


Godkjent
Kjeller 22 juli 1994


G N Johnsen
Forskningsjef


INTRODUKSJON TIL SIMULERING

SKJELLAND Nils Espen,
FEET Else Helene,
FRIHAGEN Jon

FFI/RAPPORT-94/03698

FORSVARETS FORSKNINGSINSTITUTT
Norwegian Defence Research Establishment
Postboks 25, 2007 Kjeller, Norge

REPORT DOCUMENTATION PAGE

1) PUBL/REPORT NUMBER FFI/RAPPORT-94/03698 1a) JOB REFERENCE FFISYS/651/410	2) SECURITY CLASSIFICATION UNCLASSIFIED 2a) DECLASSIFICATION/DOWNGRADING SCHEDULE	3) NUMBER OF PAGES 41		
4) TITLE INTRODUKSJON TIL SIMULERING (Introduction to Simulation)				
5) NAMES OF AUTHOR(S) IN FULL (surname first) SKJELLAND Nils Espen, FEET Else Helene, FRIHAGEN Jon				
6) DISTRIBUTION STATEMENT Distribution unlimited (Offentlig tilgjengelig)				
7) INDEXING TERMS IN ENGLISH: <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> a) <u>Operations Research</u> b) <u>Simulation</u> c) <u>Modelling</u> d) <u>Simulation models</u> e) <u>System development</u> </td> <td style="width: 50%; vertical-align: top;"> IN NORWEGIAN: a) <u>Operasjonsanalyse</u> b) <u>Simulering</u> c) <u>Modellering</u> d) <u>Simuleringsmodeller</u> e) <u>Systemutvikling</u> </td> </tr> </table>			a) <u>Operations Research</u> b) <u>Simulation</u> c) <u>Modelling</u> d) <u>Simulation models</u> e) <u>System development</u>	IN NORWEGIAN: a) <u>Operasjonsanalyse</u> b) <u>Simulering</u> c) <u>Modellering</u> d) <u>Simuleringsmodeller</u> e) <u>Systemutvikling</u>
a) <u>Operations Research</u> b) <u>Simulation</u> c) <u>Modelling</u> d) <u>Simulation models</u> e) <u>System development</u>	IN NORWEGIAN: a) <u>Operasjonsanalyse</u> b) <u>Simulering</u> c) <u>Modellering</u> d) <u>Simuleringsmodeller</u> e) <u>Systemutvikling</u>			
THESAURUS REFERENCE:				
8) ABSTRACT (continue on reverse side if necessary) <p>Simulation is an operations research method (OR) for analysing problems in large and complex systems, for example military combat. During more than 20 years, the Norwegian Defence Research Establishment (NDRE) has built up knowledge in the development, evaluation, and use of simulation models.</p> <p>Simulation models can also be used for other military purposes than traditional OR. For example, simulation is central to computer assisted exercises (CAX) and real-time systems for decision support.</p> <p>Use of simulation requires knowledge of the main ideas and concepts in this area. The report introduces these ideas and concepts, illustrated with a small example. The report also describes some of the simulation models in use at NDRE.</p>				
9) DATE 22 July 1994	AUTHORIZED BY This page only  N Johnsen	POSITION Chief Scientist		

INNHold

	side
1	INNLEDNING 5
2	HOVEDIDÉENE BAK SIMULERING 6
2.1	Systemer 6
2.2	Modeller 6
2.3	Simulering 8
2.4	Simulering på datamaskiner 8
3	UTVIKLING AV SIMULERINGSMODELLER 10
3.1	Prinsipiell oppbygning av datasystemer for simulering 10
3.2	Utvikling av simuleringsmodeller 11
3.2.1	Systemstudie 11
3.2.2	Modellering 12
3.2.3	Programmering 13
3.2.4	Verifikasjon og validering 13
3.2.5	Systemutvikling – en iterativ prosess 15
3.3	Anvendelse av simuleringsmodeller til analyseformål 15
4	KARAKTERISTIKA VED SIMULERINGSMODELLER 17
4.1	Oppløsning 17
4.2	Håndtering av usikkerhet 18
4.3	Operasjonsmodus 19
4.4	Oppsummering 19
5	ILLUSTRASJON AV SIMULERING 20
5.1	Beskrivelse av modellen 20
5.2	Datainnsamling 21
5.3	Simulering med én stridsvogn og ett PB-våpen 22
5.4	Bruk av modellen 27
6	EKSEMPLER PÅ SIMULERINGSMODELLER VED FFI 28
6.1	Sjøinvasjonsmodellen 28
6.2	Simuleringsmodell for duell mellom helikopter og MTB i lurkeposisjon ... 29
6.3	Sjøstridsmodellen SLOC 29
6.4	Sonarprediksjonsmodellen Generic Sonar Model 30
6.5	Sårbarhetsberegningsmodellen General Vulnerability Assessment Model .. 31
6.6	Luftstridsmodellen TAC THUNDER 31
6.7	Luftstridsmodellen GILMINI 32
6.8	Avskjæringsmodellen TORNADO 32

6.9	Flystasjonsskademodellen	33
6.10	Luftvernmodellen LVMOD	34
6.11	Landstridsmodellen DYNACOM	35
6.12	Landstridsmodellen PABST	35
6.13	Landstridsmodellen DIVMOD	36
6.14	Landstridsmodellen TRIAMOS	37
6.15	Kostnadsberegningssmodellen KOSTMOD	37
	LITTERATURLISTE	39
	FORDELINGSLISTE	41

INTRODUKSJON TIL SIMULERING

1 INNLEDNING

Den andre verdenskrig førte til utstrakt bruk av naturvitenskaplige metoder og teknikker i studiet av militære operasjoner – først innenfor anti-ubåtkrigføring i Atlanterhavet, siden innenfor en rekke andre sektorer. Denne virksomheten ble etter hvert kalt operasjonsanalyse.

Etter krigen ble operasjonsanalysens virkefelt utvidet fra primært å utarbeide retningslinjer for effektiv bruk av styrker, til også å omfatte utvikling av fremtidige våpensystemer og langsiktig forsvarsplanlegging. Samtidig ble operasjonsanalyse tatt i bruk på mange sivile områder som transport, luftfart, økonomi, utdanning og markedsføring. I dag bidrar operasjonsanalyse innenfor de fleste felter, og anvendelsesområdet for operasjonsanalytiske metoder og teknikker utvides stadig.

Forsvarets forskningsinstitutt (FFI) gir, og har lenge gitt, råd til forsvarsledelsen om alt fra Forsvarets oppbygging i stort hvor forsvarsgrenene og forsvarsgrenenes ulike sektorer veies mot hverandre, til valg og utforming av konkrete våpensystemer og utforming av taktikk. I disse analysene står den operasjonsanalytiske teknikken *simulering* sentralt. FFI har bygget opp et betydelig bibliotek av simuleringsmodeller og kompetanse i å bruke og evaluere slike modeller.

FFIs simuleringsmodeller og -kompetanse kan også utnyttes til andre formål enn tradisjonelle analyser. Ett eksempel er *datastøttet opplæring og øving* hvor det i dag pågår betydelig aktivitet. I mange utenlandske militære miljøer er det stor interesse for å benytte simuleringsverktøy også på andre felter, f eks *systemer for sanntids beslutningsstøtte*.

Denne rapporten søker å gi en grunnleggende forståelse av og et begrepsapparat for simulering. Hovedidéer og -begreper knyttet til simulering er beskrevet i neste kapittel. I kapittel 3 drøftes forhold rundt utvikling av simuleringsverktøy, og simuleringsmodellenes ulike karakteristika utdypes i kapittel 4. Kapittel 5 søker å illustrere idéene og begrepene gjennom et eksempel. I siste kapittel presenteres noen av FFIs simuleringsmodeller. Arbeidet bygger på en rekke kilder, men de viktigste er [1], [2], [3] og [4].

Rapporten er et resultat fra FFI/PROSJEKT 651-SYS/410 "Innføring av datastøttet opplæring og øving av ledere og staber i Forsvaret". Prosjektet arbeider mot skoler og operative ledd i alle forsvarsgrener, det ble startet i desember 1993 og avsluttes i juni 1996. Oppdragsgiver for prosjektet er FO/O.

2 HOVEDIDÉENE BAK SIMULERING

Operasjonsanalyse er en vitenskaplig metode for å gi beslutningstagere et kvantitativt grunnlag for deres beslutninger. Operasjonsanalyse forutsetter at det finnes:

- noen som skal ta en beslutning, dvs en beslutningstager
- en målsetting for beslutningen
- flere beslutningsalternativer (i motsatt fall er en videre analyse unødvendig)

Vanligvis utgjør det kvantitative grunnlaget fra operasjonsanalysen bare én del av det totale beslutningsgrunnlaget.

Simulering er en av de mange metoder som benyttes i operasjonsanalysen ved siden av f eks optimering, køteori og sannsynlighetsregning. Simulering er å gjennomføre eksperimenter med en modell av et system. En nærmere beskrivelse av simulering krever derfor kjennskap til begrepene *system* og *modell*.

2.1 Systemer

For å kunne analysere ulike beslutningsalternativer, må man forstå den del av virkeligheten som beslutningen skal påvirke. I operasjonsanalytisk sammenheng kalles den aktuelle delen av virkeligheten for et system:

Definisjon 1

Et **system** er en del av virkeligheten som kan sees på som en helhet. Det består av en samling komponenter som påvirker hverandre gjensidig.

Operasjonsanalyse av militære problemer i krigs- og fredstid er relativt forskjellig. I krigstid settes naturlig nok nåtidens problemer i fokus, og arbeidet baseres ofte på resultater fra krigføringen. Krig er imidlertid en atypisk situasjon – i hvert fall i vår del av verden. Operasjonsanalyse har derfor fått størst betydning i forbindelse med forsvarsplanlegging, våpenutvikling og -anskaffelse i fredstid. Dette krever en forståelse av hvordan eventuell krigføring i fremtiden vil forløpe. For å kunne gjennomføre en objektiv, kvantitativ analyse av et slikt fremtidig system, kreves mer enn bare forståelse av systemet – forståelsen må i tillegg beskrives vha en modell.

2.2 Modeller

I denne sammenheng er en modell følgende:

Definisjon 2

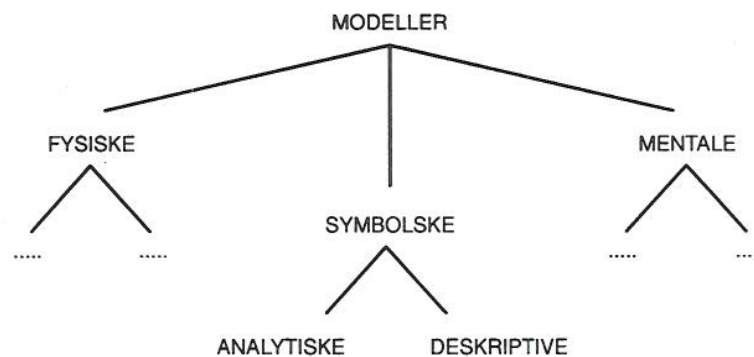
En **modell** er en forenklet representasjon av et system som har de samme relevante egenskapene som systemet.

Hvilke egenskaper som ansees å være relevante, avhenger av hvilke aspekter man ønsker å studere i systemet. Det finnes derfor vanligvis ingen universell modell av et system som kan tjene et hvert tenkelig analyseformål.

Generelt kan modeller tjene mange formål:

- gi økt forståelse av systemet gjennom organisering av tanker og idéer om systemets oppbygning og virkemåte
- støtte kommunikasjon gjennom en presis og utvetydig beskrivelse av systemet
- muliggjøre eksperimenter, siden en modell kan forandres på mange måter som ikke er mulig med selve systemet
- muliggjøre prediksjon av hvordan systemet forventes å utvikle seg i fremtiden
- støtte opplæring og øving av de som skal operere i eller i tilknytning til systemet

Modeller kan uttrykkes på ulike måter og kan derfor kategoriseres som fysiske, mentale eller symbolske modeller¹, jfr figur 2.1. En globus er et eksempel på en *fysisk* modell av jordas overflate, mens en persons oppfatning av bilmotorens oppbygning og virkemåte er et eksempel på en *mental* modell. Denne rapporten omhandler i hovedsak *symbolske* modeller der symboler benyttes for å betegne komponenter og sammenhenger i et system. Disse symbolene assosieres mentalt med det de representerer selv om det ikke er noen umiddelbar visuell overensstemmelse.



Figur 2.1 Kategorisering av modeller

Symbolske modeller kan videre deles i analytiske og deskriptive modeller. *Analytiske* modeller kan beskrives på en generell matematisk form, og løsninger på beslutningsproblemer kan utledes matematisk. Ett eksempel på en analytisk modell er et ligningssett. Ulike beslutninger kan representeres gjennom ulike koeffisienter, grensebetingelser osv, og ligningssettet må løses for hver slik konstellasjon av forutsetninger. Som oftest er det bare svært avgrensede og relativt enkle problemer som er egnet til å bli løst med analytiske modeller. En *deskriptiv* modell søker bare å gi en beskrivelse av systemet. En slik beskrivelse vil ikke nødvendigvis angi hvordan problemet skal løses, men modellen gjør det mulig å studere systemets oppførsel i ulike situasjoner. Simulering er én måte å utnytte deskriptive modeller på.

1. En annen mye brukt klassifikasjonsmåte er *ikoniske*, *analoge* og *symbolske* modeller.

2.3 Simulering

Simulering kan, som tidligere nevnt, defineres på følgende måte:

Definisjon 3

Simulering er å gjennomføre eksperimenter med en modell av et system.

Simulering står i motsetning til å eksperimentere direkte med systemet eller å løse analytiske modeller av systemet matematisk.

Et *eksperiment* innebærer å observere oppførselen til systemet – eller en modell av systemet – under gitte betingelser. Eksperimenter direkte med systemet kan, som tidligere nevnt, ofte være for kostbare, for tidkrevende eller av andre grunner uakseptable å gjennomføre. Analytiske modeller på sin side vil ofte innebære så sterke forutsetninger at modellene ikke fanger opp alle relevante egenskaper i systemet. Simulering er derfor ofte en siste utvei for å analysere store, komplekse systemer. Simulering gir ingen garanti for å finne den beste løsningen, men man finner som regel en tilstrekkelig god løsning.

Simuleringsmodellenes fremste karakteristika er at de krever input data og produserer output data, jfr figur 2.2. I motsetning til analytiske modeller *løses ikke* simuleringsmodeller, de *utføres*. Følgelig er ikke simulering noen teori men en *metode* for problemløsning.



Figur 2.2 Simulering krever input data og produserer output data

2.4 Simulering på datamaskiner

Simulering kan anvendes på et bredt spektrum av modeller – både fysiske, mentale og symbolske modeller – og krever ikke nødvendigvis datamaskiner. Når datamaskiner benyttes, (*datamaskinbasert simulering*) brukes imidlertid deskriptive modeller.

Datamaskinbasert simulering krever at modellene beskrives som dataprogrammer, dvs i et programmeringsspråk og på en måte som gjør datamaskinene i stand til å tolke og utføre modellene. Det finnes en rekke programmeringsspråk beregnet på simulering, f eks Simula, GPSS, Simscript, Modsim og Smalltalk [5].

Den viktigste grunnen til å kjøre simulering på datamaskiner, er maskinenes overlegne evne til å simulere hurtig og korrekt. (De feil som eventuelt oppstår, skyldes feil i programmene – ikke i maskinene.) Store modeller som svarer til tusener av linjer med programkode, kan i dag

kjøres i løpet av sekunder på middels kraftige datamaskiner. Dette er helt avgjørende for – innen tilgjengelige tids- og ressursrammer – å kunne gjennomføre det store antall eksperimenter som må til for å finne gode løsninger på problemer knyttet til store og komplekse systemer. Manuelle eksperimenter med tilsvarende modeller eller med det virkelige systemet, kan aldri gjennomføres på en tilsvarende komprimert og konsistent måte.

Innenfor programmeringsverdenen finnes det to hovedperspektiver på simuleringmodeller. Dette skyldes at tidsdimensjonen kan håndteres på ulike måter. I *kvasi-kontinuerlig* simulering kan modellene ofte best beskrives med et kontinuerlig tidsforløp, men modellen må diskretiseres (dvs tidsskalaen deles opp i små, jevne skritt) for å kunne simuleres vha datamaskiner. I *diskret hendelsesstyrt* simulering endres systemene som følge av hendelser som inntreffer på ulike tidspunkt. Det skjer ikke noe mellom disse hendelsene, og modellene hopper over alle mellomliggende tidsperioder. Vanligvis brukes kvasi-kontinuerlig simulering når systemenes oppførsel under hele simuleringens tidsforløp er interessant. I motsatt fall brukes som regel diskret hendelsesstyrt simulering.

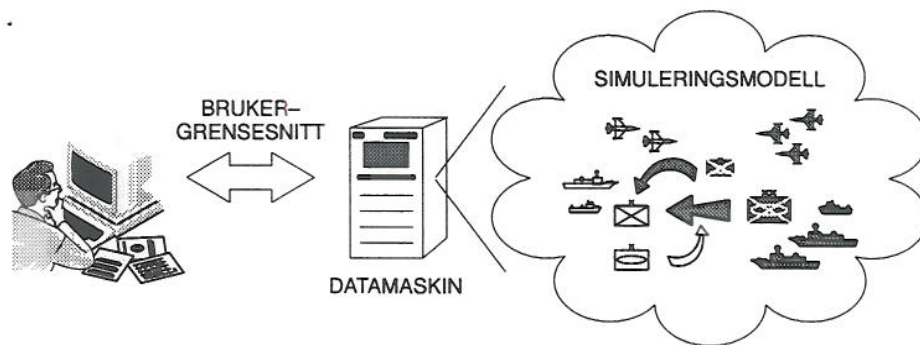
Tradisjonelt har de ulike programmeringsspråkene for simulering enten gitt muligheter for kvasi-kontinuerlig simulering eller diskret hendelsesstyrt simulering. Det er imidlertid utviklet programmeringsspråk som støtter begge tilnæringsmåter, dvs visse aspekter ved et system kan modelleres som kvasi-kontinuerlige forløp, mens andre aspekter kan modelleres vha hendelser [3].

3 UTVIKLING AV SIMULERINGSMODELLER

Utvikling av en simuleringsmodell er ofte en tid- og ressurskrevende prosess. I dette kapitlet beskrives denne utviklingsprosessen nærmere. Først omtales imidlertid simuleringsystemenes prinsipielle oppbygning. Til slutt illustreres anvendelse av simuleringsmodeller til analyseformål.

3.1 Prinsipiell oppbygning av datasystemer for simulering

Datasystemer for simulering består grovt sett av en simuleringsmodell, en datamaskin og et brukergrensesnitt, jfr figur 3.1.



Figur 3.1 Prinsipiell oppbygning av datasystemer for simulering

Simuleringsmodellen utgjør kjernen i denne type verktøy. Det er simuleringsmodellen som representerer systemet, dvs den militære virkelighet, og modellens kvaliteter er avgjørende for resultatenes gyldighet.

Datamaskinen er naturligvis nødvendig for å kjøre en datamaskinbasert simuleringsmodell. Mens maskindelen i datasystemene tidligere representerte en kostnads- eller effektivitetsmessig flaskehals, har dagens pris-/ytelsesforhold på datamaskiner langt på veg eliminert denne problemstillingen. I dag vil investering i maskinvare som regel utgjøre en liten del av de totale kostnadene forbundet med utvikling eller anskaffelse av et slikt system. Hoveddelen av kostnadene vil vanligvis gå med til utvikling eller anskaffelse av selve simuleringsmodellen med et tilhørende brukergrensesnitt.

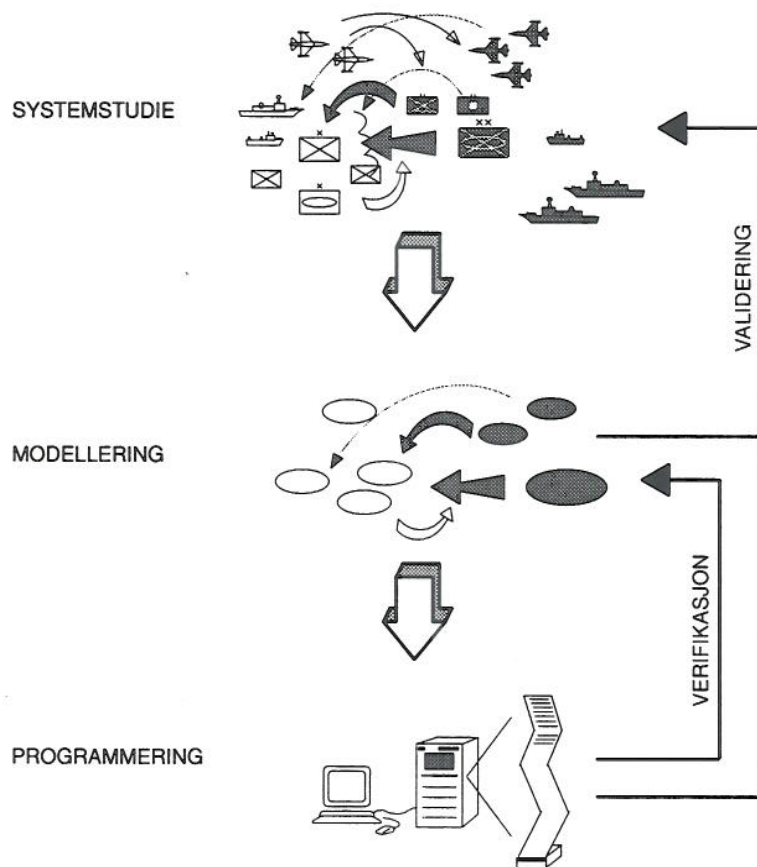
Brukergransnittet gjør det mulig å kommunisere med simuleringsmodellen, f eks å gi ordre om hvordan styrkene skal disponeres under simuleringen, samt å få tilbake resultatene fra modellen. Systemenes brukergrensesnitt har tradisjonelt vært basert på tekst og tall, og bruken av systemene har ofte vært vanskelig og tungvint. I dag er det en klar trend mot mer brukervennlige grensesnitt basert på grafiske symboler, kart og menyer.

Resten av denne rapporten vil hovedsakelig omhandle simuleringsmodeller siden modellene er og blir kjernen i denne type systemer. Utvikling av grafiske brukergrensesnitt er imidlertid programmeringsmessig sett ofte en like stor utfordring [5].

3.2 Utvikling av simuleringsmodeller

Utviklingen av en simuleringsmodell vil i grove trekk gjennomløpe fire faser, jfr figur 3.2:

- 1) Studie av systemet som skal modelleres
- 2) Utarbeidelse av en modell
- 3) Implementasjon av modellen (programmering)
- 4) Verifikasjon og validering av modellen



Figur 3.2 Faser i utviklingen av en simuleringsmodell

3.2.1 Systemstudie

Systemstudien søker å identifisere hvilke hovedfaktorer som påvirker systemets oppførsel – i militær sammenheng en gitt type stridsforløp på et gitt nivå – og hvordan disse sammenhengene arter seg. Dette krever inngående kjennskap til og forståelse av systemet, dvs analytikerens må ha et nært samarbeide med militære kompetansemiljøer og -personer. I tillegg vil analytikerens ofte kunne utnytte resultater og kanskje også modeller fra andre analyser i inn- og utland.

Alle systemer kan sees på som delsystemer i et større system. Analytikerens perspektiv bestemmer hvilke av virkelighetens aspekter som er relevante og hvilke som er irrelevante. Dette perspektivet bestemmes i sin tur langt på veg av det foreliggende problem. Systemstudien

innbefatter denne oppgangen av grensene mellom systemet og resten av virkeligheten (*systemgrensene*). De komponenter som ligger utenfor det spesifiserte systemet, men likevel har en relasjon til systemet, kalles systemets *omgivelser* eller *miljø*.

3.2.2 Modellering

Studien av systemet er utgangspunktet for konstruksjonen av selve modellen. En modell må beskrive systemets struktur og oppførsel. Dette gjøres ved hjelp av:

- komponenter
- variable
- parametre
- funksjonelle sammenhenger

Komponentene representerer systemets bestanddeler, dvs substansen i systemet. Målbare egenskaper ved substansen beskrives vha variable og parametre. *Variable* kan anta ulike verdier avhengig av simuleringens rammebetingelser og hva som skjer i modellen. Variable deles ofte i *eksogene* variable som er gitt av forhold utenfor modellen, og *endogene* variable som produseres innenfor modellen. Variablenes verdi på et gitt tidspunkt under simuleringen sies å beskrive systemets *tilstand* på dette tidspunktet. *Parametre* er størrelser som operatøren av modellen kan tilordne ulike verdier avhengig av det man ønsker å studere. Parametrene endres ikke av modellen når modellen kjøres.

Funksjonelle sammenhenger beskriver det som skjer når substansen i systemet transformeres, dvs hvordan komponentene i modellen påvirker hverandre, og hvordan dette endrer systemets tilstand. De funksjonelle sammenhengene kan deles i *deterministiske* sammenhenger der tilstandsovergangene er entydige, og *stokastiske* sammenhenger der en gitt tilstand kan lede til flere alternative tilstander som følge av usikkerhet.

Utvikling av en modell består av analyse, abstraksjon og syntese. Siden systemets enheter og oppførsel er enklere å beskrive hver for seg enn samlet, splittes vanligvis systemet opp i mindre deler som behandles separat (*analyse*). En viktig målsetting i modelleringsfasen er å skape en tilstrekkelig enkel modell som er håndterbar innenfor arbeidets tids- og ressursrammer. Etter at systemet er oppdelt i hensiktsmessige bestanddeler, må disse delene og sammenhengene derfor forenkles (*abstraksjon*). Det er imidlertid ikke mulig å sikre gode resultater for systemet som helhet ved å studere og optimalisere komponentene og delsystemene enkeltvis. Delene i systemet må derfor kombineres etter at de er modellert hver for seg (*syntese*). Dette resulterer i én modell som beskriver hele systemet med alle dets relevante egenskaper.

Det finnes ingen standard fremgangsmåte for å bestemme hvilke deler og sammenhenger som bør tas med i en modell og hvilke som kan utelates uten at resultatene påvirkes nevneverdig. Samme system kan derfor gi opphav til mange mer eller mindre gode modeller i forhold til et

gitt formål. Modellering bygger på kunnskap, idéer og intuisjon som er opparbeidet under systemstudien, samt kunnskap om modellbygging generelt. Dette arbeidet er like mye en kunst som et håndverk.

Modellering innebærer også å sørge for nødvendige input data til modellen, dvs data for eksogene variable og data som inngår i de funksjonelle sammenhengene. Gode input data er avgjørende for at simuleringen skal gi en god beskrivelse av virkeligheten. Modellens utforming må derfor sees i sammenheng med tilgangen på data. Eventuell mangel på visse typer data kan føre til at modelleringen må gjøres på en annen måte enn det som ellers er naturlig.

Modelleringens betydning kan knapt overvurderes. En god modell og gode input data er avgjørende for å finne gode og troverdige løsninger på et problem gjennom simulering.

3.2.3 Programmering

På basis av modellen konstrueres *dataprogrammet* som gjør det mulig å kjøre modellen på en datamaskin. Denne fasen bygger på kunnskap om datamaskiner og hvordan disse kan programmeres.

Simuleringsmodeller gir ofte opphav til store og kompliserte dataprogrammer. Et gjennomtenkt design, strukturert koding og god dokumentasjon er viktig for å håndtere denne kompleksiteten. Likevel må denne fasen først og fremst karakteriseres som et håndtverk.

3.2.4 Verifikasjon og validering

Selv om man aldri kan bevise at en simuleringsmodell er korrekt, må man likevel forvise seg om at simuleringen gir en relativt god beskrivelse av virkeligheten. I sammenheng med datamaskinbasert simulering kan denne prosessen deles i verifikasjon og validering²:

Definisjon 4

Verifikasjon er å forvise seg om at et dataprogram oppfører seg slik som den underliggende modellen beskriver, dvs at det er overensstemmelse mellom modellen og dens representasjon (dataprogrammet).

Definisjon 5

Validering er å forvise seg om at modellen gir en rimelig beskrivelse av virkeligheten, dvs at det er overensstemmelse mellom systemet og dets representasjon (modellen).

2. Generelt knyttes ofte *verifikasjon* til modellens indre konsistens og *validering* til modellens overensstemmelse med den ytre virkelighet. I forbindelse med datamaskinbasert simulering er det imidlertid ikke modellene, men modellenes representasjon (dataprogrammene) som er beskrevet så formelt og detaljert at verifikasjon er mulig. Dermed er det naturlig å knytte verifikasjon og validering til hhv.dataprogram og modell.

Selv om programmering langt på veg kan karakteriseres som et håndverk, gjør de fleste mennesker – også programmerere – før eller siden feil. Et dataprogram er en samling av mange små, og hver for seg enkle, programbiter. For at et program – med kanskje titusener av programlinjer – skal være feilfritt, må *alle* disse linjene være riktige.

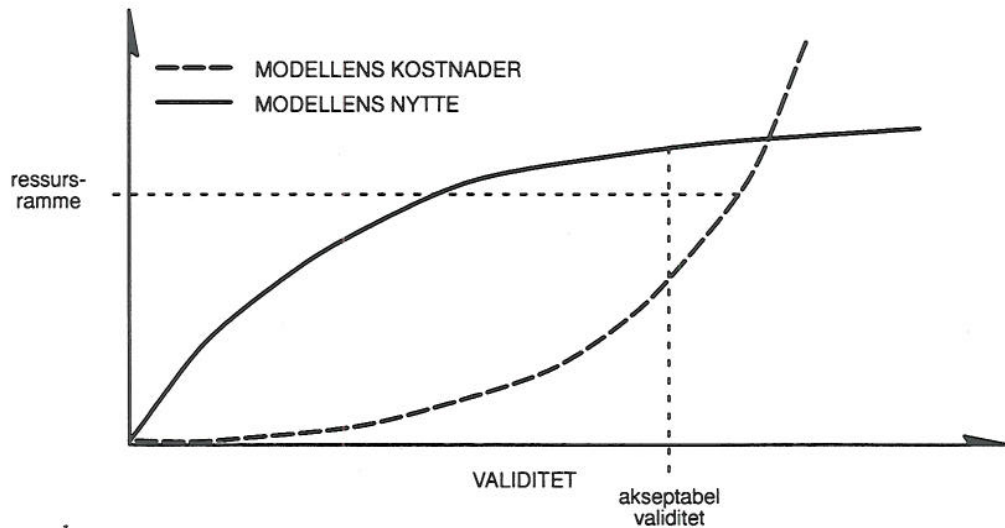
For å verifisere et program, benyttes mange metoder og teknikker, f eks testing av programmet med varierende input data, eksekvering av programmet linje for linje (eller modul for modul), manuell inspeksjon av programkoden (gjærne av andre programmerere), utskrift fra programmet i ulike feilsituasjoner osv.

Selv om validering i prinsippet er uavhengig av dataprogrammet og i størst mulig grad bør gjennomføres *før* programmeringen starter, vil det først være mulig å teste store deler av modellens validitet når modellen kan kjøres på en datamaskin, dvs *etter* at dataprogrammet er utviklet. Validering består ikke bare av å kontrollere kvaliteten til modellen og dataprogrammet, men også kvaliteten til de dataene som er benyttet. Validering av input data viser seg ofte vanskeligere enn validering av modell og programkode.

Validering er – både fra en prinsipiell og praktisk synsvinkel – problematisk. Dette skyldes primært at validering ofte skal gjøres mot en *fremtidig* og *usikker* virkelighet. Noen absolutt validitet er dermed utelukket. Likevel finnes det resultater – f eks fra tidligere krigføring, øvelser og forsøk i felt, i luft eller på sjø – som antas å ha en viss gyldighet også for fremtiden. Selv om man har resultater som gjenspeiler fremtidens virkelighet, er det likevel ikke trivielt å måle graden av overensstemmelse mellom disse og modellens resultater siden begge datasett vanligvis kommer fra prosesser med et betydelig innslag av tilfeldighet. Videre må eventuell signifikant uoverensstemmelse spores tilbake til modellen og/eller modellens input data.

Validering gjøres også gjennom vurderinger av de som kjenner virkeligheten best, dvs militære kompetansmiljøer og -personer (ekspertvurderinger). Videre analyseres ofte modellens resultater vha teori og statistiske testmetoder, og sammenlignes med tidligere forskningsresultater.

Siden simuleringens korrekthet (*validitet*) aldri kan bevises, er verifikasjon og validering et spørsmål om å bringe brukerens *tillit* til simuleringen opp til et akseptabelt nivå. Å øke simuleringens validitet krever ressurser, men samtidig vil modellens verdi for beslutningstageren også øke – dog med avtagende rate, se figur 3.3. På et tidspunkt vil det ikke være regningssvarende eller mulig å øke modellens validitet ytterligere, enten fordi kostnadene overstiger nytten eller fordi arbeidets ressursrammer er for snevre.



Figur 3.3 Sammenheng mellom validitet, kostnader og nytte, samt eksempel på nivåer for akseptabel validitet og arbeidets ressursramme

3.2.5 Systemutvikling – en iterativ prosess

Fasene i systemutviklingsprosessen forløper sjelden så sekvensielt som fremstillingen i dette kapitlet kan gi inntrykk av. I de fleste tilfeller gjentas trinnene i prosessen en rekke ganger – arbeidet starter med å beskrive deler av systemet, det utvikles en modell og deretter et data-program, programmet testes, testingen avslører feil dels i modellen og dels i programmet, feilene rettes, programmet testes på nytt etc. Deretter beskrives en ny del av systemet, det utvikles en modell for denne osv.

Denne iterative prosessen vil som regel ikke ta slutt selv om utviklingsarbeidet formelt sett er fullført og datasystemet tas i bruk. Bruk av systemet avdekker ofte feil, mangler og nye behov som leder til videreutvikling av systemet.

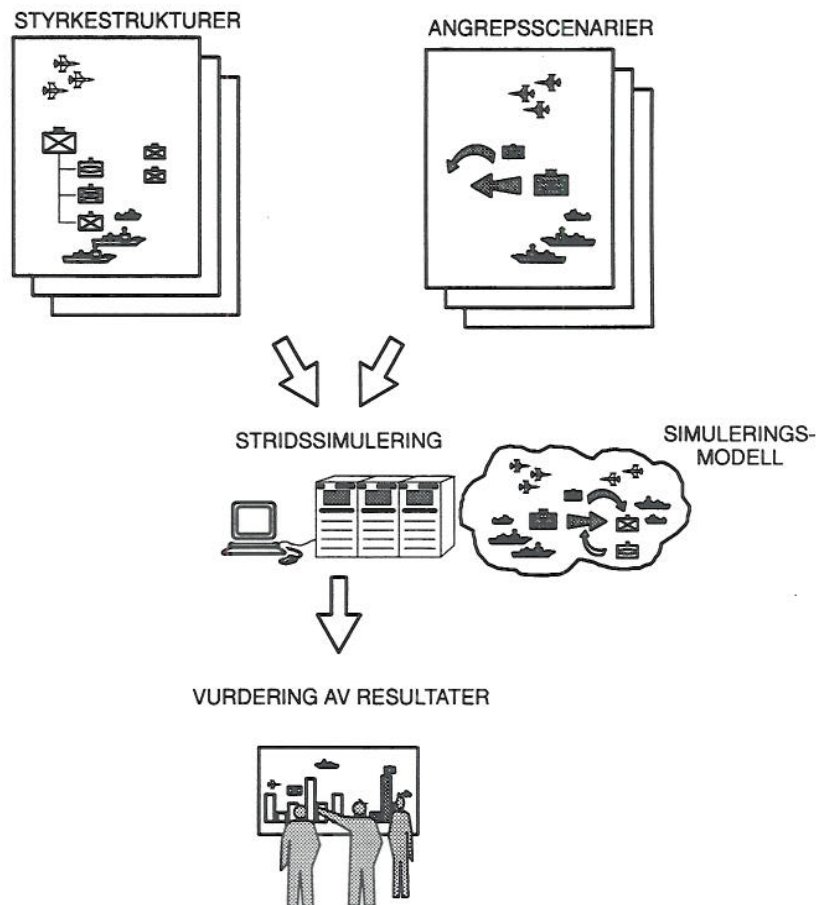
Dette kapitlet har drøftet utvikling av simuleringmodeller fra grunnen av. I praksis er man ofte i en annen situasjon. Eksempelvis kan det være aktuelt å modifisere eksisterende simuleringmodeller, å bruke deler av andre programmer (gjenbruk av programkode) eller å anskaffe simuleringmodeller som er utviklet av andre. I slike tilfeller vil systemutviklingsprosessen kunne få et helt annet forløp enn skissert i dette kapitlet.

3.3 Anvendelse av simuleringmodeller til analyseformål

Denne rapporten søker ikke å gi noen bred og dekkende fremstilling av hvordan analyser basert på simulering bør gjennomføres. Dette omfattende temaet vil bare bli illustrert på en enkel måte.

Figur 3.4 viser et eksempel på bruk av datamaskinbaserte simuleringmodeller for analyseformål. Simuleringen bygger på antagelser om norske styrker – beskrevet i ulike styrke-

strukturer – og motstanderens styrker og styrkebruk beskrevet i ulike angrepsscenarioer. Med utgangspunkt i disse antagelsene simuleres hendelsesforløpet under striden og resultater beregnes. Resultatene må imidlertid vurderes nøye siden overordnede effektivitetsmål ikke alltid kan kobles direkte til datamaskinenes resultater og ulike scenarier ofte gir ulike svar. Her spiller erfarne offiserer en viktig rolle.



Figur 3.4 Eksempel på datamaskinbasert simulering for analyseformål

Analyse vha simulering krever planlegging – både av hvordan eksperimentene skal legges opp i stort for å finne løsninger på de fundamentale problemene, og hvordan eksperimentene skal gjennomføres i detalj for å sikre statistiske tilfredsstillende resultater, holde tidsfrister o.l.

Analyse innebærer også *følsomhetsanalyse (sensitivitetsanalyse)* for å anslå følsomheten til de endelige svarene. Dette gjøres gjennom systematisk variasjon av verdiene til parametrene og/eller de eksogene variablene, og observere effektene på simuleringen. Dette viser for det første hvor sensitive resultatene er med hensyn til de parameterverdiene som er forutsatt. Hvis svarene endres radikalt med små variasjoner i én eller flere av disse verdiene, kan det forsvare bruk av mer tid eller penger for å oppnå bedre og sikrere estimater. For det andre viser sensitivitetsanalyse mulige virkninger av at systemets miljø endres. For det tredje kan sensitivitetsanalyse vise hvilke deler av modellen som er mest kritiske, og hvilke deler som kanskje kan forenkles i neste versjon.

4 KARAKTERISTIKA VED SIMULERINGSMODELLER

Simuleringsmodeller kan karakteriseres gjennom bl a deres *oppløsning*, *operasjonsmodus* og *håndtering av usikkerhet*. I dette kapitlet utdypes disse begrepene nærmere.

4.1 Oppløsning

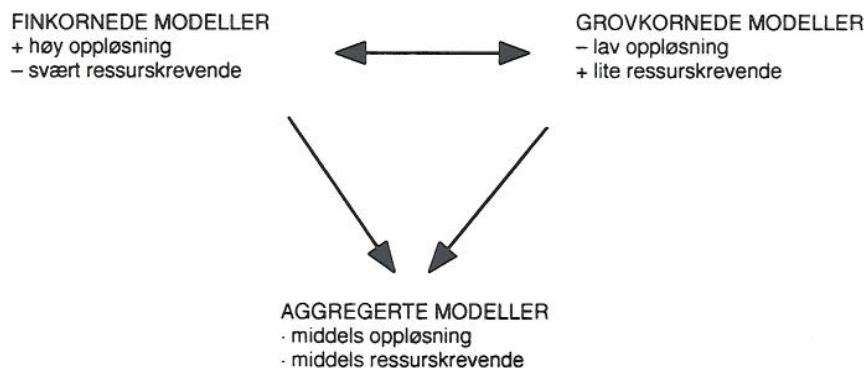
Oppløsning eller *detaljeringsnivå* inkluderer modellenes detaljeringsgrad mht styrker, våpen-systemer, terreng osv. Med hensyn til oppløsning kan modellene grovt sett deles inn i tre typer:

- finkornede modeller
- grovkornede modeller
- aggregerte modeller

Kjennetegnet ved *finkornede modeller* er høy oppløsning, dvs modellene beskriver mange av virkelighetens detaljer. Dette er gunstig siden flere detaljer ofte bidrar til en bedre beskrivelse av virkeligheten, men ulempene er at utvikling og bruk av slike modeller tenderer til å bli svært ressurskrevende. Dette skyldes både behovene på personellsiden til utvikling, bruk og vedlikehold av slike systemer, og behovene på utstyrssiden mht datakraft og datalager.

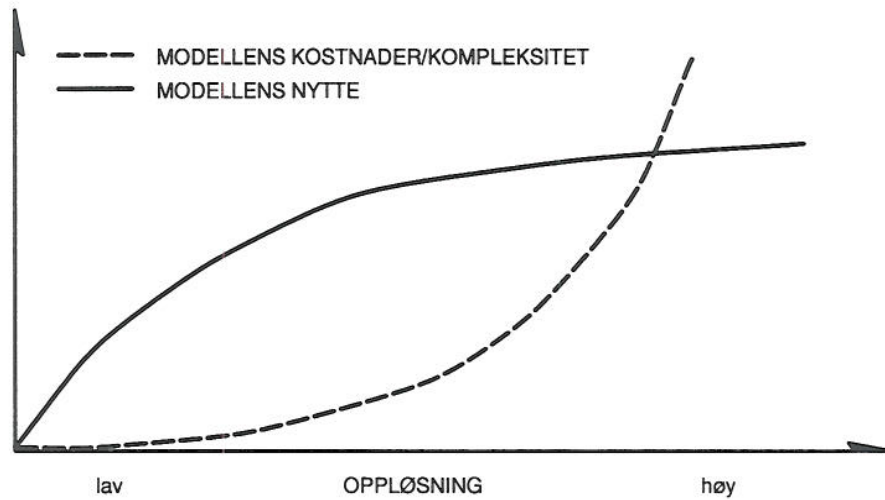
Grovkornede modeller kjennetegnes ved at de i stor grad forenkler og abstraherer virkeligheten. Dermed er det en fare for at vesentlige faktorer og sammenhenger utelates. Denne type modeller er imidlertid som oftest enkle å utvikle og bruke.

Aggregerte modeller representerer et kompromiss mellom de to ovennevnte modelltypene, jfr figur 4.1. De kjennetegnes – i likhet med grovkornede modeller – med at prosessen i stor grad abstraheres og forenkles, men dette gjøres ved hjelp av resultater fra mindre og mer finkornede modeller. På denne måten danner modellene et hierarki som fanger opp de fleste faktorer og sammenhenger som finnes i virkeligheten, uten at dette resulterer i én stor og kompleks modell.



Figur 4.1 Simulering – oppløsning

I likhet med validering er det viktig å avstemme kostnader og nytte når modellenes oppløsning fastlegges, se figur 4.2. Å øke modellens oppløsning krever ressurser, men modellens nytte kan også øke – dog med en avtagende rate. Det er dermed ikke hensiktsmessig å øke modellens oppløsning ut over et visst punkt – enten fordi de ekstra kostnadene overstiger den ekstra nytten man oppnår, eller fordi arbeidets ressursrammer er for snevre.



Figur 4.2 Sammenheng mellom en modells oppløsning, kostnader/kompleksitet og nytte

4.2 Håndtering av usikkerhet

Virkelighetens usikkerhet kan modelleres på to måter:

- deterministisk
- stokastisk

I en *deterministisk modell* baseres simuleringen på gjennomsnittsbetraktninger. Usikkerheten analyseres utenfor modellen gjennom beregning av gjennomsnittsverdier som er input data til modellen. Hver simulering med samme input data gir samme resultater, og det er derfor ikke nødvendig å gjennomføre mer enn én simulering med hver konstellasjon av input data. På den annen side kommer ikke virkelighetens variasjonsmuligheter frem i slike gjennomsnittsmodeller, og resultatet er ikke nødvendigvis en så god beskrivelse av systemets typiske oppførsel som gjennomsnittlige resultater fra stokastiske modeller.

I en *stokastisk modell* trekkes tilfeldighet inn i selve simuleringene. Små programbiter gjennomfører såkalte trekninger for å beregne utfallet av usikre begivenheter. Slike trekninger kan i praksis sammenlignes med å kaste terning, men det kastes naturligvis ikke terning helt tilfeldig – i det lange løp vil trekningene avspeile egenskapene til en sannsynlighetsfordeling som så godt som mulig beskriver det usikre fenomenet. Hver simulering med en stokastisk modell gjennomløper én av virkelighetens mange variasjonsmuligheter, og det er derfor nødvendig å gjennomføre gjentatte simuleringer for å kunne beregne gjennomsnittresultater. Denne type modeller kalles også *ikke-deterministiske* og gjentatte simuleringer med slike modeller kalles *Monte Carlo simulering*.

4.3 Operasjonsmodus

Operasjonsmodus beskriver hvordan brukerne opererer selve modellen. Med hensyn til operasjonsmodus kan modellene deles inn i:

- åpne modeller
- lukkede modeller

En *åpen modell* styres manuelt under simuleringen ved hjelp av kommandoer, menyer eller lignende. Dette betyr at simuleringen påvirkes av de beslutninger som brukeren tar underveis.

Bakgrunnen for åpne modeller er at beslutningstagere vanligvis kontrollerer viktige deler av et simulert system med komplekse beslutningsregler som er vanskelige å modellere fullgodt. Brukerinteraksjon gjør det mulig å inkludere beslutninger manuelt i simuleringen, og følgelig kan systemet likevel modelleres forholdsvis nøyaktig. Brukerinteraksjonen arter seg på to måter – noen modeller krever brukerkontroll (*interaktive* modeller), mens andre kan kontrolleres av brukeren (*avbrytbare* modeller).

I en *lukket modell* styres ikke modellen under selve simuleringen, og all beslutningstaking skjer automatisk iht forhåndsspesifiserte data. Når simuleringen er startet, kan brukeren således bare avvente det endelige resultatet.

Den primære innvendingen mot lukkede modeller er vanskene med å preprogrammere respons tilpasset alle mulige beslutningssituasjoner. Når brukerne følger simuleringen, oppdager de derfor kanskje situasjoner eller kriser som de i virkeligheten ville ha håndtert på en annen måte enn det modellen gjør. Imidlertid kan modellen likevel være en tilstrekkelig god beskrivelse av virkeligheten. Den fremste fordelen med lukkede modeller er at modellene kan kjøres forholdsvis raskt og enkelt. Dette er ofte avgjørende for å kunne gjennomføre det store antall simuleringer som er nødvendig for å finne gode løsninger på komplekse problemer innenfor begrensede tids- og ressursrammer.

4.4 Oppsummering

I dette kapitlet er simuleringsmodellene karakterisert langs tre dimensjoner:

- oppløsning (finkornet, grovkornet eller aggregert)
- håndtering av usikkerhet (deterministisk eller stokastisk)
- operasjonsmodus (åpen eller lukket)

Ofte er simuleringsmodellene åpne og deterministiske eller lukkede og stokastiske, mens oppløsningen varierer. Det kan ikke gis noe svar på hvilken type modell som generelt vil være å foretrekke uten å kjenne arbeidets formål og betingelser nærmere.

5 ILLUSTRASJON AV SIMULERING

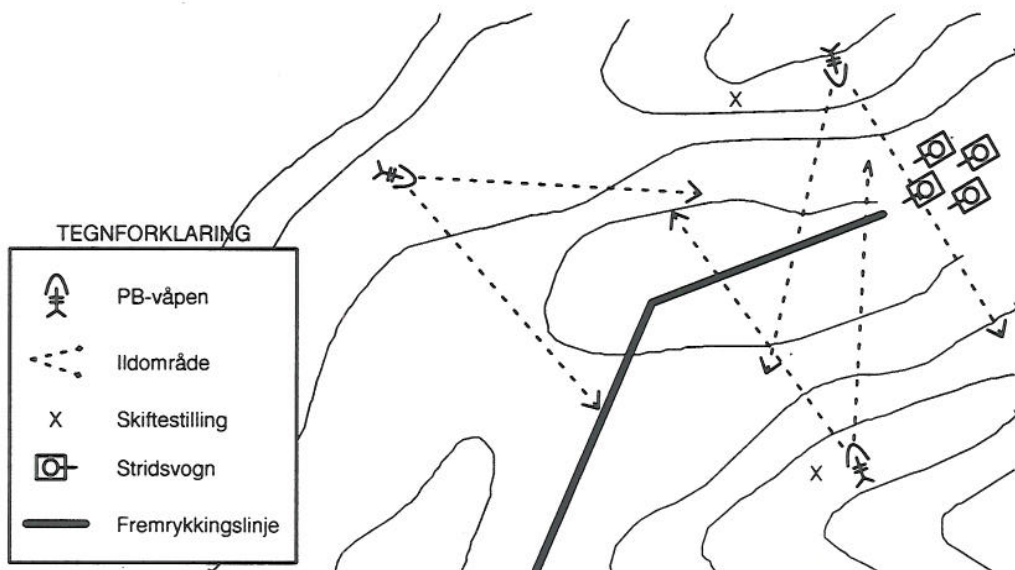
Dette kapitlet søker å illustrere noen av idéene og begrepene fra de foregående kapitler gjennom et lite eksempel. Systemet som skal simuleres, er avgrenset til et antall panserbekjempelsesvåpen (PB-våpen) i duell med stridsvogner. Systemet er representert med en modell som er diskret hendelsesstyrt (jfr kapittel 2.4), stokastisk (jfr kapittel 4.2) og lukket (jfr kapittel 4.3). Modellen er enkel og kan betraktes som en prototyp for videre utvikling.

Først gis det en kortfattet beskrivelse av modellen, deretter drøftes innsamling av inngangsdata, og en enkel simulering vises. Til slutt omtales eksempler på hvordan en slik modell kan nyttes.

5.1 Beskrivelse av modellen

Hvert PB-våpen besetter en stridsstilling med gitt posisjon, ildområde og dekningsgrad. Ved målsøking vil det ta en viss tid før eventuelle mål blir oppdaget, identifisert og våpnet er klart til første skudd. Missilet må følges til det når målet. Det går også en viss tid til observasjon av skuddets virkning og lading før et eventuelt neste skudd. Hvert skudd har en gitt sannsynlighet for å slå ut en gitt type mål. På grunn av PB-våpnets sårbarhet skiftes standplass etter et visst antall skudd. Under forflytning forutsettes våpnet å være usårbart.

Stridsvognene rykker fram med gitt fart langs gitte fremrykkingslinjer fra en gitt utgangsposisjon. Stridsvognene vil med en viss sannsynlighet oppdage PB-våpnene ved deres ildsignatur. Det går med en viss tid til lokalisering og retting av en stridsvogns våpen før første skudd, deretter skytes en serie skudd med jevne tidsintervall. Hvert skudd har en viss sannsynlighet for å slå ut PB-våpnet avhengig av avstand og PB-våpnets dekning: Sannsynligheten er konstant ut til 1000 m og avtar så med kvadratet av avstanden. Denne sannsynligheten reduseres så iht PB-våpnets dekningsgrad, eksempelvis vil en dekningsgrad på 0,3 redusere sannsynligheten med 30 %. Prosjektilenes flytid ignoreres. Situasjonen er illustrert i figur 5.1.



Figur 5.1 Beskrivelse av en enkel panserbekjempelsesmodell

Modellen er hendelsesstyrt, og aktuelle hendelser er:

- stridsvogn inn i ildområdet til et PB-våpen
- stridsvogn ut av ildområdet til et PB-våpen
- PB-våpen klar i stridsstilling
- skudd fra PB-våpen
- PB-missil når mål
- skudd fra stridsvogn

Systemets miljø er bl a terreng, værforhold, lysforhold, treningsnivå og nedholdene virkning av krumbanestøtte. Disse forholdene kan til en viss grad simuleres ved justering av parametre som skuddtakt, treffsannsynligheter og rekkevidde. Minefelt og andre hindringer for stridsvognene kan til en viss grad simuleres ved å redusere vognenes hastighet.

Modellens komponenter, parametre og variable er vist i tabell 5.1.

KOMPONENT	PARAMETRE	VARIABLE
Stridsstilling	<ul style="list-style-type: none"> – posisjon – ildområde – dekningsgrad (for PB-våpen i stillingen) 	<ul style="list-style-type: none"> – mål i ildområdet – besatt/ikke besatt av PB-våpen
PB-våpen	<ul style="list-style-type: none"> – opprinnelig ammunisjonsbeholdning – rekkevidde – missilhastighet – tid til første skudd på nytt mål – tid til påfølgende skudd – maksimalt antall skudd på samme mål – antall skudd før skifte av stilling – tid til skifte av stilling – sannsynlighet for å slå ut en stridsvogn med ett skudd 	<ul style="list-style-type: none"> – stridsstilling – ammunisjonsbeholdning – antall skudd fra samme stilling – antall skudd på samme mål – søker nytt mål/beskyter mål – avstand til målet – missilets flytid
Stridsvogn	<ul style="list-style-type: none"> – utgangsposisjon – hastighet – fremrykkingsvei – opprinnelig ammunisjonsbeholdning – sannsynlighet for å lokalisere et PB-våpen – tid til første skudd på nytt mål – tid til påfølgende skudd – antall skudd i skuddserien – sannsynlighet for å slå ut et udekket PB-våpen med ett skudd på 1000 m 	<ul style="list-style-type: none"> – ammunisjonsbeholdning – avstand til mål

Tabell 5.1 Modellens komponenter, parametre og variable

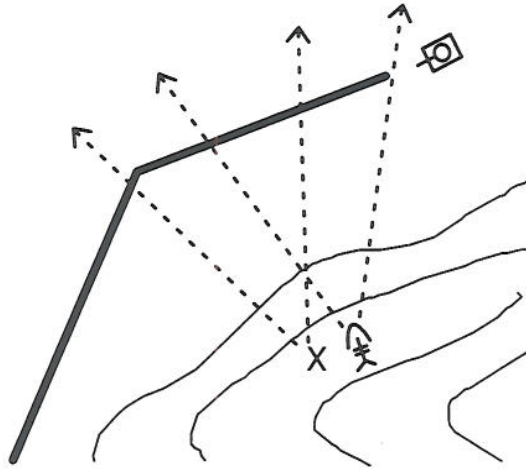
5.2 Datainnsamling

Inngangsdata til modellen, dvs de verdiene som settes på parametrene i tabell 5.1, er dels våpentekniske, dels stridstekniske og dels en blanding av disse.

Våpentekniske data

For tilgjengelige våpentyper kan en skaffe forsøks- og erfaringsdata fra produsenter og brukere av systemet. I tillegg gjøres ofte feltforsøk under varierende forhold for de mest kritiske data. Det gjøres en teknologisk gjennomgang av våpnet, bl a for å ha et bedre grunnlag for å ekstrapolere eksisterende data til forhold som det ikke foreligger data for.

Det kan imidlertid være flere grunner til at våpen ikke kan utprøves i felt. En grunn kan være at våpnet – og en potensiell fiendes våpen – ikke er ferdig utviklet. I slike tilfeller må data skaffes



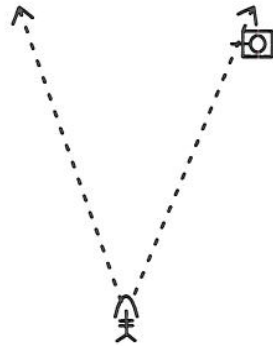
Figur 5.3 Eksempel på simulering av en enkel stridssituasjon

KOMPONENT	PARAMETER	VERDI
Ildstilling	<ul style="list-style-type: none"> - posisjon - ildområde - dekningsgrad ildstilling 1 - dekningsgrad ildstilling 2 	<ul style="list-style-type: none"> - se figur 5.3 - se figur 5.3 - 0,3 - 0,2
PB-våpen	<ul style="list-style-type: none"> - opprinnelig ammunisjonsbeholdning - rekkevidde - missilhastighet - tid til første skudd på nytt mål - tid til påfølgende skudd - maksimalt antall skudd på samme mål - antall skudd før skifte av stilling - tid til skifte av stilling - sannsynlighet for å slå ut en stridsvogn med ett skudd 	<ul style="list-style-type: none"> - 6 - 2000 m - 200 m/s - 12 s - 40 s - 2 - 3 - 100 s - 0,42
Stridsvognen	<ul style="list-style-type: none"> - utgangsposisjon - hastighet - fremrykkingsvei - opprinnelig ammunisjonsbeholdning - sannsynlighet for å lokalisere et PB-våpen - tid til første skudd på nytt mål - tid til påfølgende skudd - antall skudd i skuddserien - sannsynlighet for å slå ut udekket PB-våpen med ett skudd på 1000m 	<ul style="list-style-type: none"> - se figur 5.3 - 4 m/s - se figur 5.3 - 22 - 0,7 - 5 s - 15 s - 3 - 0,4

Tabell 5.2 Komponenter, parametre og eksempel på parameterverdier

Systemet kan bare endre tilstand som følge av en hendelse. Alle hendelser skjer på bestemte tidspunkt. For å holde styr på hendelsene, settes de inn i en liste som er ordnet etter økende tid (*hendelsesliste*). En hendelse kan lede til at nye hendelser oppstår eller at fremtidige hendelser blir uaktuelle. Før simuleringen starter etableres en hendelsesliste med minst én hendelse. I dette eksemplet vet man ut fra figur 5.3 og stridsvognens hastighet, når stridsvognen går inn og ut av de to ildområdene.

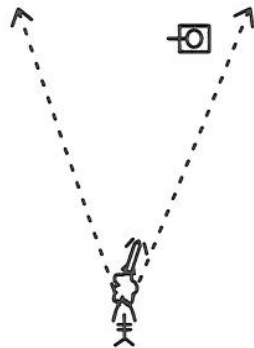
Figurene 5.4 - 5.10 viser sekvensielt hvordan en simulering av engasjementene mellom stridsvognen og PB-våpnet kan tenkes å forløpe.



TID	HENDELSE
0	STRV 1 inn i ildområde 1
95	STRV 1 inn i ildområde 2
230	STRV 1 ut av ildområde 1
320	STRV 1 ut av ildområde 2

Figur 5.4 Tid = 0: STRV 1 inn i ildområde 1

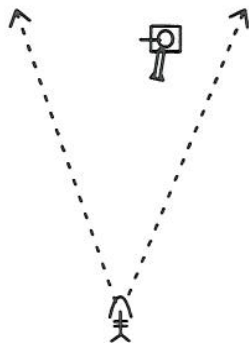
Ved tidspunkt 0 observeres stridsvognen av PB 1. I følge tabell 5.2 vil første skudd bli fyrt om 12 sekunder. En ny hendelse introduseres derfor i hendelseslisten slik som vist på figur 5.5.



TID	HENDELSE
12	PB 1 skyter mot STRV 1
95	STRV 1 inn i ildområde 2
230	STRV 1 ut av ildområde 1
320	STRV 1 ut av ildområde 2

Figur 5.5 Tid = 12: PB 1 skyter mot STRV 1

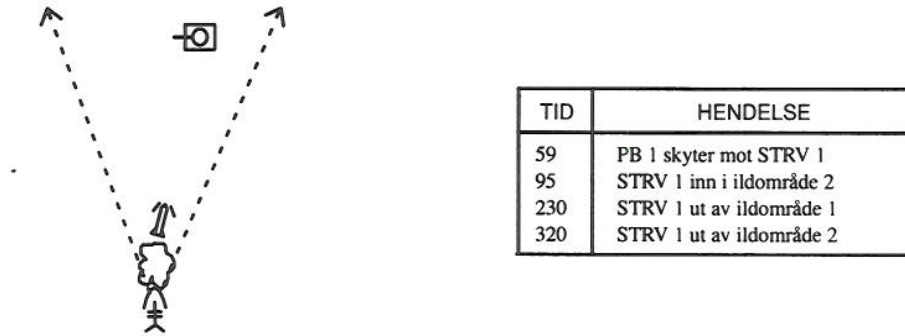
Ved tidspunkt 12 må det tas stilling til om PB-våpnets ildsignatur oppdages av stridsvognen. Det første tilfeldige tallet mellom 0 og 1,0 trekkes. Gitt at det trukne tallet er 0,85 (jfr listen på side 22), dvs at tallet er større enn 0,7 (jfr tabell 5.2), oppdages ildsignaturen ikke. Modellen beregner nå missilets flytid og bestemmer når neste hendelse inntreffer, dvs når missilet når frem til stridsvognen.



TID	HENDELSE
19	missil når STRV 1
95	STRV 1 inn i ildområde 2
230	STRV 1 ut av ildområde 1
320	STRV 1 ut av ildområde 2

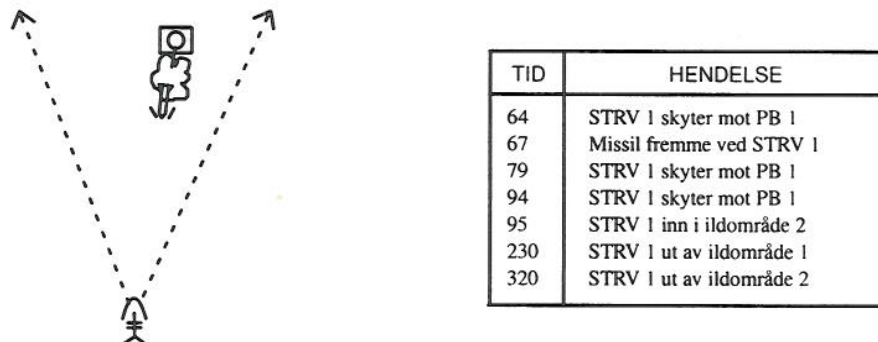
Figur 5.6 Tid = 19: Missil når STRV 1

Figur 5.6 viser situasjonen ved tidspunkt 19, dvs når missilet er fremme ved stridsvognen. Ved å trekke et nytt tilfeldig tall mellom 0 og 1,0 bestemmes om stridsvognen slås ut. Det trukne tallet viser seg å være 0,70, dvs større enn 0,42 (jfr tabell 5.2), og dermed blir stridsvognen ikke slått ut av missilet denne gang. Tidspunktet for neste skudd fra PB-våpnet bestemmes. I følge tabell 5.2 er reaksjonstiden for påfølgende skudd 40 s hvilket betyr at firing finner sted ved tidspunkt 59. Situasjonen ved dette tidspunktet er vist i figur 5.7.



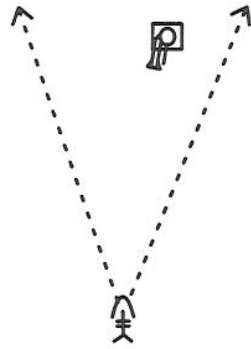
Figur 5.7 Tid = 59: PB 1 skyter mot STRV 1

På nytt må det bestemmes om ildsignaturen fra PB-våpnet oppdages av stridsvognen. Trekning av tilfeldig tall gir verdien 0,23 hvilket er mindre enn 0,7 (jfr tabell 5.2), som igjen betyr at PB-våpnet oppdages. Tidspunktene for hvert av de tre skuddene fra stridsvognen og PB-missilets flytid beregnes på grunnlag av verdiene i tabell 5.2. Fire nye hendelser introduseres dermed i hendelseslisten slik som vist på figur 5.8.



Figur 5.8 Tid = 64: STRV 1 skyter mot PB 1

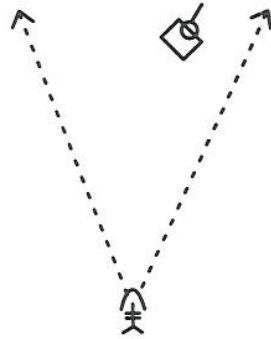
Siden stridsvognens kanonprosjektil har en høy hastighet, blir flukthastigheten relativt liten. Dermed kan det tas stilling til om PB-våpnet slås ut ved tidspunkt 64. Neste tilfeldige tall er 0,32. Siden ødeleggelsessannsynligheten avtar med kvadratet av avstanden til målet, finnes at 0,32 er større enn $0,4 \cdot (1-0,3) \cdot (1000/1400)^2$, og dermed kan det fastslås at PB-våpnet ikke blir slått ut. I følge hendelseslisten på figur 5.8 inntreffer neste hendelse ved tidspunkt 67 (missilet fremme ved stridsvognen). Situasjonen ved tidspunkt 67 er illustrert på figur 5.9.



TID	HENDELSE
67	Missil fremme ved STRV 1
79	STRV 1 skyter mot PB 1
94	STRV 1 skyter mot PB 1
95	STRV 1 inn i ildområde 2
230	STRV 1 ut av ildområde 1
320	STRV 1 ut av ildområde 2

Figur 5.9 Tid = 67: Missil fremme ved STRV 1

Ved tidspunkt 67 må det tas stilling til om stridsvognen slås ut. Trekning av tilfeldig tall gir verdien 0,21 som er mindre enn 0,42 (jfr tabell 5.2), og dermed er det avgjort at stridsvognen er ødelagt av PB-missilet. Alle videre hendelser hvor stridsvognen inngår kanselleres, og stridsvognen trekkes også ut av lister over aktuelle mål etc. Figur 5.10 viser at simuleringen er over.



TID	HENDELSE
—	—

Figur 5.10 Hendelseslisten er tom

Resultatene er som følger:

- PB 1 overlever
- STRV 1 er slått ut av PB 1 etter 67 s
- PB 1 har skutt to skudd
- STRV 1 har skutt ett skudd

En simulering gir – hvis man har tro på modellen – ett mulig resultat av denne type strid. Til spillformål kan ett resultat være tilstrekkelig, men til analyseformål kreves resultater fra mange simuleringer for å bestemme middelveier og resultatenes spredning. Antall simuleringer er som regel et kompromiss mellom høy statistisk signifikans og snevre tids- og kostnadsrammer. Siden tallet på simuleringer vanligvis er betydelig, ville en slik modell i virkeligheten ha blitt beskrevet vha et dataprogram og kjørt på en datamaskin.

5.4 Bruk av modellen

Modellen representerer bare et illustrasjonsverktøy uten noen praktisk betydning. En mer raffinert modell av tilsvarende type kan imidlertid anvendes til ulike praktiske formål:

- 1) Modellen kan støtte krigsspill gjennom troverdige bedømmelser av stridsforløp.
- 2) Modellen kan inngå i et beslutningsstøttesystem, som f.eks. kan gi råd om hvor store avdelinger som kreves for, med en viss sikkerhet, å oppnå et ønsket resultat i en gitt situasjon.
- 3) Modellen kan tjene som et opplæringsverktøy der offiserer gis anledning til å utvikle sin forståelse for sammenhengene mellom stridsteknikk, taktikk, teknologi og andre kvantitative forhold.
- 4) Modellen kan tjene som analyseverktøy. Selv om simulering ikke kan sikre optimale løsninger, kan man finne det beste av en rekke alternativer, vise hvorfor det er best og hvor mye bedre det er. (Alternativene representeres gjennom ulike konstellasjoner av parameterverdier.) Som eksempler på analyseoppgaver kan nevnes:
 - vurder effektiviteten av ulike typer våpen
 - vurder hvilke krav som bør stilles til et slikt våpen mht observasjonsutstyr, siktemidler, ammunisjon osv. (Høye krav fører ofte til høye kostnader, og simulering over en rekke aktuelle scenarier kan bidra til å vise de ulike kravenes viktighet.)
 - utform og velg mellom ulike prosedyrer for bruk av våpentypen

6 EKSEMPLER PÅ SIMULERINGSMODELLER VED FFI

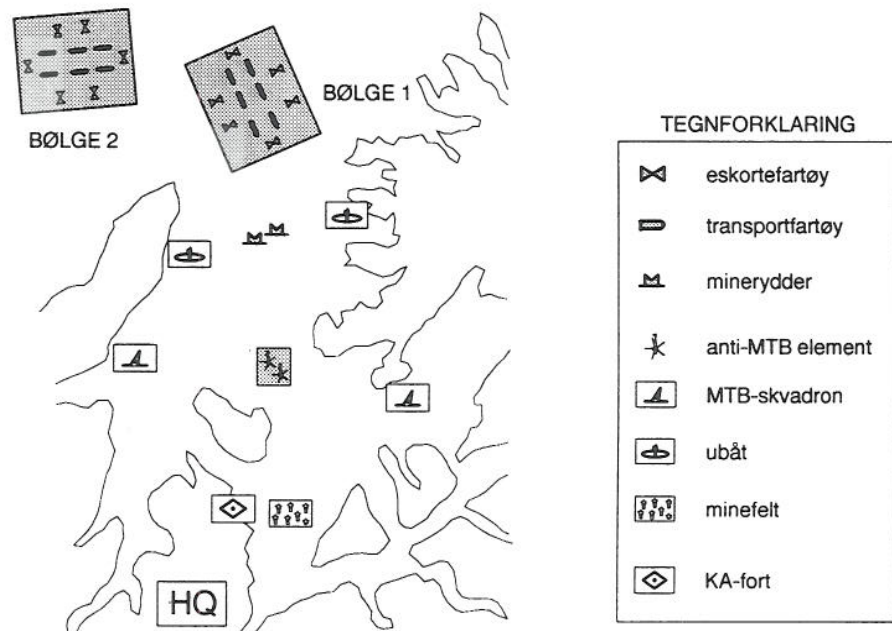
I dette kapitlet gis det en kort oversikt over de simuleringmodeller som er benyttet – og til dels også utviklet – ved FFI Avdeling for systemanalyse (FFISYS) i de senere år. For en mer fullstendig beskrivelse av modellene henvises det til refererte kilder.

6.1 Sjøinvasjonsmodellen

Under Sjøforsvarsanalysen ble det utviklet en lukket, stokastisk modell for simulering av anti-sjøinvasjonsoperasjoner. Modellen integrerer effekten av de enkelte styrkekomponentene i en operativ, dynamisk sammenheng under hensyn til komponentenes antall og type, strategisk/taktisk overvåking og evne til å utnytte informasjon for kommando og kontroll. Simuleringsmodellen omfatter delmodeller for beregning av stridsutfall i ulike stridssituasjoner for alle Sjøforsvarets anti-invasjonsenheter: Missiltorpedobåter (MTBer), undervannsbåter (UVBer), kystartilleri (KA) og miner. Utfallet av en gitt stridssituasjon i form av styrkekomponentenes tapspåførende kapasitet og egen overlevelsessevne, beregnes vha den delmodell som er relevant for stridsprosessen. Egne fly i luft-til-overflate rollen inngår ikke i modellen.

Modellen inneholder også en hovedkvarterfunksjon (HQ) som på basis av informasjon om invasionsstyrkens bevegelser, redeploys de mobile styrkekomponentene etter definerte beslutningsregler. Modellen er illustrert i figur 6.1.

Simuleringsmodellen er programmert i Simula og ble i hovedsak utviklet i løpet av 1989. Kildekoden er på ca 10 000 linjer. For videre detaljer henvises det til referansene [6] og [7].

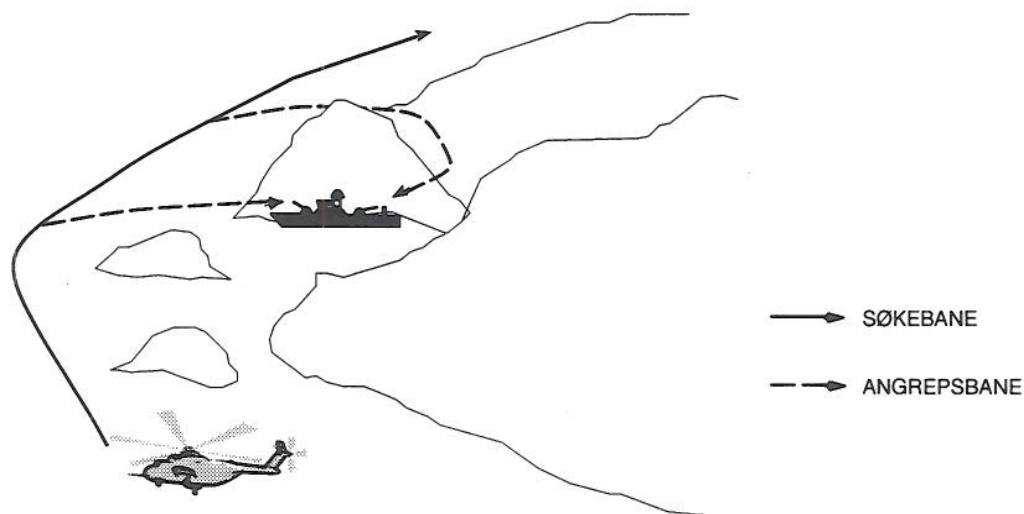


Figur 6.1 Hovedelementene i sjøinvasjonsmodellen

6.2 Simuleringsmodell for duell mellom helikopter og MTB i lurkeposisjon

Under delprosjektet "Signaturreduksjon av ny MTB" ved FFI Avdeling for våpen og materiell (FFIVM) ble det utviklet en simuleringsmodell for duell mellom egne MTBer i lurkeposisjon og fiendtlige helikoptre på oppklaring. Modellen er stokastisk og lukket. Hensikten med modellen er å vurdere hvilken betydning kamuflasje har for overlevelsessevnen til MTBene, og kosteffektiviteten til ulike kamuflasjetiltak. Modellen tar derfor utgangspunkt i en situasjon der et antall MTBer er plassert i *faste* posisjoner i et mulig invasionsområde. Helikoptrene i simuleringen flyr langs bestemte søkebaner som er spesifisert ut fra terrenget i det aktuelle området, se figur 6.2. Helikoptrene kan gjennomføre missilangrep og kanonangrep, mens MTBene kan forsvare seg med ulike typer luftvern. Ut fra den gitte taktiske situasjonen, hvor posisjonene til både MTBene og de fiendtlige helikoptrene inngår, beregner duellmodellen MTBenes overlevelsessannsynlighet.

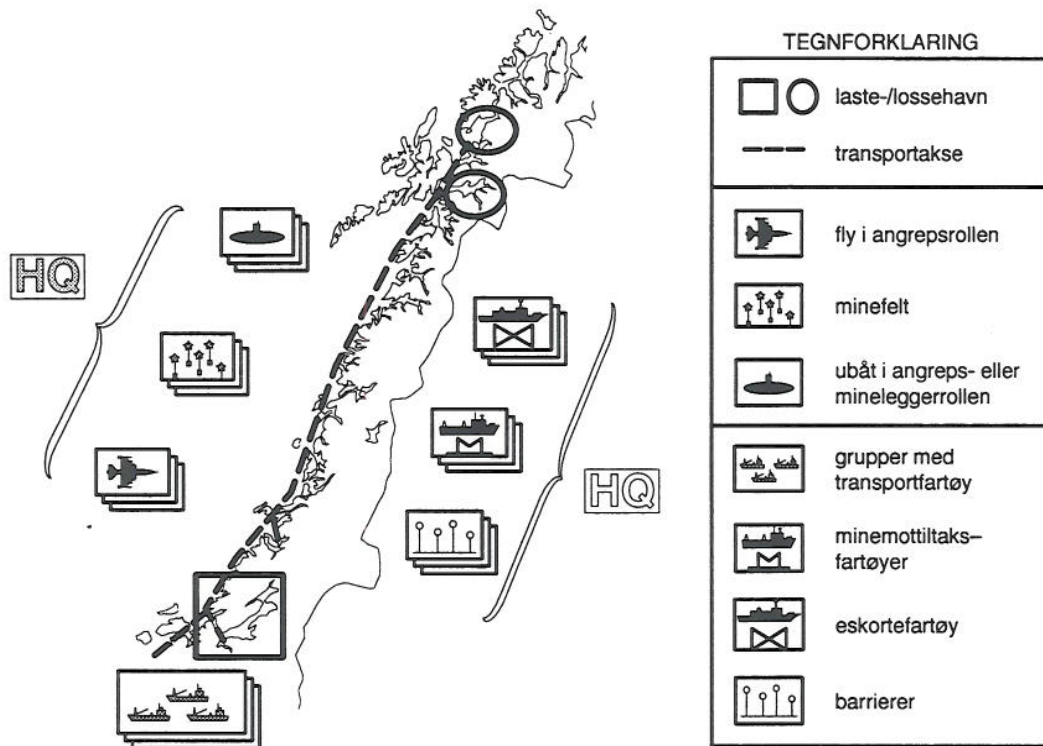
Kjernen i modellen er programmert i Simula, mens brukergrensesnittet er skrevet i C. Modellen er utviklet i 1992, se referansene [8] og [9], og bygger på en modell utviklet på FFISYS (jfr [10] og [11]).



Figur 6.2 Eksempel på taktisk situasjon i duell mellom MTB og helikopter

6.3 Sjøstridsmodellen SLOC

For å finne kosteffektive styrkestrukturer for sikring av sjøtransporter ble det under Sjøforsvarsanalysen utviklet en simuleringsmodell, SLOC, som integrerer ulike trussel- og styrkekomponenter i en slik operasjon. Modellen er lukket og stokastisk, og simulerer sjøtransporter fra angitte lastehavner til ønskede lossehavner i et gitt tidsrom. Langs transportaksen kan det operere tre typer trusselkomponenter: Ubåter, miner og fly. For å sikre sjøtransportene er det modellert anti-ubåt enheter og minemottiltaksenheter på forsvarerens side, mens effekten av anti-fly enheter antas å være evaluert utenfor modellen. Hovedelementene i SLOC er vist i figur 6.3.



Figur 6.3 Hovedelementer i simuleringsmodellen SLOC

Modellen beskriver også hovedkvarterfunksjoner. Angriperens hovedkvarter (HQ) styrer trusselementene, mens forsvarerens hovedkvarter styrer transportgruppene og sikrings-elementene. Begge hovedkvarterene handler på basis av informasjon fra egne enheter og informasjon om selve transportaksen.

Simuleringsmodellen SLOC er skrevet i programmeringsspråket Simula og var ferdig utviklet i 1992. Programmet er på ca 11 500 linjer. For videre detaljer henvises det til referansene [12] og [13]. Et system for grafisk fremvisning av resultatene er under utarbeidelse og foreligger som prototyp [14].

6.4 Sonarprediksjonsmodellen Generic Sonar Model

Generic Sonar Model (GSM) er utviklet av Naval Underwater Warfare Center, USA i 1985. Modellen er lukket og deterministisk. Primært beregner modellen sannsynligheten for å oppdage en ubåt ved hjelp av sonar som funksjon av avstand. Denne sannsynligheten blir bla beregnet ut fra faktorer som sonarenes egenstråling, gjenklang fra bunn og havoverflate, volumgjenklang, sonarens trykkfelt i området, signaloverskudd osv. Disse ulike parametrene er resultater fra beregninger i ulike programmoduler og kan gis eksplisitt som delresultater. Beregningene i modellen blir gjort på grunnlag av spesifiserte inndata angående sonar- og mål-karakteristikk, bunntype, bølgehøyde og vind, lydshastighetsprofil og støy. Modellen kan brukes til beregninger for både aktiv og passiv sonar.

GSM er skrevet i programmeringsspråket FORTRAN. Modellen er brukt på FFI Avdeling for undervannsforsvar (FFIU). Referanse [15] beskriver modellen nærmere.

6.5 Sårbarhetsberegningsmodellen General Vulnerability Assessment Model

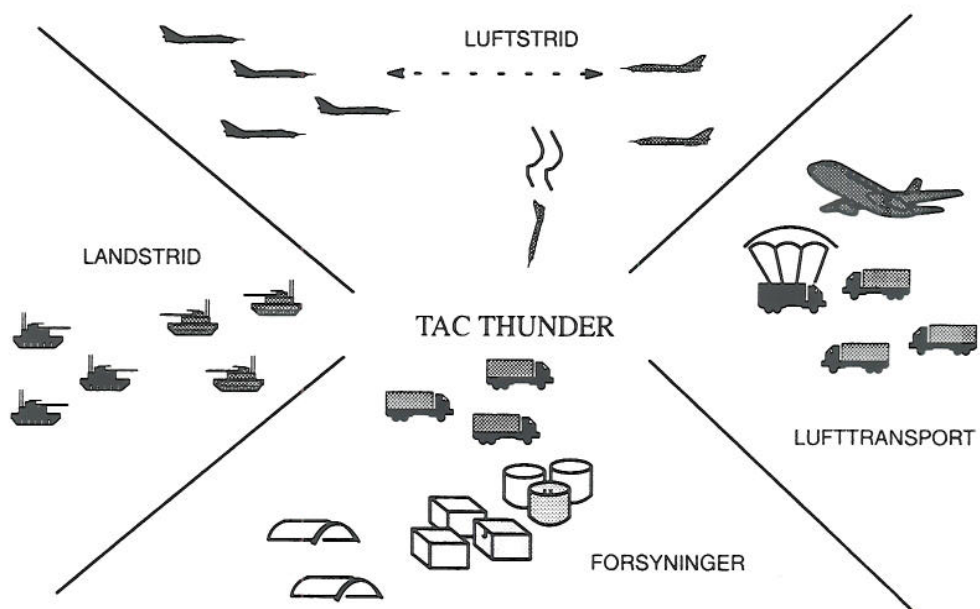
På begynnelsen av åtti-tallet utviklet det kanadiske Defence Research Establishment Valcartier (DREV) en modell, General Vulnerability Assessment Model (GVAM), for å gjøre sårbarhetsberegninger. Modellen beregner virkningene av treff fra overflate- eller luftleverte våpen i et mål. GVAM består av tre hoveddeler; Sea VAM (SVAM), Vehicle VAM (VVAM) og Air VAM (AVAM) som ser på sårbarhet for henholdsvis fartøyer, kjøretøyer og luftbårne plattformer.

FFIVM disponerer delmodellen SVAM. I SVAM modelleres målfartøy vha rektangulære blokker som kan fylles med utstyr, komponenter og personell. Videre har SVAM moduler som beregner skade som følge av trykket etter en intern eller ekstern detonasjon og skade som følge av splinter fra både eksterne og interne detonasjoner.

GVAM er en lukket, deterministisk modell som er programmert i FORTRAN, jfr [16] og [17].

6.6 Luftstridsmodellen TAC THUNDER

Simuleringsmodellen TAC THUNDER er utviklet av Det amerikanske flyvåpenet (USAF). Den er stokastisk, og kan kjøres i både åpen og lukket modus. Den kan simulere luftstrid over et stort område (teater) og et lengre tidsrom. Modellen inkluderer også logistikk og enkle modeller for landstrid, se figur 6.4. TAC THUNDER kan således simulere bl a følgende dimensjoner: Luft-til-luft, luft-til-bakke, bakke-til-luft, bakke-til-bakke, flystasjonsoperasjoner, logistikk, kommando, kontroll, samband og etterretning.



Figur 6.4 Hovedelementene i simuleringsmodellen TAC THUNDER

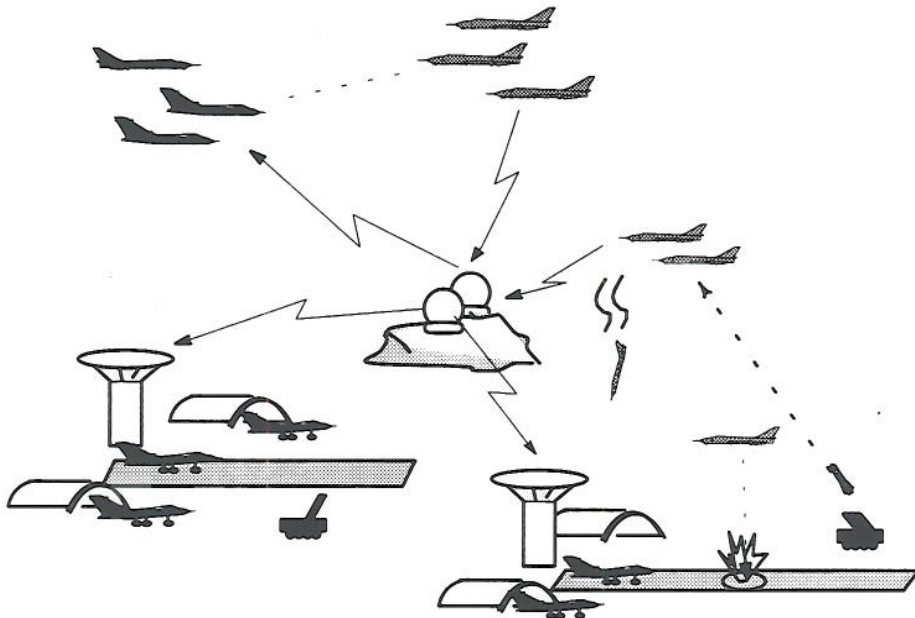
TAC THUNDER er skrevet i programmeringsspråket SIMSCRIPT II.5. Den versjonen som FFISYS disponerer ble ferdig utviklet i 1989, jfr [18]. Modellen er senere videreutviklet.

6.7 Luftstridsmodellen GILMINI

Under Luftforsvarsanalysen ble det på grunnlag av erfaringer med bruken av TAC THUNDER utviklet en forenklet modell – GILMINI – for simulering av luftstrid på kampanjenivå. GILMINI er en stokastisk og lukket modell. Modellen er asymmetrisk, dvs den ene siden angriper mål hos den andre siden, mens den andre siden forsøker å avskjære disse angrepene. Målene er typisk flystasjoner e l, forsvart av luftvern.

GILMINI inneholder egne algoritmer for å beregne skaden som påføres flystasjoner. Når det gjelder utfall av luft-til-luft og bakke-til-luft strid, baserer modellen seg på oppslag i tabeller som er utarbeidet på grunnlag av resultater fra andre modeller. Siden modellen er lukket, defineres oppdragene for angriperens og forsvarerens fly på forhånd for hele stridsforløpet. Modellen er illustrert i figur 6.5.

Modellen er skrevet i programmeringsspråket SIMSCRIPT II.5, og en versjon ble ferdig i 1991. Denne versjonen er på ca 3 500 linjer. GILMINI ble videre utviklet i 1993 under FFI/PRO-SJEKT 616-VM/139 "Passivt forsvar av flystasjoner". Modellen er nærmere beskrevet i [19].



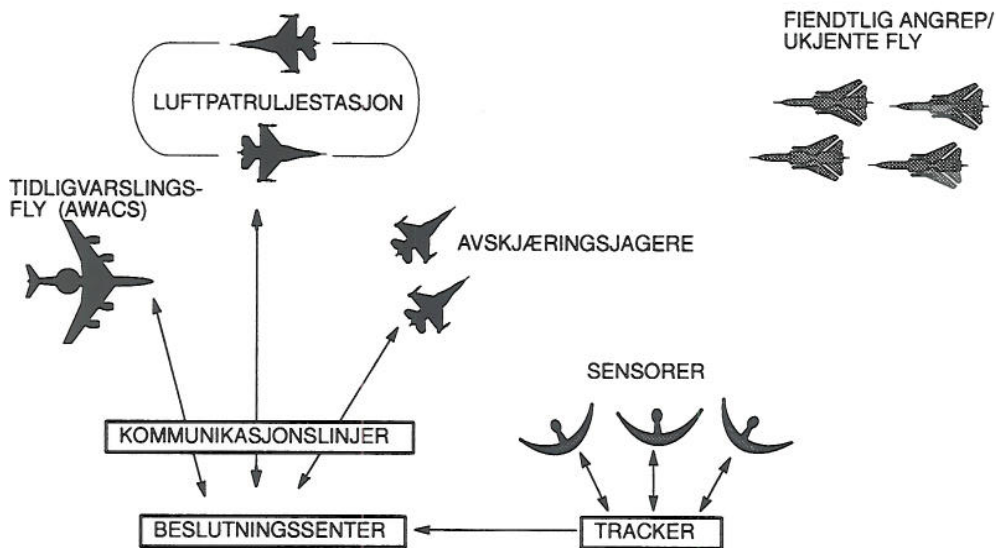
Figur 6.5 Hovedelementer i simuleringsmodellen GILMINI

6.8 Avskjæringsmodellen TORNADO

Avskjæringsmodellen (TORNADO) ble utviklet under Luftforsvarsanalysen for å kvantifisere ytelsen til luftkontrollsystemer. Modellen er stokastisk og lukket. Hver av de stridende partene

i modellen kan disponere et antall sensorer av forskjellige typer (f eks bakkebaserte/flybårne, radarer/ESM/PJL osv). På grunnlag av data fra sensorene bygger hver av sidene opp sine luftbilder, som brukes til å allokere avskjæringsjagere mot angripende fly, samt å lede disse frem til avskjæring. Radarsensorer kan degraderes ved at den andre siden anvender jamming mot, eller fysisk angriper sensorene. Siden modellen er lukket, må det på forhånd utarbeides en plan for gjennomføringen av alle tokt. Det er derfor ikke praktisk å gjennomføre simuleringer av lengre varighet enn én angrepsbølge (dvs et par timers strid). Modellen er illustrert i figur 6.6.

Avskjæringsmodellen er skrevet i programmeringsspråket C++. Versjon 2.1 ble gjort ferdig i oktober 1990 og er på ca 10 000 linjer, jfr [20]. Det eksisterer et grafisk animasjonsprogram til modellen. Modellen ble videreutviklet under FFI/PROSJEKT 604-E/161.3 "Fremtidig luftkontrollsystem i Norge NORWACCS".

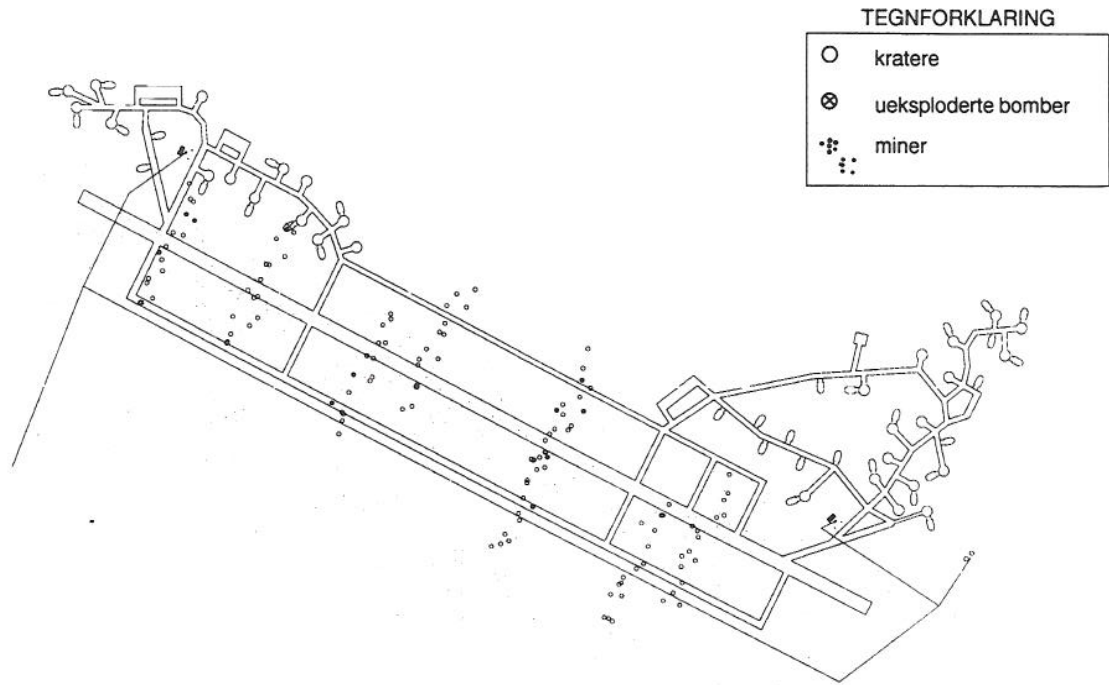


Figur 6.6 Hovedelementer i Avskjæringmodellen

6.9 Flystasjonsskademodellen

For å studere sammenhengen mellom flystasjoners reparasjonskapasitet og flystasjoners sårbarhet ble Flystasjonsskademodellen utviklet under FFI/PROSJEKT 616-VM/139 "Passivt forsvar av flystasjoner". Modellen er stokastisk og lukket. Etter et flyangrep kan flystasjonen være påført skader i form av kratere på rullebaner og taxeveier, ødelagte drivstofftanker og avkuttete sambandskabler, se figur 6.7. I tillegg kan miner gjøre områdene utilgjengelige for stasjonens personell og kjøretøyer. Modellen evaluerer deretter om forsvareren lykkes i å åpne flystasjonen for egne flyoperasjoner før neste angrep kommer. Dette vil variere ut fra angriperens antall fly, våpnenes innsiktning og nøyaktighet, og egen kapasitet på flystasjonen til å rekognosere skade, rydde miner, reparere kratere og annen skade.

Modellen ble ferdig utviklet i 1993. Den er skrevet i programmeringsspråket C++ og har et relativt godt grafisk grensesnitt. Modellen er på ca 6 000 linjer. Referanse [21] beskriver modellen nærmere.



Figur 6.7 Eksempel på situasjonsbilde fra Flystasjonsskademodellen etter et simulert angrep

6.10 Luftvernmodellen LVMOD

Siden luftvernmodellen LVMOD ble påbegynt i 1978, er den benyttet i omkring 10 prosjekter og oppdrag. Den er derfor en av FFIs eldste og mest benyttede modeller som har gjennomgått en rekke endringer og utvidelser.

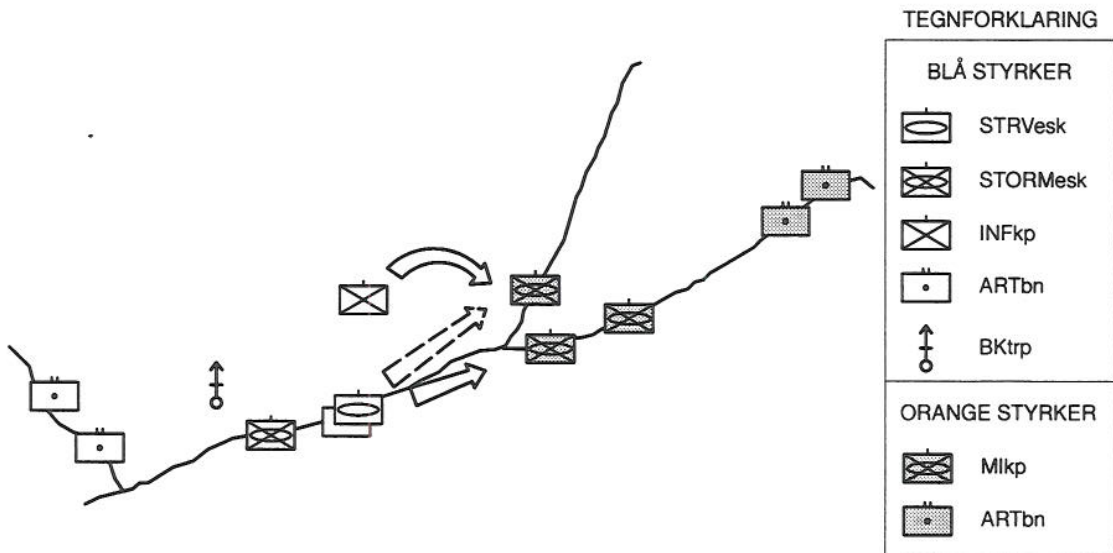
Luftvernmodellen LVMOD 2.0 beregner antall nedskutte fly/helikoptre under et luftangrep mot et vilkårlig objekt som er forsvart med kortholdsluftvern (SHORAD). Modellen bygger på en tre-dimensjonal beskrivelse av terrenget rundt målet. Luftangrepene spesifiseres med bla antall fly/helikoptre, flyenes/helikoptrenes formasjon og bevegelse. LVMOD 2.0 er stokastisk og lukket.

For å evaluere rakettluftvern med middels rekkevidde (MSAM) for flystasjonsforsvar ble det under FFI/PROSJEKT 2603–SYS/161 "Evaluering av rakettluftvern for Østlandet", utviklet en variant av Luftvernmodellen, kalt LVMOD 3.0. Modellen simulerer ett angrep på en flystasjon med utgangspunkt i definerte grupper av angrepsfly på trusselsiden og ildenheter på forsvarerens side. Radarer, som er en integrert del av hver ildenheter, fremskaffer måldata. Ildenheterne er sammenkoblet på en slik måte at de vet hva de andre enhetene gjør og kan utveksle måldata. Et jammende mål kan trianguleres, dvs at målposisjonen kan beregnes på grunnlag av retningsangivelser fra to radarer med ulike posisjoner. Modellen beregner missilenes effektivitetsmål og gir i tillegg en beskrivelse av hendelsesforløpet. Også LVMOD 3.0 er stokastisk og lukket.

LVMOD 3.0 er programmert i Simula og kildekoden er på ca 15 000 linjer. Modellen er utviklet i tidsrommet 1987 - 1988. For videre detaljer henvises det til referansene [22] og [23].

6.11 Landstridsmodellen DYNACOM

Under FFI/PROSJEKT 463 –SYS/161 "Brigade 90 – Sammensetning og strid" ble det utviklet en simuleringsmodell – DYNACOM – for landstrid på bataljonsnivå. Slik strid krever et samspill mellom bataljons underavdelinger, og spesielt viktig er bruk av forskjellige typer støttevåpen. Modellen simulerer enhetenes bevegelser, stridsutfallet i form av tap og den dynamiske virkningen av nedholdende ild. Modellen er deterministisk og kan kjøres i både åpen og lukket modus. DYNACOM er illustrert i figur 6.8.



Figur 6.8 Eksempel på stridssituasjon i simuleringsmodellen DYNACOM

DYNACOM er programmert i Simula og første versjon var ferdig i 1984, se referanse [24]. Kildekoden til første versjon av DYNACOM er på ca 1 500 linjer. Det finnes i tillegg en programmodul som muliggjør interaktiv kjøring. DYNACOM er senere videreutviklet og brukt i Hæranalysen.

6.12 Landstridsmodellen PABST

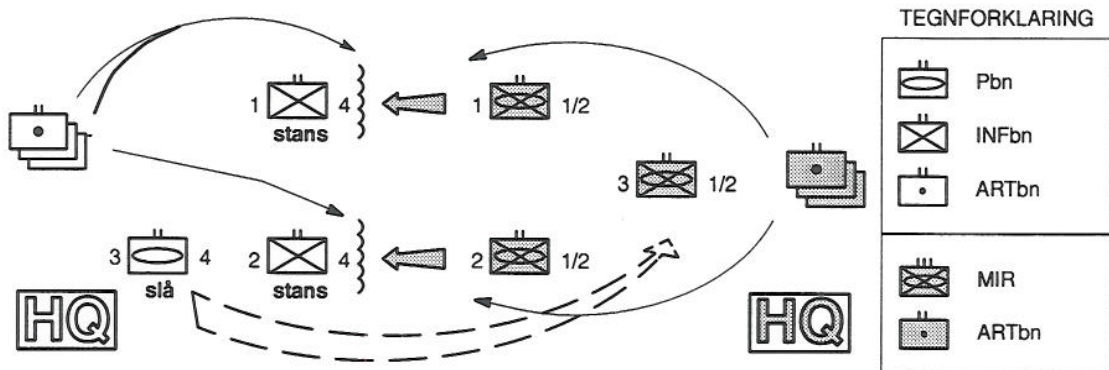
PABST (Das Panzerabwehr-Bewertungsinstrument) er en lukket, stokastisk modell for simulering av panserstrid og panserbekjempelse på taktisk nivå, dvs inntil bataljon/regiment. Modellen bygger på en detaljert representasjon av terrenget både mht høyde og vegetasjon. Likeledes beskrives våpensystemene og bruken av disse i stor detalj. PABST er utviklet ved IABG (Industrieanlagen - Betriebsgesellschaft mbH) over en 10-års periode.

Det finnes også et hjelpeverktøy for PABST som kalles INTAG (Interaktiver Arbeitsplatz für Gefechtssimulationen). Dette brukes bl a for å visualisere terrenget, beregne dekningsdiagrammer og visualisere et stridsforløp simulert med PABST.

PABST er programmert i FORTRAN og kildekode er på ca 20 000 linjer. For videre detaljer henvises det til referansene [25] og [26].

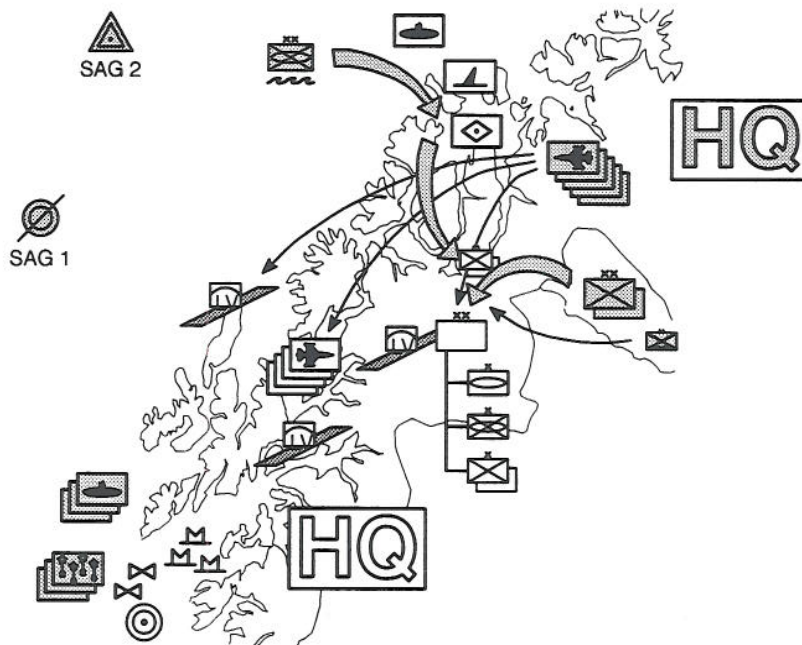
6.13 Landstridsmodellen DIVMOD

For å finne frem til kosteffektive styrkestrukturer innen Hæren, ble simuleringsmodellen DIVMOD utviklet under Hæranalysen. DIVMOD er en åpen deterministisk modell som simulerer operasjoner på brigade- og divisjonsnivå ut fra en beskrivelse av operasjonsområdet, partenes styrker og partenes disposisjoner, jfr figur 6.9. Modellens stridsevalueringer gjøres ved hjelp av oppslag i datatabeller som er utarbeidet fra resultater med simuleringsmodellen DYNACOM, jfr kapittel 6.11.



Figur 6.9 Eksempel på stridssituasjon i simuleringsmodellen DIVMOD (brigadeoperasjon)

Basert på resultater fra en rekke av de tidligere omtalte modellene for simulering av land-, luft- og sjøstrid, kan DIVMOD brukes for å analysere kosteffektiviteten til *totale* styrkestrukturer. Dette ble gjort under Forsvarsanalysen. Resultatene fra de andre modellene utgjør da input data til DIVMOD og legges inn manuelt. Under slike simuleringer kompliseres beslutningsproblematikken betydelig, jfr figur 6.10.



Figur 6.10 Eksempel på stridssituasjon i simuleringsmodellen DIVMOD (fellesoperasjoner)

DIVMOD er programmert i Simula og er på ca 20 000 linjer. Modellutviklingen ble avsluttet i 1991, se for øvrig referansene [27] og [28].

6.14 Landstridsmodellen TRIAMOS

Simuleringsmodellen TRIAMOS (Trend Indication and Analysis Model for Open Systems) kan i likhet med DIVMOD brukes for å evaluere effektiviteten av ulike styrkestrukturer og ulike disposisjoner på operativt nivå. TRIAMOS er imidlertid langt mindre spesialisert og langt mer fleksibel enn DIVMOD. Modellen kan derfor anvendes for analyser også på taktisk nivå.

Ytterligere to forskjeller mellom de to modellene kan fremheves: For det første er DIVMOD en åpen, deterministisk modell, mens TRIAMOS er lukket og deterministisk. Dette betyr at de til dels omfattende taktiske disposisjonene kan utformes underveis i en DIVMOD-kjøring, mens de må være utarbeidet på forhånd før en TRIAMOS-kjøring. For det andre er DIVMOD i stor grad avhengig av aggregerte resultater fra andre simuleringsmodeller, spesielt under simulering av fellesoperasjoner hvor både land-, luft- og sjøstrid inngår. Simuleringene i TRIAMOS baseres derimot på effektivitetstall fra våpen mot våpen.

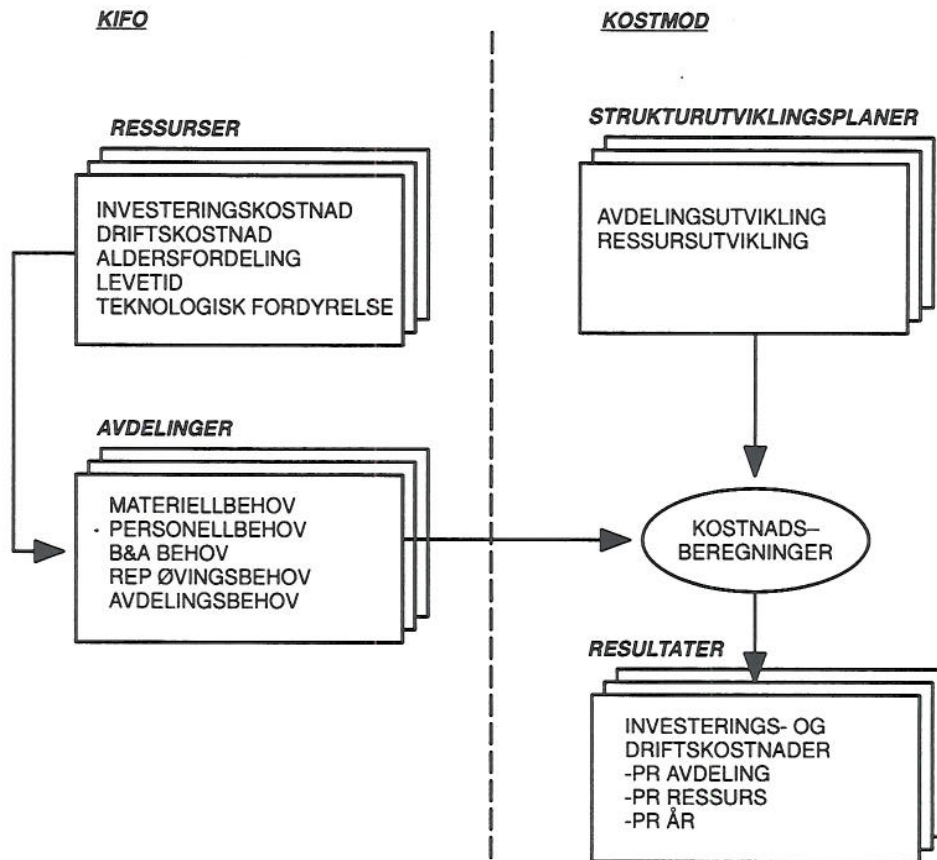
Modellen er utviklet av IABG i Tyskland. Den er skrevet i Turbo Pascal og er på anslagsvis 20-30 000 linjer. For videre detaljer henvises det til referanse [29].

6.15 Kostnadsberegningsmodellen KOSTMOD

KOSTMOD er en kostnadsberegningsmodell som beregner drifts- og investeringskostnader for forsvarsstrukturer over tid. Modellen ble utviklet ved FFISYS i perioden 1990-1991, og er en videreføring av BUDSJ som ble utviklet i 1975 og 1976. En videreutvikling av modellen (KOSTMOD versjon 2.0) ble ferdigstilt i 1993.

KOSTMOD var et viktig verktøy i FFI/PROSJEKT 601-SYS/161 "Forsvarsanalysen". Modellen brukes for tiden i forbindelse med pågående oppdrag og prosjekter på FFISYS og i tillegg av Forsvarets overkommando (FO) i forbindelse med langtidsplanlegging.

Input data til KOSTMOD er avdelinger, ressurser og strukturutviklingsplaner. De to førstnevnte utgjør en database og beskriver sammensetningen av materiell, personell og bygg og anlegg i ulike avdelinger. Databasen oppdateres årlig av forsvarsgrenene gjennom kostnadsinformasjonssystemet KIFO. På grunnlag av planleggers beskrivelse av hvilke avdelinger og ressurser som skal inngå i forsvarsstrukturen over en periode (strukturutviklingsplan), beregner modellen drifts- og investeringskostnadene for denne strukturen. Figur 6.11 viser modellens overordnede struktur og grensesnittet til KIFO.



Figur 6.11 Struktur for kostnadsberegningsmodellen KOSTMOD og grensesnittet til KIFO

Resultatene fra KOSTMOD kan tas ut i form av standardrapporter eller skreddersys ved hjelp av regnearkprogrammet EXCEL. Modellen kjøres på en PC-plattform og er basert på database-systemet dBASE IV. KOSTMOD er lukket og deterministisk. En nærmere beskrivelse av modellen finnes i referansene [30] og [31].

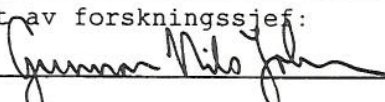
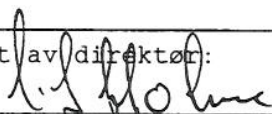
LITTERATURLISTE

- [1] Shannon R E: *Systems Simulation – the Art and Science*, Prentice-Hall, New Jersey, 1975
- [2] Bulgren W G: *Discrete System Simulation*, Prentice-Hall, New Jersey, 1982
- [3] Kreutzer W: *System Simulation: Programming Styles and Languages*, Addison-Wesley, 1986
- [4] Military Operations Research Society: *Military Modeling*, Hughes Jr W P (editor), Second Edition, 1989
- [5] Lysfjord S, Langlie E og Skjelland E: *Simulering og grensesnitt – En støtte til valg av programmeringsspråk, grafikkpakker og maskinvare*, FFI/NOTAT–94/02247
- [6] Amundsen E: *Sjøforsvarets fremtidige struktur. Sluttrapport fra sjøforsvarsanalysen*, FFI/RAPPORT–91/5023, konfidensielt
- [7] Langli E og Sandbakken J R: *Anti-sjøinvasjonsforsvar – modellering*, FFI/RAPPORT–91/5025, konfidensielt
- [8] Erlandsen E R og Skjervold J E: *Databasert simuleringsmodell for duell helikopter–MTB lurkeposisjon*, FFI/NOTAT–92/4008
- [9] Erlandsen E R: *Programbeskrivelse og listing for duell-simuleringsmodell*, FFI/NOTAT–92/4020
- [10] Amundsen E og Vik G: *Luftvern på TBK'er. Sluttrapport*, FFI/RAPPORT–83/5001, konfidensielt
- [11] Johnsen G N, Bremnes J, Dullum O, Hagenson N, Mølmann Ø og Stenersen K: *Evaluering av kortholds raketluftvern for marinen. En kosteffektivitetsanalyse av robot 70 Mk II, Stinger RMP og Mistral*, FFI/RAPPORT–90/5001, hemmelig
- [12] Otterlei J og Skjelland E: *Sikring av egne sjøtransporter – Modellering*, FFI/NOTAT–91/5011, begrenset
- [13] Otterlei J og Skjelland E: *Sikring av egne sjøtransporter – Brukerveiledning*, FFI/NOTAT–91/5012, begrenset
- [14] Ølberg E: *Grafisk grensesnitt til SLOC-modellen*, FFI/RAPPORT–94/01614, begrenset
- [15] Weinberg H: *Generic Sonar Model*, NUSC, TD 5971D, 1985
- [16] *Computer Simulation Models for Vulnerability Assessment of Land, Sea and Air Targets*, Weapon Effectiveness Group, DREV
- [17] Gass N et al: *A Simulation Model for the Assessment of Fragment Damage*, CDRV Rapport 4432/87

FFISYS
 VÅR REFERANSE:
 TIL SENTRALSTABEN
 GJENPART: FFI-BIBL (MED VEDLEGG)

OVERSENDELSE
 AV RAPPORTER

Dato: 22 juli 1994

Rapporttype (kryss av):			Rapport nr:	Referanse:	Rapportens dato:			
<input checked="" type="checkbox"/>	RAPP	<input type="checkbox"/>	NOTAT	<input type="checkbox"/>	RR	94/03698	FFISYS/651/410	22 juli 1994
Fylles bare ut når rapporten er beskyttelsesgradert			Beskyttelsesgrad:	Antall eks:	Sider:			
			UGRADERT	50	41			
Rapportens tittel:				Forfatter(e):				
INTRODUKSJON TIL SIMULERING				SKJELLAND Nils Espen, FEET Else Helene, FRIHAGEN Jon				
Godkjent av forskningssjef:				Godkjent av direktør:				
								

FORSLAG TIL EKSTERN FORDELING

FORDELT INTERNT

Antall	Eks nr	Til	Antall	Eks nr	Til
1		FO/O/v Maj M Ringheim	11		FFIBIBL
1		FO/HST/v Maj B G Øistad	1		FFIS/NH
1		FO/SST/v OK T Stubberud	1		FFIS/O Grøtta
1		FO/LST	1		FFISYS/RHS
1		FKS/v Oblt M H Ness	1		FFISYS/GNJ
1		HSTS	1		FFISYS/HaA
1		SSTS	1		FFISYS/OlB
1		LSTS	1		FFISYS/ThB
1		KNMT/v OK G F Sivertsen	1		FFISYS/NHa
1		LTS/v LT J O Akerjordet	1		FFISYS/IJ
			1		FFISYS/RFK
			1		FFISYS/TLa
			1		FFISYS/JHL
			1		FFISYS/OPM
			1		FFISYS/JMO
			1		FFISYS/BR
			1		FFISYS/KSo
			1		FFISYS/AsT
			1		FFISYS/JFr
			1		FFISYS/ElF
			1		FFISYS/ENS
			9		FFISYS AVDKTR

Til fordeling vedlegges:
 Antall: nr: