

FFI RAPPORT

SIMBA - BRUKERVEILEDNING

HALSØR Marius, EIDE Morten

FFI/RAPPORT-2003/01712

FFI-III/798/139

SIMBA - BRUKERVEILEDNING

HALSØR Marius, EIDE Morten

FFI/RAPPORT-2003/01712

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

FORSVARETS FORSKNINGSINSTITUTT (FFI)
Norwegian Defence Research Establishment

UNCLASSIFIED

P O BOX 25
 NO-2027 KJELLER, NORWAY

SECURITY CLASSIFICATION OF THIS PAGE
 (when data entered)

REPORT DOCUMENTATION PAGE

1) PUBL/REPORT NUMBER FFI/RAPPORT-2003/01712	2) SECURITY CLASSIFICATION UNCLASSIFIED	3) NUMBER OF PAGES 53
1a) PROJECT REFERENCE FFI-III/798/139	2a) DECLASSIFICATION/DOWNGRADING SCHEDULE -	
4) TITLE SIMBA – BRUKERVEILEDNING SIMBA – USER MANUAL		
5) NAMES OF AUTHOR(S) IN FULL (surname first) HALSØR Marius, EIDE Morten		
6) DISTRIBUTION STATEMENT Approved for public release. Distribution unlimited. (Offentlig tilgjengelig)		
7) INDEXING TERMS IN ENGLISH: IN NORWEGIAN:		
a) <u>User manual</u>	a) <u>Brukermanual</u>	
b) <u>Simulation model</u>	b) <u>Simuleringsmodell</u>	
c) <u>SIMBA</u>	c) <u>SIMBA</u>	
d) <u>Anti-tank warfare</u>	d) <u>Panserbekjempelse</u>	
e) _____	e) _____	
THESAURUS REFERENCE:		
8) ABSTRACT SIMBA is a simulation model for simulating combat between armour and anti-armour units. It was developed under FFI-project 701, "Fremtidige panserbekjempelsesvåpen" (Future anti-tank weapons). It has been further refined during FFI-project 798, "Panserbekjempelse 2000+" (Anti-tank warfare 2000+). The model is based on monte-carlo simulations, and consists of four programs: A preprocessor where the scenario is defined, a simulator, which does the actual simulation, and two different postprocessors; one for graphically examining one particular simulation run in detail, and one for examining the statistical results of several simulations. This document is a user manual for the simulation model. In addition, the calculation methods are briefly discussed.		
9) DATE 2004-10-10	AUTHORIZED BY This page only Johnny Bardal	POSITION Director

ISBN 82-464-0905-0

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE
 (when data entered)

FORSVARETS FORSKNING SINSTITUTT (FFI)
Norwegian Defence Research Establishment

UNCLASSIFIED

P O BOX 25
 NO-2027 KJELLER, NORWAY
REPORT DOCUMENTATION PAGE

SECURITY CLASSIFICATION OF THIS PAGE
 (when data entered)

1) PUBL/REPORT NUMBER FFI/RAPPORT-2003/01712	2) SECURITY CLASSIFICATION UNCLASSIFIED	3) NUMBER OF PAGES 54
1a) PROJECT REFERENCE FFI-III/798/139	2a) DECLASSIFICATION/DOWNGRADING SCHEDULE -	
4) TITLE SIMBA – BRUKERVEILEDNING SIMBA – USER MANUAL		
5) NAMES OF AUTHOR(S) IN FULL (surname first) HALSØR Marius, EIDE Morten		
6) DISTRIBUTION STATEMENT Approved for public release. Distribution unlimited. (Offentlig tilgjengelig)		
7) INDEXING TERMS IN ENGLISH: IN NORWEGIAN:		
a) <u>User manual</u>	a) <u>Brukermanual</u>	
b) <u>Simulation model</u>	b) <u>Simuleringsmodell</u>	
c) <u>SIMBA</u>	c) <u>SIMBA</u>	
d) <u>Anti-tank warfare</u>	d) <u>Panserbekjempelse</u>	
e) _____	e) _____	
THESAURUS REFERENCE:		
8) ABSTRACT SIMBA is a simulation model for simulating combat between armour and anti-armour units. It was developed under FFI-project 701, "Fremtidige panserbekjempelsesvåpen" (Future anti-tank weapons). It has been further refined during FFI-project 798, "Panserbekjempelse 2000+" (Anti-tank warfare 2000+). The model is based on monte-carlo simulations, and consists of four programs: A preprocessor where the scenario is defined, a simulator, which does the actual simulation, and two different postprocessors; one for graphically examining one particular simulation run in detail, and one for examining the statistical results of several simulations. This document is a user manual for the simulation model. In addition, the calculation methods are briefly discussed.		
9) DATE 2004-10-10	AUTHORIZED BY This page only Johnny Bardal	POSITION Director

ISBN 82-464-0905-0

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE
 (when data entered)

INNHOOLD

	Side	
1	INNLEDNING	9
1.1	Hovedstruktur	9
1.2	Katalogstruktur	10
2	MODELLEN	10
2.1	Hvordan starte en simulering	10
2.2	SIMBA's virkemåte	11
2.2.1	Input til SIMBA	11
2.2.2	Deteksjon	12
2.2.3	Skuddprosedyren	13
2.2.4	Bevegelse	13
2.2.5	Indirekte ild	14
2.3	Mulige ordre og deres betydning	14
2.4	Et enkelt scenario	16
2.4.1	Scenariefila	16
2.4.2	Hendelser som logges	16
2.4.3	Plattformfila	17
2.4.4	Kommandofila	17
2.4.5	Hendelsesforløpet	18
2.5	Matematiske modeller	19
2.5.1	Deteksjon	19
2.5.2	Treff- og killsannsynligheter	19
2.5.3	Artilleri	19
3	SIMBA PREPROSESSOR	20
3.1	Innledning	20
3.2	Hva er SIMBA Preprocessor	20
3.3	Oppstart	21
3.4	Innlasting av kart	21
3.5	Innlasting av scenario	23
3.6	Generering/Endring av scenario	23
3.6.1	Opprette en plattform	23
3.6.2	Innlegging av ordre	25
3.6.3	Lagre scenario	25
3.7	Andre knapper i "Elements Control"-vinduet	26
3.8	Siktbarhetsundersøkelser	27
3.9	Innlegging av siktbegrensninger	27
4	BRUKERVEILEDNING FOR SIMBA KARTLOGG	27
4.1	Generelt	27
4.1.1	Kravspesifikasjon	28
4.2	Hovedvindu	28

4.2.1	Tittellinjen	28
4.2.2	Menylinjen	28
4.2.3	Kartfeltet	32
4.2.4	Loggnummerboksen	32
4.2.5	Plattformboksen	32
4.2.6	Posisjonsboksen	32
4.2.7	Logglinjeboksen	32
4.2.8	Styrefeltet	33
4.2.9	Statuslinjen	33
4.3	Koordinatvindu	33
4.4	Hendelsesvinduet	34
4.4.1	Avspillingsmodus	34
4.4.2	Hendelseskategorier	35
4.5	Plattformvinduet	35
4.5.1	Plattformlisten	35
4.5.2	Plattformvalgene	36
4.5.3	Vis øverst	36
4.5.4	Lukk vindu	37
4.6	Kortinfovinduet	37
5	BRUKERVEILEDNING FOR SIMBA POSTPROSESSOR	38
5.1	Kravspesifikasjon	38
5.2	Hovedvinduet	38
5.2.1	Tittellinjen	38
5.2.2	Menylinjen	38
5.2.3	Filspesifikasjonsfelt	40
5.2.4	Plattformfelt	40
5.2.5	Hendelsesfelt	42
5.2.6	Filvisningsknapp/filvalgfelt	42
5.2.7	Matriseknapp	42
5.2.8	Resultatknapp	42
5.2.9	Lesfilknapp	42
5.2.10	Avslutningsknapp	42
5.2.11	Statusfelt	43
5.2.12	Antallsfelt	43
5.3	Resultatvindu	43
5.3.1	Tittellinje	44
5.3.2	Menylinje	44
5.3.3	Resultattypeboks	44
5.3.4	Resultatutskriftsboks	44
5.3.5	Avbryttnapp	44
5.3.6	Skrivfilknapp	45
5.3.7	Filspesifikasjonsfelt	45
5.3.8	Statusfelt	45
5.4	Matrisevinduet	46
5.4.1	Tittellinje	46
5.4.2	Menylinje	46
5.4.3	Mål/Våpenknapp	47
5.4.4	Resultattypeboks	47
5.4.5	Poengtypeboks	47

5.4.6	Matrisefeltet	47
5.4.7	OK-knapp	47
5.4.8	Skrivfilknapp	47
5.4.9	Filspekifikasjonsfelt	47
5.4.10	Statusfelt.	47
6	FILER SOM BRUKES AV SIMBA	48
6.1	Kartfiler	48
6.1.1	Hekkefil (.wall)	48
6.1.2	Kartbildefiler (.jpg/.bmp)	48
6.1.3	Kartdatafil (.dat)	48
6.1.4	Koordinatfil (.krd)	48
6.1.5	Matrisefil (.con)	49
6.1.6	Terrengfil (.ter)	49
6.1.7	Vegetasjonsfil (.veg)	49
6.2	Våpenfiler	49
6.2.1	Direkteskytende våpen	49
6.2.2	Konvensjonelt artilleri	50
6.2.3	Smart artilleri	51
6.3	Deteksjonsfiler	51
6.4	Andre filer	52
	Litteratur	53

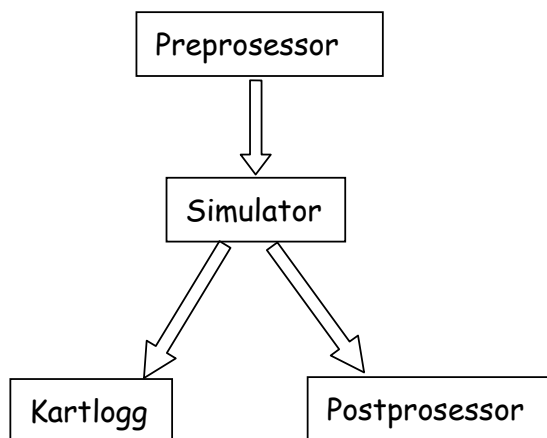
SIMBA - Brukerveiledning

1 INNLEDNING

SIMBA ble utviklet under prosjekt 701, *fremtidige panserbekjempelsesvåpen*, og skulle støtte prosjektet med kvantifisering av stridsutfall. Man prøvde først å finne eksisterende simuleringmodeller som man kunne bygge videre på for dette formålet, men ga etter hvert opp dette og begynte utviklingen av et nytt simuleringprogram; SIMBA. SIMBA står for SimuleringsModell på BAaljonsnivå, men dette er egentlig en liten overdrivelse. Opprinnelig var riktignok SIMBA ment å være en modell på bataljonsnivå, men i sin nåværende form er modellen best egnet for strid på lavere nivåer (kompaninivå), hovedsakelig på grunn av manglende AI (Artificial Intelligence, eller kunstig intelligens) i modellen. Det er for øvrig ikke noe problem å legge inn mange nok elementer, problemet er at i SIMBA baserer enhetene seg på en på forhånd gitt ordre-liste, som de ikke kan fravike i særlig grad, og riktigheten av denne tilnærmelsen minker med økende antall elementer og økende tid i engasjementet. For strid på kompaninivå med de rette komponentene er derimot SIMBA et godt verktøy.

Formålet med dette dokumentet er å gi en brukerveiledning for SIMBA, og å forklare kort hvordan striden blir modellert. SIMBA består egentlig av 4 forskjellige programmer, som vil bli beskrevet hver for seg.

1.1 Hovedstruktur



Figur 1.1: Hovedstrukturen i SIMBA.

Det er i preprosessoren man lager scenariene i SIMBA. Her genereres et sett med filer som beskriver scenariet. Disse filene er input til simulatoren. I simulatoren foregår selve simuleringen av scenariet. Her er det lite for brukeren å gjøre. Simulatoren genererer et sett med loggfiler, der hver loggfil beskriver hendelsesforløpet i én enkelt simulering.

Kartlogg og postprossessor er to programmer for å tolke resultatene fra simulatoren. Disse tar

loggfilerne fra simulatoren som input. I kartloggen vises hendelsesforløpet i én enkelt simulering grafisk. I postprosessoren presenteres resultatene fra en serie simuleringer av samme scenario, i form av statistikk fra disse simuleringene.

1.2 Katalogstruktur

For at SIMBA skal fungere, kreves det en bestemt katalogstruktur. Den øverste katalogen heter "Simba" (den kan i prinsippet hete hva som helst). Under denne katalogen må følgende underkataloger ligge:

- **Data.** Her ligger scenariofiler, plattformfiler, kommandofiler, våpenfiler og lignende.
- **Kartlog.** Her ligger alt som har med kartlogg-programmet å gjøre.
- **Map.** Her ligger diverse filer knyttet til kart.
- **Postprossessor.** Her ligger alt som har med postprosessoren å gjøre.
- **Preprossessor.** Her ligger alt som har med preprosessoren å gjøre.
- **Result.** Her ligger resultatfilene i egne underkataloger.
- **Simulator.** Her ligger filene som tilhører selve simulatoren.

2 MODELLEN

SIMBA's oppgave har vært å kvantifisere stridsutfall i scenarier der det hovedsakelig er panserbekjempelsesvåpen (PB-våpen) på en den ene siden og stridsvogner (Strv) og/eller stormpanservogner (SPV) på den andre siden. Modellen er derfor laget med tanke på slik strid, og er best egnet til å simulere strid med PB-våpen (herunder både direkteskytende og indirekteskytende) på en side, og Strv (eller SPV) samt evt artilleri på den andre siden. Forøvrig er programmet programmert på en slik måte at det skal være mulig å legge inn nye plattformer med nye egenskaper, dersom dette skulle være ønskelig. Begrensningene som ligger i SIMBA, blant annet når det gjelder manglende evne til å endre taktikk underveis, gjør modellen mindre egnet for større slag og bruk av mange forskjellige våpensystemer på begge sider. Det er derfor anbefalt å holde seg på kompaninivå eller lavere.

De forskjellige delene av SIMBA er programmert i forskjellige programmeringsspråk, og tidligere kjørte de også under forskjellige operativsystem. I den siste versjonen, versjon 2, kjøres alle delene under Windows.

2.1 Hvordan starte en simulering

Før man kan starte en simulering, må man generere et scenario. Dette gjøres i preprosessoren, og dette er nærmere beskrevet i kapittel 3.

Simuleringen startes ved å dobbeltklikke på "SIMBA"-ikonet. Dette starter et DOS-vindu, og brukeren blir bedt om å skrive inn navnet på scenariofila. Denne skal skrives inn uten filendelser (skriv "mitt_scenario", ikke "mitt_scenario.scn"). Scenariofila genereres i preprosessoren, se kapittel 3. Programmet leser nå inn scenariofila og starter simuleringene i tråd med instruksjonene i denne fila.

Dersom man ikke ønsker å logge alle inntrufne hendelser, kan man editere "loggfil", eventuelt lage en ny. Hvilken loggfil som benyttes, spesifiseres i scenariofila. Dette gjør at loggfilene som simuleringene presenterer blir kortere, og inneholder mindre informasjon. Selve simuleringene tar (for alle praktiske formål) like lang tid uavhengig av hvilke hendelser som skal logges; motivet for å kutte ned på loggede hendelser er enten et ønske om en mer oversiktlig loggfil (for de som leser denne), eller at postprosesseringen tar kortere tid. Det er imidlertid svært lite å spare, og det anbefales at man logger alle hendelser.

Ønsker man å editere noen av parametrene som SIMBA benytter, kan man åpne fila med de relevante parametre i en tekst-editor og editere tallene der. Våpenfilene har filendelse wpn. Fila som inneholder generelle parametre heter "parameters.dat".

2.2 SIMBA's virkemåte

Sentralt for SIMBA er prosedyrer for deteksjon og avfiring av skudd, samt reaksjoner på, og effekter av, skudd. Simulering av virkningen av skudd tar lang tid, og er derfor ikke direkte en del av modellen. I stedet er det gjort simuleringer av skudd med aktuell ammunisjon mot aktuelle mål på forhånd i modellen TANKKILL (1), og resultatene av disse simuleringene ligger lagret i tabeller som SIMBA slår opp i for å finne effekten av et skudd. SIMBA er således ikke en modell for å finne effekten av enkeltskudd, men baserer seg på vekselvirkningen mellom ulike våpensystemer i en strid med flere enheter, opp til kompaninivå, for å finne den totale effekten av en slik strid.

2.2.1 Input til SIMBA

Før SIMBA starter simuleringene, leser den en scenariofil spesifisert av brukeren, som beskrevet i kapittel 3.6. Denne inneholder informasjon om størrelsen på tidssteget, som bestemmer hvor lang tid det går mellom hver oppdatering, og hvilket kartområde simuleringen foregår i. Videre inneholder den informasjon om hvilken plattformfil og kommandofil som skal benyttes under simuleringen. Scenariofil, plattformfil og kommandofil genereres i preprosessoren. Hvordan dette teknisk gjøres vil bli beskrevet i kapittel 3.6, i dette kapitlet konsentrerer vi oss om betydningen av disse filene. I tillegg leser SIMBA inn noen filer som inneholder en god del parametre av mer generell karakter som også brukes under simuleringen.

Plattformfilen inneholder informasjon om hvilke plattformer som deltar på Oransje og Blå side, og hvilke våpen disse er utstyrt med. Kommandofilen er en dreiebok som gir hver enkelt plattform de ordre som skal utføres i løpet av scenariet. En ordre kan være hvilken retning en plattform skal observere i, hvor raskt og i hvilken retning den skal bevege seg, at den skal vente på et sted i en gitt tid, og mye annet. Kun indirekteskytende plattformer gis ordre om å skyte; direkteskytende plattformer skyter så snart de har identifisert en fiendtlig plattform og kriteriene som skal til for at den kan skyte er oppfylt.

Kommandofilen kan ikke endres i løpet av scenariet. Det kan naturligvis oppstå en situasjon som tilsier at en plattforms ordre burde endres, men dette vil, med enkelte unntak, ikke bli tatt hensyn til i SIMBA. Dette gjør at SIMBA bør brukes til trefninger som er begrenset i tid og omfang. Årsaken til at slike endringer underveis i et scenario ikke tillates manuelt, er at SIMBA er en stokastisk modell, og at det derfor kreves et relativt stort antall (over 100)

simuleringer for å få statistisk signifikante resultater. Det vil derfor ta alt for lang tid om man skal følge hver enkelt simulering manuelt

2.2.2 Deteksjon

For at en blå plattform skal kunne detektere en oransje plattform, må en del kriterier være oppfylt: Det må være en fri siktelinje mellom plattformene (LOS), ingen av plattformene kan være i dekning, og oransje plattform må være i blå plattformers deteksjonssektor. Så snart alle disse kriteriene er oppfylt, trekkes det fra en sannsynlighetsfordeling om oransje plattform blir detektert. Sannsynligheten for deteksjon beregnes ut fra avstanden mellom plattformene, størrelsen på oransje plattform, siktsektoren til blå plattform, og hvilket siktemiddel som benyttes. Målets hastighet er så langt ikke en parameter her, selv om bevegelse opplagt vil være viktig for deteksjonssannsynligheten. I våre scenarier har vi kompensert for dette ved å gi mål som er i bevegelse i scenariet en høyere sannsynlighet for å bli oppdaget, enn mål som holder seg i ro i scenariet. Dersom man ønsker det, og finner en brukbar deteksjonsmodell som tar hensyn til målets hastighet, er det enkelt å legge inn denne i SIMBA.

Så snart oransje plattform er detektert, må den identifiseres som en fiendtlig plattform. Dette tar et forhåndsbestemt antall sekunder. Så skal blå plattform velge hvilken av de identifiserte plattformene som skal velges som mål. Dette gjøres ved at det målet den har størst sannsynlighet for å slå ut velges, dersom ikke et annet mål allerede er innsiktet. Når et mål er valgt, blir dette beskutt dersom det er innen rekkevidde og blå plattform har mulighet for, og tillatelse til, å skyte. Modellen er på alle måter symmetrisk mellom blå og oransje, så dersom man bytter om «blå» og «oransje», gjelder det samme.

2.2.2.1 Deteksjon av røyk

En PB-stilling vil som regel være så liten og godt skjult at det vil være vanskelig for en stridsvogn å detektere den dersom PB-skytterne holder seg i ro. Men når PB-stillingen avfyre et missil, avgir den en skuddsignatur som er lettere å oppdage. For at en stridsvogn skal kunne detektere denne skuddsignaturen (heretter omtalt som røyk), må PB-stillingen være innenfor deteksjonssektoren til stridsvognen, og det må være LOS (Line of sight, altså fri sikt) til et punkt noen meter over PB-stillingen. Selv om PB-stillingen går i dekning umiddelbart etter avfiring, kan røyken fortsatt detekteres. Røyken forsvinner etter et forhåndsbestemt antall sekunder, og kan etter dette ikke lenger detekteres. Når røyk er detektert, kreves ingen identifikasjon før skudd kan avfyres. Skuddet avfyres da mot et område, ikke et enkeltmål, og har derfor vesentlig mindre sjanse for å treffe enn om målet hadde vært detektert direkte.

2.2.2.2 Innskrenking av siktsektor

En stridsvogn som detekterer en fiendtlig plattform eller røyk, vil konsentrere seg om å observere i dette området. Dette implementeres ved at siktsektoren innskrenkes. Dette er mulig for enkelte observasjonsmidler, men ikke nødvendigvis for alle. Konsekvensen av at siktsektoren innskrenkes er at det blir mye lettere å detektere fiendtlige plattformer innen den nye siktsektoren, men at plattformer utenfor denne lille siktsektoren ikke kan detekteres.

2.2.2.3 Kommunikasjon

Kommunikasjon er i liten grad implementert i SIMBA. Det som er implementert, er at en stridsvogn som detekterer en fiendtlig plattform eller røyk, vil fortelle enkelte andre, vennligsinnede stridsvogner i hvilket område den har detektert fienden. Hvilke plattformer som skal få vite dette, er forhåndsbestemt i plattformfila (alle stridsvogner som har de samme 2 første sifrene i ID-nummeret sitt, deler denne informasjonen). Når en stridsvogn får beskjed om hvor en fiende befinner seg, konsentrerer den seg om å observere i dette området, og innskrenker siktsektoren som om den skulle ha detektert fienden selv. Men den får ikke lov til å skyte før den selv har detektert fienden (eller røyk fra denne).

2.2.3 Skuddprosedyren

Alle plattformer som ikke har ladet, men har ammunisjon igjen, vil begynne å lade. Tiden det tar for en plattform å lade varierer fra våpentype til våpentype. Alle plattformene starter simuleringen med alle våpen ladd. Dette representerer en liten feil for de tilfeller der forskjellige våpen egentlig betyr forskjellig ammunisjon for samme våpen.

Før en plattform kan skyte, må den sikte seg inn. Som for lading varierer tiden det tar å sikte fra våpentype til våpentype. PB-missiler må siktes inn på nytt før hvert skudd, mens stridsvognskanoner kun trenger å sikte seg inn på et mål én gang, dersom målet ikke endrer posisjon.

Etter at skuddet er avfyrt, tar det en viss tid før det når målet, avhengig av hastigheten til missilet (enkelte våpentyper har instantan effekt). Når missilet når målet, trekkes det hvilken effekt det får. Det er følgende muligheter: Bom, treff uten virkning, mobility-kill (Mkill), firepower-kill (Fkill) og total-kill (Tkill = Mkill U Fkill). PB-systemer kan kun påføres Tkill eller bom (eller treff uten virkning, dersom målet allerede var slått ut før det ble beskutt). Sannsynligheten for de forskjellige resultatene er avhengig av typen mål og våpen, avstand til målet, missilets angrepsvinkel, om målet er i dekning eller ikke, og eventuelt om kun skuddsignaturen til målet er detektert (kun PB-stillinger avgir skuddsignaturen). Alle mål som ikke er Tkilled, vil kunne bli beskutt på nytt. Skyter man mot et mål som er Mkilled og påfører det en Mkill, blir dette registrert som et treff uten virkning. Det vil ta en viss tid fra man påfører målet en Tkill til man registrerer at målet er Tkilled. I løpet av denne tiden kan målet fortsatt bli beskutt, men selvsagt uten at det kan bli påført ytterligere skade. Eventuelle treff vil da registreres som treff uten virkning. Plattformer som er Fkilled kan selvfølgelig ikke skyte, og plattformer som er Mkilled kan ikke bevege seg.

2.2.4 Bevegelse

Plattformen i SIMBA kan endre sin posisjon på to måter; enten ved normal bevegelse eller ved instantan forflytning. Ved normal bevegelse får plattformen en hastighet og en posisjon den skal bevege seg til, og plattformens posisjon blir oppdatert hvert tidssteg. Instantan forflytning foregår ved at plattformen flyttes direkte fra en posisjon til en annen. I virkeligheten er det naturligvis ingen som er i stand til instantan forflytning. Årsaken til at dette likevel av og til blir brukt er følgende: Når en stridsvogn detekterer røyk, skyter den mot et område (definert ved punktet PB-stillingen hadde da den avfyrt skuddet), og ikke mot en plattform direkte. Måten dette er implementert på i SIMBA, er at en PB-stilling kan bli truffet dersom den befinner seg i dette punktet, og ellers ikke. Ofte går den i dekning umiddelbart etter å ha

avfyrt et skudd, og kan derfor ikke detekteres. Så dersom PB-stillingen skulle bevege seg normalt bort fra stillingen, ville den allerede etter ett tidssteg ha endret sin posisjon, og vil ikke kunne bli truffet. For å unngå dette kan man gi den ordre om å forbli i stillingen den tiden det antas at den vil være sårbar, og den kan etter denne tiden flyttes instantant ut fra stillingen til den posisjonen PB-stillingen da vil være i.

2.2.5 Indirekte ild

Indirekte plattformer kan selv ikke bli truffet, kun effekten av indirekte ild tas hensyn til. En artilleri-plattform har et antall våpen, som kan sees på som antall skyts i et batteri eller en bataljon. I motsetning til direkteskytende plattformer, som først skyter på mål når de har detektert dem, gis direkteskytende plattformer ordre om når og hvor de skal skyte. Og i motsetning til direkteskytende plattformer, skyter de ikke mot et bestemt mål, men mot en bestemt koordinat.

Denne koordinaten er sentrum av siktemønsteret. I tillegg kan hvert enkelt våpen (skyts) sikte en bestemt avstand fra dette senteret, slik at man kan lage seg det siktemønsteret man måtte ønske. Indirekte ild har en bomavstand, som er felles for alle skytsene til en plattform, samt en spredning, som er et individuelt avvik mellom treffpunkt og siktepunkt for hvert enkelt skyts.

Det finnes to forskjellige typer indirekte ild, konvensjonell ammunisjon (HE) og smart ammunisjon. Konvensjonell ammunisjon har en viss sannsynlighet for å slå ut hvert mål innenfor en bestemt radius, der sannsynligheten er avhengig av måltype og avstanden til nedslagsfeltet. Smart ammunisjon søker etter mål innenfor et søkeområde, og dersom den oppdager et mål, har den en bestemt sannsynlighet for å slå ut dette, avhengig av måltype.

2.3 Mulige ordre og deres betydning

Dette kapitlet beskriver alle ordrene som en plattform kan få i SIMBA. Noen ordre utføres umiddelbart, og plattformen fortsetter da med å lese og utføre neste ordre. Andre ordre tar tid å utføre, og plattformen venter da med å lese neste ordre til denne ordren er utført. Enkelte ordre må med for å sette initialbetingelsene til plattformen; dersom disse ordrene ikke er med, kan det oppstå feil under simuleringen. Hvilke ordre som må være med, kommer frem i listen nedenfor.

Ordre gis i kommandofila. Et eksempel på en kommandofil er gitt i kapittel 2.4.4. Her kommer en liste over alle mulige ordre som er mulig å gi til plattformer i SIMBA.

- ID x . Dette forteller at de påfølgende ordrene gjelder for plattformen med ID-nr. x . Må være med. Dersom denne plattformen ikke finnes i plattform-fila, vil det oppstå en feil under kjøring.
- START x, y, a . Dette er den første ordren til alle direkteskytende plattformer (etter ID). x og y angir plattformens posisjon i UTM-koordinater, mens a er vinkelen mellom øst og plattformens retning i grader i positiv omløpsretning. Denne ordren må være med for direkteskytende plattformer.
- STOP. Plattformen er ferdig med alle sine ordre, og fjernes fra simuleringen. Denne ordren må være med.

- `ADD_AMMO y x`. Plattformens våpen nr. y får x ammunisjonsenheter. Denne ordren må være med for alle direktskytende plattformers våpen.
- `SET_VIEW_DIRECTION y x`. Plattformens observasjonsmiddel nr. y settes til å observere i en retning som er x grader i forhold til øst. Denne ordren må være med for alle direktskytende plattformers observasjonsmidler.
- `SET_VIEW_SECTOR y x`. x er her den sektoren som plattformens observasjonsmiddel nr. y observerer i. Jo mindre denne er, dess lettere er det å detektere noe i sektoren. Denne ordren må være med for alle direktskytende plattformers observasjonsmidler.
- `SET_ELEVATION y x`. Som over, men x er nå sektoren i planet normalt på bakken. Denne ordren er kun med for å kunne justere deteksjonssannsynligheten; plattformen kan detektere mål i alle høyder uavhengig av størrelsen på denne parameteren. Denne ordren må være med for alle direktskytende plattformers observasjonsmidler.
- `SET_VIEW_RANGE y x`. x forteller her hvor langt plattformens observasjonsmiddel nr. y «ser». Ingenting som ligger utenfor denne rekkevidden kan detekteres. Denne ordren må være med for alle direktskytende plattformers observasjonsmidler.
- `WAIT x`. Betyr at plattformen skal vente i x sekunder før den utfører neste ordre. Den får beskytte eventuelle mål som detekteres mens den "venter".
- `WAIT_FOR_SHOTS x`. Plattformen skal vente til den har avfyrt x skudd før den utfører neste ordre. Den må fortsatt detektere mål før den kan skyte.
- `HIDE x`. Betyr at plattformen skal gå i dekning, og vente i x sekunder med å utføre neste ordre.
- `MOVETO x y z`. Plattformen starter å bevege seg mot posisjonen med UTM-koordinater $x:y$ med en hastighet z m/s.
- `GOTO x y`. Plattformen flyttes instantant til det forhåndsbestemte punktet med UTM-koordinater $x:y$.
- `SET_FIRE_DENIED x`. Forbyr plattformen å skyte til den får kontraordre. x er her navnet på flagget som settes.
- `SET_FIRE_ALLOWED x`. Tillater plattformen å skyte (opphever ordren «`SET_FIRE_DENIED`»).
- `SET_FLAG x`. Setter flagget x til TRUE. Dette flagget kan brukes til å trigge hendelser hos andre plattformer.
- `RESET_FLAG x`. Setter flagget x til FALSE.
- `WAIT_FOR_FLAG x`. Plattformen venter til flagget x er TRUE før den utfører neste ordre.
- `ART x y a b c d e`. $x:y$ er artilleriets siktepunkt, a er skuddtakten (skudd pr sekund), b er antall skudd hvert skyts i artilleriet skal skyte, c er skuddretningen (gitt i grader i positiv

omløpsretning i forhold til øst), og d og e er bomavstanden i henholdsvis x - og y -retningen.

2.4 Et enkelt scenario

Dette kapitlet beskriver hvordan et enkelt scenario kan være bygd opp. Det gis eksempler på scenariofil, plattformfil og kommandofil, og SIMBAs tolkning av disse.

2.4.1 Scenariofila

RUNS	2
START	0
STOP	3600
STEP	1
MAP	mrud
PLATFORMS	overgard.el
COMMANDS	overgard.cm
LOG	default.log
DIRECTORY	result1

Dette betyr følgende: Det skal kjøres 2 simuleringer, tiden skal være 0 ved starten av scenariene, scenariene skal avsluttes ved tiden 3600 (sekunder) dersom ikke andre avbruddskriterier er oppfylt. Hvert tidssteg skal være 1 sekund. Kartfilene som skal brukes heter mrud (med diverse filendelser), plattformfilen som skal brukes heter overgard.el, kommandofilen som skal brukes heter overgard.cm, hendelser som skal logges hentes fra filen default.log, og katalogen som loggfilene skal lagres på heter result1.

2.4.2 Hendelser som logges

I scenariofila spesifiseres en fil som inneholder informasjon om hvilke hendelser som skal loggføres. I eksempelet over heter denne fila "default.log". Denne fila ser ut som følger:

LOG_DEBUG	(Informasjon som ikke passer i de andre kategoriene)
LOG_DETECTION	(Data som omhandler deteksjon)
LOG_MISSILE_MOVES	(Data som omhandler missilbevegelse)
LOG_FIRE	(Data som omhandler skudd, lading og sikting)
LOG_VIEW	(Data som omhandler sikting)
LOG_SCREEN	(Hjelpelinjer for kartlogg)

Kommentarene er ikke en del av fila. Dette er alle hendelstypene det er mulig å logge. I tillegg kommer en rekke hendelser som alltid logges. Det anbefales å alltid logge alle hendelser.

2.4.3 Plattformfila

```
BEGIN
```

```
TYPE PB JAVELIN
ID 101
HEIGHT 2.0
COVER 0.0
OBS_DEV 1 EYE eye
WEAPON 1 FF javelin
BRIG 1
```

```
END
```

```
BEGIN
```

```
TYPE TANK T80
ID 201
HEIGHT 2.0
COVER 0.0
OBS_DEV 1 EYE eye
WEAPON 1 GUN t80
BRIG 2
```

```
END
```

Beskrivelsen av alle plattformer starter med linja BEGIN og avsluttes med linja END. Denne plattformfila inneholder to plattformer, en med navn JAVELIN og en med navn T80, av typene PB og TANK. Plattformen JAVELIN har ID-nr 101, en høyde på 2,0 meter, en dekning på 0,0 meter, et observasjonsmiddel nr. 1 beskrevet i filen eye.dat (øye), våpen nr. 1 av typen FF (Fire & Forget), som er beskrevet i filen javelin.wpn, og den tilhører brigade 1 (brigade 1 er «blå» og brigade 2 er «oransje»). Tilsvarende informasjon er gitt for plattformen T80. Flere plattformer kan ha samme navn, men alle må ha forskjellige ID-numre.

2.4.4 Kommandofila

```
BEGIN
```

```
ID 101
START 451114 7674966 104.5
ADD_AMMO 1 3
SET_VIEW_DIRECTION 1 0
SET_VIEW_SECTOR 1 4
SET_ELEVATION 1 4
SET_VIEW_RANGE 1 2500
WAIT_FOR_SHOTS 1
```

```
HIDE 50
STOP
END
```

```
BEGIN
ID 201
START 450956 7676899 192.39
ADD_AMMO 1 50
SET_VIEW_DIRECTION 1 0
SET_VIEW_SECTOR 1 20
SET_ELEVATION 1 8
SET_VIEW_RANGE 1 4000
MOVETO 450357 7676842 5
MOVETO 449569 7676556 5
STOP
END
```

Kommandoene for alle plattformer starter med linja BEGIN og avsluttes med linja END. Ettersom betydningen av de forskjellige ordrene ble beskrevet i kapittel 2.3, vil ikke dette bli gjennomgått nærmere her.

2.4.5 Hendelsesforløpet

Først blir den ene plattformen (201) plassert i start-posisjonen sin, våpenet får ammunisjon, observasjonsmiddelet får sikteretning, siktsektor (både høyde og bredde) og sikt-rekkevidde som beskrevet i kommandofila. Den neste kommandoen er en MOVETO-kommando, som det tar tid å utføre. Når denne ordren utføres, begynner plattformen å flytte seg, og vil ikke lese neste ordre før denne ordren er utført. Så (samtidig, ettersom ingen av disse hendelsene tok noe tid) plasseres plattform 101 i sin startposisjon, og får sine initialbetingelser som beskrevet i kommandofila. Når den kommer til ordren WAIT_FOR_SHOTS 1, venter den med å lese neste ordre til denne ordren er utført. Den holder seg altså i ro inntil den har fått skutt ett skudd. Et unntak er hvis den selv skulle bli beskutt først. Da vil den automatisk lese neste ordre (dette gjelder kun for klassen PB). Alle plattformer starter med alle sine våpen ladet.

Etter hvert som plattform 201 beveger seg, vil den komme ut av hindringene beskrevet i kartfilen, slik at det er Line-Of-Sight (LOS) mellom plattformene. På grunn av retningen og størrelsen på siktsektorene, vil begge plattformene nå kunne se hverandre. Nå trekkes det med en viss sannsynlighet hvert sekund om plattformene oppdager hverandre. La oss anta at 101 oppdager 201 (dette er det mest sannsynlige, ettersom 101 har en mindre siktsektor og 201 er det største målet). Den vil så bruke noen sekunder på å identifisere 201 som en fiendtlig plattform, så vil den velge denne som sitt mål, sikte, og skyte.

Nå er ordren WAIT_FOR_SHOTS 1 utført, og neste ordre leses. Her står det HIDE 50. Denne ordren må utføres før neste ordre kan leses. Mens den utfører denne ordren, begynner den å lade, ettersom den ikke har ladet, men har ammunisjon igjen. 101 går umiddelbart i dekning, men må være der i 50 sekunder. Når 101 har avfyrt sitt missil, kommer det røyk som 201 kan detektere. La oss anta at 201 klarer å detektere denne røyken. Den får da en smalere

siktsektor, men dette har ingen betydning i dette tilfellet, ettersom det ikke er noen plattformer den kan detektere (101 er i dekning). Den trenger ikke å identifisere røyken, og ettersom den allerede har ladet, begynner den å sikte. Når den har siktet, skyter den. Ettersom den ikke har sett selve plattform 101, kun røyken fra denne, er sjansen for å treffe liten. Det trekkes om den treffer eller ikke, med sannsynlighet hentet fra våpenfila til 201 sitt våpen (t80.wpn). La oss anta at resultatet ble bom. Umiddelbart etter at den har skutt, vil den begynne å lade. Mens 201 lader, vil det ha gått så lang tid at missilet fra 101 kommer frem til målet. Det vil så trekkes, med sannsynligheter hentet fra våpenfila til plattform 101 sitt våpen (javelin.ver2) hvilken effekt skuddet har. La oss anta at resultatet ble en Fkill. Plattform 201 har da ikke lenger mulighet til å skyte.

Etter 50 sekunder har plattform 101 utført ordren HIDE 50, og leser neste ordre. Her står det STOP, og plattform 101 fjernes umiddelbart fra simuleringen (den antas å ha kommet i sikkerhet). Nå er det ikke flere plattformer igjen i brigade 1. At det ikke er flere (levende) plattformer igjen i en av brigadene, er ett av avbruddskriteriene i SIMBA. Men det vil fortsatt gå 50 sekunder før simuleringen avsluttes, for å få med effekten av eventuelle skudd avfyrt fra en plattform rett før denne ble fjernet. I dette tilfellet er det ingen slike skudd, og simuleringen endte altså med en Fkilled T80, og et uskadd Javelin-lag.

2.5 Matematiske modeller

2.5.1 Deteksjon

Parametrene som bestemmer deteksjonssannsynligheten for et mål, er målets kontrast mot bakgrunnen, målets areal (A), avstanden til målet (r) og størrelsen på skytterens observasjonssektor (θ ganget med Ω , der θ er vinkelen i horisontalplanet og Ω vinkelen i vertikalplanet). Formelen for deteksjon ser slik ut:

$$P_{det} = 1 - \exp\left(-\frac{C_m \cdot A}{r^2 \cdot \theta \cdot \Omega} \Delta t\right) \quad (1)$$

der C_m er en konstant som blant annet tar hensyn til kontrast, og Δt er lengden på tidssteget i simuleringene. Det trekkes hvert tidssteg om målet detekteres eller ikke.

2.5.2 Treff- og killsannsynligheter

Når et missil har kommet frem til sitt mål, må det avgjøres om det treffer eller ikke, og eventuelt hvilken skade det påfører målet. Simuleringen av et skudd som treffer et mål er meget komplisert og for tidkrevende til å kunne utføres i SIMBA. Derfor er disse simuleringene gjort på forhånd, bl.a. ved hjelp av programmet TANKKILL (1), og resultatene foreligger i tabellform i våpenfilene til SIMBA. Disse er representert som sannsynligheter for å treffe målet som funksjon av avstand, og sannsynligheter for å påføre målet forskjellige typer skade som funksjon av vinkelen mellom missil og mål.

2.5.3 Artilleri

I SIMBA kan ikke artilleri bli beskutt, bare skyte. Artilleri trenger ikke å detektere mål for å skyte, så det eneste som modelleres for artilleri, er effekten i nedslagsfeltet. Parametre som

påvirker denne effekten er antall rør som skyter, skuddtakt, bomavstand, spredning og ammunisjonens effekt mot forskjellige mål. Måldeteksjon antas å være gjort av en OP eller annen ildleder på forhånd.

Artilleri treffer ikke nøyaktig der det sikter. Nedslagsfeltet forskyves derfor en avstand i x- og y-retning. Disse avstandene trekkes fra en normalfordeling med den angitte bomavstanden som standardavvik. Tilsvarende forskyves treffpunktet til hver enkelt granat en avstand i x- og y-retning, som trekkes fra en normalfordeling med spredningen som standardavvik. Spredningen er definert i våpenfila (se kapittel 2.4.3). Alle skytsene til en artilleriplattform trenger ikke å sikte på samme punkt. I plattformfila kan man derfor gi hvert skyts et ønsket avvik fra senterpunktet, for å generere ønsket siktemønster.

Tidspunktet en granat lander på er bestemt av skuddtaket. Dersom skuddtaket er 0.1, vil hvert skyts avfyre ett skudd hvert tiende sekund. Nøyaktig når granaten lander innenfor en tisekunders periode trekkes tilfeldig.

Konvensjonell og smart ammunisjon fungerer på to forskjellige måter. For konvensjonell ammunisjon har alle mål innenfor en maksimumsavstand en bestemt sannsynlighet for å bli slått ut. Denne sannsynligheten er avhengig av måltype og avstanden til nedslagspunktet. Det trekkes så for hvert enkelt mål om det blir slått ut eller ikke.

Smart ammunisjon, derimot, søker etter mål innenfor et søkeområde. Vanligvis søker de utenfra og innover i en spiral. Dette er implementert slik at det først trekkes om granaten oppdager målet som ligger ytterst i søkeområdet. Hvis den ikke gjør det, trekkes det om det neste målet oppdages, og så videre, helt til den angriper et mål eller har forsøkt å oppdage alle målene i søkeområdet uten å lykkes. Et allerede utslått mål gis mindre sannsynlighet for å bli oppdaget enn et som ikke er slått ut, for å ta hensyn til granatens overkillmekanismer. Når et mål oppdages, trekkes det om dette slås ut. En smart granat kan kun slå ut ett mål.

3 SIMBA PREPROSESSOR

3.1 Innledning

For å gjøre bruken av SIMBA Preprocessor så enkel som mulig, vil brukerveiledningen for denne modulen følge et oppgavebasert mønster. Med andre ord: Det beskrives hvordan man løser bestemte oppgaver, isteden for den noe mindre intuitive framgangsmåten der man beskriver programkomponentene. Dette betyr i praksis at man slår opp på kapittelet "Generere Scenarior" når det er dette man ønsker å gjøre, isteden for å slå opp på "Plattformvinduet", for så å lese alt om dette før man fortsetter. For nærmere beskrivelse av programmets virkemåte og vinduenes implementasjon henvises det til kildekodedokumentasjonen (2).

3.2 Hva er SIMBA Preprocessor

Kort fortalt er SIMBA Preprocessor et verktøy for å sette opp scenarier som skal simuleres i SIMBA Simulator. Programmodulen har liten nytteverdi uten resten av SIMBA, men den kan isolert sett brukes til å undersøke siktbarhet i et gitt terreng, samt å vise kartbilder. For å få mest mulig ut av SIMBA Preprocessor i simuleringssammenheng, er det viktig at modulen er

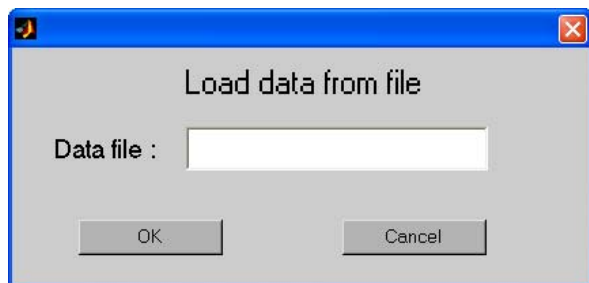
installert sammen med resten av SIMBA på den måte (i de kataloger) som er foreskrevet i kapittel 1. Så er det i grunnen bare å starte.

Programmet er skrevet i MatLab, og man må ha MatLab oppe for å bruke programmet. Man må stå i riktig katalog og skrive "sp", som er navnet på den kjørbare filen.

3.3 Oppstart

Når programmet først startes, ønskes man velkommen av ungløven Simba. Vinduet som rommer Simba, meldingsområdet (under) og hovedmenyen (over) kalles hovedvinduet. Det området som nå brukes til å vise Simba, er Kartområdet. Dette området vil senere vise det kartet som er lastet. Menyene brukes til å styre programmets funksjonalitet på vanlig måte, og de enkelte menyvalg er forklart i den sammenheng de brukes. Meldingsområdet brukes i vekslende grad til å fortelle brukeren hva som opptrår programmet og hva som forventes av brukeren.

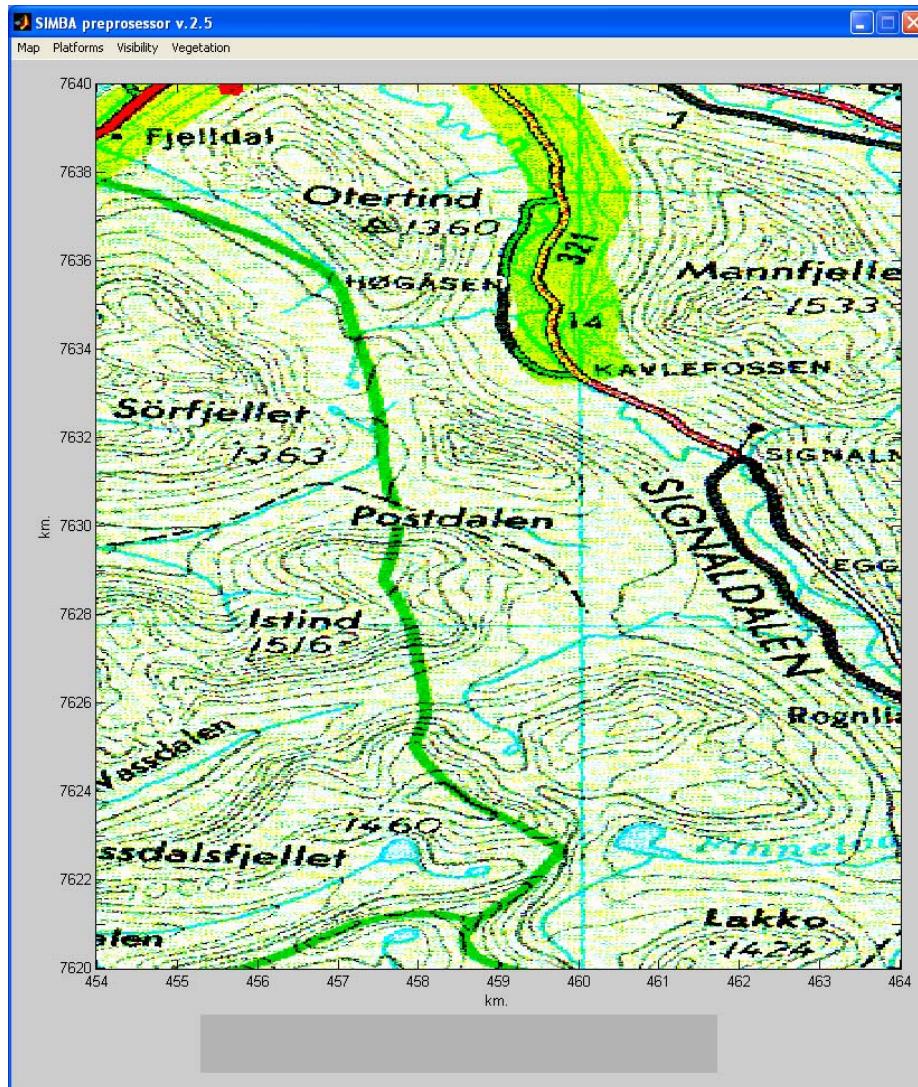
3.4 Innlasting av kart



Figur 3.1: Vinduet der kartfilen oppgis.

Innlasting av kart kan initieres på flere måter. Du kan velge "Load" fra "Map"-menyen, eller du kan umiddelbart etter oppstart klikke i kartområdet (klappe Simba). Uansett får du spørsmål om hvilket kart du ønsker å jobbe med, se Figur 3.1. Kartfilene antas å ligge under "map" i katalogstrukturen, og det er ikke nødvendig å ta med filendelsen. Kartbildet (hvis et slikt finnes) må være lagret på JPG-format. Når kartet skal leses inn, undersøkes det om det finnes en kartmatrise for det nevnte kart. Hvis dette finnes, går det fort og greit, hvis ikke tar det noe lenger tid. Grunnen til at dette nevnes her er at det ved gjenbruk av kartnavn er viktig at disse matrisefilene fjernes. SIMBA Preprocessor kontrollerer kun filnavnene, og vil således bruke en gammel matrise hvis et kart overskrives av et nytt.

Når kartet er lastet, vil man få frem et høydekoteplott i hovedvinduet. Hovedvinduet er vist i Figur 3.2. Her er det mange muligheter. Man kan nå bruke "Map"-menyen til å tilpasse kartet slik man ønsker det. Valget "Load" laster et nytt brukerspesifisert kart, og valget "Zoom" anbefales ikke brukt. Imellom de to finnes et valg ved navn "Show". Her kan man velge følgende:



Figur 3.2: Hovedvinduet i SIMBA preprocessor.

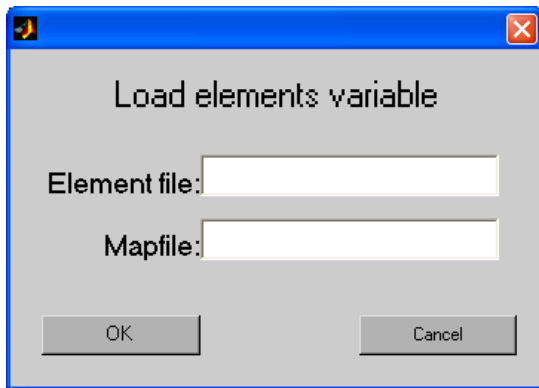
- "Contour plot", med eller uten vegetasjon: Det kartet man initielt får opp er "Contour plot" uten vegetasjon ("without veg"). Velger man å vise med vegetasjon, vil et eventuelt digitalkart bli liggende under koteplottet hvis dette var valgt fra før. Dette siste valget medfører altså bare en opptegning av koteplottet oppå det som måtte finnes i vinduet fra før. Dette valget fungerer dårlig for 3-dimensjonale plott, og brukes hovedsaklig når det digitale kartet er aktivt.
- "Digital map": Her kan man velge å vise et kartbilde i digitalisert form. Et scannet veikart er det vi oftest har brukt. Kilden spiller ingen rolle bare det er lagret på JPG-format. Det området som vises på kartet må selvsagt være samme område som konturplottet for at det skal gi mening.
- "Pcolor", "Surf", "SurfL" og "Mesh": Dette er standard Matlab 3D-plott. Basert på dataene fra høydekoteplottet kan man her få forskjellige 3-dimensjonale representasjoner av terrenget.

- "Change viewpoint": Dette valget anbefales ikke brukt på 2-dimensjonale plott. Det åpner en kontrollboks for å styre retningen plottet ses fra. Denne styremekanismen virker, men er neppe særlig nyttig, og den er i alle fall ikke spesielt rask.

En annen måte å få lastet et kart på er å gå via "Platforms" menyen, og velge "Load" der. Da vil et tidligere generert scenario lastes inn med kart og det hele. Fremdeles kan man leke med "Map" menyen hvis man ønsker. For detaljer rundt innlasting av scenario, se kapittel 3.5.

3.5 Innlasting av scenario

Hvis det finnes scenariofiler som tidligere er generert av SIMBA Preprocessor, kan disse enkelt lastes inn igjen ved å velge "load" fra "Platforms"-menyen. Her angir man navn på plattformfil og kartfil. Enten man ønsker å bruke dem som mal eller for viderearbeid er dette en god måte å spare tid på. Når kartet vises med tidligere innlagte plattformer, velges "Control" fra "Platforms"-menyen for å vise detaljer angående plattformene og deres ordre. Den typiske måten å ferdigstille et scenario på er å sette opp et forslag i Preprocessor, simulere dette i SIMBA Simulator og så sjekke hva som hendte i SIMBA Kartlogg. Deretter lastes scenariet på nytt inn i SIMBA Preprocessor, der ytterligere plattformer kan legges til og eventuelle feil rettes.



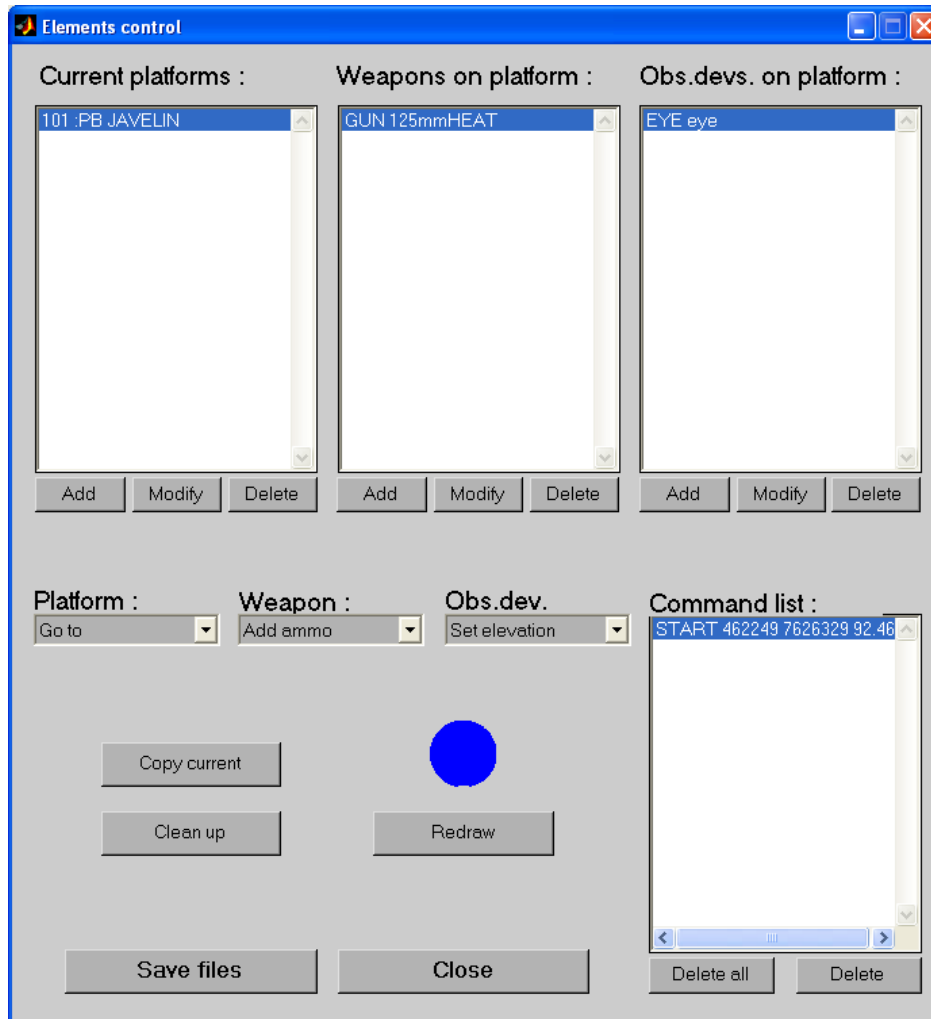
Figur 3.3: Vindu for innlasting av gamle scenariofiler.

3.6 Generering/Endring av scenario

For å generere et scenario fra scratch, må man laste inn et kart (på den ene eller den andre måten) og så velge "Control" fra "Platforms"-menyen. Her kan man sette opp plattformer og gi dem kommandoer. Angivelse av filnavn og slikt kommer ikke inn før man skal lagre. Endring av eksisterende innlastet scenario gjøres på samme måte, bortsett fra at det da finnes plattformer i plattformlisten allerede.

3.6.1 Opprette en plattform

De tre øverste listeboksene i "Elements Control"-vinduet, Figur 3.4), er beregnet på å spesifisere plattformer og deres våpen og siktemidler. Man velger først "Add" under plattformlisten. I vinduene som dukker opp angir man ID-nummer (første siffer angir side; 1 er blå og 2 er oransje) og type plattform, samt våpenets høyde og dekningsgrad i et eget vindu. Flere plattformer kan genereres før "New Element"-vinduet lukkes.



Figur 3.4: "Elements Control"-vinduet: Vindu for å legge til og endre plattformer, våpen og siktemidler, og for å gi ordre til plattformen.

Nå skal man ha en eller flere plattformer i plattformlisten. En av disse velges, og da vises denne plattformens tildelte våpen og siktemidler i de to listene til høyre. Disse vil på det nåværende tidspunkt være tomme. Ved å velge "Add" under våpenlisten får man opp "New Weapon" vinduet, der man kan velge blant registrerte våpen. Flere våpen kan tilordnes samme plattform før vinduet lukkes. For siktemidler gjelder det samme som for våpen.

Hvis det i "New Element"-vinduet velges "Indirect" som plattformstype, er det artilleri som skal opprettes. Ettersom artilleri i SIMBA ikke har noen posisjon, er heller ikke plattformens høyde eller dekningsgrad videre interessant. Derfor nyttes anledningen her til å legge inn artilleriets systematiske feil parallelt med og på tvers av skyteretningen. Videre vil det når det forsøkes opprettet et våpen av typen KonvArt eller SmartArt bli innsamlet koordinater for spredning. Koordinatene gis i forhold til skuddretning. Man kan altså generere ønsket siktemønster ved å opprette ett våpen for hvert antatte nedslagspunkt. Artilleri kan ikke tildeles siktemidler.

Da skulle plattformene være opprettet, men ingenting vises på kartet i hovedvinduet før plattformene tillegges ordre (artilleristillinger representeres på kartet i hovedvinduet som et

kryss for hvert rørs siktepunkt). De andre knappene under de tre øverste listene skulle forklare seg selv. Det nevnes bare at det selvsagt er det til enhver tid markerte element i listen over "Mod"/"Del"-knappene som påvirkes.

3.6.2 Innlegging av ordre

Under de tre øverste listene finnes det tre rullegardinmenyer som brukes for å spesifisere plattformenes oppførsel. I listen nederst til høyre vises kommandoene til plattformene, for én plattform av gangen. Aktiv plattform i plattformlisten øverst til venstre bestemmer hvilken plattforms kommandoer som vises. Det kan slettes én enkelt kommando (den markerte), eller alle kommandoer for angitt plattform ved hjelp av knappene under kommandolisten. For plattformer som ikke er artilleristillinger, SKAL første kommando være "Start". Derfor tillates ikke noe annet. Etter dette kan det legges inn det som måtte ønskes. Det anbefales å gi en "Stop"-kommando til slutt for alle plattformer, men SIMBA Preprocessor legger til denne ved lagring hvis den ikke finnes. For artilleri er det litt spesielt også her. Hver artillerienhet er forutsatt å skyte bare én gang. Selv om det er mulig å angi flere "Art"-kommandoer, så anbefales det ikke. Man er nødt til å gjøre en del antagelser i forhold til artilleriet i SIMBA, så det skulle ikke være mye ekstra arbeid å sette opp ett artilleri for hver serie (se "copy current" under kapittel 3.7).

De fleste kommandoene er greie å forstå, men noen trenger kanskje litt forklaring. Alle ordre er tidligere beskrevet i kapittel 2.3

- "Wait for shots" angir hvor mange skudd en plattform skal prøve å skyte før den stikker av. Brukes typisk for PB-stillinger som står i ro og venter på action.
- Alle kommandoer med "flag" i navnet har med kommandoflyt å gjøre. En plattform (eller flere) kan settes til å vente på at en hendelse inntreffer.
- "Look at point" er med i kommandorullegardinene, men kan ikke angis i kommandolisten. Dette er en kommando som plattformene selv initierer når noen med samme to første siffer i ID-nummeret oppdager en fiendtlig plattform eller skuddsignatur fra denne. Ordren er med i rullegardinene kun med tanke på senere implementasjon.
- "Edit Start" er ikke en kommando, men en mulighet til å endre gjeldende "Start" kommando for den aktuelle plattform uten å måtte slette hele listen.
- For artilleri er det kommandoen "Art" som angir når det startes å skyte.
- Forskjellen på "Go to" og "Move to" er at for den sistnevnte flyttes plattformen med angitt fart mot punktet som angis, mens det for den førstnevnte er snakk om en instantan flytting av en plattform til ny posisjon.

3.6.3 Lagre scenario

Nederst til venstre i "Elements Control"-vinduet finnes en knapp med påskriften "Save files". Ikke uventet er det her man trykker for å lagre det arbeidet man har gjort. Man får da opp et vindu som ber om navn på filsettet som skal genereres. I tillegg kan det velges hvorvidt det skal opprettes en ny scenariofil. Dette valget er på som "default", og vil typisk kun slås av hvis man holder på å editere et tidligere lagret scenario, og lagringsprosessen er i så fall

ferdig. Ellers får man opp et nytt vindu, Figur 3.5, der man må spesifisere noen data for styring av simuleringene. Fra toppen (venstre til høyre) er feltene:

- Katalog for loggfiler fra SIMBA Simulator.
- Antall simuleringer som ønskes kjørt i SIMBA Simulator.
- Tidsavstanden mellom simuleringstegene.
- Simuleringens starttid.
- Simuleringens stopptid.



Figur 3.5: Vindu for spesifisering av data ved lagring av scenario.

Deretter velges "Create file" når man er fornøyd. Da får man spørsmål om logsettings. Dette er ikke noe annet enn en godt kamouflert måte å spørre brukeren om hvor mye informasjon om simuleringen SIMBA Simulator skal ta med i sine loggfiler. For spesifisering av logfilkategorier, se kapittel 2.4.2. Normalt bør det angis "default" her (bruker alle kategorier) før man trykker på "Use existing". Skriver man et filnavn og velger "Create new" kan man lage sin egen, med kun de kategorier man ønsker. Dette er et valg for avanserte brukere. En slik fil vil senere kunne gjenbrukes ved å angi navnet og så velge "Use existing".

3.7 Andre knapper i "Elements Control"-vinduet

"Copy current" er en knapp for å forenkle innlegging av lignende plattformer. Trykker man på denne, vil det opprettes en kopi av den plattformen som er aktiv i plattformlisten. Denne plattformen kopieres med ID-nummer, våpen, siktemidler og kommandoer. Selv om det er mulig i SIMBA Preprocessor å ha flere plattformer med samme ID-nummer, er det ikke noe SIMBA Simulator ser med på vennlige øyne. ID-nummeret må derfor endres manuelt før lagring. Normalt vil man også gjøre noen mindre endringer i kommandolisten, men dette er helt opp til brukeren.

"Clean up" brukes for å rydde kartet. Kartet tegnes da opp på nytt som et høydekoteplott, og alle plattformer tegnes inn, men nå bare i siste kjente posisjon (i følge kommandolisten).

”Redraw” brukes også for å rydde, men nå blir kun symbolene (ikke selve kartet) fjernet og tegnet på nytt. Her tegnes de i alle kjente posisjoner (i følge kommandolisten). Dette kan med hell gjøres etter at man for eksempel har lagt til en ”Move to”-kommando midt i en eksisterende liste med bevegelser.

”Close” bare lukker vinduet. Ingenting lagres, og ingenting slettes. Åpner man ”Elements Control” vinduet igjen uten at MatLab er avsluttet først, vil alle data fremdeles være der.

3.8 Siktbarhetsundersøkelser

Som tidligere nevnt kan SIMBA Preprocessor brukes til å undersøke siktbarhet i terrenget. Dette kan gjøres under byggingen av et scenario, eller på helt frittstående grunnlag. Det eneste som kreves er at en høydematrise er lastet inn, men det blir kanskje lettere å orientere seg hvis man også laster opp et digitalt kart. Menyene ”Visibility” i hovedvinduet brukes til dette. For alle valgene på denne menyen vil en eventuell vegetasjonshøydematrise tas med i betraktningen hvis en slik finnes.

”Profile plot” brukes for å tegne en profil av landskapet fra et punkt til et annet. Over store avstander blir dette noe unøyaktig for det blotte øye.

”In sector” brukes for å undersøke hva man kan se fra et gitt punkt (med en gitt høyde over bakken) i en gitt sektor. Resultatet vises som gul skravering på kartet i hovedvinduet.

”Along line” brukes til å undersøke om man fra en gitt høyde over et punkt kan se noe med en gitt høyde over et annet punkt. Dette er i praksis et profilplott med innlagt ”line of sight”-linje, samt at det gjøres en utregning på om det er sikt eller ikke.

3.9 Innlegging av siktbegrensninger

I mangel av gode data for vegetasjonshøyde trenger man en måte å blokkere sikt på. Man kan for eksempel ved befaring ha funnet at det ikke er fri sikt i et kritisk område der det etter høydekoteplottet skulle være det. Da kan man legge inn en vegetasjonshekk. En slik hekk kan gis den høyde man måtte ønske, slik at man i teorien kan se over den hvis det skulle være ønskelig. Til dette brukes ”Vegetation”-menyen. Denne menyen inneholder bare ett valg, så vi satser på at det er ”Make” som skal benyttes. Man får da mulighet for å angi fra- og tilpunkter på kartet i hovedvinduet, samt angi hekkens høyde over terrenget (i cm). For å fjerne en hekk, må man slette filen med opplysningene (den såkalte .wall-filen). Hekkene lagres ikke før scenariet lagres, så inntil det er gjort, kan man i teorien bare starte SIMBA Preprocessor.

4 BRUKERVEILEDNING FOR SIMBA KARTLOGG

4.1 Generelt

SIMBA KartLogg gir brukere av SIMBA mulighet til å vise grafisk hva som foregikk under en enkelt simulering. SIMBA KartLogg viser plattformene som svarte sirkler fylt med brigadens farge, og artillerinedslagsfelt som tomme sirkler i brigadens farge. Skudd vises som fargede streker mellom de aktuelle plattformene. I tillegg kan SIMBA KartLogg vise hva hver enkelt plattform ser, i form av gule og lilla siktsektorer. For å bruke SIMBA KartLogg trengs

plattform-, scenario- og kommandofilen som var inn-data til SIMBA, samt loggfilen som SIMBA genererte. I tillegg trengs et kart over området (2D, Windows bitmap-format), og hvis det finnes en "wall-fil" (siktbegrensningsfil), vil begrensningene i denne vises som grønne linjer på kartet. Programmet brukes for eksempel til å verifisere at plattformenes oppførsel harmonerer med det som var hensikten da man genererte simuleringen, og således sikre at utdataene er pålitelige. Programmet begrenser selv vindusstørrelsen (vertikalt) til gjeldende skjermoppløsning, dersom denne er mindre enn maksimumsstørrelsen på programmet.

4.1.1 Kravspesifikasjon

SIMBA KartLogg kan kjøres på alle Win32-plattformer (95, 98, NT og WinCenter). Ettersom SIMBA KartLogg selv krymper vindushøyden til å passe på skjermen, anbefales 1024x768 som minimum oppløsning, og fargedybden bør minimum tilsvare fargedybden i kartet som brukes. Fargedybden i kartet bestemmes når man skanner det (eller genererer det på annen måte). Fargedybden for Windows bitmap filer er normalt 256 farger (8 bit). Programmet bruker kun 5 siffer til å angi posisjon. Derfor må kartet som brukes ikke inneholde linjer der posisjonens 6. minst signifikante siffer endres.

4.2 Hovedvindu

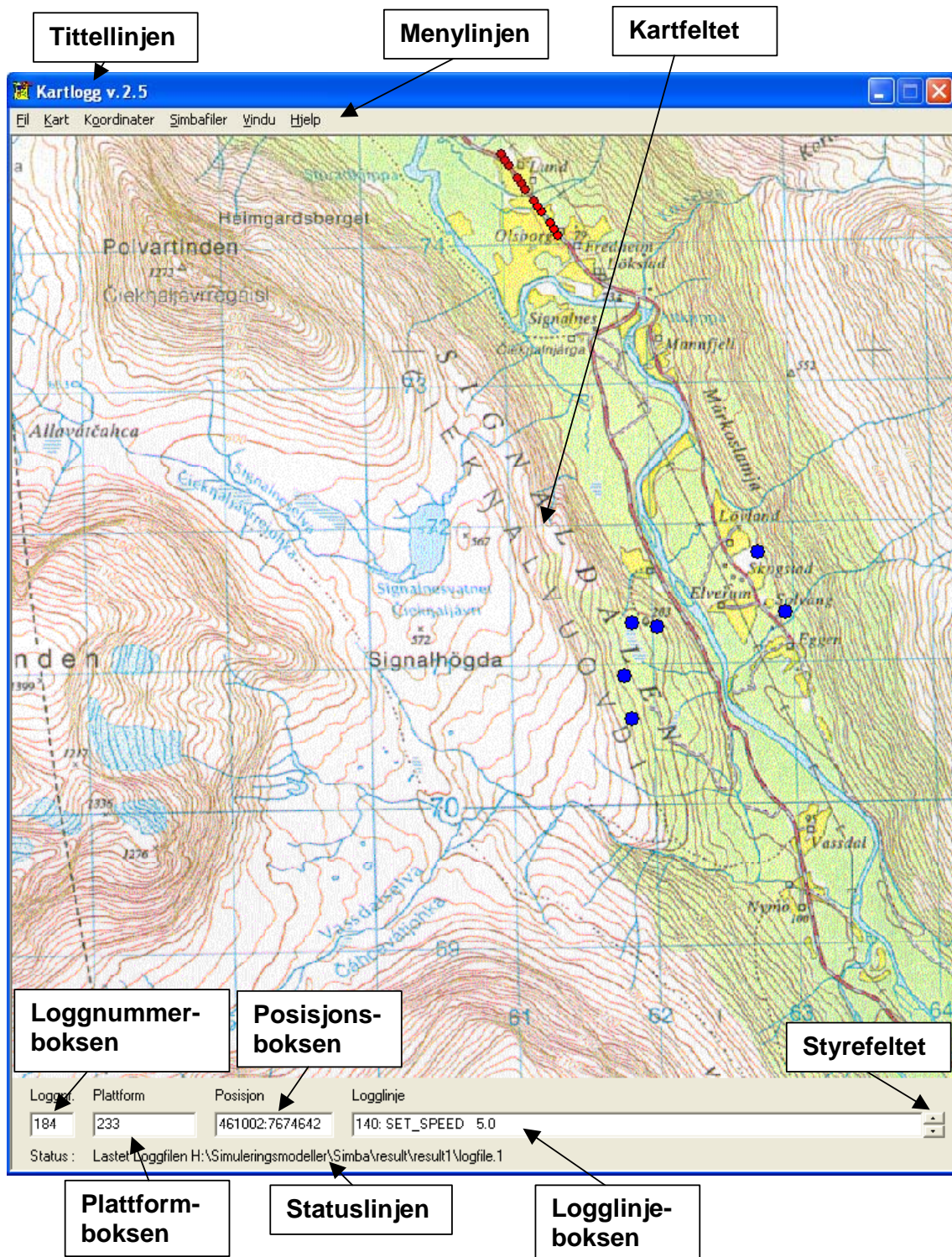
Hovedvinduet er vist i Figur 4.1. For alle beskrivelser i kapittel 4.2 henvises det til denne figuren. Hovedvinduet består av en tittellinje (4.2.1), en menylinje (4.2.2), et kartfelt (4.2.3), en loggnummerboks (4.2.4), en plattformboks (4.2.5), en posisjonsboks (4.2.6), en logglinjeboks (4.2.7), et styrefelt (4.2.8), og en statuslinje (4.2.9).

4.2.1 Tittellinjen

Tittellinjen viser programnavnet og versjonsnummeret. I tillegg finnes de vanlige 4 knappene her; Systemmenyen i venstre kant, og minimerings-, maksimerings- og avsluttknappen i høyre kant (ramset opp fra venstre). Skulle brukeren komme til å dobbeltklikke på tittellinjen vil hovedvinduet (4.2) vokse til å dekke hele skjermen. Dette er ikke hensiktsmessig i SIMBA KartLogg, derfor gjøres brukeren herved oppmerksom på at et nytt dobbelklikk vil tilbake stille vinduet til opprinnelig størrelse.

4.2.2 Menylinjen

Menylinjen ligger like under tittellinjen (4.2.1), som den alltid gjør i Windows, og har 6 kategorier. Kategoriene og deres innhold beskrives under.



Figur 4.1 Hovedvinduet i SIMBA Kartlogg.

4.2.2.1 Filmenyen

Valgene her har med standardoppsett og programavslutning å gjøre. For forklaring til hvert enkelt menyvalg, se under.

Fil/Kjøre standard oppsett...

Åpner Windows sitt standardvindu for valg av fil. Filen kan ligge hvor som helst, og det navigeres i filsystemet på vanlig måte. Standardoppsettsfiler inneholder navn (inkludert sti) på kartfil, koordinatfil, scenariofil og loggfil. Ingenting annet. For informasjon om koordinatfiler, se forklaring av menykategorien koordinater (4.2.2.3). Standardoppsettsfiler har endelse ".dsf". Dette er en forkortelse for det engelske "default setup file".

Fil/Lagre standard oppsett...

Åpner Windows sitt standardvindu for valg av filnavn for lagring. Stien angis ved å navigere i filsystemet på vanlig måte, og filnavnet gis automatisk endelsen ".dsf". Standardoppsettsfilen kan lagres hvor som helst, men den bør enten gis et selvforklarende navn eller lagres sammen med logg- og scenariofilen som standardoppsettsfilen skal laste. Det bør genereres en koordinatfil (4.2.2.3) før lagring av standardoppsettsfil.

Fil/Avslutt

Avslutter programmet uten å lagre noen ting.

4.2.2.2 Kartmenyen

Inneholder valg angående manuell lasting av kart. Foreløpig finnes det kun ett valg i denne menyen.

Kart/Åpne kartfil...

Igjen er det Windows sitt standardvindu for valg av fil som dukker opp. Det navigeres som vanlig. Kartfiler har normalt endelsen ".bmp", og derfor er det i utgangspunktet kun slike filer som vises. Manuelt valg av kartfil anses som alternativ til å laste standardoppsett. Programmet forventer derfor å få koordinater (4.2.2.3), scenariofil og loggfil (4.2.2.4) angitt manuelt etter bruk av dette menyvalget, før det startes avspilling (4.2.8) av loggfilens innhold.

4.2.2.3 Koordinatmenyen

Menyen består av 3 valg, som alle omhandler koordinater på en eller annen måte. To for filhåndtering og ett for manuell spesifisering av koordinater.

Koordinater/Åpne koordinatfil...

Nok en gang kommer Windows sitt standardvindu for filvalg opp på skjermen. Ettersom koordinatfiler normalt har endelsen ".krd" vises kun slike filer i utgangspunktet. Det må lastes kartfil (4.2.2.2) før koordinatfil, ellers vises en feilmelding på statuslinjen (4.2.9).

Koordinater/Lagre koordinatfil...

Windows sitt standardvindu kommer opp og ber om filnavn og plassering. Det anbefales å lagre koordinatfilen sammen med kartfilen koordinatene gjelder.

Koordinater/Ta ut koordinater...

Når dette menyvalget brukes, er det viktig å følge med på statuslinjen. Der kommer det informasjon om hva som skal gjøres. Kort fortalt brukes musen til å klikke på kartet (4.2.3) i et punkt der man kjenner den virkelige posisjonen. Programmet finner skjermkoordinatene og åpner et koordinatvindu (4.3) der brukeren taster inn den virkelige posisjonen. Dette gjentas så for et annet punkt, og basert på disse to punktene regner programmet ut alle posisjoner

senere i kjøringen. Koordinatene som tas ut på denne måten, anbefales lagret i en koordinatfil, slik at denne prosessen ikke må gjennomgås hver gang det aktuelle kartet skal brukes. Videre anbefales det å ta ut punkter som ligger så langt fra hverandre som mulig, da dette vil minske feilen som skyldes at man ikke treffer nøyaktig på det punktet man ønsker med musa.

4.2.2.4 Simbafilmenyen

Ved hjelp av valgene på denne menyen kan brukeren manuelt velge scenariofil og loggfil. Valgene på denne menyen kan brukes også etter lesing av standardoppsettsfil (4.2.2.1), og det er da opp til brukeren å se til at det er riktig kart (4.2.2.2) som er lastet.

Simbafilmer/Åpne Scenariofil...

Windows sitt standardvindu for filvalg dukker opp igjen. Ettersom scenariofiler ikke har noen bestemt endelse, vises alle filer. Det blir således opp til brukeren å sikre at scenariofilen omhandler det området som vises på kartet (4.2.2.2) i kartfeltet (4.2.3). Scenariofiler inneholder som kjent navn på tilhørende plattformfil, kommandofil og såkalt wallfil. Den siste trenger ikke være tilstede, men de to første må finnes. Hvis viktige filer ikke finnes, vil det vises en feilmelding på statuslinjen (4.2.9). Hvis alt går etter planen, vil statuslinjen be om en loggfil.

Simbafilmer/Åpne Loggfil...

Brukeren får velge fil i Windows sitt standardvindu. Alle filer vises, så det er opp til brukeren å sørge for at filen som spesifiseres er en gyldig loggfil generert på bakgrunn av den angitte scenariofil. Når loggfilen er hentet inn, kan man starte avspilling (4.2.8) ved bruk av piltaster eller mus. Avspillingsmodus (4.4.1) kan velges i hendelsesvinduet (4.4).

4.2.2.5 Vindumenyen

I denne menyen er alt som styrer fremvisning på skjerm samlet. Hvert av menyvalgene er beskrevet under.

Vindu/Hendelser...

Dette menyvalget åpner hendelsesvinduet (4.4) for valg av avspillingsmodus (4.4.1) og spesifisering av hendelseskategorier (4.4.2).

Vindu/Siktsektorer

Dette menyvalget er en av-/på-funksjon. Hvis valget er på, vises en hake foran teksten. Siktsektorene vises ut til maksimal rekkevidde, men plattformens faktiske sikt vil være begrenset av eventuelle vegger (spesifisert i wallfil) markert med grønne streker på kartet (4.2.3), samt terreng og eventuell vegetasjon dersom det finnes en vegetasjonsfil. For å se siktsektorer må det i tillegg spesifiseres for den enkelte plattform (4.5.2.1) i plattformvinduet (4.5). Det vises ikke siktsektor for utslåtte plattformer.

Vindu/Plattformer...

Dette valget åpner plattformvinduet (4.5), der brukeren kan angi siktsektor- (4.5.2.1), kortinfo- (4.5.2.3), fullinfo- (4.5.2.2) og ID-nummerskjuling (4.5.2.4) for hver enkelt plattform. Årsaken til at man kan ønske å skjule ID-numrene, er at de kan hindre brukeren i å se plattformenes posisjon. Bare de valgene som er relevante for den aktuelle plattform, kan settes. I tillegg kan man i plattformvinduet velge hvilken plattform som skal vises øverst (4.5.3). Dette kan være lurt hvis man lurer på hvor en bestemt plattform er.

Vindu/Kortinfo vindu

Dette menyvalget åpner kortinfovinduet (4.6). I likhet med menyvalget Vindu/Siktsektorer er dette et av-/på-valg. Hvis det er en hake foran valget, er vinduet på og skal dekke den delen av skjermen som ikke dekkes av hovedvinduet (4.2). Kortinfovinduet kan flyttes etter behag, men det vil alltid vises foran hovedvinduet, slik at man ikke mister det, så det står nok best der det står opprinnelig. Kortinfovinduet kan bare lukkes fra menyen. Hvilke plattformer som skal vises i kortinfovinduet styres fra plattformvinduet (4.5).

Vindu/Oppdater

Tegner kartfeltet (4.2.3) på nytt. Det hender at det blir noe krøll på skjermen hvis vinduer lukkes, åpnes eller dras over hverandre. Skulle det bli noe galt med fremvisningen, velges dette alternativet, og kartfeltet slettes før det tegnes opp på nytt. Fortsetter problemet må SIMBA KartLogg avsluttes for så å startes på nytt.

4.2.3 Kartfeltet

Kartfeltet er der ting skjer. I bunnen vises et brukerspesifisert kart, og oppå der vises alle plattformene som var med i simuleringen. Eller nesten alle da. Artilleriplattformer vises ikke. For dem markeres det kun skudd, og dette gjøres med fargede sirkler. I tillegg vises skudd og siktsektorer (4.2.2.5) for alle andre plattformer. Kartfeltet brukes også i forbindelse med uttak av koordinater (4.2.2.3). Klikker man på kartfeltet ellers, vises terrengposisjon på statuslinjen (4.2.9). Posisjonen oppgis med 5 siffer i hver retning. Hvis skjermopløsningen er 1024x768 eller mindre, vil loggnummerboksen (4.2.4), plattformboksen (4.2.5), posisjonsboksen (4.2.6), logglinjeboksen (4.2.7), styrefeltet (4.2.8) og statuslinjen vises oppå kartfeltet, og kartet klippes i nedre kant slik at det passer i kartfeltet.

4.2.4 Loggnummerboksen

Her vises den aktive logglinjens linjenummer. Linjenummeret settes internt og samsvarer ikke uten videre med linjenummer i loggfilen. Kun hvis det i hendelsesvinduet (4.4) ikke er lagt noen begrensninger på hvilke hendelser som skal vises (4.4.2), vil det interne linjenummeret og linjenummeret i loggfilen være det samme. Dette feltet brukes for å se hvor langt man har kommet i avspillingen (forutsetter at man kjenner det totale antall linjer) eller til å finne igjen bestemte hendelser.

4.2.5 Plattformboksen

Dette feltet forteller hvilken plattform den aktive logglinjen (4.2.7) gjelder. Hvis to plattformer er involvert, viser dette feltet den plattformen som er den aktive. Hvis for eksempel plattform 203 skyter på plattform 106, viser plattformboksen plattform 203.

4.2.6 Posisjonsboksen

I dette feltet vises (hvis relevant) den aktive plattformens posisjon i det den aktive logglinjen ble skrevet fra SIMBA. Aktiv plattform finnes i plattformboksen (4.2.5), og aktiv logglinje finnes i logglinjeboksen (4.2.7).

4.2.7 Logglinjeboksen

Her skrives den aktive logglinjen. Som en del av teksten i logglinjeboksen vises tiden i sekunder siden simuleringen startet. Hvis man tar en titt i loggfilen, vil man fort se at linjene der inneholder mer informasjon enn det som står i logglinjeboksen. Dette skyldes at linjene deles opp og skrives ut på forskjellige steder. For eksempel skrives posisjonen (hvis den

finnes) i posisjonsboksen (4.2.6). All informasjon fra linjene i loggfilen kan finnes igjen i hovedvinduet, så sant hendelseskategorien (4.4.2) linjen tilhører ikke er valgt bort i hendelsesvinduet (4.4). I så fall vil linjen ikke vises i det hele tatt.

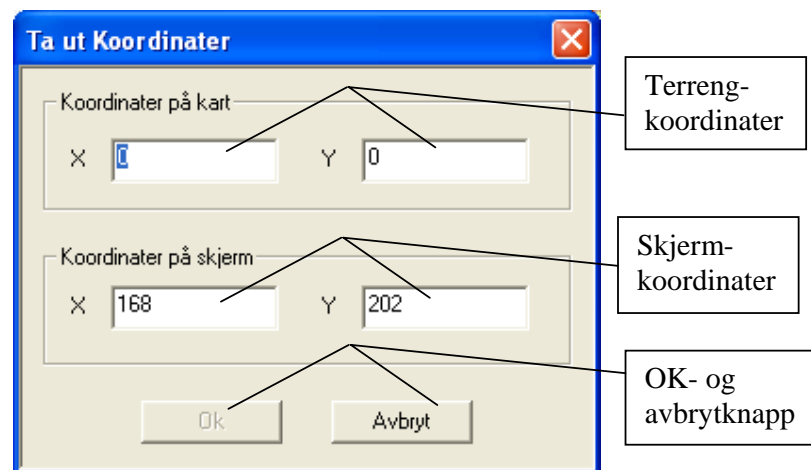
4.2.8 Styrefeltet

Styrefeltet er de to pilene til høyre for logglinjeboksen (4.2.7). Klikk på opp-pilen spiller fremover i loggfilen, og klikk på ned-pilen spiller bakover i loggfilen. Dette feltet kan også opereres ved hjelp av piltastene på tastaturet så lenge hovedvinduet (4.2) er aktivt. Skulle hovedvinduet miste fokus (bli inaktivt), må vinduet aktiveres før piltastene kan brukes igjen. De fleste vinduer som åpnes fra hovedvinduet krever lukking før avspilling kan fortsette. Vinduer som kan være åpne under avspilling, omfatter kortinfovinduet (4.6) og Windows sine standardvinduer for filvalg. Førstnevnte tar ikke fokus fra hovedvinduet med mindre brukeren klikker på det, og sistnevnte vil normalt lukkes av brukeren før avspilling ønskes fortsatt. Når de sistnevnte lukkes, returneres fokus automatisk til hovedvinduet. Når styrefeltet opereres endres innholdet i loggnummerboksen (4.2.4), plattformboksen (4.2.5), posisjonsboksen (4.2.6) og logglinjeboksen (4.2.7), samt at innholdet i kartfeltet (4.2.3) endres slik at det gjenspeiler situasjonen på det tidspunkt den aktive logglinjen ble skrevet. Informasjon om forskjellige avspillingsmodi (4.4.1) kommer i kapitlet om hendelsesvinduet (4.4).

4.2.9 Statuslinjen

Statuslinjen viser feilmeldinger, bekreftelser på utførte handlinger, i enkelte tilfeller beskjed om hva som forventes utført av brukeren, og koordinater for det angitte punkt ved klikk i kartfeltet (4.2.3). Eventuelle feilmeldinger kommer i klartekst, så noen oversettelsestabell for feilmeldinger er ikke nødvendig. Virker feilmeldingene kryptiske, vil man som regel kunne tolke dem ved å lese kapitlet som omhandler den aksjonen som avstedkom feilmeldingen. I de fleste tilfeller er begrensinger som kan resultere i feilmeldinger beskrevet der.

4.3 Koordinatvindu



Figur 4.2 Koordinatvinduet

Koordinatvinduet (Figur 4.2) består av fire innfyllingsfelder og to knapper. Vinduet åpnes to ganger ved uttak av koordinater. Først velges "Koordinater/Ta ut koordinater..." (4.2.2.3), så velges et punkt på kartet, og da først kommer koordinatvinduet frem. De to nederste

innfyllingsfeltene er da allerede satt til skjermkoordinatene for punktet som brukeren klikket på, og de to øverste feltene må fylles inn med kartkoordinatene (altså punktets terrengkoordinater). Først når de to øverste feltene er fylt inn, blir OK-knappen aktiv. Avbryt-knappen er aktiv hele tiden og brukes hvis man ikke ønsker å fortsette uttak av koordinater. Innholdet i de to nederste innfyllingsfeltene kan også endres manuelt, selv om det er litt vanskelig å se at det skulle være behov for det. Terrengkoordinatene må spesifiseres, selv om de skal være lik det som står i feltene når vinduet åpnes. Når OK-knappen trykkes, forventer programmet klikk på punkt nummer to (dette står da også på statuslinjen), og når klikket foreligger, åpnes koordinatvinduet på nytt for inntasting av terrengkoordinater for dette punktet.

4.4 Hendelsesvinduet



Figur 4.3 Hendelsesvinduet

Hendelsesvinduet (Figur 4.3) består av to samlinger med hakebokser og en OK-knapp. Den øverste samlingen med hakebokser spesifiserer avspillingsmodus (4.4.1), og den nederste angir hvilke hendelser utover basis som skal tas med i avspillingen (4.4.2). OK-knappen brukes for å lukke vinduet med de valg som er foretatt. Her finnes ingen avbryt-knapp, fordi det er så få muligheter at gjeldende konfigurasjon lett lar seg gjenopprette manuelt.

4.4.1 Avspillingsmodus

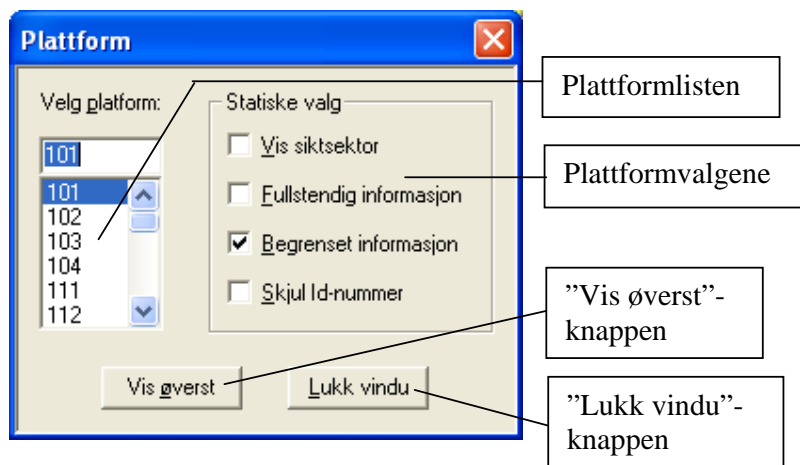
Den øverste samlingen med hakebokser i hendelsesvinduet (4.4) styrer hvilken avspillingsmodus som er aktiv. Hvis ”Vis bare første hendelse i hvert sekund” er haket av, vil avspillingen hoppe til første logglinje med tid ulik den i den aktive logglinjen (4.2.7). Spiller man bakover, får man således siste logglinje i forrige sekund der det finnes logglinjer, og spiller man fremover, får man første logglinje i neste sekund der det finnes logglinjer. De sekundene som hoppes over, finnes det ikke noen logglinjer for. Hvis valget er av (blank hakeboks), vises alle logglinjene som ble lastet inn. Retning på avspilling styres via styrefeltet

(4.2.8). Automatisk avspilling er ikke implementert. Derfor er hakeboksen for dette valget ikke mulig å slå på. Avspillingsmodus endres umiddelbart når vinduet lukkes.

4.4.2 Hendelseskategorier

Den nederste samlingen med hakebokser i hendelsesvinduet spesifiserer hvilke hendelser som skal tas med fra loggfilen. Hendelsene er delt inn i kategorier som spesifisert i 2.4.2. Basiskategorien inneholder alle de linjene som må være med for at SIMBA KartLogg skal få nok informasjon til å vise simuleringen i reprise. Denne kategorien kan derfor ikke velges bort. Alle de andre kan velges bort enkeltvis, og endringer i denne delen av hendelsesvinduet vil først gi effekt når neste loggfil leses inn i SIMBA KartLogg ved hjelp av menyvalgene "Simbafiler/Åpne Loggfil..." (4.2.2.4) eller "Fil/Kjøre standard oppsett..." (4.2.2.1) i hovedvinduet (4.2).

4.5 Plattformvinduet



Figur 4.4 Plattformvinduet

I plattformvinduet (Figur 4.4) spesifiseres hva som skal vises for den enkelte plattform. Vinduet består av en liste over alle plattformene (4.5.1), en samling hakebokser (4.5.2), en "Vis øverst"-knapp (4.5.3) og en "lukk vindu"-knapp (4.5.4). De fleste endringer som gjøres i dette vinduet blir implementert først når aktiv logglinje endres. Dette vinduet inneholder ingen "avbryt"-knapp. Endringer som utføres blir således aktivisert med mindre brukeren manuelt omgjør dem før vinduet lukkes.

4.5.1 Plattformlisten

Plattformlisten er altså feltet til venstre i plattformvinduet (4.5). Øverst finnes et felt som er tomt (til brukeren velger en plattform), og under finnes en liste over alle plattformene som finnes i simuleringen. Også plattformer som ikke sees på skjermen, er med i denne listen. Hvis listen blir lengre enn plattformlistens fysiske størrelse, aktiveres piler i høyre kant av listen som tillater blaing opp og ned i listen. For å velge en plattform brukes musen (klikk på ønsket plattform) eller piltastene (hvis listen er aktiv). I denne listen kan man (når den er aktiv) faktisk også bruke tastene "page up" og "page down". Det kan være greit hvis listen blir virkelig lang. Plattformene i listen er ordnet i nummerrekkefølge.

4.5.2 Plattformvalgene

Fire valg finnes for hver plattform. Noen valg kan ikke gjøres for alle plattformer. For eksempel kan siktsektor (4.5.2.1) ikke vises for plattformer av typen artilleri. Logisk nok. Valgene som finnes er beskrevet under. Alle tekstene gjenspeiler aktiv status når det finnes en hake i boksen de tilhører.

4.5.2.1 Vis siktsektor

Bestemmer om siktsektor skal vises for den valgte plattformen. Selv om dette valget aktiveres, vil siktsektoren ikke vises hvis siktsektorer er slått av i vindumenyen (4.2.2.5). Fra start er dette valget slått av for alle plattformer, så man må manuelt slå det på før noen siktsektorer vises.

4.5.2.2 Fullstendig informasjon

Denne funksjonaliteten er ikke implementert ennå. Spesifisering av dette valget vil derfor ikke ha noen effekt. Når det blir implementert, vil dette valget åpne et nytt vindu med fullstendig oversikt over den valgte plattforms status og historie. Standard av.

4.5.2.3 Begrenset informasjon

Hvis plattformen skal vises i kortinfovinduet (4.6), spesifiseres dette valget. Resultatet av dette valget avhenger av forrige status. Hvis status går fra på til av, vil effekten vises ved at plattformen forsvinner fra kortinfovinduet så snart det velges en ny plattform, eller plattformvinduet (4.5) lukkes. Hvis derimot status går fra av til på, vil plattformen først dukke opp i kortinfovinduet når den trenger å tegnes på nytt, det være seg fordi den flytter seg eller hva som helst annet. Valget er på ved oppstart.

4.5.2.4 Skjul Id-nummer

Alle plattformene har i utgangspunktet dette valget av, og det handler om det nummeret som vises over og litt til venstre for plattformsirkelen i kartfeltet (4.2.3). Det har svart skrift på hvit bunn og kan i enkelte tilfeller skjule plattformer som ligger veldig nært hverandre. Da er det altså mulig å fjerne det.

4.5.3 Vis øverst

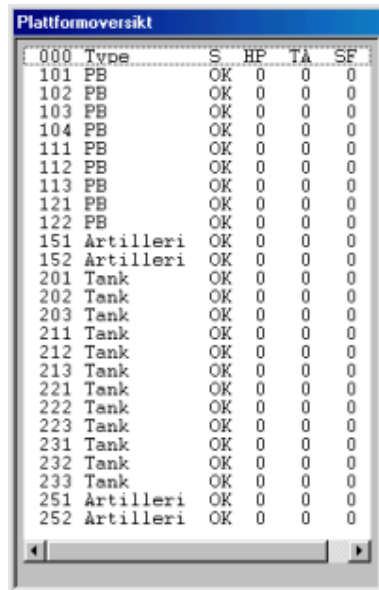
Knappen nederst til venstre er merket med ”Vis øverst”. Klikk på denne knappen medfører da også at den plattformen som er valgt, vises øverst neste gang plattformen tegnes på skjermen. Knappen er ikke aktiv før en plattform er valgt, og alle plattformer kan gis denne kommandoen. For plattformer som representeres ved nedslagsfelt vil det være nedslagsfeltet som da tegnes øverst.

Når en plattform gis kommandoen ”Vis øverst”, er det rimelig klart hva som skjer. Men hva skjer hvis en ny plattform får den samme kommandoen etterpå? For eksempel, plattform 101, 102 og 103 ligger tett inntil hverandre, slik at de tegnes delvis oppå hverandre. 101 øverst, 102 i midten og 103 underst. Hvis nå 103 får kommandoen ”Vis øverst”, vil rekkefølgen fra toppen bli 103, 101 og 102. Nå gir vi plattform 102 kommandoen ”Vis øverst”, og da blir rekkefølgen (fremdeles fra toppen) 102, 103 og 101. På denne måten kan brukeren altså sette visningsrekkefølgen for samtlige plattformer, ikke bare den som vises øverst.

4.5.4 Lukk vindu

Denne knappen befinner seg til høyre for ”Vis øverst”-knappen (4.5.3) og lukker plattformvinduet (4.5). Den sørger også for å utføre endringene som ble gjort for den siste plattformen i vinduet. Endringene gjort for andre plattformer på et tidligere tidspunkt, ble utført da neste plattform ble valgt.

4.6 Kortinfovinduet



ID	Type	S	HP	TA	SF
000	Type	S	HP	TA	SF
101	PB	OK	0	0	0
102	PB	OK	0	0	0
103	PB	OK	0	0	0
104	PB	OK	0	0	0
111	PB	OK	0	0	0
112	PB	OK	0	0	0
113	PB	OK	0	0	0
121	PB	OK	0	0	0
122	PB	OK	0	0	0
151	Artilleri	OK	0	0	0
152	Artilleri	OK	0	0	0
201	Tank	OK	0	0	0
202	Tank	OK	0	0	0
203	Tank	OK	0	0	0
211	Tank	OK	0	0	0
212	Tank	OK	0	0	0
213	Tank	OK	0	0	0
221	Tank	OK	0	0	0
222	Tank	OK	0	0	0
223	Tank	OK	0	0	0
231	Tank	OK	0	0	0
232	Tank	OK	0	0	0
233	Tank	OK	0	0	0
251	Artilleri	OK	0	0	0
252	Artilleri	OK	0	0	0

Figur 4.5 Kortinfovinduet

Kortinfovinduet (Figur 4.5) åpnes og lukkes utelukkende fra vindumenyen (4.2.2.5), og viser et antall plattformer og litt informasjon om dem. Informasjonen som vises er (i rekkefølge fra venstre til høyre) ID-nummer, plattformtype, status, poeng, tildelt ammunisjon og avfyrte skudd. Den øverste linjen hjelper til med å huske hva kolonnene står for. De to første kolonnene er selvforklarende, og de plattformtyper som eksisterer i skrivende stund er Tank, PB og artilleri. At det står ”000” i første kolonne på øverste linje har med sorteringen å gjøre. Plattformene er nemlig sortert etter ID-nummer i stigende rekkefølge. Kolonne 3 har overskriften S for status (eng. Status) og kan inneha verdiene OK (ikke skadet), MK (bevegelsesudyktig, eng. mobilitykilled), FK (skuddudyktig, eng. firepowerkilled) og TK (MK **U** FK, eng. totalkilled)¹. En plattform som har status TK, regnes som helt utslått. Kolonne 4 har overskrift HP. HP står for treffpoeng (eng. hit points), og plattformene tildeles 1 poeng for hver MK og FK de påfører motstanderne. Hvis en blå plattform treffer en oransje plattform som allerede var FK, og skader den slik at den blir TK, får den blå plattformen bare 1 poeng. Kolonne 5 har overskriften TA, som står for tildelt ammunisjon (eng. total ammo). Kolonnen viser hvor mye ammunisjon plattformene har tilgang til. Kolonne 6 har overskriften SF, som står for skudd avfyrte (eng. shots fired), og gjenspeiler selvsagt hvor mange skudd hver plattform har avfyrte. Hvilke plattformer som skal vises i kortinfovinduet, bestemmes av brukeren i plattformvinduet (4.5).

¹ MK, FK og TK tilsvarer henholdsvis Mkill, Fkill og Tkill i kapittel 2

5 BRUKERVEILEDNING FOR SIMBA POSTPROSESSOR

SIMBA PostProcessor er et program utviklet med tanke på prosessering av SIMBA resultatdata. Én eller flere resultatfiler (loggfiler) fra kjøring med likt oppsett (samme plattformer og samme dreiebok etc) brukes som inndata, og resultater av postprosesseringen kan skrives til skjerm og fil. I tillegg til loggfilene trenger postprosessoren tilgang til scenario- og plattformfilen som var grunnlag for produksjon av de aktuelle loggfilene.

Bruksområdet er i hovedsak å finne gjennomsnittsverdier og standardavvik for et sett med simuleringer. SIMBA PostProcessor kan også brukes til å gjennomgå loggfilene på linjenivå for hver enkelt plattform, samt til å se på resultater for enkeltplattformer i enkeltkjøring.

5.1 Kravspesifikasjon

For å kjøre programmet trengs en maskin med Windows 95 eller nyere versjoner av Windows. Det anbefales å kjøre programmet på en lokal maskin, men det fungerer også via nettverk med WinCenter. Atskillig høyere prosesseringstider må da påregnes.

5.2 Hovedvinduet

Hovedvinduet (Figur 5.1) består av en tittellinje (5.2.1), en menylinje (5.2.2), to felt for filspesifikasjon (5.2.3), tre plattformfelt (5.2.4), ett hendelsesfelt (5.2.5), en filvisningsknapp med tilhørende filvalgfelt (5.2.6), en matriseknapp (5.2.7), en resultatknapp (5.2.8), en lesfilknapp (5.2.9), en avslutningsknapp (5.2.10), et statusfelt (5.2.11) og et antallsfelt (5.2.12).

5.2.1 Tittellinjen

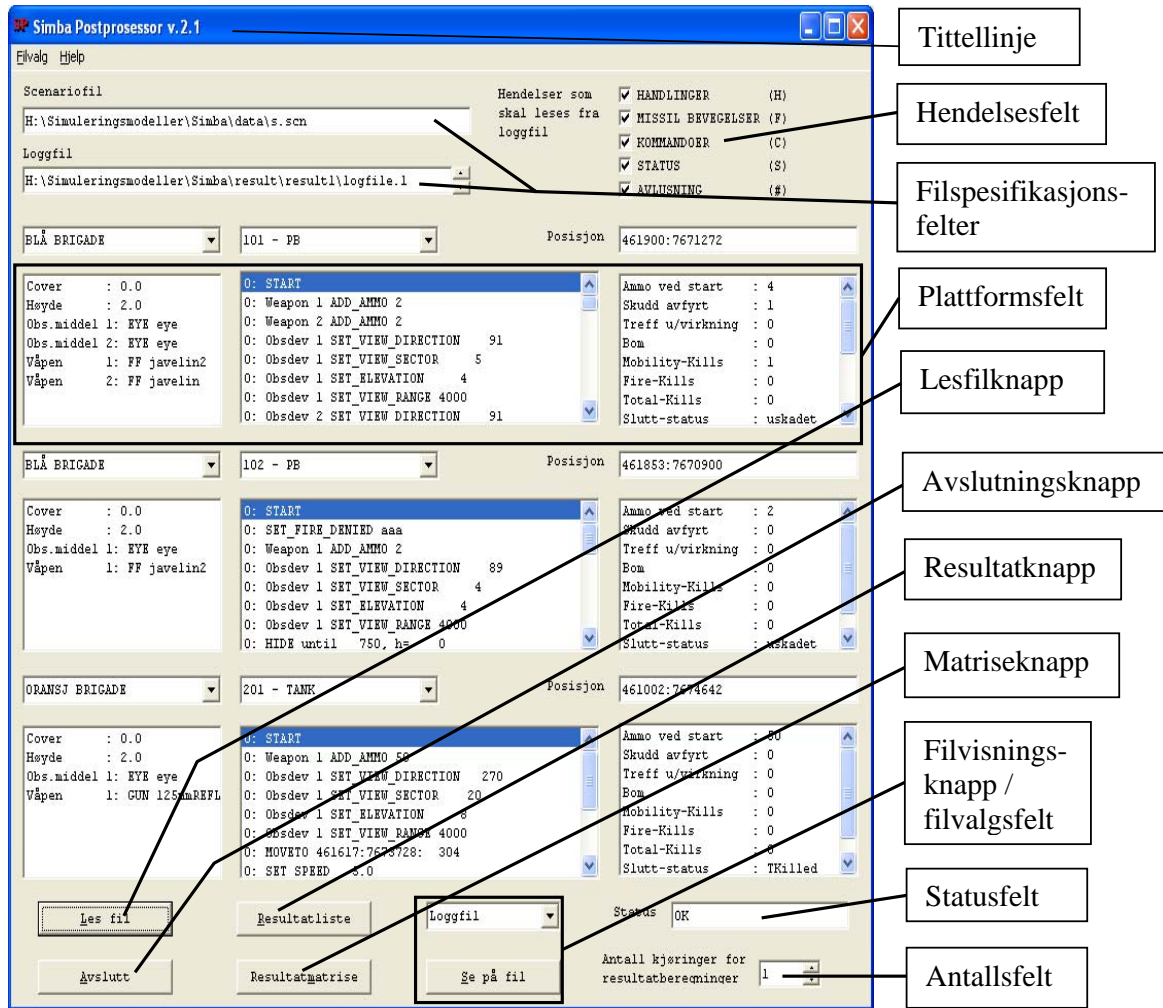
Tittelfeltet er den blå linjen øverst i programvinduet. Den inneholder minimerings-, maksimerings- og avslutningsknapper og systemmeny. Tittelfeltet viser navnet på programmet og hvilken versjon som kjøres.

5.2.2 Menylinjen

Menylinjen inneholder en meny for styring av programmet. Det eneste som kun kan gjøres fra menyen er å lagre standard oppsett, samt å velge filer grafisk. Menylinjen består av to rullegardiner: Filvalg og Hjelp. Alle menyvalg i hovedvinduet er beskrevet under.

5.2.2.1 Filvalg/Velge Scenariofil...

Dette menyvalget åpner Windows sin standardboks for innhenting av filnavn. Filen som velges må være en scenariofil, ellers vil programmet krasje. Programmet kontrollerer ikke hva slags fil som velges, det er opp til brukeren. Hvis brukeren vet hvilken fil som skal brukes, samt hvor den ligger (stien til filen), kan filnavnet med fullstendig sti like godt skrives inn i filspesifikasjonsfeltet (5.2.3) direkte.



Figur 5.1 Hovedvinduet i SIMBA postprocessor

5.2.2.2 Filvalg/Velge Loggfil...

Dette menyvalget åpner Windows sin standardboks for innhenting av filnavn. Filen som velges må være en loggfil, ellers kan programmet "henge". Programmet kontrollerer ikke hva slags fil som velges, det er opp til brukeren. Hvis brukeren vet hvilken fil som skal brukes, samt hvor den ligger (stien til filen), kan filnavnet med fullstendig sti like godt skrives inn i filspesifikasjonsfeltet (5.2.3) direkte. I tillegg er det et krav at det er samsvar mellom scenariofilen og loggfilen for at programmet skal fungere. Det vil i praksis si at loggfilen må være generert med angitt scenariofil. Det spiller ingen rolle hvilken fil som velges først så lenge det er samsvar mellom dem når enten lesfil- (5.2.9), matrise- (5.2.7) eller resultatknappen (5.2.8) trykkes.

5.2.2.3 Filvalg/Lagre standard oppsett

For å lagre navnene på de angitte filene i en ny fil som lastes automatisk ved oppstart, brukes dette menyvalget. Filen lagres under navnet "defaultsetting.dat". Det er imidlertid noe forskjellig hvor denne filen blir lagret i de forskjellige versjonene av Windows. Det er den

filen som ligger i den katalogen programmet kjøres fra som lastes ved oppstart, og vanligvis må lagrede standardoppsettsfiler kopieres dit manuelt.

5.2.2.4 Filvalg/Avslutt

Dette valget avslutter programmet på lik linje med klikk på krysset i høyre kant av tittellinjen (5.2.1) eller dobbelklikk på ikonet til venstre på tittellinjen (5.2.1).

5.2.2.5 Hjelp/om programmet

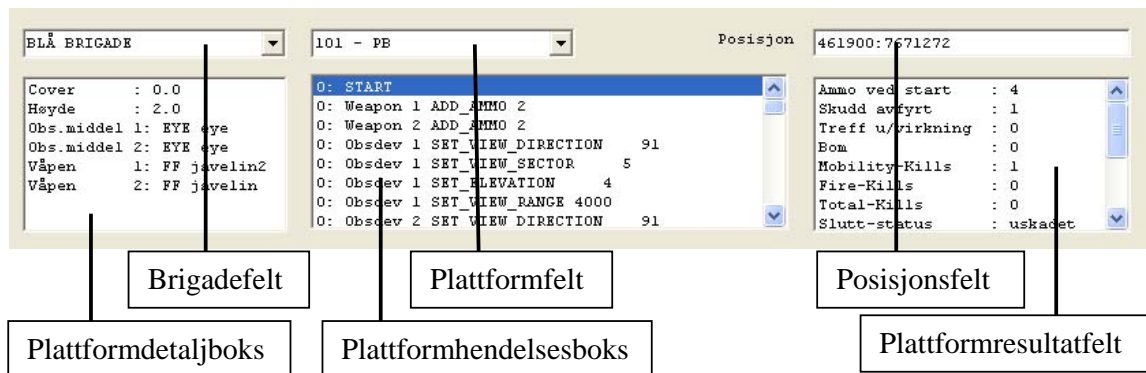
Dette menyvalget bringer frem et vindu som forteller hvilken versjon som kjøres (finnes også i tittellinjen (5.2.1) for enkel aksess), samt hvem som har skrevet programmet.

5.2.3 Filspesifikasjonsfelt

Det er ett felt for scenariofil og ett felt for loggfil. Begge kan velges fra menyen (5.2.2) eller skrives inn manuelt. Scenariofilen må være den filen som loggfilen ble generert på grunnlag av, og loggfilen som angis er den som prosesseres ved trykk på lesfilknappen (5.2.9). Loggfilene har navn som ender med ".xxx" der xxx er et tresifret tall. Pilene på enden av loggfilboksen gir mulighet til å endre nummeret én opp eller ned, slik at et antall filer kan gjennomgås enkelt, uten at filnavnet må spesifiseres for hver fil.

Når man skal produsere en resultatliste eller resultatmatrise (5.2.8 og 5.2.7), må man angi et antall filer som skal være med i beregningen. Velger man n filer i antallsfeltet (5.2.12), og filen i loggfilfeltet har endelse $.m$, vil det være loggfilene med endelse fra og med m til og med $m+n-1$ som resultatlisten og resultatmatrisen beregnes på bakgrunn av.

5.2.4 Plattformfelt



Figur 5.2 Plattformfelt

Hvert av de tre plattformfeltene består av 6 felt, og hvert av disse er i det følgende beskrevet i egne underkapitler. Plattformfeltene viser informasjon for en enkelt plattform i en enkelt loggfil (den som er angitt i filspesifikasjonsfeltet (5.2.3)). Hvilken plattform som skal vises, velges fritt.

5.2.4.1 Brigadefeltet

Angir hvilken brigade sine plattformer som skal vises i plattformfeltet (5.2.4.2). Dette feltet endres ved hjelp av pilen i høyre kant og kan kun inneha verdiene blå og oransje brigade. Hvis feltet er aktivt (i Windows forstand), kan piltastene brukes til å skifte mellom de to tillatte verdiene.

5.2.4.2 Plattformfeltet

Angir hvilken plattform som skal vises. Informasjonen som vises i plattformdetalj- (5.2.4.4), plattformhendelses- (5.2.4.5) og plattformresultatboksen (5.2.4.6) gjelder alltid plattformen som er angitt i dette feltet, og data hentes fra filene angitt i filspesifikasjonsfeltene (5.2.3). Innholdet i feltet endres på samme måte som brigadefeltet (5.2.4.1), og kan kun inneholde nummer og navn på plattformer som eksisterer i gjeldende scenariofil.

5.2.4.3 Posisjonsfeltet

Posisjonsfeltet er et fremvisningsfelt. Det betyr at innholdet ikke kan endres manuelt. Innholdet vil være en posisjonsangivelse (eller blankt felt hvis slik informasjon ikke er relevant). Verdien som vises avhenger av hvilken linje som er aktiv i plattformhendelsesboksen (5.2.4.5). Det som vises er plattformens posisjon på det tidspunktet den aktive logglinjen ble skrevet.

5.2.4.4 Plattformdetaljeboksen

Her vises detaljer angående plattformen spesifisert i plattformfeltet (5.2.4.2). Data i boksen er hentet fra scenario- og plattformfilen og kan ikke endres manuelt.

5.2.4.5 Plattformhendelsesboksen

Her vises alle logglinjene som angår plattformen som er angitt i plattformfeltet (5.2.4.3), dog bare de handlinger som er utført av denne plattformen. Dette betyr at andre plattformers deteksjon av eller skudd mot den angitte plattform ikke kommer med i denne listen. Hvis angitt plattform skulle være så uheldig å bli truffet, vil det selvsagt gjenspeiles i plattformhendelsesboksen. Innholdet kan ikke endres manuelt, men når feltet er aktivt, kan man velge aktiv linje ved bruk av piltaster eller museklikk. Innholdet i posisjonsfeltet (5.2.4.3) avhenger av hvilken hendelse som er aktiv. Hvilke typer hendelser som skal vises i plattformhendelsesboksen, spesifiseres i hendelsesfeltet (5.2.5). Når plattformhendelsesboksen aktiveres, settes Hendelsesfeltet til gjeldende konfigurasjon.

5.2.4.6 Plattformresultatboksen

I plattformresultatboksen vises resultatet for den plattformen som er angitt i plattformfeltet (5.2.4.2). Boksens innhold kan ikke endres manuelt. Med resultatet menes for eksempel ammunisjonsstatus, plattformens avslutningsstatus og plattformens effekt på andre plattformer. I tillegg inneholder boksen en liste med detaljer for hvert skudd plattformen har avfyrt.

5.2.5 Hendelsesfelt

Feltet består av 5 hakebokser. Vises en hake i boksen, blir den aktuelle hendelsestypen vist. Dette feltet har effekt kun på innholdet i plattformhendelsesboksene (5.2.4.5), der kun hendelser fra kategorier som er slått på i hendelsesfeltet vil vises. Feltet kan endres når som helst ved å klikke på en eller flere av hakeboksene, men effekten gjøres ikke gjeldende før neste fil leses inn (kan godt være den samme, bare den leses på nytt). Ny fil leses inn ved trykk på lesfilknappen (5.2.9). (Kun valg av ny fil er ikke nok til å endre aktiv konfigurasjon). Man kan sette hendelsesfeltet tilbake til gjeldende konfigurasjon automatisk ved å aktivere en av plattformhendelsesboksene (5.2.4.5).

5.2.6 Filvisningsknapp/filvalgfelt

Denne knappen brukes for å ta en titt på de filene som brukes til beregningene. Man har valget mellom loggfilen, scenariofilen og plattformfilen (den siste er spesifisert i scenariofilen). Foreløpig kan man ikke velge kommandofilen. Hvilken fil som skal vises, velges i filvalgfeltet ved hjelp av pilen i høyre kant (eller piltaster hvis feltet er aktivt) før knappen trykkes. Filene som vises er de som er spesifisert i filspesifikasjonsfeltet (5.2.3).

5.2.7 Matriseknapp

Resultatmatrisen vises i eget vindu, matrisevinduet (5.4). Opplysninger om hvilke filer som skal tas med i beregningen av matrisen, hentes fra filspesifikasjonsfeltene (5.2.3) og antallsfeltet (5.2.12). Matrisen viser skuddresultater for hver plattformtype mot hver av fiendens plattformtyper. Hvem som i denne sammenheng skal anses som fiende, kan velges i matrisevinduet.

5.2.8 Resultatknapp

Resultatlisten vises i eget vindu, resultatvinduet (0). Data til beregningen er de samme som for matriseknappen (5.2.7). Resultatlisten viser resultater og sluttstatus for hver plattformtype, eller for alle plattformtypene samlet, på en noe mer summarisk måte enn resultatmatrisen (5.2.7).

5.2.9 Lesfilknapp

Klikk på denne for å lese inn filene i filspesifikasjonsfeltet (5.2.3), med de begrensninger som er gitt i hendelsesfeltet (5.2.5). Denne knappen berører kun fremvisning i plattformfeltene (5.2.4), og trenger således ikke benyttes før matrise- (5.2.7) og resultatknappen (5.2.8). Det er kun loggfilen i filspesifikasjonsfeltet (5.2.3) som leses. Antallsfeltet (5.2.12) har ingen innvirkning på resultatet av trykk på denne knappen.

5.2.10 Avslutningsknapp

Avslutter programmet uten å lagre verken oppsett eller utregnede resultater.

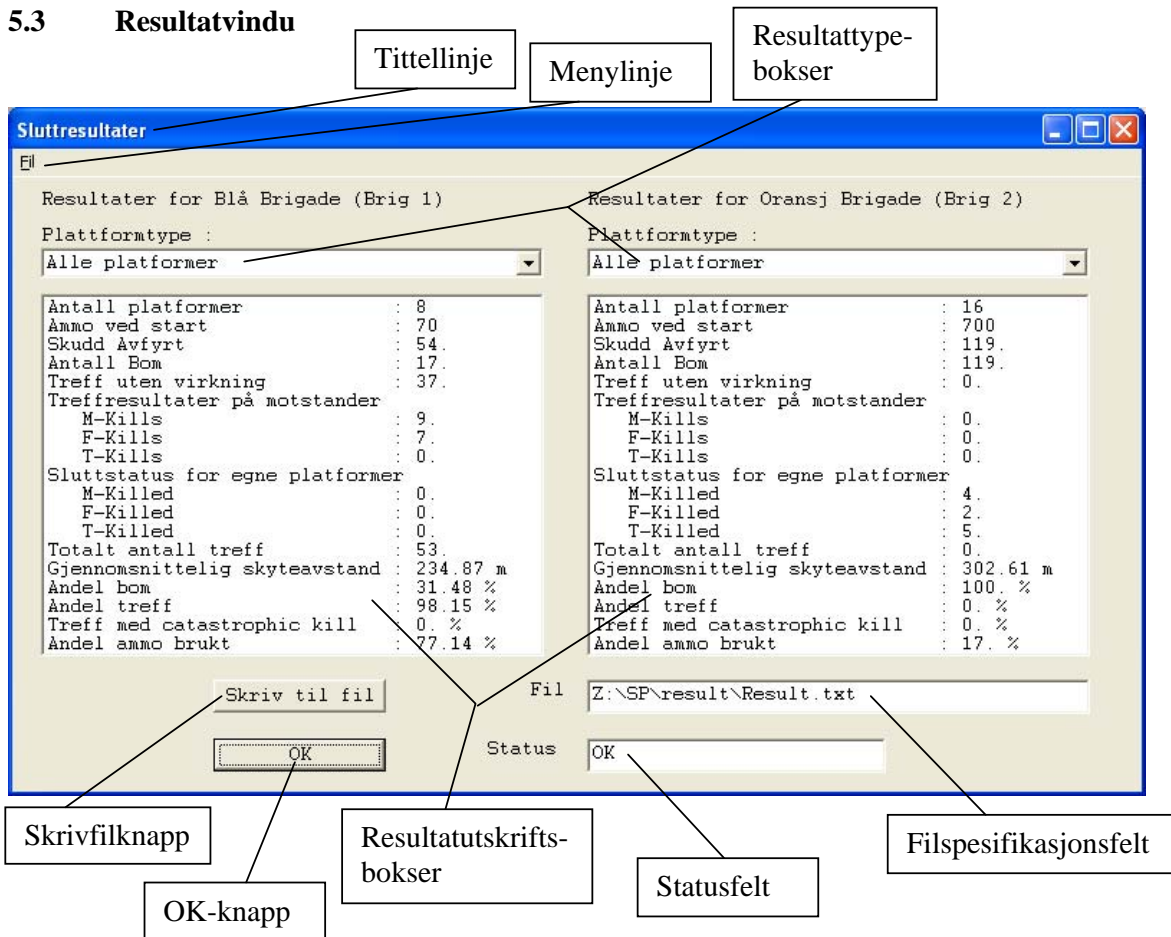
5.2.11 Statusfelt

Dette feltet brukes til å gi tilbakemelding om hendelser i programmet. Hvis et tastetrykk ikke gir forventet respons, kan det lønne seg å ta en titt på statusfeltet. Her vil det ofte komme frem en feilmelding i slike situasjoner. Bekreftelser på utførte oppgaver kommer også her.

5.2.12 Antallsfelt

Angir hvor mange loggfiler som skal brukes som basis for kalkulasjonene når matrise- (5.2.7) eller resultatknappen (5.2.8) trykkes. Hvis det i filspesifikasjonsfeltet (5.2.3) er angitt en loggfil ved navn "loggfil.130" og antallsfeltet står på 7, vil filene til og med "loggfil.136" brukes som data. Antallsfeltet kan endre ved bruk av pilene i høyre kant eller, hvis feltet er aktivt, ved hjelp av piltaster. En tredje måte er å skrive inn det ønskede antall manuelt; markér (aktiviser) feltet og skriv som i en tekstbehandler.

5.3 Resultatvindu



Figur 5.3 Resultatvinduet

Resultatvinduet (Figur 5.3) åpnes ved å klikke på resultatknappen (5.2.8). Det består av en tittellinje (5.3.1), en menylinje (5.3.2), en resultattypeboks (5.3.3) og en resultatutskriftsboks (5.3.4) for hver side, en avbryttnapp (5.3.5), en skrivfilknapp (5.3.6), et filspesifikasjonsfelt (5.3.7) og et statusfelt (5.3.8). Hver av disse er beskrevet i eget kapittel under. De feltene i

hovedvinduet (5.2) som setter grunnlaget for utregningene i resultatvinduet er filspesifikasjonsfeltene (5.2.3) og antallsfeltet (5.2.12).

5.3.1 Tittellinje

Tittellinjen er, som alltid, den blå linjen øverst på vinduet. Der står navnet på vinduet, slik at man vet hvor man er. I tillegg er det fra tittellinjen tilgang til systemmenyen via ikonet i venstre kant, og minimering/maksimering gjøres med de vanlige Windowsknappene (2. og 3. knapp fra høyre på tittellinjen). Det som er litt spesielt med tittellinjen i dette vinduet, er at knappen helt til høyre (den som normalt lukker det aktive vinduet) i dette vinduet er utskrift. Klikk på denne knappen for å skrive resultatene til filen som er angitt i filspesifikasjonsfeltet (5.3.7).

5.3.2 Menylinje

Menylinjen i dette vinduet inneholder kun én rullegardin, og den har bare 2 valg; ”Velg fil for skriving” (5.3.2.1) og Avslutt (5.3.2.2). Se under for forklaring av de enkelte menyvalg.

5.3.2.1 Resultatvindu - Fil/Velg fil for skriving

Dette menyvalget åpner standardboksen for filvalg, og gir dermed brukeren mulighet til grafisk å velge fil. Filnavnet blir overført til filspesifikasjonsfeltet (5.3.7) og brukes ved klikk på skrivfilknappen (5.3.6) eller krysset i tittellinjen (5.3.1).

5.3.2.2 Resultatvindu – Fil/Avslutt

Dette menyvalget avslutter vinduet. Det var vel neppe noen overraskelse. Data fra resultatvinduet blir ikke automatisk skrevet til fil.

5.3.3 Resultattypeboks

Her kan det velges hvilken plattformstype det skal vises resultater for. Det finnes én slik boks for hver side (blå og oransje), og de står i utgangspunktet slik at de viser alle typer og dermed det totale resultatet for hver side. Feltene endres med klikk på pilen i høyre kant og så klikk på typen man ønsker, eller ved hjelp av piltastene hvis feltet er aktivt. Her kan man velge om man vil se på resultatene for én enkelt plattformstype, eller om man vil se på resultatene for alle plattformene.

5.3.4 Resultatutskriftsboks

Dette er feltet der resultatene faktisk skrives ut (på skjerm). Innholdet i dette feltet kan ikke endres manuelt. Det oppdateres automatisk når det velges ny plattformstype i resultattypeboksen (5.3.3).

5.3.5 Avbryttnapp

Samme funksjonalitet som menyvalget ”Resultatvindu - Fil/Avslutt” (5.3.2.2).

5.3.6 Skrivfilknapp

Trykk på denne knappen for å skrive resultatene til filen som er angitt i filspesifikasjonsfeltet (5.3.7). Det skrives resultater for hver plattformtype og for alle sammenlagt, for begge sider. Hva som for øyeblikket vises på skjermen har ingen innvirkning på hva som skrives til fil.

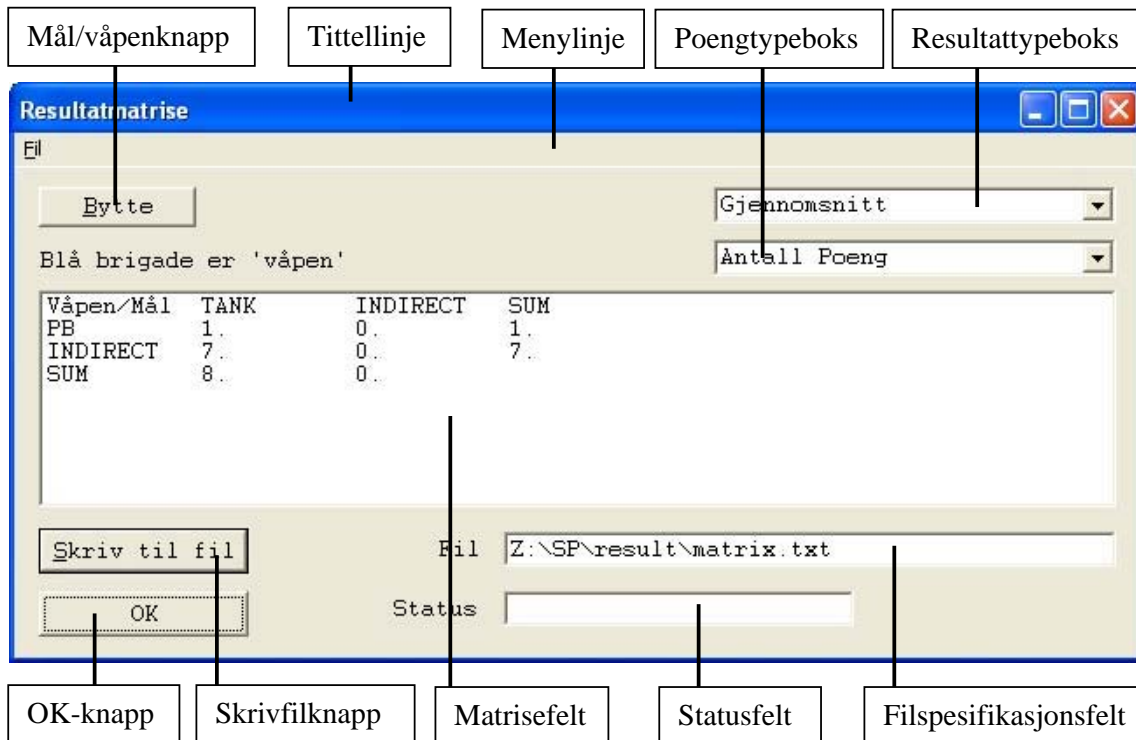
5.3.7 Filspesifikasjonsfelt

Her vises navnet på filen det skal skrives til (hvis det skal skrives til fil). Feltet kan aktiveres, og endres som i en vanlig tekstbehandler, eller man kan bruke menyvalget ”Resultatvindu - Fil/Velg fil for skriving” (5.3.2.1) for å endre filnavnet. Filnavnet må oppgis med fullstendig sti.

5.3.8 Statusfelt

Her vises status for vinduet. Stort sett er det informasjon om hvorvidt filen det skal skrives til finnes eller ikke.

5.4 Matrisevinduet



Figur 5.4 Matrisevinduet

Matrisevinduet åpnes ved å klikke på matriseknappen(5.2.7). Det består av en tittellinje(5.4.1), en menylinje(5.4.2), en mål/våpenknapp(5.4.3), en resultattypeboks(5.4.4), en poengtypeboks(5.4.5), et matrisefelt(5.4.6), en skrivfilknapp(5.4.8), en OK-knapp(5.4.7), et filspesifikasjonsfelt(5.4.9) og et statusfelt(5.4.10). Hver av disse er beskrevet i egne avsnitt i det følgende. De feltene i hovedvinduet(5.2) som setter grunnlaget for utregningene i matrisevinduet er filspesifikasjonsfeltene(5.2.3) og antallsfeltet(5.2.12). Matrisevinduet gir en matrise med alle plattformtypene på den ene siden som våpen og alle på andre siden som mål. Vinduet starter med blå brigade som våpen, men dette kan byttes om ved hjelp av mål/våpenknappen(5.4.3). Tallene i matrisen(5.4.6) viser data for våpenbrigaden, og (hvis relevant) deres effekt på målbrigaden. Poeng telles i hele og halve poeng. Halve poeng gis for ”del-kills” (Mkill eller Fkill).

5.4.1 Tittellinje

Tittellinjen i matrisevinduet fungerer på samme måte som i resultatvinduet (5.3.1). Det gjelder også krysset ytterst til høyre, som gir utskrift.

5.4.2 Menylinje

Menylinjen i dette vinduet er lik den i resultatvinduet (5.3.2) og virker på samme måte. Begge menyvalgene er også nøyaktig lik de i resultatvinduet (5.3.2.1 og 5.3.2.2).

5.4.3 Mål/Våpenknapp

Trykk her for å se målbrigaden som våpen, og omvendt.

5.4.4 Resultattypeboks

Resultattypeboksen i matrisevinduet (5.4) gir mulighet til å velge om man vil se gjennomsnittsverdier eller standardavvik i matrisefeltet (5.4.6). Det opereres med pilen ytterst til høyre i feltet.

5.4.5 Poengtypeboks

I poengtypeboksen kan det velges å se på antall poeng totalt, eller antall treff med forskjellig virkning, eller antall skudd avfyrt. Det finnes flere kategorier, og de skulle alle være relativt selvforklarende. Resultatene vises i matrisefeltet (5.4.6) så snart de er ferdig utregnet. Boksen opereres med pilen ytterst til høyre i feltet, eventuelt piltaster hvis feltet er aktivt.

5.4.6 Matrisefeltet

Her vises den valgte matrisen. Hvilken matrise som vises avhenger av hvem som er mål og våpen (mål/våpenknappen (5.4.3)), samt hva som er valgt i resultat- (5.4.4) og poengtypeboksene (5.4.5).

5.4.7 OK-knapp

Samme funksjonalitet som menyvalget "Resultatvindu - Fil/Avslutt" (5.3.2.2).

5.4.8 Skrivfilknapp

Skriver matrisene til filen som er angitt i filspesifikasjonsfeltet (5.4.9). Det skrives resultater for alle poengtypene, men kun for den resultattypen som er valgt i resultattypeboksen (5.4.4). Hva som ellers vises på skjermen har ingen innvirkning på hva som skrives til fil.

5.4.9 Filspesifikasjonsfelt

Feltet er nøyaktig likt i funksjon til det i resultatvinduet (5.3.7).

5.4.10 Statusfelt.

Også dette feltet er likt tilsvarende i resultatvinduet (5.3.8).

6 FILER SOM BRUKES AV SIMBA

6.1 Kartfiler

Det som omtales som ”kartet” i sammenheng med SIMBA, består av opp til syv filer. De angir hver på sin måte noe om det terrenget simuleringen skal foregå i. Filer i samme sett (som har samme ”fornavn”) bør i det lengste inneholde informasjon om det samme området, men det ligger ikke inne noen form for kontroll av dette. Filene som inngår er beskrevet i det følgende.

6.1.1 Hekkefil (.wall)

Denne filen går også under navnet wallfilen. Den er ikke nødvendig å ha med, men hvis den finnes, inneholder den informasjon om en rekke ”vegger” i terrenget. Dette brukes som en erstatning for vegetasjonsinformasjon der man ved befarung har funnet at det ikke er fri sikt, samt andre steder der man av forskjellige årsaker ønsker å umuliggjøre observasjon og deteksjon. Filen leses av SIMBA Preprocessor, SIMBA Simulator og SIMBA Kartlogg, og kan opprettes ved tegning på kartet i førstnevnte.

6.1.2 Kartbildefiler (.jpg/.bmp)

Kartbildefilen til SIMBA Preprocessor skal ha filnavnendelsen *jpg*, mens kartbildefilen til SIMBA Kartlogg skal ha endelsen *.bmp*. Begge filene må selvsagt inneholde et bilde på det format filendelsene angir. Begge filene skal inneholde et bilde av et kart. For SIMBA Preprocessor er poenget at det blir noe enklere å orientere seg etter et kartbokblad enn etter et høydekoteplott. Dette kartet er i midlertid ikke nødvendig for å kjøre SIMBA Simulator, og brukes heller ikke andre steder. Når det gjelder *.bmp*-kartet, så er dette absolutt nødvendig for å få noe fornuftig ut av SIMBA Kartlogg, men det brukes ingen andre steder. Hvordan disse bildene genereres er for så vidt likegyldig for SIMBA, men det som i all hovedsak har vært brukt til nå, er rett og slett veikart som digitaliseres (scannes).

6.1.3 Kartdatafil (.dat)

Innholdet i denne filen angir hvor i verden kartet er fra (utm_{x0}/utm_{y0}), hvor stort område det dekker (representert ved nx/ny, som er antallet punkter i hver retning), og hvilken oppløsning punktene i Terrenghen og Vegetasjonsfilen har (avstanden mellom punktene er gitt av dx og dy). Filen skal ha etternavn *.dat*, og skrives (og leses) på ASCII-format. Filen kan ikke editeres/opprettes fra SIMBA Preprocessor, det må gjøres manuelt i en editor som kan lagre ren tekst.

6.1.4 Koordinatfil (.krd)

Dette er filen som holder referansen mellom Kartbildefilen (.bmp) i SIMBA Kartlogg og den virkelige verden. Med andre ord beskriver filen hvilket område kartbildefilen representerer. Dette gjøres ved at det tas ut to punkter som det angis UTM-koordinater for. Disse lagres og sammenholdes med tilsvarende skjermkoordinater. Etersom det antas at kartet er tilpasset et todimensjonalt kartesisk koordinatsystem der nord er opp i skjermbildet, er det nok med to punkter i denne sammenheng. Punktene kan tas ut og lagres fra SIMBA Kartlogg, og brukes kun av denne modulen.

6.1.5 Matrisefil (.con)

Denne filen fylles med data regnet ut på bakgrunn av terrengfilen. Dette gjøres for å slippe å vente på utregning av nødvendige størrelser hver gang man laster et kartfilsett i SIMBA Preprocessor. Hvis denne filen finnes, brukes den automatisk. Filen skal ha etternavn *con*, og den opprettes automatisk. En smule varsomhet må utvises ved endring av et eksisterende kart, eventuelt opprettelse av et nytt kart med samme navn som et eksisterende, fordi det ikke foretas noe kontroll før matrisefilen aksepteres og benyttes. Man kan altså ende opp med å bruke matrisefilen fra et gammelt kart i slike tilfeller.

6.1.6 Terrengfil (.ter)

Denne filen skal inneholde terrengets høyde i punkter med lik avstand (punktavstanden i lengde- og bredderetning kan være ulik). Avstanden mellom punktene i terrengfilen angis i Kartdatafilen. Informasjonen leses av både SIMBA Preprocessor og SIMBA Simulator, men ingen av postprosesseringsverktøyene bruker den. Informasjonen lagres digitalt. Denne filen kan altså ikke uten videre leses av oss mennesker. I tillegg bør man legge merke til at høydepunktene i SIMBA Preprocessor strekkes for å tilpasse terrengomrisset til størrelsen på vinduet. I praksis betyr dette at en kilometer i nord/syd-retning ikke er like lang på skjermen som en kilometer i øst/vest retning. Det kan se litt rart ut på skjermen, særlig når man slår på siktsektorer (vinkelen blir tilsynelatende feil), men det regnes riktig likevel.

6.1.7 Vegetasjonsfil (.veg)

Denne filen skal ha filnavnendelsen *veg*, og må ikke forveksles med *.wall*-filen. Spesielt kan en slik misforståelse oppstå fordi konstruksjon av *.wall* filer gjøres via menyvalget *Vegetation/Make* i SIMBA Preprocessor sitt Hovedvindu. Filen skal inneholde data for vegetasjonens høyde i det aktuelle området, og skal være bygget opp på samme måte som terrengfilen med samme avstand mellom punktene, men med ett punkt mindre i hver retning. Her er det nemlig vegetasjonens høyde i senter av et rektangel dannet av fire punkter i terrengfilen som skal spesifiseres. For øvrig gjelder de samme bemerkningene angående format, tilnærminger og avvik. Vegetasjonsfilen er ikke påkrevd, og ettersom de nødvendige data har vært vanskelige å få tak i, har denne funksjonaliteten vært lite brukt. Filen brukes av SIMBA Simulator og SIMBA Preprocessor hvis den finnes.

6.2 Våpenfiler

SIMBA simulerer ikke effekten av enkeltskudd. Disse dataene er beregnet på forhånd, og er lagret i diverse våpenfiler som SIMBA benytter. Disse filene er lagret på ASCII-format, så det er enkelt å endre disse dersom det skulle være behov for det. Det finnes tre forskjellige typer våpenfiler i SIMBA: Én for direkteskytende våpen, én for konvensjonelt artilleri og én for smart artilleri. Alle våpenfilene har endelse *wpn*.

6.2.1 Direkteskytende våpen

Et enkelt eksempel på en våpenfil for et direkteskytende våpen kan se slik ut:

```
MIN_RANGE      100
MAX_RANGE      2500
MAX_SHOTS      4
AIM_TIME       10
LOAD_TIME      20
SPEED          300
```

BEGIN_TARGET	T72							
ASPECT_ANGLE	7	0	30	60	90	120	150	180
COVER	<140							
RANGE	1000							
M_KILL		0.23	0.31	0.35	0.42	0.40	0.38	0.45
F_KILL		0.14	0.22	0.24	0.27	0.25	0.24	0.26
MFK_KILL		0.31	0.42	0.47	0.51	0.48	0.47	0.49
P_HIT		0.78	0.80	0.83	0.88	0.83	0.80	0.75
END_TARGET								

De øverste linjene beskriver egenskaper ved våpenet som ikke har direkte med sannsynligheten for å ødelegge et mål å gjøre. Disse parametrene beskriver henholdsvis våpenets minimumsrekkevidde, maksimumsrekkevidde, maksimalt antall skudd dette våpenet vil skyte mot samme fiende, tiden det tar å sikte, tiden det tar å lade og hastigheten til ammunisjonen (missilet).

Alt som står mellom "BEGIN_TARGET" og "END_TARGET" beskriver effekten mot ett bestemt mål, i dette tilfellet T72 (tallene er ikke reelle). Linjen som starter med "ASPECT_ANGLE" kommer bare én gang for hvert mål – dette er en overskrift for det som følger. "COVER" beskriver hvilken dekning målet kan ha for de følgende dataene. I dette tilfellet har målet under 140 cm dekning. Vanligvis opererer vi med over eller under 140 cm, som beskriver om man ser hele kjøretøyet eller bare tårnet. "RANGE" beskriver rekkevidden de påfølgende dataene gjelder for.

Man kan oppgi data for to forskjellige dekningsnivåer (over/under 140 cm) og så mange rekkevidder som man ønsker. Når man skyter mot et mål med en bestemt dekning på en bestemt avstand, slår SIMBA opp i tabellen, og finner virkningsdata for den aktuelle dekningsgraden og rekkeviddene som er over og under den faktiske skyteavstanden. Så interpoleres disse verdiene for å finne effekten på den aktuelle avstanden.

Formatet på virkningsdataene er litt spesielt. Tallene angir sannsynligheter for å påføre målet henholdsvis en M_kill (mobility-kill, ødelagt bevegelsessevne), F_kill (firepower-kill, ødelagt evne til å skyte), MFK_kill (enten M_kill eller F_kill), samt sannsynligheten for å treffe målet (p_hit). Disse sannsynlighetene er ikke uavhengige, og SIMBA regner om dataene til sannsynligheter for følgende hendelser: Bom, treff uten virkning, kun m_kill, kun f_kill og både m_kill og f_kill.

6.2.2 Konvensjonelt artilleri

Eksempel på våpenfil for konvensjonelt artilleri:

MAX_RANGE	100	
SPREAD_X	20	
SPREAD_Y	20	
BEGIN_TARGET	BMP	
COVER	<140	>140
0	1	1

10	0.3	0.2
50	0.05	0.02
100	0	0

END_TARGET

Nok en gang er de øverste linjene parametre som ikke har direkte å gjøre med sannsynligheten for å slå ut mål. De beskriver den maksimale avstanden som ammunisjonen har effekt innenfor, samt spredningen til ammunisjonen i x- og y-retning. Dataene mellom "BEGIN_TARGET" og "END_TARGET" beskriver sannsynligheten for å slå ut det aktuelle målet på en gitt avstand med en gitt dekning. Det skilles ikke mellom m_kill og f_kill, det antas at målet enten blir fullstendig ødelagt eller ikke skadet i det hele tatt.

6.2.3 Smart artilleri

Eksempel på våpenfil for smart artilleri:

```

RADIUS      90
SPREAD_X    20
SPREAD_Y    20

BEGIN_TARGET T72

COVER        <140 >140
P_DET        0.80 0.50
P_OVERKILL   0.60 0.40
M_KILL       0.70 0.50
F_KILL       0.30 0.40
MFK_KILL    0.90 0.85

```

END_TARGET

De øverste linjene beskriver her radien på søkeområdet til granaten, samt ammunisjonens spredning i x- og y-retning. Parametrene som beskriver effekten mot et bestemt mål, er for smart ammunisjon kun avhengig av dekningen, ikke av avstanden fra nedslagspunktet (som egentlig er søkeområdets sentrum). P_DET er sannsynligheten for å finne og velge et mål som ikke allerede er fullstendig slått ut, P_OVERKILL er sannsynligheten for å finne og velge et mål som er fullstendig slått ut, M_KILL, F_KILL og MFK_KILL er tilsvarende sannsynligheter som for direkteskytende våpen, gitt at målet er detektert og valgt. En smart granat kan kun velge ett mål.

6.3 Deteksjonsfiler

Foreløpig finnes det bare én type deteksjonsmiddel implementert i SIMBA, nemlig direkte observasjon. Man kan likevel ha forskjellige deteksjonsmidler innenfor denne kategorien. Deteksjonsfilene er veldig enkle, og ser slik ut:

```

TYPE      EYE
HEIGHT    0.0
TARGETS   Target1      Target2      osv
AREAL     10           3           osv
DCONST    4.0e6        4.0e6        osv

```

TYPE angir navnet på observasjonsmiddelet. HEIGHT forteller hvor høyt over plattformen observasjonsmiddelet sitter. For hvert mulige mål oppgis AREAL, som er arealet til målet (i kvadratmeter), og DCONST, som er en konstant SIMBA bruker i beregningene sine.

6.4 Andre filer

Scenariofila, plattformfila og kommandofila er beskrevet i kapittel 2.4. Resultatfilene som SIMBA genererer, er dokumentert i (2).

Litteratur

- (1) Logica Limited (1973): Tankkill System, Programmer Guide/User guide.
- (2) HALSØR Marius, EIDE Morten (2003): SIMBA – Dokumentasjon av kildekode, 2003/01711, Ugradert