



# RENDEZVOUS



*Nr 3 december 1999*



**Gott Nytt År**



## *Innehåll*

Ordföranden har ordet .....	3
Rapport från Ag/Teknik/mikroprojekt'99 .....	4
Sammandrag av resultat från Ag Metodik mikroprojekt under 1999 .....	6
Saab NINS .....	12
Du besöker väl vår websajt? .....	22
SESAM möten .....	22

## SESAM

---

### Vad är SESAM?

SESAM har tillkommit för att organisera och stimulera samarbete och samverkan inom programvaruområdet mellan försvarsindustrin, FMV och FOA.

Det avtalsfästa syftet med SESAM är ”att genom organiserat samarbete mellan användargruppens medlemmar främja tillförlitlighet och effektivitet i utveckling och vidmakthållande av programvarusystem i Ada inom försvarssektorn”. Inom ramen här för skall SESAM även anpassa, profilera och förnya sin verksamhet med hänsyn till ändrade tekniska och andra omständigheter av betydelse för intresseområdet.

Följande kommer att ske under den närmaste 2-3-årsperioden.

1. SESAM skall allmänt verka för att sprida information om faktorer som påverkar möjligheterna till tillförlitlig och effektiv utveckling och vidmakthållande av programvarusystem. Särskilt skall härvid Adas betydelse i sammanhanget klargöras.

2. SESAM skall i sin verksamhet fortlöpande bevaka möjligheterna att samla, skapa och sprida information om objektiva mät- och andra resultat och erfarenheter vunna vid användning av ”software engineering”-principer och Ada.

3. SESAM behandlar tillvägagångssättet vid utveckling och vidmakthållande av programsystem. Implicit i detta ligger givetvis att använda processer skall tillförsäkra de resulterande produkterna efterfrågade egenskaper. Produkt-egenskaper som påverkas av processerna är därför av primärt intresse att bevaka i SESAMs verksamhet.

4. SESAM skall i sin verksamhet fästa stor vikt vid att underlätta samexistens mellan Ada-program och programvara skriven i andra språk. Speciellt skall aspekter vid användning av COTS beaktas.

5. SESAM skall där så är möjligt sätta konkretiserade och mätbara mål för sin verksamhet under avgränsade tidsperioder.

SESAM styrs av ett Råd med representanter för gruppens medlemmar. Rådet har till sin hjälp ett Verkställande Utskott (VU) och ett sekretariat.

Rådets ordförande är Claes Wadsten, Celsius Aerotech, tel .

### VU

Bengtsson Christopher, FMV  
chben@tranet.fmv.se

Brandt Roger, FMV

robra@fmv.se

Eckfeldt Sune, Enator Telub AB

sune.eckfeldt@enator.se

Johansson Billy, CelsiusTech Electronics AB

bijo@celsiustech.se

Källberg Björn, CelsiusTech Systems AB

bjkae@celsiustech.se

Arbetet utförs i ett två arbetsgrupper:

Ag Metodik

Håkan Edler, CTH/Datorteknik

edler@ce.chalmers.se

Ag Teknik

Torbjörn Andreasson,

Ericsson Microwave Systems AB

torbjorn.andreasson@emw.ericsson.se

### Vilka kan vara med i SESAM?

Medlemmarna i SESAM är svenska företag, organisationer och myndigheter (förvaltningar, utbildningsinstitutioner etc) med anknytning till försvarssektorn. Medlemmarna indelas i följande kategorier

- ordinarie medlemmar
- arbetsgruppsmedlemmar
- informationsmedlemmar.

Enskild person kan endast komma ifråga som informationsmedlem.

### Inträde i SESAM

För samtliga medlemskategorier gäller att inträde beslutas av Rådet.

För inträde som ordinarie- och arbetsgruppsmedlem krävs status som leverantör till FMV. Dessutom krävs en skriftlig förbindelse att uppfylla åtagande som ordinarie- och arbetsgruppsmedlem.

För inträde som informationsmedlem (erhåller endast informationsbladet) krävs status som leverantör till FMV eller status som myndighet inom totalförsvaret. Rådet kan emellertid anta annan part som informationsmedlem.

För ansökan om medlemskap i SESAM vänd er till sekretariatet.

### SESAM-Sekretariatet

AerotechTelub AB

c/o Kåsjös Kontor

Ytterspåret 14

187 54 TÄBY

# Ordföranden har ordet

## Ett aktivt SESAM

Undertecknad har sedan årsmötet tillträtt som ordförande i SESAM. Och vill därför presentera mig och mina ambitioner för framtiden.

Jag arbetar på Celsius Aerotech sedan 1986 och har sedan två år tillbaka hand om teknikkoordinering vid företaget inklusive våra dotterbolag. Det innebär bland annat att jag följer upp all teknisk utveckling vid företaget samt har en mängd externa kontakter. Våra teknikområden är många, varför jag sysslar med allt från mekanik, elektronik till programvara fördelade på en mängd produktområden. Större delen av min verksamhet berör det militära området.

Innan min tid i försvarsindustrin arbetade jag som lärare på tekniska gymnasiet och högskolan i telekommunikation och datorämnen. Datorämnena var mycket maskinnära som t.ex. mikrodatorer, OS, grafisk presentation, styr och mät med mikrodatorer samt programmering i Assembly.

Jag är mycket resultatnriktad varför jag också vill överföra detta på SESAM. Jag vill därför ha ett SESAM som är aktivt och att alla medlemmar skall känna nyttan med att delta. Det innebär för mig att deltagarna skall knyta tvärkontakter mellan olika företag och institutioner genom gemensamma projekt och personliga relationer. SESAM är för mig ett kontaktorgan inom svensk försvarsindustri, högskolor och försvarsmakten där man utbyter erfarenheter och bevakar nya teknologier inom programvara. Utvecklingen går idag fort så grunden till framgång hänger till stor del på hur man bildar olika typer av nätverk för att snabbt ta till sig nya idéer och andras erfarenheter. Detta gäller speciellt inom dator och programmeringsområdet.

Användningen av COTS, högnivåspråk och utvecklingsverktyg tillsammans med kraven på återanvändning, verifiering och test samt korta utvecklingstider underlättas om man på ett strukturerat sätt kan utbyta erfarenheter.

Grunden för medlemskap i SESAM är att man ställer upp med eget arbete och deltar i projekt vars resultat sedan kan delges de andra i nätverket.

Tanken med detta är just att bilda nätverk samt att alla skall få mer tillbaka än den egna insatsen. För att detta skall fungera måste vi alla hjälpas åt till att få ett aktivt SESAM. Ett annat mål med SESAM är också kontakter mellan industrin, försvarsmakten och utbildningsanstalter. Genom att ha dessa kontakter samt stödja verksamheter på utbildningsanstalter kommer vi vid industrin att få nya medarbetare som också vid sin första anställning har en utbildning som passar för vår verksamhet.

Jag kommer som ordförande att försöka skapa förutsättningarna för att våra mål kan uppfyllas. För att detta skall vara möjligt måste alla bidra till verksamheten och uppfylla sin del i medlemskapet. Jag vill därför att ni på olika sätt hör av Er till mig med information, tips och problem som berör SESAM så skall jag göra min del av åtagandet för att få ett SESAM där alla känner nyttan av att delta.

Seminarier med externa föreläsare, redovisning av egna mikroprojekt samt kontakter med liknade nätverk är några metoder för att nå målen.

Höstseminariet som genomfördes i år visade hur medlemmarna kan samarbeta och producera ett resultat som alla har nytta av. Jag vill därför framföra ett tack till er alla som på olika sätt bidrog till detta seminarium. Jag hoppas att vi kan upprepa detta på liknade sätt också under detta verksamhetsår.

Förutom den ekonomiska nyttan så vill jag att Ni får nya personliga vänner via kontaktnätet i SESAM och därmed 'förenar nytta med nöje'.

Jag vill också framföra ett tack till Christopher Bengtsson som med kort varsel fick hoppa in som ordförande då Björn Källberg blev sjukskriven.

Med dessa ord vill jag önska Er välkommen till ett nytt verksamhetsår och hoppas att alla känner nyttan av SESAM.

Med hälsningar  
Claes Wadsten

# Rapport från Ag/Teknik/mikroprojekt'99

Under året som gått har teknik-gruppen fokuserat på att studera, skapa och sprida erfarenheter kring begreppet HLA (High Level Architecture). HLA anses allmänt få mycket stor betydelse för modellering och simulering i det framtida högteknologiska försvaret. I USA har man fattat beslut om en fullständig övergång till HLA och man säger till och med 'No Pay - No Play', vilket innebär att det inte kommer att läggas ut några beställningar på simuleringsprogram som inte stödjer HLA. Efter 2001-01-01 kommer inga sammansatta simuleringar att förekomma annat än med hjälp av HLA. Det svenska försvaret följer den amerikanska utveckling och ser också ett införande av HLA som en trolig väg. Det är med anledning av denna bakgrund som Ag/Teknik beslutade att under året belysa begreppet utifrån de olika medlemsföretagens perspektiv. Här följer korta aptitretare på vad som sedan döljer sig bakom portalen på den lösenordsskyddade medlems-webben. Dessutom passar undertecknad på att under rubriken övrigt, informera om status i det av FMV initierade demonstratorprojektet öppna adaptiva gränssnitt.

## FOA - Sven Sköld

Sven arbetar vid FOA och institution för modellering och simulering. Han presenterar pågående projektverksamhet där simuleringsmodeller med olika abstraktionsnivå samverkar i en gemensam HLA-baserad distribuerad simulering. Erfarenheter från HLA-anpassningen av dessa modeller presenteras tillsammans med framtida planer inom HLA-området. Sven har producerat följande rapporter och dokument:

1. Vad är HLA? (doc)
2. Erfarenheter från FOA (doc)
3. OH-presentation från SESAM höstseminarium 21-22 okt 1999 (ppt)

## Celsius Aerotech AB - Magnus Öhlund

Magnus har under året som gått studerat följande frågeställningar

- Vad är HLA?
- Varför HLA?
- Hur kommer man igång?
- Hur interfacar HLA mot programvarukomponenter, programmeringsspråk och middleware?
- HLA och UML?

Arbetet är avhandlat i följande rapport:

1. OH-presentation från SESAM höstseminarium 21-22 okt 1999 (doc) (ppt)

## Bofors Missiles AB - Jörgen Riisom och Alf Rundberg

Jörgen och Alf har under året jobbat med följande frågeställningar:

- Vad är det som HLA tillför som inte går att åstadkomma med moderna middleware-teknologier som t.ex CORBA?
- Varför inte använda CORBA och HLA tillsammans?
- Hur många certifierade RTI:er existerar?
- Flaskhals och prestandaproblem?

Resultaten har dokumenterats i följande rapporter och dokument:

1. Sammanfattning (pdf)
2. OH-presentation från SESAM höstseminarium 21-22 okt 1999 (pdf)

## TietoEnator - Bengt Andersson

Bengt har i sitt arbete studerat HLA ur ett säkerhetsperspektiv. Rapporten behandlar inledningsvis generella säkerhetsfrågor och går därefter in på att diskutera säkerheten i HLA. Syftet med rapporten är att belysa komplexiteten och påtala vikten av att man tidigt tar upp säkerhetsaspekterna vid utveckling av modellering och simulerings-system baserade på HLA.

1. HLA och säkerhet? (doc)

## **Ericsson Microwave Systems AB - Henrik Wassenius**

Henrik har under året studerat HLA med anledning av att konceptet skall användas i ett skarpt projekt, dvs de nyutvecklade komponenterna skall ha HLA-gränssnitt. I sin presentation under höstens SESAM-seminarie definierade han de grundläggande koncepten i HLA och därefter delade han med sig av sin erfarenheter från det skarpa projektet.

1. OH-presentation från SESAM höstseminarium 21-22 okt 1999 (pdf)

## **SAAB - Dag Folkesson**

Dag jar under året studerat HLA ur perspektivet ledning och styrning av flygande farkoster. Under seminariet delade han med sig av resultaten från analysen om huruvida och i så fall hur HLA skulle kunna användas i flygsammanhang.

1. OH-presentation från SESAM höstseminarium 21-22 okt 1999 (ppt)

## **Övrigt**

Under seminariet passade undertecknad på att informera om läget i det av FMV initierade demonstratorprojektet öppna adaptiva gränssnitt. Arbetet har genomförs av programvarulabbet (Softlab) vid Ericsson Microwave Systems AB (EMW) och fokuserat på att utvärdera och demonstrera dynamik och adaptivitet som bas för utveckling av komplexa informationssystem. Resultaten presenteras delvis och "pö om pö" på www-servern rma.nmac.ericsson.net (alt. www.rma.ericsson.net) fram till och med årsskiftet. Servern innehåller aktiva komponenter som exekverar i både Netscape och Explorer. Komponenterna och applikationerna kräver installation av Java-plugin, systemet hjälper dig automatiskt när dessa komponenter skall installeras. Dessutom behöver du ett speciellt säkerhetscertifikat för komponenterna över huvud taget skall starta - hör av dig till undertecknad.

Torbjörn Andreasson  
ordf. Ag Teknik'99

# Sammandrag av resultat från Ag Metodik mikroprojekt under 1999

Håkan Edler, ordf Ag Metodik

Aktiviteten har varit god under året inom arbetsgruppen. De ursprungliga 19 förslagen till mikroprojekt kondenserades till fyra och tre projekt har givit signifikanta resultat till vårt mål, att samla, skapa och sprida information om metodik för systemutveckling. Det första projektet studerade hur man hanterar de stora informationsmängder, som skapas i ett utvecklingsprojekt. Man ställde upp en frågelista och har för varje fråga angivit hur man gör idag och hur man vill göra i framtiden. Det andra projektet studerade effekten av granskningar av dokument i kravhantering och konstruktion. Man har studerat vad som är bra och vad som är dåligt i dagens metodik och hur mycket man vunnit på att hitta fel tidigt. Det tredje projektet studerade metod och praktik för provning av datorbaserade system. Man har sammanfattat och beskrivit tekniker för provning och studerat hur dessa tekniker används. Vidare har man mätt vilken effekt man får av att hitta fel tidigt och studerat hur man kan använda formella metoder vid konstruktion av system.

## 1. Dynamiska informationsstrukturer

Frågorna kretsar kring "informationstrukturen", som är den information vi bygger upp i ett systemarbete, i en databas eller i ett annat lagringsmedium, med all information om och runt komponenterna i ett system under arbete. Idén är alltså att "informationstrukturen" skall byggas under utvecklingens gång och vara komplett för varje utgåva (version) som levereras av ett system.

Det "dynamiska" är då att strukturen förändras tillsammans med produkten under utvecklingsarbetets gång. Arbetet har genomförts genom att gruppen ställt samman en frågelista med faktorer att ta hänsyn till under utvecklingen av ett system. Speciellt har man tänkt på hanteringen av krav.

Deltagare var:

- Ingmar Ögren, Romet
- Harriet Borgman, Ericsson Microwave Systems
- Peter Camitz, Ericsson Microwave Systems
- Billy Johansson, CelsiusTech Electronics.

### 1.1 Representation

1. Hur beskriver vi systemstrukturer (SA, UML klassdiagram, UML komponentdiagram, annat)?
2. Vilka informationsstrukturer har att göra med krav, direkt eller indirekt?
3. Hur bygger vi dessa informationsstrukturer (papper & penna, ordbehandlare, CASE-verktyg, annat)?

4. Hur representerar vi kundkrav?
5. Hur representerar vi övergripande krav på högsta nivå (Uppdrag, förmågor, användningsfall, scenarior, annat)?
6. Arbetar vi med formalisering av krav t. ex. genom att uttrycka kraven i boolesk syntax?
7. Utnyttjar vi formaliserade beteendebeskrivningar (t.ex. tillståndsdigram, pseudokod) som en del i kravarbetet och i så fall hur?
8. Hur hanterar vi krav med icke-textuella delar, som t. ex. diagram?

### 1.2 Kravkopplingar

9. Hur fördelar vi krav?
10. Hur relaterar vi olika krav till varandra, t ex kundkrav, härledda krav, funktionella krav, egenskapskrav?
11. Hur relaterar vi krav till provfall?
12. Hur hanterar vi härledda krav, som framkommer under ett utvecklingsarbete?

### 1.3 Ändringshantering

13. Hur hanterar vi information, som uppkommer under utvecklingsarbetet som t. ex. problem, användarsynpunkter, annat?
14. Hur hanterar vi förändringar i systemstrukturen under en utveckling med säkerställande att krav är korrekta och konsistenta samt hur säkerställer vi att hela strukturen är konsistent?
15. Hur hanterar vi svårigheter att ställa rätt krav,

som kan uppstå t.ex. vid långt och nära samarbete med kund, när rollerna användare - kund - leverantör, är oklara och/eller det finns motstridiga uppfattningar?

## 1.4 Beslutsvägar

16. Hur prioriterar vi mellan krav, t.ex. funktionella krav och icke-funktionella krav? (Konstruktören vet normalt vad systemet ska klara men ibland inte varför)
17. Hur hanterar vi ändringsstyrning av krav, vilka beslutsvägar finns?

## 1.5 Övrigt

18. Hur stödjer vi återanvändning av krav?
19. Hur hanterar vi krav för att stödja varianter av produkter?
20. Hur hanterar vi krav vid stegvis utveckling med ökad funktionalitet?
21. Hur genererar vi relevanta dokument (krav och andra dokument)?
22. Hur säkerställer vi att programmerare och slutanvändare bägge förstår dokumentationen och varandra?
23. Hur hanterar vi krav vid krav på
  - korta ledtider (specificering och utveckling överlappar varandra)?
  - utveckling av system i flera utgåvor med successivt ökad funktionalitet?
  - flera simultana varianter av ett system (utveckling och underhåll), dvs. för kundanpassade system?
24. Hur hanterar vi krav från underlev ?

## 2. V&V – granskningsmetodik

Granskning av dokumenten i en utvecklingsprocess är en effektiv metod att åstadkomma en given kvalitet i en produkt. Mikroprojektets mål var att samla in erfarenhet om vilka metoder och verktyg man använder vid granskning och data om hur effektiv granskningen är.

Deltagare var:

- Mari Persson, Celsius, Kockum Naval Systems, som gjorde en genomgång av en problemrapportdatabas för ett Ada programutvecklingsprojekt i syfte att finna andelen fel som borde upptäckts vid tidigare granskningstillfällen samt konsekvenserna och slutsatserna av detta.

- Billy Johansson, CelsiusTech Electronics, som samlade in erfarenheter från ett förbättringsprojekt i syfte att ta fram en anpassad och ensad metod för genomförande av formella granskningar samt skaffa erfarenheter från verktygsstöd.
- Mats Forsman, Enator Telub, som har följt upp effekterna av granskning enligt den relativt formella metod man har i företagets handbok för systemutveckling.

## 2.1 Några resultat

Några motiv man har till att genomföra formella granskningar är:

1. Formella granskningar är den mest kostnads-effektiva metoden att hitta avvikelser och förbättringsförslag. Studier visar på att granskningar har en utväxlingspotential i nytta/kostnad med en faktor >11.
2. Man kan i en enda granskning hitta mer än 80% av felen (granskningseffektivitet)
3. Utvecklingskostnaderna minskade med 50% efter införande av granskningar (IBM studie)
4. Underhållskostnader minskade med en faktor >10 genom införande av granskning (Tom Gilb studie)
5. Granskningar kan minska den totala projekt-tiden med 50%.
6. Formella granskningar tar tid i början av utvecklingen, men den totala LCC-kostnaden minskar genom att nedlagd tid i prov- och verifieringsfasen samt i underhållsfasen för en produkt reduceras.
7. Formella granskningsmöten är effektivare - man hittar fler fel tillsammans än vid en individuell (informell) granskning.
8. Oklarheter och missförstånd undviks vid formella granskningar genom att det är svårt för t.ex. författaren av ett dokument att avgöra vad som är fel om vederbörande inte har hört granskarens beskrivning av felet.
9. Viktigt att se till nyttan med granskningar, dvs. mäta effekterna av effektiva granskningar inte bara se till kostnaden för granskningen.
10. Största vinsten av genomförande av granskningar erhålls i de tidiga faserna av ett projekt - granskning av kravdokument.

Några resultat av arbetet är:

- Ungefär hälften av avvikelserna borde upp-

täckts vid något tidigare granskningstillfälle.

- Andelen specifikationsrelaterade problem är över 50%, såväl för “normala” problemrapporter som för de man borde upptäckt tidigare. Det borde löna sig att granska kravdokumenten ännu bättre
- Flertalet avvikelser har passerat två eller flera granskningar utan upptäckt.
- Ju senare felupptäckt desto större tidsåtgång för åtgärdande.
- Korrigeringstiden är längst för kravrelaterade felrapporter.
- Man skall ha en väl etablerad process för datainsamling, sammanställning till mätetal och återkoppling för att höja kvaliteten på produkterna och ge den önskade kompetensuppbyggnaden.

## 2.2 Sammanfattning

Att genomföra granskningar på ett formellt sätt är inte någon “silver bullet” som löser alla problem med produktutveckling, men det är ett kostnadseffektivt sätt att hitta fel i produkter och ger andra positiva effekter såsom “teamlärande”, dvs. kompetensutveckling. Genom att ta del av andras synpunkter och erfarenheter undviks många problem och ger, på sikt, kortare utvecklingstider.

Med formell granskning avses en procedur som omfattar planering av granskningen, en faktagenomgång inför ett granskningsmöte. Vid granskningsmötet rapporteras alla funna (potentiella) problem som identifierats vid den individuella granskningen. Vid eller efter granskningsmötet beslutas vilka åtgärder som ska vidtas. Därefter sker en uppföljning av vidtagna åtgärder.

Många studier och undersökningar visar på att det existerar brister framförallt vid granskning av kravdokument, bl.a. krav som inte är uppfattade på rätt sätt, förbisedda krav m.fl. Detta för med sig ett antal omtag i projekten som är kostnadsdrivande och ofta medför förseningar. En attitydförändring krävs ofta och att genom styrning se till att mer resurser på granskning läggs i början av produktutvecklingsfasen i allmänhet och i systemspecificeringsfasen i synnerhet.

Saknas någon av de aktiviteter som måste ingå i en effektiv formell granskningsprocess upplevs granskningen som ineffektiv. Mycket tid tas i anspråk för att diskutera t.ex. en konstruktionslösning istället för att ge synpunkter på befintligt underlag. Dessutom förbrukas ett antal timmar i

onödan för dem, som kallats som granskare. De sitter överksamma på granskningsmötet och upplever granskningen som onyttig. Så kanske man slutar att ha granskningsmöten, baserat på felaktiga slutsatser. Rätt genomförda granskningar är ett medel att minska kostnader och genomloppstider.

Skilj på att läsa ett dokument och att granska det, dvs. granskning måste ske mot källdokument. Checklistor och “design rules” är ofta ett stöd för främst oerfarna granskare.

Viktigt att ha ett granskningsteam sammansatt av flera olika roller/kompetenser (dock inte nödvändigtvis flera olika personer, en person kan ha flera roller) för att fokusera på vad som är viktigt vid det aktuella granskningstillfället och för det aktuella objektet.

Gör klart målet innan man inleder en granskning. Generella mål för granskningar:

- Säkerställa att man kan påbörja nästa fas.
- Förbättra produktkvaliteten.
- Förbättra processerna (arbetsmetoderna).
- Öka kompetensen, motivationen, utbilda medarbetare etc.

## 3. V&V, – provningsmetodik

Dynamisk verifiering av programvara innebär att man kör ett programsystem eller en modell av det. Detta till skillnad mot statisk verifiering, då man t ex granskar resultatet av en konstruktion mot förutsättningarna eller kör en kodverifierare. Det finns ett stort antal metoder för såväl dynamisk som statisk provning, men de kan vara svårt att hitta beskrivningar av dem. Målet med mikroprojektet har varit, att kartlägga några metoder för provning, se hur provning tillämpas i praktiken och att relatera detta till utveckling med matematisk formalism.

Deltagare har varit:

- Lena Sporre, FOA  
Val av indata för provning, litteraturstudie
- Håkan Edler, Chalmers  
Black-box-, white-box-, threadtesting, litteraturstudie
- Dag Folkesson. Saab AB, Gripen  
Provning resp övervakning av distribuerade RT-system i farkoster
- Bertil Lundgren, Celsius, Kockums  
Provning / felupptäckt i olika faser
- Per Larsson, Industrilogik L4i AB  
Formell programvaruutveckling med B-metoden



Arbetet har resulterat i fem rapporter enligt följande kapitel.

### 3.1 Några tekniker för provning med varierande indata samt beskrivningar av debuggning-tekniker

#### 3.1.1 Provning och provfall

Provning är en process att exekvera ett program med avsikten att finna fel.

Provfall och provdata är inte samma sak. Provdata är indata som har valts ut för att prova systemet med. Det är ibland möjligt att generera provdata automatiskt men omöjligt att generera provfall.

Ett provfall för provning med varierande indata består av en

- specifikation av indata
- beskrivning av systemfunktionerna provade med dessa indata
- utsaga om det förväntade resultatet

Eftersom man med provning inte kan påvisa frånvaron av felaktigheter utan bara visa på att programvarufel finns, bör provfallen konstrueras så att olika klasser av fel systematiskt avslöjas med ett minimum av ansträngning och tid.

Ett bra provfall bör vara så konstruerat att det

- har en hög sannolikhet att finna ett ännu inte upptäckt fel
- inte är redundant
- varken är för enkelt eller för komplext

Black-box-tekniker fokuserar på provning av funktioner. Två viktiga tekniker finns för att välja provfall:

- Indelning i ekvivalensklasser.
- Val av värden i gränserna mellan ekvivalensklasser.

#### 3.1.2 Debuggning

Till skillnad mot provningsprocessen, som består av att exekvera ett program med avseende att finna fel, är debuggning en process som utnyttjas när en indikation har erhållits på att det existerar ett fel. Indikationen kan vara resultatet från ett lyckat testfall. Provning och debuggning har således en nära koppling till varandra.

Debuggning består av aktiviteter i en tvådelad process:

1. bestämning av karaktären på och lokaliseringen av felet
2. rättning av felet

Punkt 1 omfattar här upp mot 95% av det totala arbetet.

Det finns olika typer av metoder för debuggning:

- debuggning by brute force
- debuggning by induction
- debuggning by deduction
- debuggning by backtracking
- debuggning by testing

### 3.2 Om några metoder att prova programvara

Provning av system har i grunden två olika syften: Att visa att systemet svarar mot kraven och att hitta fel. Det första syftet är dels validering av systemet, när man vill visa, att systemet fungerar så som en användare tänkt sig, dels verifiering, när man antingen granskar en konstruktion och jämför den med dess specifikation eller kör systemet mot provningsföreskrifter och visar att system beter sig som förväntat. Det andra syftet är att finna och ta bort konstruktionsfel och att förutsäga förekomsten av kvarvarande fel.

Statisk provning är när man verifierar eller validerar ett system enbart med hjälp av dess dokumentation. Det är: Granskningar, statisk programanalys och formella metoder.

Dynamisk provning gör man för att finna fel och för att uppskatta systemets tillförlitlighet och då kör man ett programsystem eller en exekverbar modell av det. Stegen i dynamisk provning är: Enhetsprov, modulprov, integrationsprov och acceptansprov.

I integrationsproven kan man använda flera olika tekniker: Top-down, bottom-up, trådad provning och back-to-back.

Gjorda experiment visar att statisk provning är mer effektiv än dynamisk provning. Man fann ett större antal fel och fler fel per tidsenhet vid statisk provning.

Statistisk provning gör man för att bedöma tillförlitligheten i ett system. Man bestämmer användningsprofilen för systemet, väljer provdata slumpmässigt ur användningsprofilen och kör provfall med dessa data och notera tiden varje gång man

observerat ett fel. När man samlat tillräckligt många noteringar om fel kan man med hjälp av statistiska metoder bedöma antalet kvarvarande fel.

Det finns två grundläggande principer för felupptäckande provning: strukturstyrd provning - white-box testing - och beteendeprovning - black-box testing. I strukturstyrd provning utnyttjar man kunskap om ett systems interna struktur för att prova alla möjliga exekveringsvägar. Det är en metod att konstruera provfall utifrån kunskapen om hur ett system är uppbyggt. I beteendeprovning utgår man från specifikationerna av ett systems gränssnitt och beteende och provar om transformationerna från indata till utdata är korrekta. Beteendeprovning används för att verifiera ett system både mot funktionskrav och övriga krav. Vid beteendeprovning kör man systemet utan kunskap om dess interna uppbyggnad, bara indata och utdata är kända. Man väljer en mängd provdata, kör systemet och observerar resultatet.

### 3.3 Provning stödd av inbyggd realtidövervakning för distribuerade system i farkoster

Effektiv provning av realtidssystem erfordrar en kvalificerad detektering/ mätning av feltillstånd, inte endast observation av felyttring/haveri (failure).

Inbyggd övervakning är effektiv och nödvändig för att garantera säkerhet och funktion med varning eller reservinkoppling vid problem. Lång erfarenhet visar även att sådan inbyggd övervakning i distribuerade realtidssystem är en värdefull grund att utnyttja även för utprovning (verifiering och validering) av realtidsexekveringen.

Inbyggd övervakning utför detektering och utvärdering av systemet under ordinarie drift.

Inbyggd test utför reguljärt "provning" av utrustning (mest hårdvara) genom speciell excitering under icke-drift, och även sådan "test"teknik är intressant för såväl ordinarie provning (verifiering/validering) som funktionsövervakning.

Inbyggd test och inbyggd exekveringsövervakning enligt ovan ingår således permanent i det ordinarie systemet.

Den inbyggda övervakningen, feldetekteringen

1. har stora likheter med utvecklingsprovningen,
2. har dessutom stora fördelar av att ha tillgång till tillstånd och data i realtidsexekutiven under exekveringsförloppet och
3. är tillgängligt under betydligt större (drift- och kalender)tid och med betydligt fler och

varierande driftfall än "normal" utvecklingsprovning kan vara m a p realtidförloppet.

Av ovanstående kan då inses att

- utvecklingsprovning/utprovning har stora likheter med den redan inbyggda övervakningen och (konstlast-) testningsfunktionerna.
- och till stor nytta kan utnyttja dessa

### 1.4 Provning / felupptäckt i olika faser

Hos Kockums delas testverksamheten för programvarusystem, som är utvecklade i Ada, in i följande faser:

- Integrationstest, INT  
Test i referenssystemmiljö med vissa teststubar, men framför allt PC-simulatorer
- Formell test , FQT  
Test i referenssystemmiljö med specificerade testplaner och testbeskrivningar, godkända av kund. PC-simulatorer används
- Systemtest i fabrik, FAT  
Test i uppbyggd landanläggning, huvudsakligen mot verkliga utrustningar, kompletterade med simulatorer för att skapa scenarier
- Systemtest ombord, SAT  
Test hos kund ombord fartygen i verklig miljö. Normal användning ombord.

Det är allmänt känt att ju tidigare man upptäcker ett fel desto enklare och snabbare är det att åtgärda felet. Tiden för att felsöka och åtgärda ett fel ökar med en 'fasfaktor' mellan varje fas.

Med dessa förutsättningar ansätter vi faktorn 4 mellan faserna integrationstest och formell test. Vid övergång till systemtest i fabrik och systemtest ombord är det sämre kontakt mellan den som upptäcker och beskriver felet, och den som skall analysera och åtgärda det. Här ansätter vi den något högre faktorn 5.

Vi gör således följande ansats för en felrättningsinsats under de olika faserna. I tiden ingår arbetet med att identifiera, analysera och korrigera ett programvarufel:

- |                               |       |
|-------------------------------|-------|
| • Under integrationstest, INT | 1 h   |
| • Under formell test , FQT    | 4 h   |
| • Systemtest i fabrik, FAT    | 20 h  |
| • Systemtest ombord, SAT      | 100 h |

Man kan då göra följande räkneexempel på en trolig besparing vid tidigare felupptäckt:

- Om 8 st SAT-fel i stället hade upptäckts under FAT, blir besparingen:  $8 \times (100-20) = 640$  h
- Om 7 st FAT-fel i stället hade upptäckts under FQT, blir besparingen:  $7 \times (20-4) = 112$  h

Kan man undvika fel i sena faser gör man avsevärda besparingar. Uppflyttningar i de tidiga faserna ger endast marginella besparingar.

### 3.5 En snabb introduktion till B-metoden

Med formella metoder i systemutveckling menar man i allmänhet användningen av logik och matematik i ett beskrivningsspråk med formell syntax och semantik. Vid användningen av formella metoder utgår man från informella krav och modellerar dem på hög abstraktionsnivå. Vinsten med detta är, dels att man beskriver vad systemet skall göra men inte hur det skall göras, dels att man får en entydig specifikation som underlag för det fortsatta utvecklingsarbetet. Man kan få en precis förståelse för kraven, men gapet till det färdiga programmet, som skall realisera funktionerna, är stort, då detaljeringsgraden i ett programspråk är för stor jämfört med det logiska beskrivningsspråket. Man kan lösa problemet med att använda delmängder av programspråket som pseudokod för att gradvis förfina specifikationerna från krav till färdigt program

B-metoden är en formell metod, som är avsedd att stödja både specifikationer på en hög abstraktionsnivå och utveckling av program på lägre nivåer. Den täcker en stor del av utvecklingskedjan till skillnad från de flesta andra formella metoder, som enbart kan användas för abstrakta beskrivningar av krav. En komplett systembeskrivning i B består av en hierarki av specifikationer med den

abstrakta systemspecifikationen på högsta nivån och en så gott som exekverbar konstruktion på den lägsta. B-metoden är avsedd att stödja utveckling av system som skall skrivas i något av de vanliga imperativa programspråken som Ada eller Pascal.

Metoden har använts i ett antal tillämpningar, där de mest kända är från det franska företaget GEC-AHLSTROM-Transport. Företaget tillverkar utrustning och levererar signalanläggningar för järnvägar. Under det senaste decenniet har man utvecklat flera stora system, där B-metoden använts i hela utvecklingsprocessen, och man rapporterar goda resultat av detta.

I Sverige har B-metoden använts i ett forskningsprojekt om användningen och integrationen av formella metoder i industriell systemutveckling. I projektet deltog både industri och akademi och ett av målen var att prova några formella metoder i praktisk användning, däribland B-metoden. Den tillämpning man valde var dokumenterad med en mängd tillståndsgrafer blandat med C-program och informella beskrivningar. Med hjälp av systemets konstruktör fann man det ganska enkelt att göra om beskrivningarna i enbart B-språket, trots att ingen av projektdeltagarna hade djupare kunskaper i vare sig metod eller språk.

Åtskilliga positiva erfarenheter finns alltså av praktisk användning av B-metoden. Förutom att vara en formell metod har den fördelarna av att ha syntax och semantik, som liknar ett imperativt språk och att det kan användas både på hög abstraktionsnivå och för förfining och detaljering ned till programkonstruktion. En van systemkonstruktör kan relativt lätt börja använda metoden och språket utan att ha lärt sig alla detaljer.

Rapporten från mikroprojektet går igenom B-metodens fundamenta.

# Forskning och Teknikutveckling för Anpassning (FoTA) –aktuellt läge

Bror-Arne Ersson, FMV sammanhållande för FoT-programmet för anpassning, rapporterar att sedan en tid är beställningar lagda till industrin på alla programvaruprojekten utom P.12, vilket av naturliga skäl måste ligga litet senare i tiden. P.12 är ju ett slags uppsamlings- och kunskapsspridningsprojekt för de övriga projekten med avseende på säkerhetskritiska aspekter. P.12 är dock på god väg vilket framgår av inlägget från FMV handläggare Inga-Lill Bratteby-Ribbing.

Följande projekt är alltså nu i full verksamhet:

## **P. 3 COTS och objektorientering som bas för konstruktion av realtidsapplikationer**

FMV handläggare: Gustaf Myhrman, FMV:UVsystS, gumyh@fmv.se

Industrins kontaktperson: Lennart.Bie@emw.ericsson.se

## **P. 4 Patterns och komponentåteranvändning**

FMV handläggare: Christopher Bengtsson, FMV:INFOSYST, chben@fmv.se

Industrins kontaktperson: Barbro Sjöland, Sjöland och Thyselius Datakonsulter, barbro.sjöland@st.se

## **P. 7 COTS-produkter i militära lednings- och informationssystem**

FMV handläggare: Jan Flodi, FMV:ELEKTRO, jaflo@fmv.se

Industrins kontaktperson: Bengt Gustafsson, begu@celsiustech.se

## **P. 9 Experimentell verifiering av feltolerans**

FMV handläggare: Gunnar Fredriksson, FMV:FlygE, gufre@fmv.se

Industrins kontaktperson: Håkan Edler, edler@hisafe.se

## **P. 10 Projekt MANA - Ett run-time system för säkerhetskritiska komplexa system (Kompletteringsprojekt till NUTEK Komplexa Tekniska Systems projekt "A Run-Time System for Safety Critical Complex Systems")**

FMV handläggare: Gunnar Fredriksson, FMV:FlygE, gufre@fmv.se

Industrins kontaktperson: Lars Asplund, asplund@Minsk.DoCS.UU.SE

## **P. 11 Formalisering, analys och hantering av krav på säkerhetskritiska system**

FMV handläggare: Per-Henrik Persson, VapenP, phper@fmv.se

Industrins kontaktperson: Göran Anger, anger@L4i.se

Medan alltså

## **P. 12 Överföring till industrin av programvaruteknik för säkerhetskritiska system**

FMV handläggare: Inga-Lill Bratteby-Ribbing, FMV:InfosystT, ilbra@fmv.se

Industrins kontaktperson: Dag.Folkesson@saab.se

är under intensiv beredning, se Inga-Lills inlägg.

Det sker en hel del organisatoriska förändringar på FMV vid millennieskiftet och möjligen också någon handläggarförändring vad gäller FoTA-projekten. Vi hoppas kunna återkomma med uppgifter om detta i nästa nummer.

# Status FoTA P12: Överföring till industrin av programvaruteknik för säkerhetskritiska system

Ett första projektmöte inom P12 hölls 99-10-20 med information om projekt, projektplan, rapportmallar samt uppgifter och ansvarsfördelning (knutet till olika roller inom projektet). Deltagarna fick i uppgift att till 5.11 ange vilken insats man är villig att bidra med som teknikupptagande företag och till den 30.11 -i mån av intresse- inkomma med anbud på huvudleverantörskapet, som innebär att inom givna ramar ombesörja samfakturering av deltagande företags kostnader samt att vid behov avlasta FMV projektledning med insamling och distribution av information mellan berörda parter.

Parallellt med detta hade även P9 och P11 projektmöten i angränsande lokaler.

Under eftermiddagen gav P12:s samarbetsprojekt inom programvaruteknik en halvtimmes presentation var. Syftet var att ge dessa en uppfattning om hur P12 kan hjälpa till samt att ge P12 (i sin roll som bollplank och RefG) klarhet i när producerat material kan finnas tillgängligt för granskning, yttranden, provning etc.

Sammanfattningsvis konstaterades att modellen med en heldag för enskilda projektvisa

f.m.-möten och gemensamma e.m.-genomgångar varit givande och bör tillämpas i fortsättningen (förslagsvis en gång per halvår). H-projekten uppmanades lämna förslag på lämpliga datum för år 2000 samt projekt-/arbetsplaner (för P12:s planering).

Anteckningar från ovanstående sammankomster samt P12:s uppdaterade projektplan har efter mötet distribuerats per mail till den koordineringsgrupp, som definierats för info-spridning mellan P12 och berörda projekt.

Samtidigt har nästa P12-möte planerats (99-12-07 från kl 09.30).

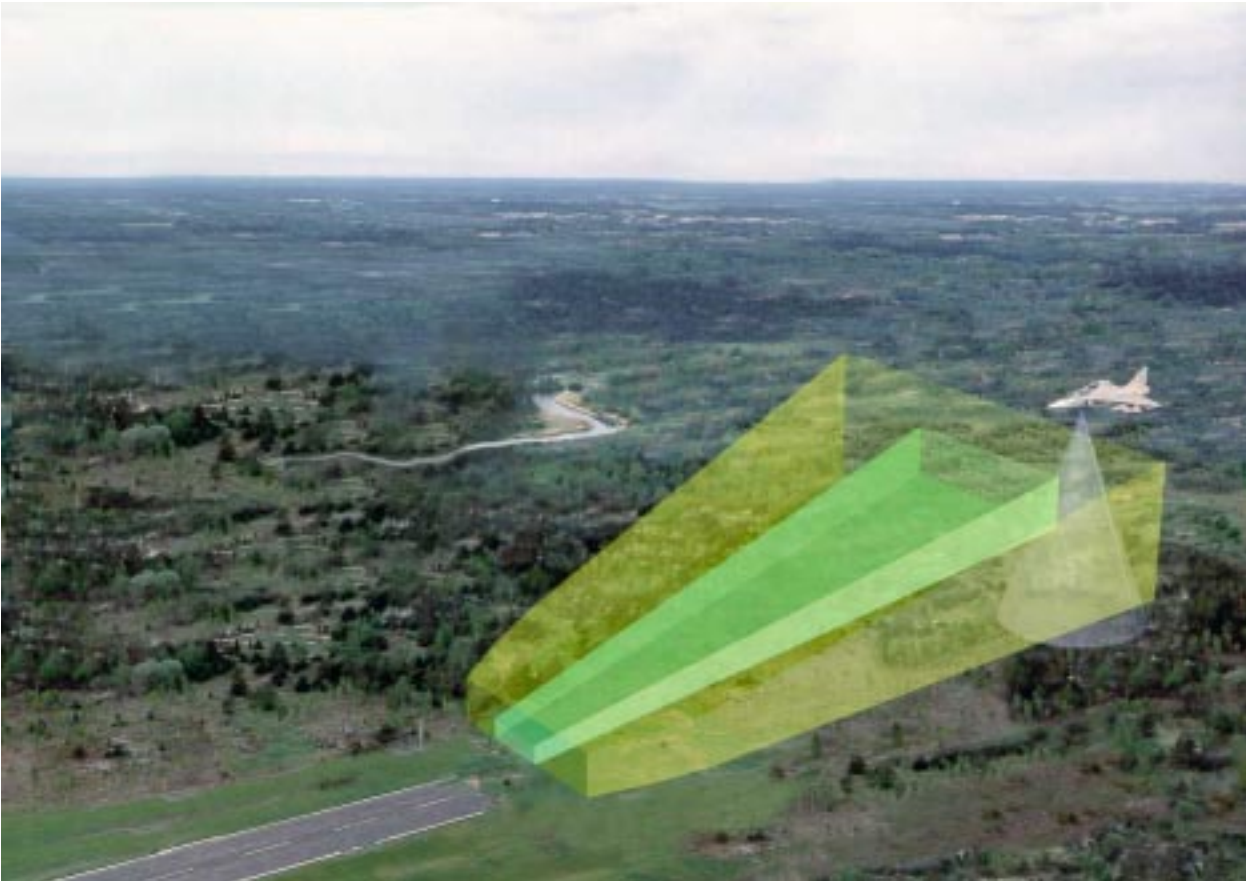
Eftermiddagsprogrammet är öppet även för övriga projekt. En introduktion i formella metoder (FM) ges som förberedelse till P12:s samverkan med P11. Det inleds med en presentation av FoTA P3 (utgick p g a sjukdom vid föregående FoTA-möte). Det avslutas med erfarenheter från ett antal FoT-projekt på hårdvarusidan av FM vid maskinvarukonstruktion.

Inga-Lill Bratteby-Ribbing  
AOL FoTA P12  
ilbra@fmv.se  
tel 018-120263

# Saab NINS

## Ett autonomt navigeringssystem

Peter Bergljung, Stefan Sjögren  
Saab Dynamics AB, SE-581 88 Linköping, Sweden  
E-mail: peter.bergljung@dynamics.saab.se  
stefan.sjogren@dynamics.saab.se



### SAMMANFATTNING

Denna artikel presenterar ett nytt angreppssätt för integrerade navigeringssystem för det 21:a århundradet. Tillämpliga avioniksystem är militära flygplan, helikoptrar, kryssningsrobotar och obemannade flygfarkoster.

Saabs Nytt Integrerat Navigerings System (NINS) är certifierbart som det primära navigeringssystemet och kan användas både för taktiska och säkerhetskritiska tillämpningar. Typiska tillämpningar för Saab NINS är noggrann godtycklig brytpunktsnavigering (RNAV) under instrumentflygregler,

indata till autopiloten, markkollisionsvarningssystem (GCAS), terrängföljning, landningssystem och vapenintialisering. Ytterligare tillämpningar är indata till syntetisk terrängpresentation på siktlinjes- eller panelindikatorer, kollisionsvarning farkostfarkost (MCAS) och passiv målinmätning.

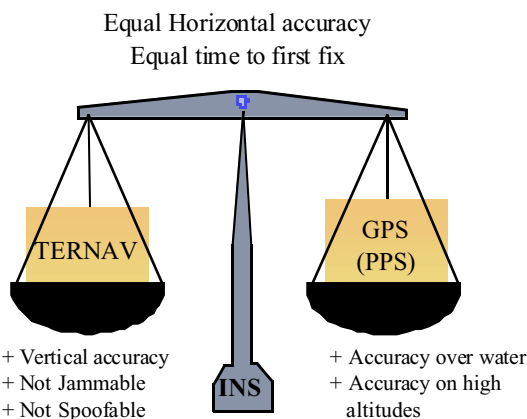
Saab NINS är världsunikt och såvitt känt det enda integrerade navigeringssystemet i världen, som är kapabelt att uppfylla dagens stränga taktiska och säkerhetskritiska krav på noggrannhet, integritet och kontinuitet.

## INTRODUKTION

Saab Dynamics har verkat inom navigeringsområdet med egenutvecklade och patenterade lösningar sedan 1970.

Integrerade navigeringssystem är inte ett nytt koncept, men det är först på senare tid som denna teknologi har mognat för både taktiska och säkerhetskritiska tillämpningar. Integrerad navigering är processen att estimeras navigeringsvariablerna position, hastighet och attityd ur en följd av mätningar med olika sensorer. Det är huvudsakligen två kategorier sensorer som används i integrerade navigeringssystem: dödräkningssensorer och positionsbestämmande sensorer. Saab NINS baseras på ett tröghetsnavigeringssystem (TN, eng. INS) som dödräkningssensor och de huvudsakliga positionsmätningssensorerna är Terräng Refererad Navigering (TRN) och satellitnavigeringssystemet Global Positioning System (GPS).

Saab Dynamics TRN-algoritm, TERNAV, har i dag nått en prestandanivå motsvarande militär GPS (PPS, Precise Positioning Service). Detta gör TERNAV till en attraktiv programvarukomponent i militära tillämpningar när GPS är föremål för avsiktlig eller oavsiktlig påverkan (d v s störning (jamming), vilseledning (spoofing) eller att den s k "selective availability" inte är påkopplad), se figur 1. Å andra sidan är GPS ett gott komplement till TERNAV över vatten och på hög höjd.



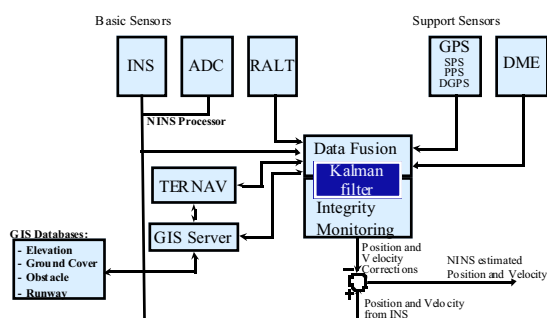
Figur 1. Den synergistiska kombinationen av positionsuppdateringar av TN-systemet från både TERNAV och GPS (PPS).

## SYSTEMÖVERSIKT

Saab NINS baseras på tröghetsnavigering tätt integrerad med ett terrängrefererat navigeringssystem (TERNAV). Systemet drar också nytta av navigeringsinformation som erhålls från luftdatasystemet (eng. ADS) och från infrastruktursystem som GPS och DME (Distance Measuring Equipment).

Den grundläggande principen bakom det integrerade navigeringssystemet, som syftar till att förbättra navigeringsprestanda, är att stödja TN-systemets navigeringsberäkningar med mätningar från de övriga positionssensorerna. Detta sker i ett utvidgat Kalmanfilter. Estimat av hastighets- och positionsfel i TN-systemet beräknas för att ge flygplanet bästa möjliga navigeringsprestanda.

Figuren nedan visar blockschemat för NINS-systemet.



### Abbreviations & Acronyms

<b>INS:</b> Inertial Navigation System	<b>DGPS:</b> Differential GPS
<b>ADC:</b> Air Data Computer	<b>TERNAV:</b> Terrain Referenced Navigation
<b>RALT:</b> Radar Altimeter	<b>GIS:</b> Geographical Information System
<b>PPS:</b> Precise Positioning Service	<b>NINS:</b> New Integrated Navigation System
<b>GPS:</b> Global Positioning System	<b>DME:</b> Distance Measuring Equipment
<b>SPS:</b> Standard Positioning Service	

Figur 2. Blockschemat över NINS-systemet.

Saab NINS består av fyra större programvarukomponenter:

**GIS** (Geografiskt Informations System) består av databaser för terränghöjd, hinder, markanvändning och start/landningsbanor. Databaserna för höjddata och markanvändning utnyttjas i TERNAV-algoritmen. Databaser för hinder och start/landningsbanor behövs inte för den grundläggande funktionaliteten, men kan läggas till senare för att förbättra systemprestanda och för utvidgade funktioner i framtiden (se t ex om NILS-tillämpningen längre fram).

**TERNAV** är en terrängrefererad algoritm utvecklad av Saab Dynamics. TERNAV utnyttjar terrängvariationer för att uppskatta flygplanets position i tre dimensioner. Som ingångsdata använder TERNAV höjd- och markanvändningsdatabaserna, höjdmätningar från radarhöjdmätaren samt information om flygplanets positionsändring mellan radarhöjdmätningarna. TERNAVs prestanda är nära relaterat till mängden terrängvariation i de områden som man flyger över.

Ett **Kalmanfilter** används som ett sensorfusionsfilter. Det använder mätningar från TERNAV, GPS och andra sensorer tillsammans med matematiska modeller av TN-plattformen och andra delar av systemet för att skatta fel i sensorerna. Kalmanfiltret levererar t ex ett estimat av TN-systemets positions- och hastighetsvektorfel. Om mätningarna avbryts av någon orsak, kan felskattningar fortfarande levereras och användas för att kompensera TN-systemet. Korrektionerna baseras då på den bästa tillgängliga kunskapen dittills beträffande TN-felen, och noggrannheten i korrigeringarna kommer att vara god så länge de verkliga TN-felen inte ändrar sig (vilket är en långsam process) från de skattade.

En viktig egenskap hos Kalmanfiltret är att det levererar osäkerheten i sin skattning tillsammans med skattningen. Detta utnyttjas i **Integritetsövervakningssystemet**. Detta är ett diagnostiskt system för felupptäckt och för att utesluta felaktiga sensorer. När ett fel upptäcks, omkonfigureras systemet till den nivå som ger bästa möjliga prestanda med de återstående felfria sensorerna ("graceful degradation").

Sensorerna som används i NINS är indelade i två grundläggande kategorier: grundläggade sensorer och stödsensorer.

**Grundläggande sensorer (Basic Sensors)** utgör basen i NINS-konceptet och ger systemet dess autonoma egenskaper.

TN-systemet är en tröghetssensor som ger de grundläggande navigeringsdata: attityd, position och hastighet. TN-systemet består av gyron och accelerometrar med hjälp av vilka hastighet, position och attityd erhålls genom död räkning.

TN-systemet har tre grundläggande styrkefaktorer. Dess skattningar levereras i hög takt, med hög noggrannhet och under mycket dynamiska flygplanmanövrar. Dessutom är det omöjligt att störa eller

vilsleda ett TN-system. Den största nackdelen med ett TN-system är dess feltillväxt, d v s med tiden innehåller positionsuppskattningen från TN-systemet ett växande fel. Ett typiskt TN-system i flygplan har en feltillväxt på 1 nautisk mil i timmen.

Luftdatasystemet (LD) innehåller trycksensorer, vilka ger barometrisk höjd, och en temperatursensor. LD-systemet levererar bl a storheterna barometrisk höjd och ytterlufttemperatur. I NINS kombineras höjden från LD-systemet med höjd från TN-systemet och TERNAV, och bildar tillsammans NINS skattade höjd.

LD-systemet är liksom TN-systemet helt autonomt. Det blir en feltillväxt i mätvärdena beroende på väderleksändringar under flygningen, d v s på grund av ändringar i statiskt tryck och temperatur.

Radarhöjdmätaren (RHM, eng. RALT) har den grundläggande funktionen hos en avståndsmätande radar. En modulerad signal sänds ut mot marken. Tiden mellan utsändning och mottagning av den från marken reflekterade signalen omvandlas till höjd. RHM mäter egentligen inte flygplanets höjd över marken utan avståndet till det närmaste föremålet inom dess strålningsdiagram (typiskt 50 grader).

Därför har RHM större svårighet att upptäcka ändringar i höjdprofilen med ökande flyghöjd. Vidare minskar RHMs noggrannhet under manövrar med stora roll- och tippändringar. För mycket stora sådana vinklar tappar RHM följningen. Den maximalt mätbara höjden över marken för en radarhöjdmätare är typiskt 1500-2500 m vid strålningsdiagram på 50 grader.

**Stödsensorer (Support Sensors)** är sådana sensorer som förbättrar systemsäkerhet och tillgänglighet, men som inte är nödvändiga för NINS-konceptets noggrannhet. En gemensam egenskap hos stödsensorerna är att de är beroende av något slags infrastruktur (på marken eller i rymden).

De viktigaste stödsensorerna är GPS (med civil, eller där den är tillgänglig, militär noggrannhet) och differentiell GPS (DGPS). Även data från det befintliga avståndsmätande systemet DME kan användas.

NINS kan uppgraderas för att utnyttja information också från andra stödsensorer som Dopplerradar, bildalstrande system, elektroniska kompasser m fl.



## NINS DETALJERAD BESKRIVNING

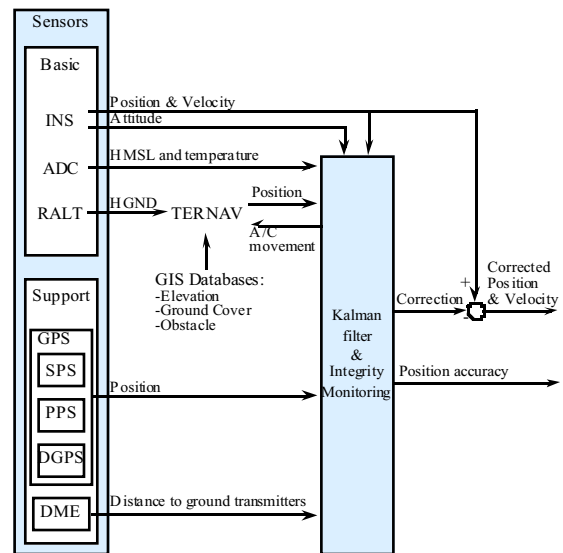
NINS datorsystem är uppbyggt kring en 225 MHz PowerPC 740 RISC-processor. Erfoderligt minne är 32 Mb RAM för programexekvering och 220 Mb Flashminne för program och GIS-databasen (täcker Sverige, Norge, Danmark och Finland). Laddning av program och GIS-data till Flashminnet sker via en Ethernetanslutning.

NINS programvara kan delas upp i realtidsexekutivet och tillämpningarna. All programvara är skriven i Ada83 under Rationals utvecklingsmiljö VADS. Planering och genomförande av test med en Ada 95 kompilator pågår.

Realtidsarkitekturen delar in tillämpningarna i ett antal periodiska och aperiodiska processer. Dessa processer styrs av realtidsexekutivet, vilket använder en prioritetsbaserad schemaläggare (scheduler) som säkerställer effektiv resursutnyttning. Därför är det lätt att integrera andra taktiska och säkerhetskritiska tillämpningar ovanpå Saab NINS.

NINS-datorn har en högupplösande realtidsklocka synkroniserad till UTC-tid. Varje sensor är tilldelad ett tidsmärkesbeskrivning som innehåller information om sensorns tidskaraktäristika. Ett fält i beskrivningen innehåller information om tidpunkten när det första datat i sensorn skapats. Hela det sensordrivna systemkonceptet baseras på det förhållandet att alla sensordata relateras till samma tidbas. I ett högprestanda navigeringssystem är det av största vikt att tidsnoggrannheten är extremt hög. Därför är det viktigt att exakt kännedom upprätthålls om tidsfördröjningen i varje sensor från den tid när data skapas och till det ögonblick då tidsmärkningen sker.

Grundprincipen bakom NINS är att kontinuerligt korrigera TN-systemets navigeringslösning, genom att bearbeta mätdata från andra sensorer i flygplanet. Detta sker genom sensordatafusion i ett utvidgat Kalmanfilter med primärt TERNAV och LD-systemet, men också med GPS och DME, om de är tillgängliga. Skattningar av hastighet och position (Kalmanfiltret skattar internt attitydfel och vissa TN-fel) beräknas för att ge flygplanet bästa möjliga navigeringsprestanda. Ett högnivåblockschema över navigeringssystemet framgår av figur 3.



Figur 3. Högnivå blockschema över NINS.

GIS programvarukomponent innehåller en topografisk databas, en markanvändningsdatabas, en hinderdatabas och en databas över start/landningsbanor.

Dessa databaser innehåller komprimerade data, av vilka vissa är komprimerade med Huffmankodning, som säkerställer att ingen information går förlorad. GIS-serverns programkomponent hanterar operationer på och kommunikation med dessa databaser.

Operationerna kan vara dekomprimering, linjär interpolation mellan mellan punkter i databasen och olika typer av datamängdsreducering. Den viktigaste funktionen hos GIS är att i förväg prediktera behovet av data i de ovan definierade databaserna, i realtid dekomprimera de data som erfordras och att upprätthålla interna cachar vilka möjliggör snabb dataåtkomst. GIS har den funktionalitet som fordras för att hantera samtidiga avläsningar i alla databaser

GIS har utvecklats för att använda ett gemensamt referensnät som inte är begränsat till Sveriges geograf. Det är allmänt i den bemärkelsen att det använder Gauss konform avbildning, vilket gör det enkelt att byta kartprojektion genom att endast ändra några konstanter.

I terrängrefererad navigering utnyttjas variationerna i terrängen för att uppnå hög noggrannhet. TERNAVs algoritm som ger uppskattningar med stor noggrannhet av flygplanets position, är en mycket

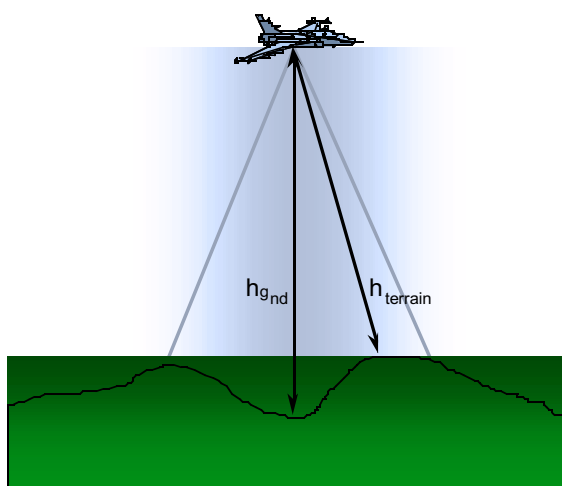
olinjär algoritm baserad på mängdteori och Bayesiansk statistik, där markprofilen under flygplanet, mätt med radarhöjdmätaren och NINS uppskattade höjd, jämförs med den i flygplanet lagrade höjddatabasen. I varje punkt  $(X, Y)$ :

$$h_{msl} - h_{gnd} - h_{top}(X, Y) = 0 \quad (1)$$

Det terrängrefererade navigeringssystemets uppgift är att hitta positionen  $(X, Y)$  hos flygplanet, givet det uppmätta värdet på höjden över marken  $h_{gnd}$ .

I praktiken kan ingendera av dessa storheter uppmätas exakt. Flygplanets höjd över havsytans medelnivå (mean sea level),  $h_{msl}$  kan inte mätas exakt vare sig av höjddaten i TN-systemet, eller av den barometrisk höjdmätningen i luftdatasystemet.

Den grundläggande funktionen hos en typisk radarhöjdmätare framgår av figur 4.



Figur 4. Fundamental funktion hos en radarhöjdmätare.

Om marken inte är helt platt, är närmaste punkt inte alltid rakt under flygplanet. Även om den är det, kan radarhöjdmätaren mäta in trädtopparna om det finns vegetation inom radarstrålen. I båda dessa fall, mäter radarhöjdmätaren ett avstånd som är för kort,  $h_{terrain}$ . Detta mätfel betecknas i det följande  $N$ . Vidare finns det alltid ett ytterligare mätbrus,  $e$ , som består av en "bias" och ett linjärt fel.

Det sanna avståndet till marken begränsas av:

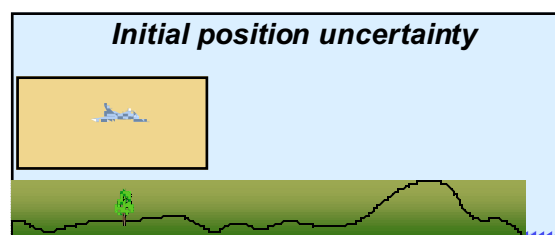
$$h_{terrain} - e < h_{gnd} < h_{terrain} + e + N \quad (2)$$

där  $N$  är det största förväntade mätfelet beroende på

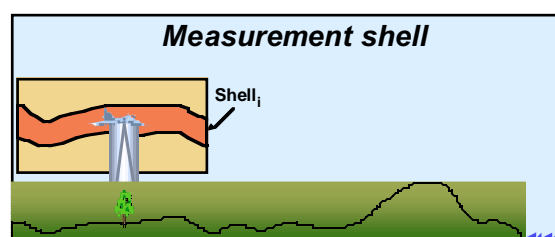
terrängen och vegetationen och  $e$  är det största förväntade mätbruset.

Topografin,  $h_{top}$ , lagras i den digitala höjddatabasen i något nät. Höjden i punkter mellan nätpunkterna kan endast interpoleras och den lagrade höjden i en nätpunkt är inte heller exakt den sanna höjden, utan snarare den sanna höjden förvanskad av databas-konstruktionsfel.

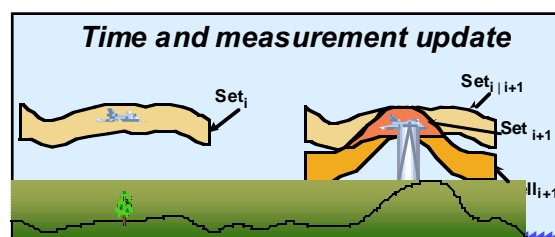
Här följer en mycket förenklad förklaring av den mängdteori som används i TERNAV-algoritmen. Funktionen hos osäkerhetsmängdens (uncertainty set) filter illustreras bäst av figurerna 5 till 7.



Figur 5. Initial positionsosäkerhet.



Figur 6. Mätskal.



Figur 7. Tids- och mätuppdatering.

I figur 5 börjar vi med en given initial osäkerhetsmängd med en tjocklek och en radie. Tjockleken och radien måste vara stora nog för att flygplanets sanna position ligger inom mängden (typiskt en 10 km radie). I figur 5 är osäkerhetsmängdens centrum flygplanets uppfattning om sin position, medan den sanna positionen indikeras av flygplansymbolen.

Antag att en mätning görs med radarhöjdmätaren vid tiden  $t_{i+1}$ , som ger avståndet till den underlig-

gande terrängen  $h_{\text{terrain}}$ . Alla punkter inom osäkerhetsmängden i den topologiska databasen slås upp. I varje punkt  $(X, Y)$ , kalkyleras en undre och en övre gräns. D v s om flygplanets laterala position är  $(X, Y)$ , begränsas dess vertikala position av gränser som ges av

$$\begin{aligned} \text{Shell}_{\text{down}}(X, Y) &= h_{\text{top}}(X, Y) + h_{\text{terrain}} - e \quad (3) \\ \text{Shell}_{\text{up}}(X, Y) &= h_{\text{top}}(X, Y) + h_{\text{terrain}} + e + N \quad (4) \end{aligned}$$

Volymen mellan dessa två ytor kallas "mätskalet",  $\text{Shell}_i$  i figur 6.

Man vet nu att flygplanet finns i både osäkerhetsmängden och i mätskalet, d v s i skärningen av osäkerhetsmängden och mätskalet. Således är skärningen den nya osäkerhetsmängden,  $\text{Set}_i$  i figur 7.

Osäkerhetsmängden flyttas med flygplanet till tiden  $\text{time}_{i+1}$ , när nästa radarhöjdmätning inträffar. Emellertid försämras mätningen av flygplanets förflyttning av osäkerheten i hastighet. Därför måste osäkerhetsmängden utvidgas i alla riktningar i enlighet med osäkerheten i hastighet.

Den flyttade och utvidgade osäkerhetsmängden betecknas  $\text{Set}_{i+1}$  ( $\text{Set}_i$  vid  $\text{time}_{i+1}$ ). Detta är den nya osäkerhetsmängden innan nästa mätuppdatering.

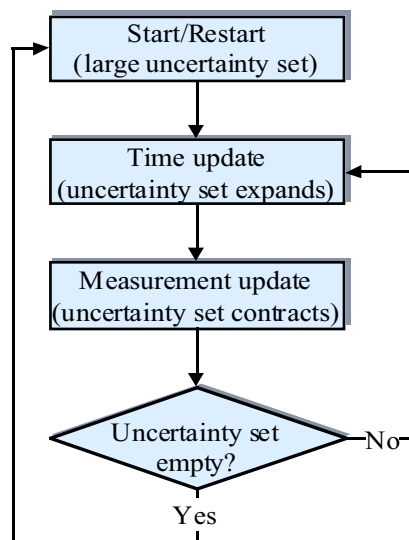
Radarhöjdmätningen ger ett nytt mätskal,  $\text{Shell}_{i+1}$  och skärningen mellan  $\text{Shell}_{i+1}$  och  $\text{Set}_{i+1}$  ger den nya osäkerhetsmängden  $\text{Set}_{i+1}$ . I figur 7 kan man se att osäkerheten i flygplanets position har minskat.

Detta förfarande med tids- och mätuppdateringar fortsätter och flygplanets positionsuppskattning förbättras gradvis. Emellertid utvidgar en tidsuppdatering mängden, medan en mätuppdatering minskar den vid karaktäristiska terrängvariationer. Om marken är platt, kommer en mätuppdatering inte att ge någon ny information och mängden påverkas ej. Därför är terrängrefererad navigering högst beroende av variationer i den underliggande terrängen.

För att uppnå hög noggrannhet i positionsskattningen, är det nödvändigt med en liten osäkerhetsmängd. Man kan se i figur 7 att osäkerhetsmängden skulle vara mindre om mätskalet vore tunnare. Dock kan större mätfel än det predikterade  $N$  orsaka att skärningsprocessen kan sluta med en tom mängd. Om detta inträffar måste TERNAV starta om med en större osäkerhetsmängd. Valet av  $N$  är en kom-

promiss mellan önskemålet om en liten osäkerhetsmängd och risken att få en tom mängd.

Den beskrivna processen sammanfattas i figur 8.



Figur 8. TERNAV processen.

För att förbättra systemprestanda använder TERNAV-algoritmen flera parallella filter med olika modellering av de definierade felkällorna. Avancerade statistiska metoder kombinerade med logik används för att kombinera olika filterskattningar för att finna den optimala flygplanpositionsskattningen. Det är en nyckel till att kunna uppnå prestanda i paritet med militär GPS.

Det **Kalmanfilter**, som är det centrala i systemet, används som ett sensorfusionsfilter. Det innefattar tillståndsmodeller för alla sensorerna. Baserat på en statistisk beskrivning av felkällorna, skattar Kalmanfiltret position och hastighet genom att jämföra informationen från olika sensorer med modellerna i Kalmanfiltret. Kalmanfiltret beräknar också osäkerheten i dessa skattningar, d v s NINS är alltid medvetet om flygplanets positionsnoggrannhet. Denna medvetenhet är en nyckelegenskap hos NINS-systemet.

På grund av den icke-linjära karaktären hos dynamiken i sensorfelen i förhållande till observerade storheter, i det här fallet position, måste en lineariserad felmodell utvecklas för att implementera Kalmanfiltrets estimator. Denna linearisering görs omkring det skattade tillståndet och den resulterande implementationen av Kalmanfiltret är såle-

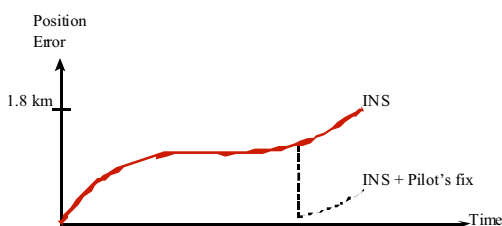
des ett utvidgat Kalmanfilter.

Kalmanfiltret jämför den skattade navigeringslösningen med ytterligare ingångsdata från de stödjande sensorerna och den "optimala" navigeringslösningen beräknas. Sensorfusionsfiltret tillförs en mätrest (residual) som beräknas som skillnaden mellan NINS position och den position som observeras av den stödjande sensorn. Skillnaden kompenseras för åldern hos mätningen, d v s algoritmen beräknar mätskillnaden vid en tidigare tidpunkt, som bestäms av det observerade datats ålder.

Om det uppstår "realtidsträngsel", väljer Kalmanfiltret att bearbeta den mest ändamålsenliga mätningen. Urvalsprocessen utvecklas för att uppnå bästa möjliga navigeringsprestanda vid varje tidpunkt oberoende av vilka sensorer som fungerar.

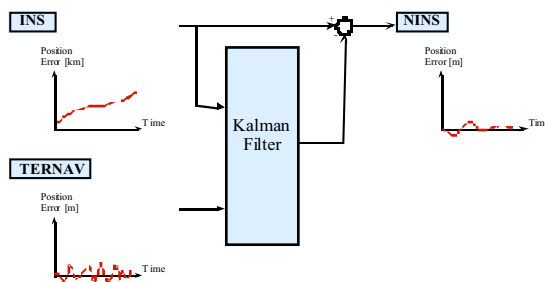
De fel Kalmanfiltret beräknar förs in i den grundläggande navigeringslösningen från TN-systemet. Detta sker som korrekationer (externa, d v s utanför TN-systemet) till positions- och hastighetsvärdena.

Den normala proceduren utan avancerade integrerade navigeringssystem är att korrigera TN-systemets position med fixtagningar som görs av piloten när flygplanet passerar kända landmärken som kyrkor eller en sjökant. Resultatet är en omedelbar återställning av TN-systemets position. Detta illustreras i figuren nedan. Observera dock att positionsfelet ökar i samma takt efter fixen.



Figur 9. Pilotens fix korrigerar TN-positionen.

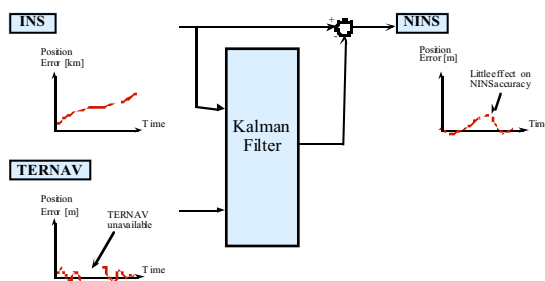
Ur figur 9 kan man lätt se att om man kunde göra fixar som de piloten gör, fast oftare, skulle man kontinuerligt kunna korrigera TN-positionen. Detta är precis vad NINS gör, men i stället för att endast använda rena positionsfixar, utnyttjar NINS både de grundläggande sensorerna och tillgängliga stödsensorer. Detta visas i nästa figur. För enkelhets skull visas endast TN-systemet och TERNAV.



Figur 10. TN-system stöttat av TERNAV.

Märk att felet i NINS är mindre än den stöttande sensorns fel (här TERNAV).

Vidare uppskattar NINS felet i TN-systemets gyron och accelerometrar. Under kortare perioder kan dessa fel anses konstanta. Med hjälp av de skattade felet, predikterar NINS den fortsatta utvecklingen av TN-systemets fel. Kom ihåg hur positionsfelet i figur 9 fortfarande ökade i samma takt efter pilotens fixtagning. Det NINS gör kan beskrivas som att lutningen skattas och elimineras. Följaktligen kan NINS fortsätta att korrigera TN-systemets utdata med noggranna justeringar, även i situationer när vissa av sensorerna inte är tillgängliga. Figuren nedan illustrerar inverkan på NINS noggrannhet beroende på en tillfällig förlust av TERNAV, t ex under överflygning av en sjö.

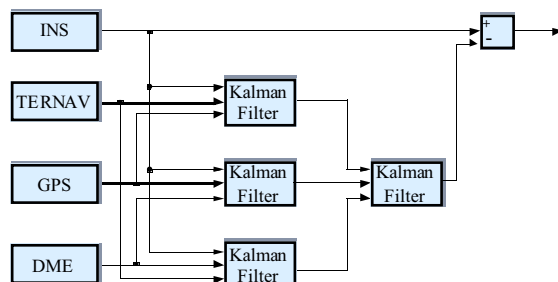


Figur 11. Tillfälligt sensorbortfall ger liten inverkan på NINS noggrannhet.

Integritetsövervakning är en komponent med syftet att säkerställa att den skattade felkovariansen från Kalmanfiltret gäller, d v s att den predikterade positionsnoggrannheten är giltig.

Noggrannhet är nödvändigt för möjligheten att utnyttja flygplanet på ett avancerat sätt, men hög noggrannhet är inte tillräckligt. Hög integritet är ett annat nödvändigt krav som oundvikligen ställs i flygsäkerhetskritiska tillämpningar. Att ha information samlad i sensorfusionsfiltret möjliggör en

effektiv tillämpning av feldetekterings- och felisoleringsalgoritmer. Att använda olika informationskällor tillsammans med kunskap om felkaraktäristika och de fysikaliska relationerna mellan de fysikaliska storheterna, är nyckeln till feldetektering genom analytisk redundans. Vad som görs med övervakningsalgoritmerna är en jämförelse av sensorvärdet med matematiska modeller av värdet från en specifik sensor och att korskontrollera detta mot andra sensorer. Märk skillnaden mot fysisk redundans. Redundant utrustning måste där tas med i flygplanet vilket är ofördelaktigt bl a ur kostnads- och viktssynpunkt.



Figur 12. Generaliserad observationsmodell för ett decentraliserat Kalmanfilter.

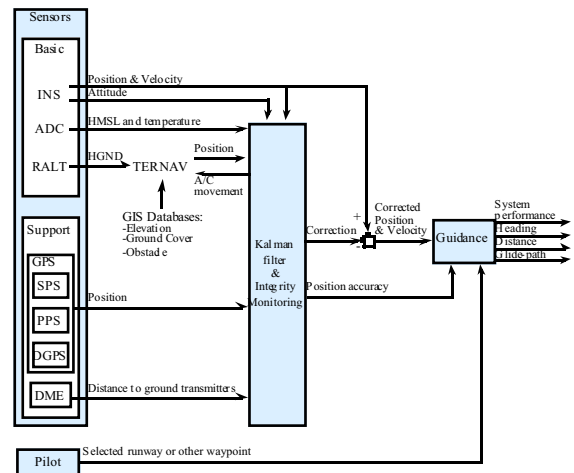
För att isolera fel, implementeras Kalmanfiltret som ett decentraliserat filter. I figur 12 visas den generaliserade observationsstrategin för ett decentraliserat Kalmanfilter. Feldetekteringsalgoritmer appliceras på utsignalerna från de olika Kalmanfiltren. Beroende på varifrån fel signaleras, kan isolering utföras. När ett felaktigt delsystem har isolerats, ger den generaliserade observatörsstrategin en lösning baserad på alla de övriga delsystemen. Alternativt, om felet kan isoleras innan det har påverkat navigeringslösningen för mycket, kan den globala lösningen användas.

Integritetsövervakningen är givetvis beroende av antalet tillgängliga positionssensorer såväl som av kvaliteten hos de stödjande sensorerna.

## TILLÄMPNING - SAAB NILS

**Saab NILS (Nytt Integrerat Landnings System)** är f n under utveckling. NILS är ett landnings-system som drar de fulla fördelarna av NINS-konceptet genom att skapa en glidbana baserad på NINS navigeringsinformation och koordinaterna för landningsbanan, se figur 13. Oberoendet av infrastruktur reducerar kostnaderna och ökar flexibiliteten genom att flygplanet under alla väder-

förhållanden kan operera från enkla och utspridda flygbaser utan markbaserad landningsutrustning. Stor positionsnoggrannhet och tillgång till information om systemprestanda är nyckelfaktorer hos NINS. Landningssystemet kommer att ha förmåga att kontinuerligt övervaka säkerhetsnivån i systemet.



Figur 13. Högnivå blockschema över NINS/NILS.

NILS kommer att certifieras för landningsminima enligt kraven för kategori 1, d v s flygplanet får gå ner till beslutshöjd (normalt 60 m , 1200 m från bantröskeln) utan att piloten visuellt ser banan. Om piloten inte får syn på banan vid beslutshöjden måste han genomföra ett omdrag (“missed approach procedure”).

Traditionellt certifieras ett landningsystem som används i ett flygplan genom certifiering av den individuella utrustningen i flygplanet. Den proceduren tar mycket lång tid för godkännande och fordrar obligatoriskt att viss specifierad utrustning är installerad ombord. Dessutom, när ett flersensor-system som NINS/NILS används, är det dess totala funktion (algoritmer, databaser, antagna statistiska egenskaper etc) som borde certifieras snarare än de enskilda utrustningarna.

De civila luftfartsmyndigheterna har utvecklat en metodik för att bestämma kraven på ett flygplans navigeringssystem, som kallas Required Navigation Performance, RNP. Det kan användas för både rutt- och terminalområdestrafik. Noggrannhet, kontinuerlighet och integritet definierar RNP-parametrarna. Detta koncept avses användas vid certifieringen av NINS/NILS

*Noggrannhet* är den parameter som beskriver avvikelser hos flygplanets position i förhållande till den ideala glidbanan.

Förlust av *kontinuitet* är parametern som beskriver risken för avbrott i funktionen under ett landningsförlopp.

Förlust av *integritet* beskriver risken att inte erhålla varning när flygplanet kommer utanför ett definierat begränsningsområde när ett fel uppstår. Ett begränsningsområde är den volym inom vilket flygplanet måste befinna sig. Den undre gränsen till begränsningsområdet är mycket nära marken. Om flygplanet penetrerar begränsningsområdet är det en stor risk för kollision med andra flygplan, omgivande terräng, höga master etc. Integritetsnivån bestäms av två termer. Den första termen är sannolikheten för att ett fel som resulterar i genombrott av begränsningsområdet ej upptäcks. Den andra är sannolikheten för ett genombrott av säkerhetsområdet med ett felfritt system (noggrannheten är för låg)

Saab NILS definierar två syntetiska tunnlar (se den inledande figuren), vilka inte presenteras för piloten. Den inre tunneln är området som är centrerat kring den önskade glidbanan där flygplanet skall befinna sig minst 95% av tiden. Den yttre tunneln är ett område centrerat kring den önskade flygbanan utanför vilket flygplanet aldrig skall befinna sig. Dessa egenskaper uttrycks normalt i kvantitativa eller probabilistiska termer (t ex 5 sigma). Det totala systemfelet, som omfattar både flygplanets och dess navigeringssystem följefel, måste uppfylla både de inre och yttre tunneldimensionerna.

Nära beslutshöjden spelar flygplanets manövrerbarhet en viktig roll när den inre tunneldimensionen bestäms. Längre bort från landningsbanan kan flygplanet flygas friare och, beroende på navigeringssystemet, kan den inre tunneln ha en annan form. Eftersom terrängen används för navigering, följer att om flygplanet flyger över ungefär samma terräng vid varje landning, kommer navigeringsprestanda inte att skilja sig påtagligt mellan landningar. Olika landningsbanor kan ha anflygningsvägar som är av helt olika karaktär, om minimering av beslutshöjden är huvudkriteriet när anflygningsvägarna väljs

Den yttre tunneln definierar det område i luften där

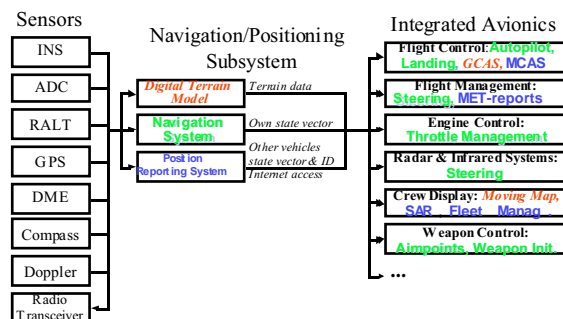
flygplanet tillåts befinna sig och där det inte är en fara för andra flygplan eller objekt på marken. Märk att dimensionerna hos den yttre tunneln inte är oberoende av den inre tunnelns dimensioner. Om den inre tunneln är för stor, ökar sannolikheten för skelfri genomträngning av den yttre tunneln och därigenom att den budget som ges av integritetskravet "äts upp".

## FRAMTIDA POTENTIAL

I denna artikel har visats ett koncept där integrerad navigering kombineras med en digital terrängmodell. Detta har potential att i framtiden ytterligare integreras med ett positionsrapporteringsystem, som levererar tillståndsvektorn, identitet och annan information (t ex DGPS-korrigeringar och anknytning till Internet) för omgivande farkoster. Möjliga ytterligare tillämpningar genom integration med positionsrapporteringsystemet är situationsmedvetenhet, "mid-air" kollisionsundvikande, management av företags flygplanspark, och räddningstjänster. Typiska positionsrapporteringsystem kan vara en taktisk datalänk som JTIDS eller VDL Mode 4.

JTIDS (Joint Tactical Information Distribution System) är ett synkront TDMA kommunikations- och navigeringssystem som arbetar i 960 – 1215 MHz-bandet.

VDL Mode 4 (VHF Data Link Mode 4) är ett självorganiserat TDMA (STDMA) kommunikations- och navigeringssystem som befinner sig i en standardiseringsprocess inom ICAO (International Civil Aviation Organisation). Systemet arbetar i 112 – 174 MHz-bandet och baseras på idéer från Håkan Lans.



Figur 14. Navigerings/positioneringssystem, dataflöde.

Figur 14 visar Navigerings- och Positionerings-

systemets dataflöde till andra avioniksystem och exempel på tillämpningar. Exempeltillämpningar som använder terrängdata från den digitala terrängmodellen är Markkollisionsvarningssystem (GCAS), Rörlig kartpresentations-, Autopilot-, Landnings-, Styrnings-, Gaspådragsstyrnings- och Vapenstyrningsfunktioner använder den egna tillståndsvektorn från navigeringssystemet. Andra farkosters tillståndsvektorer, ID och Internetkoppling levererade från positionsrapporteringssystemet används i väderrapporter (MET-reports), Mid-Air Collision Avoidance System (MCAS), Search and Rescue (SAR) och Fleet Management.

## SLUTSATSER

Ett ny ansats för integrerad navigering har beskrivits. Detta system baseras på tröghetsnavigering tätt integrerad med ett terrängrefererat navigeringssystem (TERNAV). Systemet drar också nytta av navigeringsinformation från ett luftdatasystem och från infrastrukturbaserade system som GPS och DME.

De huvudsakliga fördelarna med Saab NINS är de autonoma egenskaperna som ligger i att skapa en högprestanda tillståndsvektor (position, hastighet och attityd) hos den flygande farkosten, samt den inbyggda geografiska databasen (terränghöjd, hinder, markanvändning och start/landningsbanedata).

TERNAV har varit i tjänst i flygvapnet sedan 1995 i AJS37 Viggen. NINS och det tillhörande landningssystemet NILS har ingående flygutprovats under flera år.

## OM FÖRFATTARNA

Peter Bergljung blev civilingenjör (M) vid Linköpings Universitet 1987 och anställdes vid Saab Dynamics samma år. Sedan dess har han arbetat med utveckling av integrerade navigerings- och kommunikationssystem. Han är för närvarande Saab Dynamics projektledare för NINS/NILS-projektet i JAS 39 Gripen.

Stefan Sjögren blev civilingenjör (E) vid Linköpings Universitet 1993 och anställdes vid Saab Dynamics samma år som systemingenjör. Sedan dess har han arbetat med utvecklingen av sjömålsroboten RBS15MkII och med integrerade navigeringssystem. För närvarande är han Saab Dynamics gruppleddare för utvecklingen av NINS/NILS i JAS 39 Gripen.

Författarna vill framhålla de insatser som har gjorts i det arbete som presenterats här av de många kolleger som utgör NINS/NILS-teamet både vid Saab Dynamics och Saab Gripen, särskilt P Björklund, H Eriksson, S Eriksson, A Hagstedt, F Hofmann, K Lundberg, K-E Lundholm, D Murdin, F Neregård, J Palmqvist, N Persson, F von Post, P Pucar, M Svensson, B Olsson (nu vid FMV) and T Bergström (nu vid Saab Celsius Transponder Tech AB).

Författarna vill också tacka Bengt Fredricson vid FMV för hans stöd till forskningen inom området under nästan två dekader.

Slutligen vill författarna uttrycka sin tacksamhet för stödet från ISIS kompetenscentrum vid Linköpings Universitet; Professor Lennart Ljung och Tekn Dr Niclas Bergman et al.

\* \* \* \* \*

# SESAM möten

20 jan 2000	VU
30 mars 2000	VU
13 april 2000	Rådet
25 maj 2000	VU
25 aug 2000	VU
5 okt 2000	VU
20 okt 2000	Rådet
8 dec 2000	VU

Du besöker väl vår websajt?  
**<http://sesam.tranet.fmv.se>**

## Bidrag till Rendezvous och SESAM hemsida efterlyses

Varken Rendezvous eller hemsidan blir bättre än vad medlemmarna gör den till. Håll ögon och öron öppna för intressanta händelser i eller nära SESAM-världen och skicka in en notis till sekretariatet, [annalillk@yahoo.se](mailto:annalillk@yahoo.se). När det är aktuellt att kunna publicera rapporter eller beskrivningar från projekt som Ni är inblandade i, glöm då inte våra egna medier.

---

**SESAM-Sekretariatet:** AerotechTelub AB  
c/o Kåsjös Kontor  
Ytterspåret 14  
187 54 TÄBY

Telefon: 08-510 51866  
Telefax: 08-510 51932  
GSM: 070-716 9702  
E-post: [annalillk@yahoo.se](mailto:annalillk@yahoo.se)