T scoren går fra å være avtakende til å være stigende, siden avstanden mellom tracket og målet øker.
- 1d** Track X_2 opphører å eksistere, og OSPA-T scoren faller kraftig.

Tracker 2 - parametersett 1

- 2a** Tracker 2 begynner med en OSPA-T score på 7500. Scoren er så høy fordi trackeren bare leverer ett track til to mål Y_2 , som ikke er assosiert med noe track, bidrar med en avstand c , og trekker opp snittet.

-
-
- 2b** OSPA-T scoren holder seg stabil fram til track X_5 oppstår. Hvert mål kan nå assosieres med et nærliggende track, og OSPA-T scoren går drastisk ned. OSPA-T scoren fortsetter å avta, siden track X_5 har en minkende avstand til mål Y_2 .
- 2c** Track X_4 opphører, og OSPA-T scoren hopper, siden vi nå igjen bare har ett track til to mål.
- 2d** Track X_5 krysser mål Y_2 , og OSPA-T scoren går fra å være avtakende til å være stigende, siden avstanden mellom tracket og målet øker.
- 2e** Track X_6 oppstår, og OSPA-T scoren faller. Vi ser at scoren faller til en lavere verdi enn den var ved punkt **2c**, selv om scenarioet har en symmetri i seg som tilsier at OSPA-T verdien burde speiles om tidssteg 50, som er midt i scenarioets tidsrom. Grunnen til at dette ikke skjer er at OSPA-T først gjør en global assosiering. I dette scenarioet har X_4 og X_6 lik tidsakkumulert avstand til mål Y_1 . En av dem blir valgt, antakeligvis på bakgrunn av rekkefølge, og i dette tilfelle har altså X_6 blitt assosiert med Y_1 i den globale assosiasjonen. X_4 vil dermed få en straffeavstand α når den assosieres (lokalt) med Y_1 .
- 2f** Track X_5 opphører, og OSPA-T scoren hopper.

Tracker 1 - parametersett 2

- 1A** Tracker 1 begynner med en OSPA-T score på 4000, som er den gjennomsnittlige avstanden ved starten. Deretter faller OSPA-T scoren. Dette skyldes at track X_3 nærmer seg mål Y_2 .
- 1B** Ved dette tidspunktet oppstår track X_2 , som er et ikke-assosiert track. X_2 bidrar dermed med en avstand på $c = 5000$, og OSPA-T scoren, som er gjennomsnittet av avstandene, hopper noe.
- 1C** Track X_3 krysser mål Y_2 , og OSPA-T scoren går fra å være avtakende til å være stigende, siden avstanden mellom tracket og målet øker.
- 1D** Track X_2 opphører å eksistere, og OSPA-T scoren faller noe.

Tracker 2 - parametersett 2

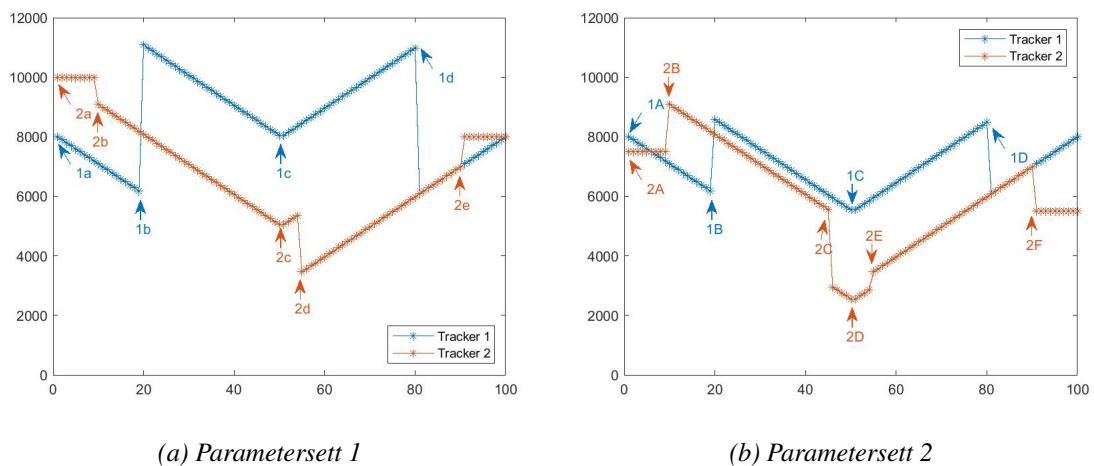
- 2A** Tracker 2 begynner med en OSPA-T score på 5000. Dette er den maksimale OSPA-T scoren som er mulig med dette parametersettet, siden vi har $c = 5000$ og OSPA-T scoren er begrenset av 0 og c . Det er to bidrag til denne scoren. Det ene er det ikke-assosierte målet Y_2 , som bidrar med en score på c . Det andre bidraget er track-mål-paret X_4 - Y_1 . Dette track-mål-paret er ikke assosiert i den globale assosiasjonen, og bidrar dermed med den euklidske avstanden dem imellom pluss straffeavstanden α . I dette tilfelle får vi $d + \alpha = 3000 + 5000 = 8000$. Dette er høyere enn den maksimale avstanden $c = 5000$, og bidraget blir dermed satt til c . Siden begge bidragene er av størrelse c er gjennomsnittet også c . Scoren er en del lavere enn for parametersett 1 fordi parameteren c har en lavere verdi.
- 2B** OSPA-T scoren holder seg stabil fram til track X_5 oppstår. Hvert mål kan nå assosieres med et nærliggende track, og OSPA-T scoren går noe ned. OSPA-T scoren fortsetter å avta, siden track X_5 har en minkende avstand til mål Y_2 .
- 2C** Track X_5 krysser mål Y_2 , og OSPA-T scoren går fra å være avtakende til å være stigende, siden avstanden mellom tracket og målet øker. Vi er nå midt i scenarioet. Track X_4 har opphørt, og mål Y_1 er ikke assosiert med noe track. For parametersett 1 resulterte dette i et hopp i OSPA-T score (se punkt **2c**). Vi har ikke noe tilsvarende hopp her for parametersett 2. Det skyldes valget av parametere og at track-mål-paret X_4 - Y_1 ikke er assosiert i den globale assosiasjonen, og som nevnt bidrar med en avstand c . Når X_4 opphører bidrar Y_1 alene med

c , og bortfallet av X_4 påvirker ikke OSPA-T scoren.

2D Track X_6 oppstår, og OSPA-T scoren faller. Dette er fordi X_6 er assosiert med Y_1 i den globale assosiasjonen, og bidraget fra Y_1 - X_6 -paret er mindre enn c .

2E Track X_5 opphører, og OSPA-T scoren hopper noe.

5.2.2 GOSPA-T



Figur 5.4 GOSPA-T score for tracker 1 og tracker 2. (a) viser resultatene for parametersett 1, (b) viser resultatene for parametersett 2.

Figurene 5.4a og 5.4b viser GOSPA-T scoren for henholdsvis parametersett 1 og parametersett 2.

Tracker 1 - parametersett 1

- 1a** Tracker 1 begynner med en GOSPA-T score på 8000. Dette er summen av avstandene mellom track-mål-parene Y_1 - X_1 og Y_2 - X_3 . GOSPA-T tar ikke gjennomsnittet, slik OSPA-T gjør. Scoren er avtakende, siden avstanden mellom Y_2 - X_3 minker.
- 1b** Det ikke-assosierte tracket X_2 oppstår, og GOSPA-T scoren hopper. Størrelsen på hoppet er på c/β .
- 1c** Track X_3 krysser mål Y_2 , og GOSPA-T scoren går fra å være avtakende til å være stigende.
- 1d** Det ikke-assosierte tracket X_2 opphører, og GOSPA-T scoren faller.

Tracker 2 - parametersett 1

- 2a** Tracker 2 begynner med en GOSPA-T score på 10000. Scoren er så høy fordi det ikke-assosierte målet Y_2 bidrar med $c/2 = 5000$, mens track-mål-paret X_4 - Y_1 , som ikke er assosiert i den globale assosiasjonen, bidrar med avstand + straffeavstand $\alpha = 3000 + 2000 = 5000$.
- 2b** GOSPA-T scoren holder seg stabil fram til track X_5 oppstår. Hvert mål kan nå assosieres med et nærliggende track, og GOSPA-T scoren går noe ned. GOSPA-T scoren fortsetter å avta, siden track X_5 har en minkende avstand til mål Y_2 .

-
- 2c** Track X_5 krysser mål Y_2 , og GOSPA-T scoren går fra å være avtakende til å være stigende. Vi er nå midt i tidsrommet dekket av scenarioet, og Y_1 er ikke assosiert med noe track. Med parametersett 1 førte dette til et hopp i OSPA-T scoren, noe som ikke skjer i GOSPA-T. Grunnen til at det ikke skjer her er valget av parametere. Y_1 , når det ikke er tracket, bidrar med en avstand på $c/\beta = 5000$. Før dette bidrar Y_1 - X_4 -paret med den euklidske avstanden de imellom pluss en straffeavstand for at X_4 ikke er assosiert med Y_1 i den globale assosiasjonen. I dette scenarioet med parametersett 1 blir det $d + \alpha = 3000 + 2000 = 5000$. Derfor er GOSPA-T scoren ikke påvirket av at X_4 opphører å eksistere.
 - 2d** Track X_6 oppstår, og GOSPA-T scoren faller. Dette skyldes at track X_6 er assosiert med Y_1 i den globale assosiasjonen.
 - 2e** Track X_5 opphører, og GOSPA-T scoren hopper.

Tracker 1 - parametersett 2

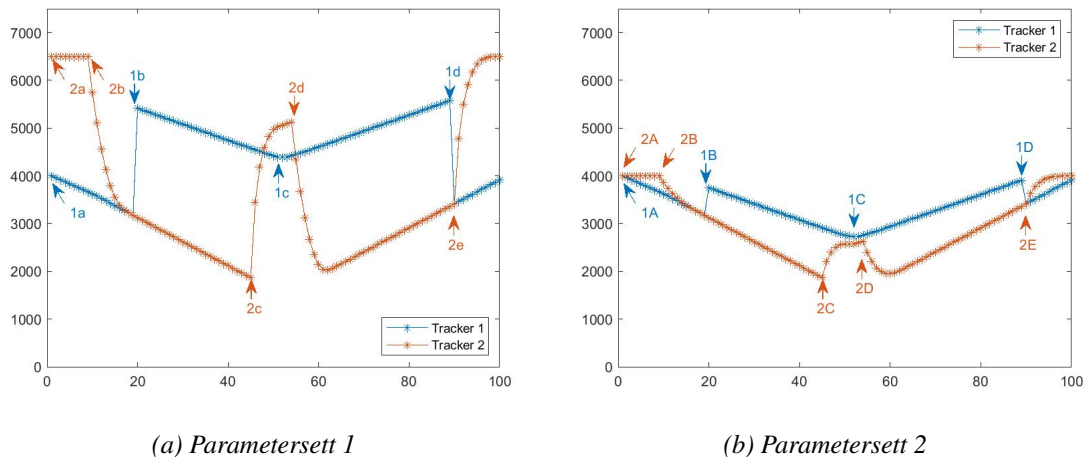
- 1A** Tracker 1 begynner med en GOSPA-T score på 8000. Dette er summen av avstandene mellom track-mål-parene Y_1 - X_1 og Y_2 - X_3 . Scoren er avtakende, siden avstanden mellom Y_2 - X_3 minker.
- 1B** Det ikke-assosierte tracket X_2 oppstår, og GOSPA-T scoren hopper. Størrelsen på hoppet er på c/β .
- 1C** Track X_3 krysser mål Y_2 , og GOSPA-T scoren går fra å være avtakende til å være stigende.
- 1D** Det ikke-assosierte tracket X_2 opphører, og GOSPA-T scoren faller.

Tracker 2 - parametersett 2

- 2A** Tracker 2 begynner med en GOSPA-T score på 7500. Track-mål-paret X_4 - Y_1 bidrar med en avstand på c , mens det ikke-assosierte målet Y_2 bidrar med $c/2$.
- 2B** GOSPA-T scoren holder seg stabil fram til track X_5 oppstår. Hvert mål kan nå assosieres med et nærliggende track. Vi ser at GOSPA-T scoren faktisk går opp. Dette skyldes at avstanden mellom track X_5 og mål Y_2 er større enn avstanden c/β , som Y_2 bidro med når det ikke var assosiert med noe track. Vi ser at GOSPA-T scoren avtar fra dette punkt, siden track X_5 har en minkende avstand til mål Y_2 .
- 2C** Track X_4 opphører, og mål Y_1 er ikke assosiert med noe track. GOSPA-T scoren faller. Dette er den samme effekten som fikk scoren til å stige i punkt **2B**. Der steg scoren når et track oppsto og alle mål kunne assosieres. Grunnen til dette fallet er at summen av den euklidske avstanden mellom mål Y_1 og track X_4 pluss straffeavstanden α er større enn avstanden c/β som mål Y_1 bidrar med når det ikke er assosiert med noe track.
- 2D** Track X_5 krysser mål Y_2 , og GOSPA-T scoren går fra å være avtakende til å være stigende.
- 2E** Track X_6 oppstår, og GOSPA-T scoren hopper. Økningen er mindre enn fallet i **2C**, siden track X_6 er assosiert med mål Y_1 i den globale assosiasjonen.
- 2F** Track X_5 opphører, og GOSPA-T scoren faller.

5.2.3 OSPA-on-OSPA

Figurene 5.5a og 5.5b viser OSPA-on-OSPA scoren for henholdsvis parametersett 1 og parametersett 2.



Figur 5.5 OSPA-on-OSPA score for tracker 1 og tracker 2. (a) viser resultatene for parametersett 1, (b) viser resultatene for parametersett 2.

Tracker 1 - parametersett 1

- 1a** Tracker 1 begynner med en OSPA-on-OSPA score på 4000, som er den gjennomsnittlige avstanden ved starten. Deretter faller OSPA-on-OSPA scoren. Dette skyldes at track X_3 nærmer seg mål Y_2 .
- 1b** Ved dette tidspunktet oppstår track X_2 , som er et ikke-assosiert track. X_2 bidrar dermed med en avstand på $c = 10000$, og OSPA-on-OSPA scoren, som er gjennomsnittet av avstandene, hopper kraftig.
- 1c** Track X_3 krysser mål Y_2 , og OSPA-on-OSPA scoren går fra å være avtakende til å være stigende, siden avstanden mellom tracket og målet øker.
- 1d** Track X_2 opphører å eksistere, og OSPA-on-OSPA scoren faller kraftig

Tracker 2 - parametersett 1

- 2a** Tracker 2 begynner med en OSPA-on-OSPA score på 6500. Scoren er så høy fordi trackeren bare leverer ett track til to mål. Y_2 , som ikke er assosiert med noe track, bidrar med en avstand c , og trekker opp snittet.
- 2b** OSPA-on-OSPA scoren holder seg stabil fram til track X_5 oppstår. Hvert mål kan nå assosieres med et nærliggende track, og OSPA-on-OSPA scoren går drastisk ned over en håndfull tidssteg. OSPA-on-OSPA scoren fortsetter deretter å avta, siden track X_5 har en minkende avstand til mål Y_2 .
- 2c** Track X_4 opphører, og OSPA-on-OSPA scoren hopper, siden vi nå igjen bare har ett track til to mål. Som ved **2b** skjer hoppet over en håndfull tidssteg.
- 2d** Track X_6 oppstår, og OSPA-on-OSPA scoren faller.
- 2e** Track X_5 opphører, og OSPA-on-OSPA scoren hopper.

Tracker 1 - parametersett 2

- 1A** Tracker 1 begynner med en OSPA-on-OSPA score på 4000, som er den gjennomsnittlige

avstanden ved starten. Deretter faller OSPA-on-OSPA scoren. Dette skyldes at track X_3 nærmer seg mål Y_2 .

- 1B** Ved dette tidspunktet oppstår track X_2 , som er et ikke-assosiert track. X_2 bidrar dermed med en avstand på $c = 5000$, og OSPA-on-OSPA scoren, som er gjennomsnittet av avstandene, hopper noe.
- 1C** Track X_3 krysser mål Y_2 , og OSPA-on-OSPA scoren går fra å være avtakende til å være stigende, siden avstanden mellom tracket og målet øker.
- 1D** Track X_2 opphører å eksistere, og OSPA-on-OSPA scoren faller noe.

Tracker 2 - parametersett 2

- 2A** Tracker 2 begynner med en OSPA-on-OSPA score på 4000. Trackeren leverer bare ett track til to mål Y_2 , som ikke er assosiert med noe track, bidrar med en avstand c , og trekker opp snittet. Scoren er en del lavere enn for parametersett 1 fordi parameteren c har en lavere verdi.
- 2B** OSPA-on-OSPA scoren holder seg stabil fram til track X_5 oppstår. Hvert mål kan nå assosieres med et nærliggende track, og OSPA-on-OSPA scoren går noe ned. OSPA-on-OSPA scoren fortsetter å avta, siden track X_5 har en minkende avstand til mål Y_2 .
- 2C** Track X_4 opphører, og OSPA-on-OSPA scoren hopper.
- 2D** Track X_6 oppstår, og OSPA-on-OSPA scoren faller.
- 2E** Track X_5 opphører, og OSPA-on-OSPA scoren hopper noe.

5.2.4 Betraktninger rundt resultatene

Med to trackere, to parametersett og tre forskjellige enkeltmetrikker får vi 12 forskjellige resultater. Over er disse gått igjennom i detalj. I dette underkapittelet forsøker vi å finne de mest interessante trekkene ved enkeltmetrikkene, være det seg hvordan en metrikk oppfører seg med de forskjellige parametersettene, forskjellen på to metrikker, eller annet. Under følger en liste over disse betraktningene.

5.2.4.1 Effekten av parametervalg: OSPA-T og GOSPA-T

Vi ser i figurene 5.3a og 5.3b at formen til OSPA-T scoren til tracker 1 er den samme i begge figurene, selv om verdien har endret seg. For tracker 2 er dette ikke tilfelle. Ved punkt **2c** opphører track X_4 , og scoren hopper. Vi har ikke tilsvarende hopp i figur 5.3b. Dette skyldes at valget av parametere er slik at straffen for manglende track (og dermed ikke-assosiert mål Y_1) er lik bidraget vi fikk fra track-mål-paret X_4-Y_1 . Dette er ikke nødvendigvis en uønsket effekt, men det illustrerer tydelig hvordan parametervalg kan påvirke enkeltmetrikkenscoren.

Vi ser en tilsvarende effekt for GOSPA-T og tracker 2, hvor vi for parametersett 1 ikke ser noe tegn på at et track har opphørt i midten av scenarioet, mens effekten er der for parametersett 2.

5.2.4.2 Effekten av scenario: OSPA-on-OSPA

Det er en veldig merkbar forskjell på OSPA-on-OSPA scoren til tracker 1 og tracker 2 i figurene 5.5a og 5.5b: Tracker 1 har markante og plutselige hopp i score, mens tracker 2 har en mer

gradvis og avrundet endring i score. Dette skyldes hvordan scenarioene er designet, og hvordan OSPA-on-OSPA-scoren håndterer problemene lagt inn i scenariene.

I scenario 1 (5.5a) har vi et ekstra track, X_2 . Dette tracket er aldri assosiert med noe mål, og vil dermed bidra med straffeavstanden Δ i hvert tidssteg hvor det kommer med bidrag. Vi ser i formel 4.14 at bidrag fra overflødige track (eller mål) ikke blir vektet med vektfunksjonen w , og vi har derfor brå hopp og fall. Tracker 2 har også tidsintervaller hvor det er et ekstra mål, med påfølgende høyere score. Forskjellen her er at det ekstra målet senere (eller tidligere) blir assosiert med et track. Dermed blir vektfunksjonen benyttet, og hoppene og fallene blir mer gradvise.

5.2.4.3 OSPA-on-OSPA har ikke noen global assosiasjon

Et av trekkene til både OSPA-T scoren og GOSPA-T scoren til tracker 2 er at den ikke er symmetrisk om midten av scenarioet, men er lavere i andre halvdel. Dette skyldes at det er track X_6 som har blitt assosiert med mål Y_1 i den globale assosiasjonen, og dette paret har dermed ikke en straffeavstand α , noe paret X_4 - Y_1 har. OSPA-on-OSPA har ikke noen global assosiasjon, og hadde vært helt symmetrisk om midten av tidsrommet hvis det ikke var for tidsvektingen, som gjør at hopp og fall ikke er umiddelbare.

5.2.4.4 GOSPA-T tar ikke gjennomsnittet

OSPA-T og OSPA-on-OSPA tar begge gjennomsnittet av alle avstandsbidragene. GOSPA-T tar derimot summen av alle bidragene. Dette gjør at store mengder falske track eller ikke-trackede mål straffes hardere. I scenariene som vi har designet her, er ikke effekten av dette veldig synlig, men hvis vi ser på tracker 1 når vi bruker parametersett 2 i de forskjellige enkeltmetrikkene ser vi at vi får et hopp når det ikke-assosierte tracket X_2 oppstår. Dette hoppet er mye høyere i GOSPA-T scoren enn det er i OSPA-T eller OSPA-ON-OSPA.

6 Oppsummering

For å kunne sammenlikne trackere med hverandre trenger vi kvalitetsmetriker som setter tall på ytelsen til en tracker. Denne rapporten beskriver noen slike kvalitetsmetriker og anvender de på et eksempelscenario.

Metrikkene kommer i to hovedtyper: multimetriker og enkeltmetriker. Multimetriker kjennetegnes ved at man har mange forskjellige ytelsesmål for en tracker og beregner alle sammen. Resultatet blir en tabell per tracker som beskriver ytelsen. Hvilke ytelsesmål vi ønsker å inkludere velger vi selv. Enkeltmetriker gir deg ett tall for kvaliteten per tidssteg og kan lett visualiseres som en graf over tidsrommet. Denne rapporten har satt opp et sett med ytelsesmål brukt i multimetriker og ser på tre forskjellige enkeltmetriker: OSPA-T, GOSPA-T og OSPA-on-OSPA.

Arbeidet er gjort i FFI-prosjekt 1467 - *Robust og modulbasert målfølgning*. Dette prosjektet lager et rammeverk for å teste ut forskjellige estimatorer og assosiasjonsalgoritmer innenfor tracking, og planen er å bruke kvalitetsmetrikkene beskrevet i denne rapporten til å vurdere ytelsen til de forskjellige metodene. Dette vil i tillegg gi oss kunnskap om og erfaring med å bruke kvalitetsmetrikkene. Vi unnlater derfor å komme med en anbefaling her om hvilken metrikk som burde brukes, men håper at en slik anbefaling kan komme i senere rapporter fra prosjekt 1467.

Referanser

- [1] Oliver E. Drummond. *Methodologies for performance evaluation of multitarget multisensor tracking*. Proceedings of SPIE, Signal and Data Processing of Small Targets 1999
- [2] A.A. Gorji, R. Tharmarasa and T. Kirubajan. *Performance Measures for Multiple Target Tracking Problems*. 14th International Conference on Information Fusion, 2011
- [3] A.S. Rahmatullah, A. F. García-Fernández and Lennart Svensson. *Generalized optimal sub-pattern assignment metric*. 20th International Conference on Information Fusion, 2017
- [4] T. Vu and R. Evans. *A New Performance Metric for Multiple target tracking based on Optimal Subpattern Assignment*. 17th International Conference on Information Fusion, 2014
- [5] M. Beard, Ba-Tuong Vo and Ba-Ngu Vo *OSPA⁽²⁾: Using the OSPA Metric to Evaluate Multi-target Tracking Performance*. 2017 International Conference on Control, Automation and Information Sciences, 2017
- [6] D. Schumacher, Ba-Tuong Vo and Ba-Ngu Vo *A Consistent Metric for Performance Evaluation of Multi-Object Filters*. IEEE Transactions on Signal Processing, 2008

About FFI

The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's MISSION

FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's VISION

FFI turns knowledge and ideas into an efficient defence.

FFI's CHARACTERISTICS

Creative, daring, broad-minded and responsible.

Om FFI

Forsvarets forskningsinstitutt ble etablert 11. april 1946. Instituttet er organisert som et forvaltningsorgan med særskilte fullmakter underlagt Forsvarsdepartementet.

FFIs FORMÅL

Forsvarets forskningsinstitutt er Forsvarets sentrale forskningsinstitusjon og har som formål å drive forskning og utvikling for Forsvarets behov. Videre er FFI rådgiver overfor Forsvarets strategiske ledelse. Spesielt skal instituttet følge opp trekk ved vitenskapelig og militærteknisk utvikling som kan påvirke forutsetningene for sikkerhetspolitikken eller forsvarsplanleggingen.

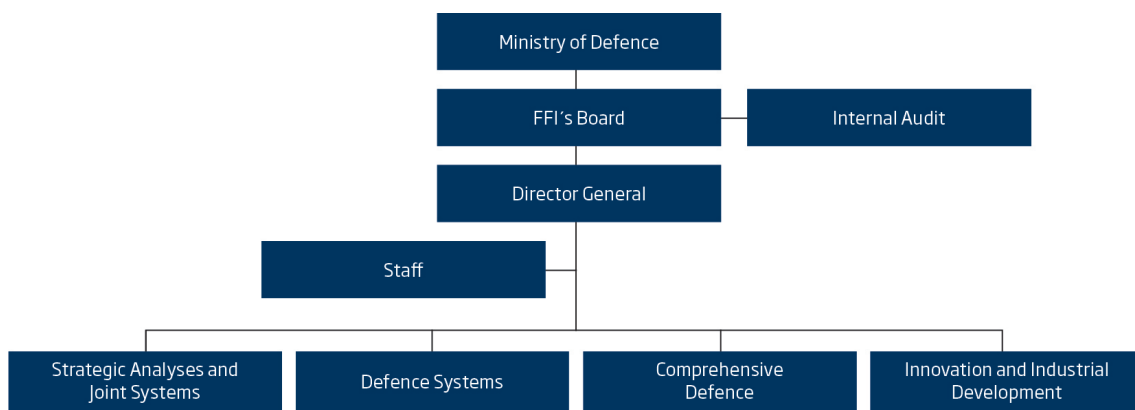
FFIs VISJON

FFI gjør kunnskap og ideer til et effektivt forsvar.

FFIs VERDIER

Skapende, drivende, vidsynt og ansvarlig.

FFI's organisation



Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: ffi@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: ffi@ffi.no