

CODEN: LUTFD2/(TFRT-5387)/1-45/(1988)

# Deriverande filter för luftmålsinmätning

Hans Öhlin

Institutionen för Reglerteknik  
Lunds Tekniska Högskola  
Oktober 1988

TILLHÖR REFERENSBIBLIOTEKET  
UTLÅNAS EJ

<b>Department of Automatic Control</b> <b>Lund Institute of Technology</b> P.O. Box 118 S-221 00 Lund Sweden		<i>Document name</i> Master Thesis	
		<i>Date of issue</i> May 1988	
		<i>Document Number</i> CODEN: LUTFD2/(TFRT-5387)/1-45/(1988)	
<i>Author(s)</i> Hans Öhlin		<i>Supervisor</i> Rolf Johansson and Torbjörn Wigren	
		<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Deriverande filter för luftmålsinmätning. (Differentiating filter for maneuvering target tracking.			
<i>Abstract</i> <p>This paper illustrates the problem of target tracking, using ordinary Kalman filters. It compares the performance of a second and third order filter. The transient behavior and the ability to track a maneuvering target, are the characteristics of interest. A way to define the time to converge, using filter parameters only, is also shown. This makes it possible to evaluate the effect of, for instance, a higher measurement frequency. A slight modification of the ordinary second order Kalman filter is proposed, to adjust the filter to the specific application.</p>			
<i>Key words</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i>			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 45	<i>Recipient's notes</i>	
<i>Security classification</i>			

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

## INNEHÅLL

1.	FÖRORD	2
2.	SAMMANFATTNING	3
3.	LUFTVÄRNSLEDNING	4
4.	GRUNDLÄGGANDE TEORI	6
4.1	MÅLBANEMODELLEN	6
4.2	MÄTPROCESSEN	7
4.3	BERÄKNING AV DERIVERANDE FILTER	8
4.4	DETEKTORER OCH ÅTGÄRDER VID ALARM	11
5.	SIMULERING AV FILTERPRESTANDA	13
5.1	FILTERPARAMETRARNAS INVERKAN	13
5.2	SIMULERING MED SIMNON	17
6.	PROGRAM	22
6.1	PASCALPROGRAM	22
6.2	SIMNONPROGRAM	31
7.	APPENDIX A	41
7.1	ANDRA ORDNINGENS KALMANFILTER	41
7.2	TREDJE ORDNINGENS KALMANFILTER	41
7.3	INITIERING AV KALMANFILTER	43
8.	REFERENSER	44

## 1. FÖRORD

I denna rapport har problem kring deriverande filter för skattning av målrörelse vid luftmålsinmätning studerats. Huvudsakligen har vanliga standard Kalmanfilter använts. Dessa har sedan modifierats för att passa tillämpningen bättre. Denna rapport jämför ett andra respektive tredje ordningens filter dels med avseende på insvängningstid och dels med avseende på följförmågan. Insvängningstiden definieras genom variansen i estimaten. Filtrens ordning innebär att man antar att målhastighet respektive målacceleration är konstanta. Vidare studeras hur filteregenskaperna beror av de olika parametrarna. Följförmågan illustreras med hjälp av simuleringsspråket Simnon. I luftvärnstillämpningar har man en mycket begränsad tid till att mäta målets position och estimerar dess hastighet. Detta gör att ett tredje ordningens filter både är för långsamt och dessutom estimerar målets hastighet för dåligt, åtminstone i system som baseras på avståndslaser. Slutsatsen blir att ett andra ordningens filter med någon slags manöverdetektor är det bästa valet. Som exempel på manöverdetektorer visas två enkla som fungerar utan att beräkningsmängden svämmar över. All teori och alla program som använts redovisas samt några simuleringar över erhållna filterekvationer. Rapporten visar också att en adekvat mätfrekvens med laser är ca 5-10 Hz. I rapporten förutsätts genomgående att man hela tiden har laserloben på målet.

Avslutningsvis ber jag att få tacka de personer som hjälpt mig. Dessa är framförallt

Torbjörn Wigren  
Bofors Aerotronics AB  
181 84 Lidingö

Rolf Johansson  
Inst. Reglerteknik  
Lunds Tekniska Högskola

### 3. LUFTVÄRNSELDLEDNING

Då man använder kanoner för att bekämpa luftmål, är det inte ovanligt med skjutavstånd på 2-4 kilometer. Vid så stora skjutavstånd kommer projektilen att behöva ett 3-8 sekunder för att träffa målet. Under denna tid hinner målet att förflytta sig så mycket, att träffsannolikheten minskar högst avsevärt. I äldre system var skytten tvungen att kompensera för detta genom att sikta framför planet. I moderna system görs detta direkt genom att man beräknar den tillskottsvinkel kanonen ska ha relativt siktet. För att detta ska vara möjligt måste man känna målets hastighet, projektilens ballistik samt andra yttre faktorer, till exempel vind förhållandena.

Den enhet som gör detta kallas eldledningssystem, där beräknas alltså målets rörelsetillstånd dvs position, hastighet och eventuellt acceleration. Det enda mätbara tillståndet är positionen som mäts vid diskreta tidpunkter. Mätningen sker i polära koordinater med hjälp av laseravståndsmätare/radar samt resolverar. Mätprocessen adderar brus till positionen och problemet är att ur brusiga mätningar derivera fram hastighet och ev acceleration. Målet med detta arbete har varit att åstadkomma en mindre sammanfattning av tidsdiskreta derivande filter samt att utföra simuleringar.

#### MODELLANTAGANDET

Alla derivande filter kräver viss kännedom av processen. Detta medför, att man måste anta en banmodell för målets rörelse. Det vanligaste är att man antar att målet rör sig i en bana med konstant fart, dvs raktbana i kartesiska koordinater. En mer komplicerad modell är till exempel en med konstant acceleration. Avsikten med denna rapport är bland annat att utröna vilken modell, som ger bäst prestanda.

#### BRUSET

Filterekvationerna medger, att osäkerheten i målbaneantagandet ingår i filterberäkningarna. Vanligtvis görs detta beräkningstekniskt genom att införa ett s.k processbrus. Ett högt värde på processbrusets varians innebär att filtret litar mindre på modellbaneantagandet, vilket leder till bättre följförmåga vid omodellerade målmanövreringar. Priset för detta blir att skattningen blir brusigare. Man har alltså gjort det klassiska valet mellan bandbredd och brus känslighet. Ett rimligt krav på bruset är att det ska vara bandbegränsat. Systembruset antas verka på accelerationen ty flygplan är accelerations styrda. En förutsättning för att hålla beräkningsmängden nere är att x-, y- och z- koordinaterna antas vara oberoende av varandra. Mätningen av positionen är störd av mätbrus. Detta beror främst på kvantiseringsbrus i mätt avstånd (typiskt 5 m), samt riktarens vinkel (typiskt 0.5 mrad). Eftersom mätningen sker i polära koordinater blir mätbruset beroende av positionen i det kartesiska koordinatsystemet. Dessa olinjäriteter försummas för att beräkningsmängden ska kunna hållas nere. Mätbruset modelleras sålunda som vitt och stationärt, dvs med konstant varians i var och en av de tre koordinatriktningarna.

## 4. GRUNDLÄGGANDE TEORI

För att kunna simulera de olika filtrens prestanda har en enkel banmodell simulerats i simnon. I denna modell kan man variera ett antal parametrar, så man erhåller den bana, som antagits vid respektive filterberäkning. Modellen ger även möjlighet att undersöka filtrens följförmåga. Härnedan redovisas teorin för ett tredje ordningens system, det vill säga att man antar att målet har konstant acceleration.

### 4.1 MÅLBANEMODELLEN

Antag att

$$\frac{d^2x_3}{dt^2} = a + \omega \quad (4.1)$$

där  $a$  är en konstant acceleration och  $\omega$  är en stokastisk störning som beror på vind och pilotens manövreringar, dvs processbruset. En given förutsättning är också att det enda mätbara tillståndet är positionen. Om nu  $\omega$  är ett bandbegränsat brus, kan man införa ett extra tillstånd så att den okända parametern  $a$  försvinner. Detta görs genom att derivera positionen ytterligare en gång så att den konstanta accelerationen försvinner. Systemet får alltså den enkla matrisformen

$$\frac{dx}{dt} = Ax + V$$

$$y(t) = Cx(t) + e(t) \quad (4.2)$$

där  $x_1$  är accelerationen,  $x_2$  hastighet samt  $x_3$  är positionen.  $y(t)$  är mätstorhet (positionen) och  $e(t)$  tillhörande mätbrus. Vidare blir matriserna  $A$ ,  $C$  samt  $V$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad C = [0 \ 0 \ 1], \quad V = \begin{bmatrix} \omega \\ 0 \\ 0 \end{bmatrix}$$

De antaganden om bruset som gjorts är

- i) Bruset i  $x, y$  och  $z$  koordinaterna är okorrelerade
- ii) Brusets medelvärde är noll
- iii) Bruset har en spektraltäthet enligt

$$\Phi_v = \frac{r}{2\pi} \left( 1 - \frac{a^2}{\omega^2 + a^2} \right) \quad (4.3)$$

Att bruset får den spektraltätheten beror på att man filtrerar ett bandbegränsat brus med ett filter, som har överföringsfunktionen,  $G(s) = 1/s$ . Om nu bruset, som verkar på accelerationen, har spektraltätheten

$$\Phi = \frac{R}{2\pi} \frac{2a}{\omega^2 + a^2} \quad (4.4)$$

appendix A. Mätbruset antages vara vitt med kovariansen  $R_2$ . Man gör antagandet att bandbredden,  $a$ , är så hög att kovariansfunktionen,  $r(\tau)$ , är i stort sett noll för tidsförskjutningar skilda från noll. Detta leder till att extra tillstånd ej behövs införas, vilket minskar beräkningsmängden.

### 4.3 BERÄKNING AV DERIVERANDE FILTER

Ett andra och ett tredje ordningens Kalman filter har jämförts då det gäller insvängningstid, följförmåga samt beräkningsmängd i realtid. Dessutom har ett sk tidsvariabelt  $\alpha\beta$ -filter jämförts med ett andra ordningens Kalmanfilter. Vidare presenteras några manöverdetektorer och olika åtgärder vid alarm. Avsikten med detta arbete har inte varit att fullständigt utreda ett optimalt val av detektor utan här behandlas endast några enkla varianter. Resultaten skall ses som en fingervisning om, vad man kan åstadkomma med något slags adaptivt filter.

#### Standard Kalmanfilter

Skillnaden mellan de två Kalmanfiltren är antagandet om målets bana. Ena filtret utgår från en modell med konstant acceleration och det andra med konstant fart, annars är modellerna identiska. Filtret bygger som bekant på en prediktor och att man minimerar kvadratfelet. Kalman's optimala enstegsprediktor ges av

$$\hat{x}(k+1|k) = \Phi\hat{x}(k) + K(k)[y(k) - C\hat{x}(k|k+1)] \quad (4.8)$$

Där  $K(k)$  är den optimala förstärkningsvektorn. Rekonstruktionsfelet blir

$$\tilde{x}(k+1) = x(k+1) - \hat{x}(k+1|k)$$

Genom insättning av  $\hat{x}(k+1)$  och  $x(k+1)$  kan detta skrivas som

$$\tilde{x}(k+1) = [\Phi - K(k)C]\tilde{x}(k) + W(k) - K(k)e(k) \quad (4.9)$$

Kriteriet för den optimala förstärkningsvektorn är minimal varians i kvadratfelet. Detta kan tecknas som

$$P(k+1) = [\Phi - K(k)C]P(k)[\Phi - K(k)C]^T + R_1 + K(k)R_2K^T(k) \quad (4.10)$$

Att minimera  $\alpha^T P(k+1)\alpha$ , där  $\alpha$  är en godtycklig vektor, görs genom kvadratkomplettering, resultatet blir

$$\begin{aligned} \alpha^T P(k+1)\alpha = & \alpha^T \{ \Phi P(k)\Phi^T + R_1 - \Phi P(k)C^T (R_2 + CP(k)C^T)^{-1} CP(k)\Phi^T \} \alpha \\ & + \alpha^T \{ [K(k) - \Phi P(k)C^T (R_2 + CP(k)C^T)^{-1}] [R_2 + CP(k)C^T] \\ & [K(k) - \Phi P(k)C^T (R_2 + CP(k)C^T)^{-1}]^T \} \alpha \end{aligned}$$

Av dessa två termer beror endast den andra på förstärkningsvektorn  $K(k)$ . Genom att välja  $K(k)$ , så att denna term blir noll, minimerar man kriteriet, dvs förstärkningsvektorn blir

Där  $\Delta p(k)$  och  $\Delta v(k)$  är slumpvariabler, oberoende av mätfel, som står för avvikelser från den linjära modellen som antagits i ekvationen ovan. De antas ha medelvärdet noll samt standardavvikelserna  $\sigma_p$ ,  $\sigma_v$  samt korskovariansen  $\mu_{pv}$ . Det optimala filtret bygger på minimal varians, dvs man ska minimera medelkvadratfelet i den predikterade positionen,  $E\{\dot{x}_s(k)\}$ . Eftersom prediktionen är en linjär kombination av de glättade tillstånden blir kriteriet ekvivalent med att minimera medelkvadratfelet i både  $x_s(k)$  och  $\dot{x}_s$ . Felet i ekvation (4.13) blir alltså

$$\begin{aligned} \varepsilon\{x_s(k+1)\} &= \alpha(k+1)\varepsilon\{x_m(k+1)\} + [1-\alpha(k+1)] \\ &\quad [\varepsilon\{x_s(k)\} + h\varepsilon\{\dot{x}_s(k)\} + \Delta p(k)] \\ \varepsilon\{\dot{x}_s(k+1)\} &= \varepsilon\{\dot{x}_s(k)\} + \Delta v(k) + \frac{1}{h}\beta(k+1)[\varepsilon\{x_m(k+1)\} - \varepsilon\{x_s(k)\}] \\ &\quad - h\varepsilon\{\dot{x}_s(k)\} - \Delta p(k) \end{aligned} \quad (4.15)$$

Kvadrering och matematisk förväntan ger resultaten

$$\begin{aligned} E\varepsilon^2\{x_s(k+1)\} &= \alpha^2(k+1)\sigma_m^2(k+1) + [1-\alpha(k+1)]^2[E\varepsilon^2\{x_s(k)\} + \\ &\quad + 2hE\varepsilon\{x_s(k)\dot{x}_s(k)\} + h^2E\varepsilon^2\{\dot{x}_s(k)\} + \sigma_p^2(k)] \\ E\varepsilon^2\{\dot{x}_s(k+1)\} &= \beta^2(k+1)/h^2[\sigma_m^2(k+1) + E\varepsilon^2\{x_s(k)\}] - \\ &\quad - [2\beta(k+1)(1-\beta(k+1))/hE\varepsilon\{x_s(k)\dot{x}_s(k)\} \\ &\quad + (1-\beta(k+1))^2E\varepsilon^2\{\dot{x}_s(k)\} - 2\mu_{pv}(k)\beta(k+1)/h \\ &\quad + \sigma_v^2(k) + \beta^2(k+1)/h^2\sigma_p^2(k)] \end{aligned} \quad (4.16)$$

De partiella derivatorna m a p  $\alpha(k+1)$  och  $\beta(k+1)$  sättes till noll och den optimala viktningen blir

$$\begin{aligned} \alpha(k+1) &= \left[ E^2\varepsilon\{x_s(k)\} + 2hE\varepsilon\{x_s(k)\dot{x}_s(k)\} + h^2E\varepsilon^2\{\dot{x}_s(k)\} + \right. \\ &\quad \left. + \sigma_p^2(k) \right] / \left[ \sigma_m^2(k+1) + E\varepsilon^2\{x_s(k)\} + 2hE\varepsilon\{x_s(k)\dot{x}_s(k)\} + \right. \\ &\quad \left. + h^2E\varepsilon^2\{\dot{x}_s(k)\} + \sigma_p^2(k) \right] \\ \beta(k+1) &= \left[ hE\varepsilon^2\{x_s(k)\dot{x}_s(k)\} + h^2E\varepsilon^2\{\dot{x}_s(k)\} + h\mu_{pv}(k) \right] / \\ &\quad \left[ \sigma_m^2(k+1) + E\varepsilon^2\{x_s(k)\} + 2hE\varepsilon\{x_s(k)\dot{x}_s(k)\} + \right. \\ &\quad \left. + h^2E\varepsilon^2\{\dot{x}_s(k)\} + \sigma_p^2(k) \right] \end{aligned} \quad (4.17)$$

Ovanstående filterekvationer förutsätter, att korskovariansen är känd och kan beräknas rekursivt. Uttrycket för denna blir tämligen långt och presenteras därför i appendix A tillsammans med beräkningar av begynnelsevärden. Liksom tidigare kan



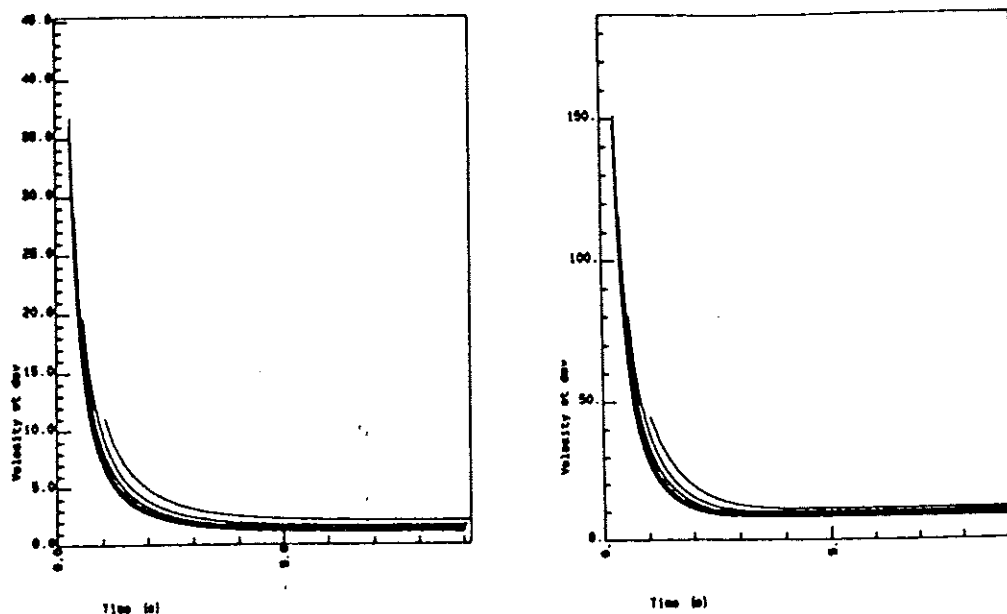
Har man en bra skattning av variansen kan man antingen jämföra den med en tröskel som tidigare eller direkt sätta in värdet i filterekvationerna.

#### Åtgärder vid alarm

Den enklaste och kanske vanligaste åtgärden vid alarm är, att man backar i iterationen av förstärkningsvektorn. Det är ett mycket rättframt sätt som bygger på antagandet att målets avvikelser från antagen banmodell inte har någon längre varaktighet. Eftersom förstärkningsvektorn ligger förberäknad i ett minne kommer detta inte att kräva några omfattande beräkningar. Ett alternativ till detta är t ex att man skattar en eller flera parametrar i filterekvationerna och använder skattningen för att datera upp efter de nya förutsättningarna.

Laser PRF (Hz) : 1.0- 20.0  
Vel. stand. dev. 2:nd order filter

Vel. stand. dev. 3:rd order filter



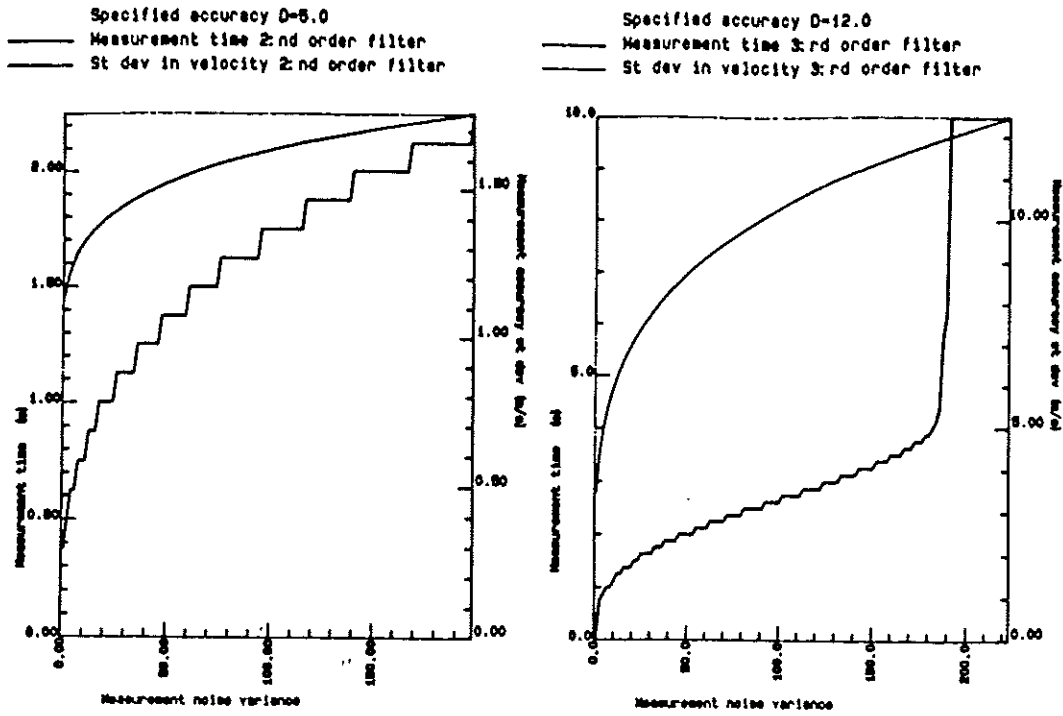
Figur 5.1.1 Teoretisk standard avvikelse i hastighetsestimaten

De parametrar som man kan variera är

- Processbrusets varians,  $R = 9 \text{ (m/s}^2\text{)}^2$
- Processbrusets bandbredd,  $A = 20.0 \text{ Hz}$
- Mätningens varians,  $R_2 = 100 \text{ m}^2$
- Samplings intervallet,  $h = 0.125 \text{ s}$

De värden som angivits är de standardvärden, som använts då kovariansmatrisen itererats. Figur 5.1.1 visar tydligt att ett andra ordningens filter är noggrannare. Detta därför att målet antas röra sig i en enklare bana.

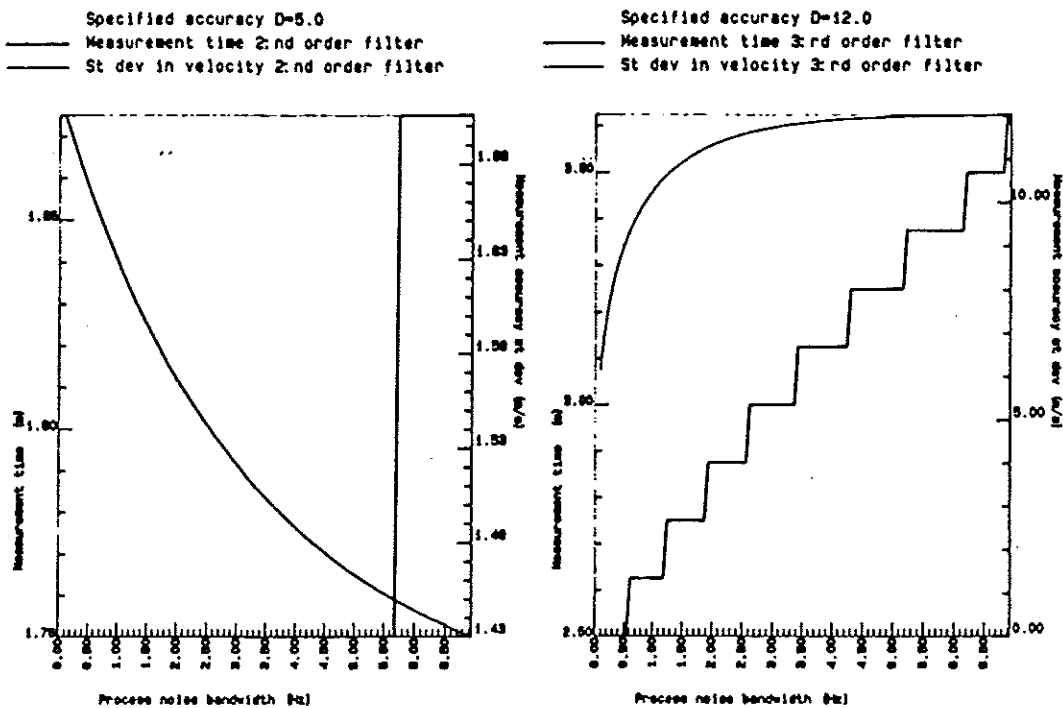
En designparameter, som har avgörande betydelse för filtrets prestanda, är mätfrekvensen. En hög mätfrekvens ger många värden och därmed ett noggrannare filter. Problemet är att snabba avståndslasrar är mycket dyra. Diagram 5.1.2 visar hur mät noggrannhet och insvängningstid hos ett andra respektive tredje ordningens Kalmanfilter beror på mätfrekvensen.



Figur 5.1.3 Insvängningstid och hastighetsestimatets noggrannhet s.f.a mätbrusvariansen.

Detta är i viss mån fingerade värde då ju mätningen sker i polära koordinater.

I figur 5.1.4 visas hur processbrusets bandbredd påverkar systemet. Det är en mycket blygsam inverkan, vilket beror på att brusets effekt inte kommer att ändras, då bandbredden ändras. Den totala effekten är konstant men fördelas endast inom ett större frekvensområde.

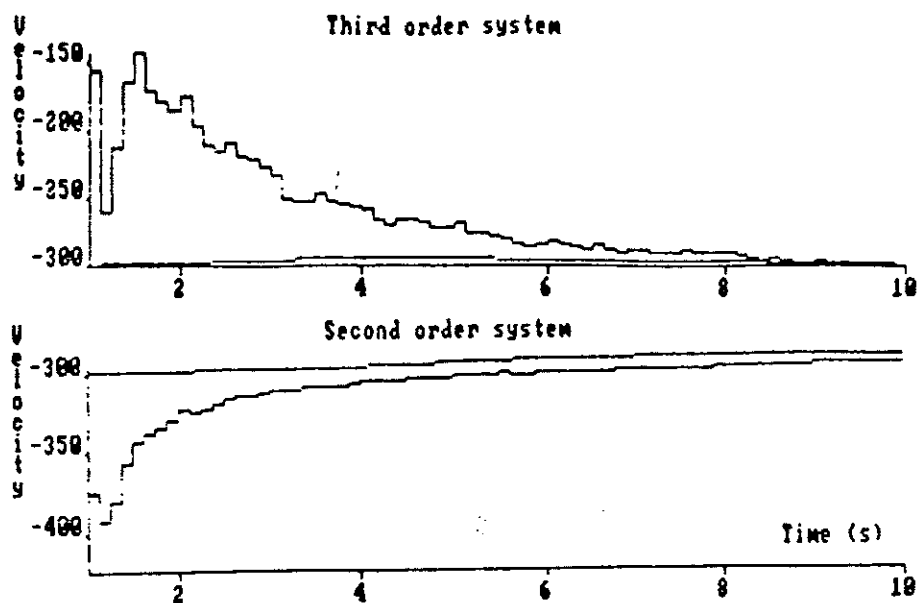


Figur 5.1.4 Inverkan av processbrusets bandbredd

## Filterinitieringar

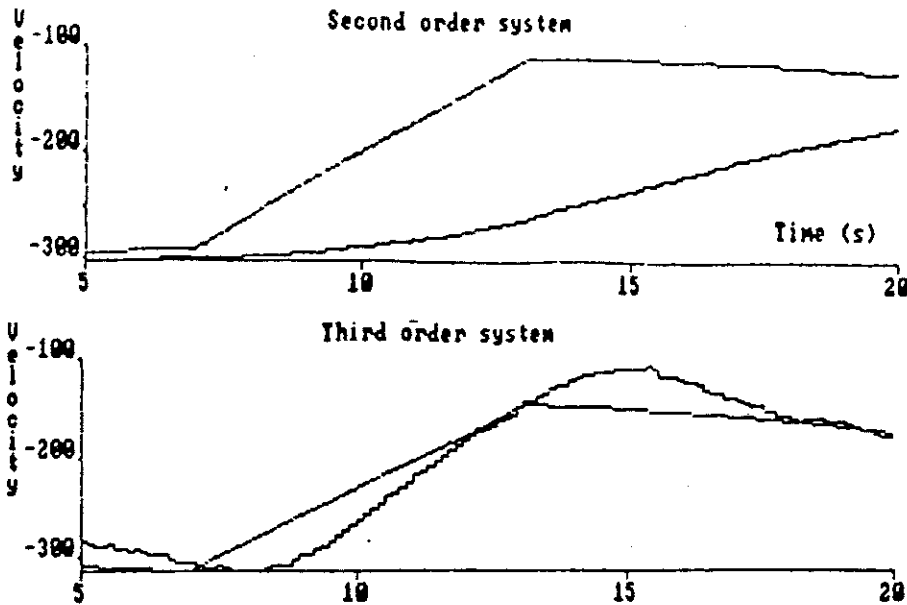
Samtliga filter måste initieras. Ett naturligt val av begynnelsevärden är det första mätvärdena. Prediktionerna av derivatorna initieras med vanlig differensbildning, och därefter är det enkelt att räkna ut motsvarande begynnelsevärde i variansmatrisen. Dessa är väntevärdet av linjärkombinationer av mätvärden med kända varianser (se appendix A).

Nedan visas som jämförelse de två standard kalmanfilter som studerats. De visas både i rakbanefallet och fallet med konstant acceleration.



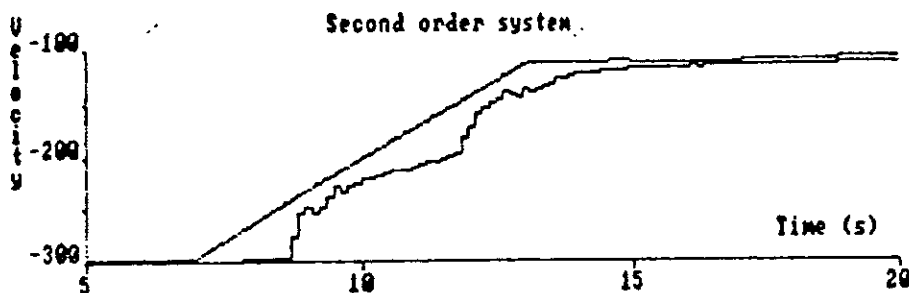
Figur 5.2.1 2:a resp 3:e ordningens kalman filter med mål som rör sig i rakbana.

Den så gott som konstanta hastigheten kring  $-300$  m/s är den sanna hastigheten. Den är endast tillgänglig vid simuleringar och visas här som referensvärde till estimerad hastighet. Här ser man klart att om approximationen som säger att målet går på rakbana, inte är för grov, finns det ingen anledning att ha ett filter med högre ordningstal. Inmätningen är både snabbare och noggrannare med ett andra ordningens filter på ett rakbanemål. Skulle däremot målet kurva kommer den ha en bana, som mer liknar approximationen med konstant acceleration. Figuren nedan visar, hur de båda filtren beter sig i detta fall.



Figur 5.2.4 2:a resp 3:e ordningens Kalmanfilter med målbana som har en ändring i accelerationen.

Denna situation klarar inte filtren av, utan att man modifierar dem. Det finns två olika tillvägagångssätt. Det första är, att man använder ett tredje ordningens filter med en annan parameteruppsättning. Tyvärr får man med ett sådant filter alltför dålig noggrannhet. Det andra sättet är att man tillfogar en manöverdetektor, som ger alarm vid målmanöver. Vid alarm har man ett par olika åtgärder att välja mellan. Figur 5.2.5 visar ett andra ordningens filter, som har en FIR-detektor och som åtgärd itererar man förstärkningsvektorn på nytt. Denna manöverdetektor tar de sju senaste residualerna och bildar medelvärdet av dessa. Då detta medelvärde är större än en fix tröskel ges alarm.



Figur 5.2.5 2:a ordningens FIR-detektor, ny iteration av förstärkningsvektorn vid alarm

Problemet med denna är att det är svårt att praktiskt bestämma en tröskel, som inte ger för många falska larm. Vid simuleringarna har den bestämts empiriskt. Den kommer givetvis att variera beroende på parameteruppsättning. Positionsmätningarna kommer att ha ett fel, som ligger inom den noggrannhet som angivits. Detta är

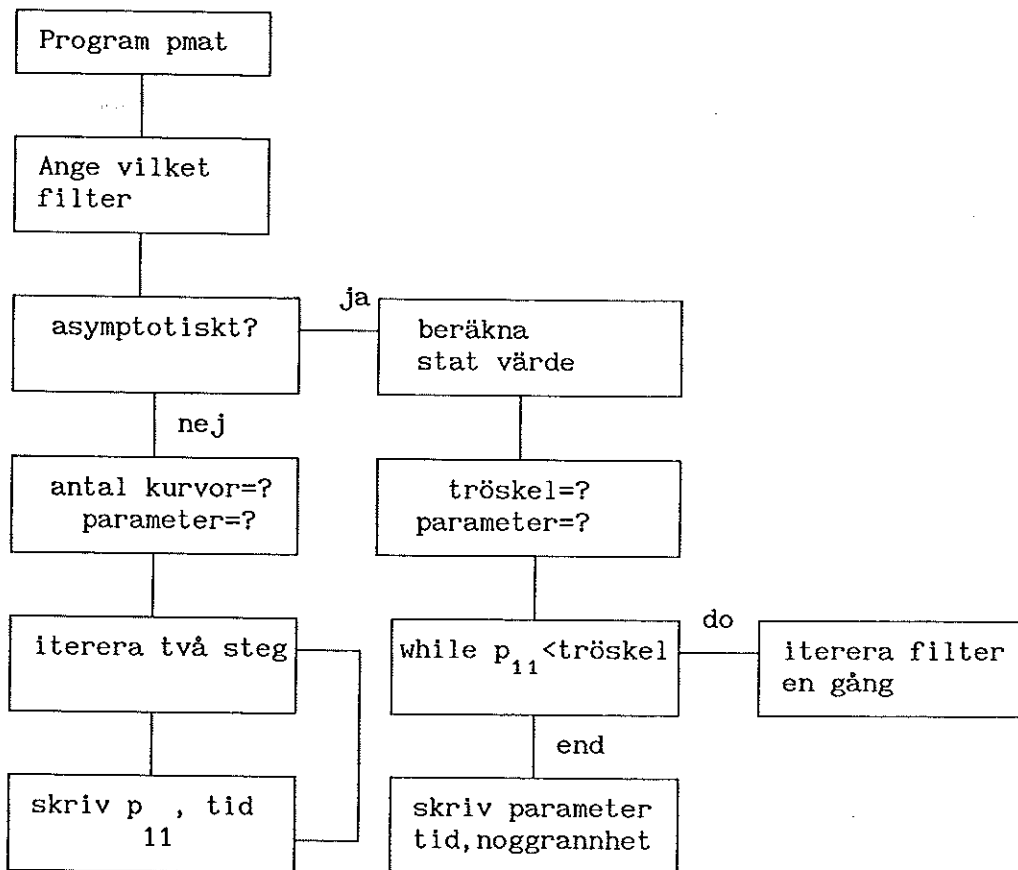
## 6. PROGRAM

I detta kapitel beskrivs de program som används för att simulera dels målbana och dels mätprocess. I avsnitt 6.1 redovisas det pascal program som beräknar konvergenstider och hur filtren konvergerar för olika parameter- val. I avsnitt 5.2 ges både ett flödesschema och en kort förklaring till de simnon program som finns.

### 6.1 PASCALPROGRAM

För att undersöka parameterberoendet i filterekvationerna har ett program skrivits i pascal. Detta itererar endast p-matrisen, antingen till en angiven noggrannhet varpå iterationstid, parametervärde samt asymptotisk noggrannhet tabelleras, eller tabelleras tidsutvecklingen hos hastighetssestimatets varians . Detta görs med avseende på någon parameter.

Nedan visas ett schematiskt blockschema till programmet.



Anledningen till att man itererar ekvationerna två steg vid det tidskontinuerliga fallet är, att ändringen mellan två iterationer är liten. Utdata skrivs i en fil som heter 'Plotdatafil.dat'. Denna fil förses även med rubriker etc som plottern sedan skriver ut. Plotprogrammet bifogas ej. I övrigt betraktas programmet som självförklarande.

```

        P23:=R2/T ;
        P33:=R2 ;
end;

(*****
(*This subroutine writes the headings in the output file      *)
(* 'plotdatafil.dat'                                         *)
)

procedure headings;

var      h      : integer;
        info    : char;

begin
  writeln('Vad {r utdata ? 1 = m{ttid(.) samt noggrannhet(.)}');
  writeln('          2 = noggrannhet(tid)      ');
  writeln('          3 = noggrannhet(.)      ');
  writeln('          4 = tid(.)');
  readln(asym);
  writeln('Asympt. st dev i hast est: p11=',sqrt(p211));
  writeln('          p22=',sqrt(p22));
  writeln;
  writeln('Ange troskeln som definierar konvergenstiden');
  write('D = ');
  read(D);
  D:=D*D;
  writeln;
  writeln('Vilket filter ? (1=b}da,2,3)');
  readln(filter);
  writeln;
  repeat
  ok := true ;
  writeln
  ('Vilken parameter onskas varieras ?(f=1,R2=2,A=3,R=4)');
  read(ans);
  writeln;
  case ans of
    1 : xlabel:='Laser PRF (Hz)';
    2 : xlabel:='Measurement noise variance ';
    3 : xlabel:='Process noise bandwidth (Hz)';
    4 : xlabel:='Process noise variance ';
    otherwise ok := false ;
  end;
  until ok ;
  writeln
  ('Ange intervall for parametervariationen (min max)');
  read(parmin);
  read(parmax);
  writeln;
  writeln('Ange antal punkter (kurvor) som ska plottas ');
  read(antal);
  if ans<>1 then
    inc := (parmax-parmin)/antal
  else
    inc := (parmax-parmin)/(parmax*parmin*antal);
  open(utdata,'plotdatafil.dat');
  rewrite(utdata);

```

```

        read(skip);
    end;
3 : ylabel:='Measurement time (s)';
4 : begin
        ylabel:='Estimation accuracy (m/s)';
        writeln('Ange tidstrlskel');
        read(Dtid);
    end;
end;
writeln(utdata,xlabel);
writeln(utdata,ylabel);
if plots>1 then
    writeln(utdata,zlabel);
end;

(*****)
(* This procedure increments the parameter. *)

procedure increment;

begin
    case ans of
        1 : begin
            T:=1/parmax+(i-1)*inc;
            parameter[1]:=1/T;
            end;
        2 : begin
            R2:=parmin+(i-1)*inc;
            parameter[2]:=R2;
            end;
        3 : begin
            A:=(parmin+(i-1)*inc)*2*3.14;
            parameter[3]:=A/(2*3.14);
            end;
        4 : begin
            R :=parmin+(i-1)*inc;
            parameter[4]:=R;
            end;
    end;
end;

(*****)
(* This procedure calculates the 2:nd order P-matrixes *)

procedure step2 ( stepsize : integer; var tid : real);

var    i
      : integer ;

begin
    for i:= 1 to stepsize do
        begin
            NP211:=P211-P212*P212/(R2+P222)+R211;
            NP212:=T*P211+P212-(T*P212+P222)*P212/(P222+R2)+R212;
            NP222:=T*T*P211+2*T*P212+P222+R222;
            NP222:=NP222-(T*P212+P222)*(T*P212+P222)/(R2+P222);

```



```

begin
  for i:=1 to antal do
    begin
      tid2:=0.0;
      tid3:=0.0;
      increment;
      initial;
      while (p211>D) and (tid2<15.0) do
        step2(1,tid2);
        step2(500,trash);
writeln(utdata,parameter[ans], ' ',tid2, ' ',sqrt(p211),2);
      end;
      for i:=1 to antal do
        begin
          increment;
          tid3:=0.0;
          initial;
          while (p22>D) and (tid3<15.0) do
            step3(1,tid3);
            step3(500,trash);
writeln(utdata,parameter[ans], ' ',tid3, ' ',sqrt(p22),4);
          end;
        end
      else if (filter=2) then
        begin
          for i:=1 to antal do
            begin
              tid2:=0.0;
              tid3:=0.0;
              increment;
              initial;
              while (p211>D) and (tid2<15.0) do
                step2(1,tid2);
                step2(500,trash);
writeln(utdata,parameter[ans], ' ',tid2, ' ',sqrt(p211),2);
              end;
            end
          else
            begin
              for i:=1 to antal do
                begin
                  increment;
                  tid3:=0.0;
                  initial;
                  while (p22>D) and (tid3<15.0) do
                    step3(1,tid3);
                    step3(500,trash);
writeln(utdata,parameter[ans], ' ',tid3, ' ',sqrt(p22),4);
                  end;
                end;
            end;
          end;
        end
      2: begin
        if filter=1 then
          begin
            for i:=1 to antal do
              begin
                increment;

```

```

3: begin
    for i:=1 to antal do
    begin
        tid2:=0.0;
        increment;
        initial;
        while (p211>D) and (tid2<15.0) do
        begin
            step2(1,tid2);
        end;
        writeln(utdata,parameter[ans],', ',tid2,4);
    end;
end;
4: begin
    for i:=1 to antal do
    begin
        tid2:=0.0;
        increment;
        initial;
        while (tid2<Dtid) do
        begin
            step2(1,tid2);
        end;
        writeln(utdata,parameter[ans],', ',sqrt(p211),5);
    end;
end;
end;      (* end case *)

close(utdata);
writeln('Reultatet finns i filen plotdatafil.dat');
end.

```

## DISCRETE SYSTEM WIND

" This system generates discrete noise to be added to the  
"cartesian components of the acceleration, that is the components  
"commanded by the pilot.

output exwind eywind ezwind

state ex ey ez  
new nex ney nez

time t  
tsamp tsa

devx = stdevx\*sqrt((1-exp(-b\*deltat))/b)  
devy = stdevy\*sqrt((1-exp(-b\*deltat))/b)  
devz = stdevz\*sqrt((1-exp(-b\*deltat))/b)

nex = devx\*norm(t)  
ney = devy\*norm(t)  
nez = devz\*norm(t)

exwind = nex  
eywind = ney  
ezwind = nez

tsa = t + deltat  
deltat : 0.05  
stdevx : 20  
stdevy : 20  
stdevz : 20  
b : 6.28

END

## CONTINUOUS SYSTEM INPUT

" This system creates the commanded acceleration to the system  
"TARGET. The acceleration is assumed to increase after a certain  
"time due to the pilot's maneuver.

output uxacc uyacc uzacc  
time t

uxacc = if t<40 then 0 else if t<80 then axcomm else 0  
uyacc = if t<40 then 0 else if t<80 then aycomm else 0  
uzacc = if t<40 then 10 else if t<80 then azcomm else 10

axcomm : 10  
aycomm : 10  
azcomm : 10

END

## CONTINUOUS SYSTEM TARGET

```

y2 = sty2
y3 = sty3

z2 = stz2
z3 = stz3

g : 10
stz3 : 300
stz2 : -10
sty3 : 2500
sty2 : -200
stx3 : 3000
stx2 : -300

```

END

#### DISCRETE SYSTEM SAMPLE

" This system takes samples of the system TARGET with a sampling  
"period of tsamp. It transforms the cartesian coordinates into  
"polary, adds noise and finally retransforms the system and supply  
"the kalman-filter with input.  
"notice that x2=velocity, x3=position etc

```

input x2 x3 y2 y3 z2 z3
output resx3 ix0 ix1 ix2

```

```

state x22 x11 x00
new nx22 nx11 nx00

```

```

time t
tsamp tsa

```

```

nr = stdevr*norm(t)
nth = stdevt*norm(t)
nph = stdevp*norm(t)

```

```

newr = sqrt (x3*x3+y3*y3+z3*z3) + nr
newth = atan2(z3,sqrt(x3*x3+y3*y3)) + nth
newph = atan2(y3,x3) + nph

```

```

resx3 = newr*cos(newth)*cos(newph)

```

```

nx00 = resx3
nx11 = x00
nx22 = x11

```

```

ix2 = x22
ix1 = x11
ix0 = x00

```

```

tsa = t + h

```

```

stdevr : 10

```

## DISCRETE SYSTEM ERAK

"This system calculates the optimal gain vector and the  
"Kalman one step ahead prediction under the assumption  
"that the plane has constant speed.

input p12 p22 messx2 ix0 ix1

state x1 x2  
new nx1 nx2

time t  
tsamp tsa

INITIAL  
tsa = 2\*h  
SORT

k1 = p12/(r2+p22)  
k2 = (h\*p12+p22)/(r2+p22)

errx2 = messx2 - x2

nx1 =if tsa>3\*h then x1+k1\*errx2 else (ix0-ix1)/h  
nx2 =if tsa>3\*h then h\*x1+x2+k2\*errx2 else ix0

tsa = t + h  
r2 : 100  
h : 0.1

end

## CONNECTING SYSTEM CONRAK

uxacc[target]=uxacc[input]  
uyacc[target]=uyacc[input]  
uzacc[target]=uzacc[input]

exwind[target]=exwind[wind]  
eywind[target]=eywind[wind]  
ezwind[target]=ezwind[wind]

x2[sample]=x2[target]  
x3[sample]=x3[target]

y2[sample]=y2[target]  
y3[sample]=y3[target]

z2[sample]=z2[target]  
z3[sample]=z3[target]

p12[erak]=pout12[prak]  
p22[erak]=pout22[prak]

messx2[erak]=resx3[sample]  
messy2[erak]=resy3[sample]

```

input r11 r12 r13 r22 r23 r33
output pout13 pout23 pout33

state p11 p12 p13 p22 p23 p33
new pn11 pn12 pn13 pn22 pn23 pn33

```

```

time t
tsamp ts

```

```

INITIAL
ts = 4*h
p11 = 6*R2/(h*h*h*h)
p12 = 3*R2/(h*h*h)
p13 = R2/(h*h)
p22 = 2*R2/(h*h)
p23 = R2/h
p33 = R2
SORT

```

```

pn11 =p11-(p13*p13)/(p33+R2)+r11
pn12 = h*p11+p12-(p13*(h*p13+p23))/(p33+R2)+r12
pne13 = (h*h)/2*p11+h*p12+p13+r13
pn13 = pne13-(h*h*p13*p13/2+h*p13*p23+p13*p33)/(p33+R2)
pne22 = h*h*p11+2*h*p12+p22+r22
pn22 =pne22-(h*p13+p23)*(h*p13+p23)/(p33+R2)
pne23 = h*h*h*p11/2+3*h*h*p12/2+h*p22+h*p13+p23+r23
pn23 = pne23-(h*p13+p23)*(h*h*p13/2+h*p23+p33)/(p33+R2)
pne33 = h*h*h*h*p11/4+h*h*h*p12+h*h*p13+h*h*p22
pa33 = 2*h*p23+p33+r33
pb33 = (h*h*p13/2+h*p23+p33)*(h*h*p13/2+h*p23+p33)
pn33 =pne33+pa33-pb33/(p33+R2)

```

```

ts = t + h
pout13 = p13
pout23 = p23
pout33 = p33

```

```

R2 : 100
h : 0.1

```

```

END

```

```

DISCRETE SYSTEM ESTIM

```

```

" This system estimates the messured vector x with the one step
"ahead Kalman predictor.

```

```

input k1 k2 k3 messx3 ix2 ix1 ix0

```

```

state x1 x2 x3
new nx1 nx2 nx3

```

```

time t

```

```

x2[sample]=x2[target]
x3[sample]=x3[target]

y2[sample]=y2[target]
y3[sample]=y3[target]

z2[sample]=z2[target]
z3[sample]=z3[target]

r11[pmat]=var11[varmat]
r12[pmat]=var12[varmat]
r13[pmat]=var13[varmat]
r22[pmat]=var22[varmat]
r23[pmat]=var23[varmat]
r33[pmat]=var33[varmat]

p13[ gain]=pout13[pmat]
p23[ gain]=pout23[pmat]
p33[ gain]=pout33[pmat]

messx3[estim]=resx3[sample]
messy3[estim]=resy3[sample]
messz3[estim]=resz3[sample]

ix2[estim]=ix2[sample]
ix1[estim]=ix1[sample]
ix0[estim]=ix0[sample]

iy2[estim]=iy2[sample]
iy1[estim]=iy1[sample]
iy0[estim]=iy0[sample]

iz2[estim]=iz2[sample]
iz1[estim]=iz1[sample]
iz0[estim]=iz0[sample]

k1[estim]=k1[ gain]
k2[estim]=k2[ gain]
k3[estim]=k3[ gain]

xm[alfa]=resx3[sample]
ix0[alfa]=ix0[sample]
ix1[alfa]=ix1[sample]

```

END

## 7.2 TREDJE ORDNINGENS KALMANFILTER KOMPLEMENTVIS

Kovariansmatrisen blir

$$r_{11} = 2R \left[ 1 - e^{-ah} \right]$$

$$r_{12} = R \left[ h - h e^{-ah} \right]$$

$$r_{13} = R \left[ \frac{h^2}{2} - \frac{h}{a} + \frac{2}{a^2} - \left( \frac{h}{2} + \frac{h}{a} + \frac{2}{a^2} \right) e^{-ah} \right]$$

$$r_{22} = R \left[ \frac{h^2}{2} - \frac{1}{2a} + \left( \frac{h}{a} + \frac{1}{2a} \right) e^{-ah} \right]$$

$$r_{23} = \frac{R}{2} \left[ h^3 - \frac{h^2}{a} + \frac{h^2}{a} e^{-ah} \right]$$

$$r_{33} = R \left[ \frac{h^4}{4} - \frac{h^3}{3a} + \frac{2}{a} - \left( \frac{h^2}{a} + \frac{2h}{a} + \frac{1}{a} \right) e^{-ah} \right]$$

Variansen i prediktionsfelet

$$P_{11}(k+1) = p_{11} - \frac{P_{13}^2}{R_2 + P_{33}} + r_{11}$$

$$p_{12}(k+1) = h p_{11} + P_{12} - \frac{h P_{13}^2 + P_{23}}{R_2 + P_{33}} + r_{12}$$

$$P_{13}(k+1) = \frac{h^2}{2} P_{11} + h P_{12} + P_{13} - \frac{h^2 P_{13}^2 / 2 + h P_{23} P_{13} + P_{13} P_{33}}{R_2 + P_{33}} + r_{13}$$

$$P_{22}(k+1) = h^2 P_{11} + 2h P_{12} + P_{22} - \frac{(h P_{13} + P_{23})^2}{R_2 + P_{33}} + r_{22}$$

$$P_{23}(k+1) = \frac{h^3}{2} P_{11} + \frac{3h^2}{2} P_{12} + h P_{22} + h P_{13} + P_{23} - \frac{(h P_{13} + P_{23})^2}{R_2 + P_{33}} + r_{22}$$



## 8. REFERENSER

- [1] K.J. Åström och B. Wittenmark (1984): *Computer controlled systems*. Englewood Cliffs N.J. : Prentice-Hall, Inc
  
- [2] T.R. Kronhamn (1985) : *Adaptive target trackin with serial Kalmanfilters*. IEEE 1985 Decision and control conference, Dec pp 1288-1293
  
- [3] C.C. Schooler (1975) : *Optimal  $\alpha\beta$ -filters for systems with modeling errors*. IEEE 1975 Trans. on aerospace and electronic systems. Vol AES-11, No.6 pp 1300-1306
  
- [4] A.S. Willsky (1976) : *A survey of design methods for failiure detection in dynamic systems*. Automatica Vol. 12, pp 601-611
  
- [5] P.O. Gutman och M. Velger : *Tracking targets with unknown noise variance using adaptive Kalman filtering*. Docentföreläsning vid Lunds Tekniska Högskola, inst för reglerteknik. Vt 1988