



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering:
Kybernetikk/signalbehandling

Vårsemesteret, 2009

Åpen / ~~Konfidensiell~~

Forfatter: Atle Gjengedal

.....
(signatur forfatter)

Faglig ansvarlig: Tormod Drengstig

Veileder(e): Tormod Drengstig

Tittel på masteroppgaven:

Sammenligning av EKF- og UKF- kalmanfilter anvendt på totankprosess

Engelsk tittel:

Comparison of EKF- and UKF- Kalman filter on two tank process

Studiepoeng: 30

Emneord:
Unscented Kalmanfilter, UKF
Utvidet Kalmanfilter, EKF
Simulering
Tilstandsestimering
Lekkasjeestimering

Sidetall: 74
+ vedlegg/annet: 27

Stavanger, 9.6.2009
dato/år

Sammenligning av EKF- og UKF- kalmanfilter anvendt på totankprosess

Atle Gjengedal

9. juni 2009

Forord

Denne hovedoppgaven markerer slutten på min mastergradsutdanning innen kybernetikk ved Universitetet i Stavanger. Jeg har tatt mastergraden ved siden av jobben min som ingeniør i Aibel AS, og det har vært svært utfordrende og krevende til tider. I tillegg til det faglige har jeg også lært mye om meg selv. Gjennom arbeidet har jeg fått god erfaring med å jobbe strukturert for å nå mine mål.

Takket være min kjære kone, Camilla B. Østebrød som har støttet meg gjennom disse årene har jeg kommet meg gjennom masterutdanningen med gode resultater. Det er ikke få helger som har gått med på lesesalen før eksamener, hvor vi nok heller skulle ønske at vi kunne brukt tiden sammen. Tusen takk for at du har vært så tålmodig, Camilla!

Når det kommer til denne oppgaven må jeg si at Tormod Drenstvig har vært en god veileder å ha. Vi har hatt faste ukemøter, og jeg har fått gode tips og tilbakemeldinger underveis både til arbeidet og utforming av rapporten.

Til slutt vil jeg også takke søsteren min, Silje L. Gjengedal, og hunden Balder for mange fine lufteturer på stranden etter lange dager på skolen.

Atle Gjengedal

Tjelta, Juni 2009

Sammendrag

Denne masteroppgaven er todelt. Del I går ut på å sammenligne ytelsen til et utvidet Kalmanfilter (EKF) og et "unscented" Kalmanfilter (UKF) på et prosessanlegg bestående av to tanker, ved Universitetet i Stavanger (UIS). I del II benyttes et utvidet Kalmanfilter for å estimere lekkasje fra den ene tanken i anlegget. I begge delene baseres Kalmanfiltrenes estimat på måling av nivået i den ene tanken og kjennskap til prosessen.

I del I har EKF og UKF blitt sammenlignet under forskjellige forhold i anlegget. Det er sett på hvordan filtrene presterer i forhold til hverandre ved forskjellige tuningparametre og evnen til å kompensere for ekstern påvirkning/forstyrrelser i systemet. Som forstyrrelse har tilsats av vann vært benyttet. UKF er i følge teorien overlegent EKF på systemer der man ikke kan anta linearitet rundt arbeidspunkt bestemt av tilstandsestimat. UKF unngår lineariseringen i EKF ved å plukke ut punkter rundt arbeidspunktene og sende disse gjennom den ulineare modellen. På denne måten unngår man feil som følge av lineariseringen. Resultatene fra forsøkene i del I viser at EKF og UKF har tilnærmet lik prestasjon på totankanlegget og at årsaken til dette er at systemet ikke er så ulineart at lineariseringen i EKF skaper problemer.

I del II blir modellen og EKF fra del I utvidet for å kunne estimere lekkasje fra den ene tanken i systemet. Denne typen filter kan for eksempel benyttes til å gi alarmer i et prosesskontrollsystem. Ved å teste dette Kalmanfilteret ved forskjellige driftsituasjoner har man vist filterets styrker og svakheter. Det har vist seg at modellfeil fører til at filteret ikke kan brukes for å oppdage lekkasje i driftsituasjoner hvor ventil- og pumpepådragene varierer ofte og brått. I situasjoner hvor pådrag variere lite og/eller sjelden gir filteret tydelig indikasjon på lekkasje og kan anvendes for å gi alarm om lekkasje.

Innhold

Forord	i
Sammendrag	iii
1 Innledning	1
1.1 Bakgrunn for oppgaven	1
1.2 Kalmanfilter	2
1.2.1 Kort historie	2
1.2.2 Teori	3
I UKF og EKF for nivåestimering	5
2 Modellering og modellverifisering	7
2.1 Differensialligninger	8
2.1.1 Differensialligning for nivå i tank 1	8
2.1.2 Differensialligning for nivå i tank 2	9
2.2 Implementering av totankmodell i Simulink	9
2.3 Modellverifisering	12
3 Implementering av Utvidet Kalman-filter (EKF)	15
3.1 Teori og historie	15
3.1.1 Ligningene for EKF	15
3.1.2 Linearisering	16
3.2 Implementering av EKF i Simulink	18

4	Implementering av ”Unscented” Kalmanfilter (UKF)	21
4.1	Teori og historie	21
4.1.1	Ligningene for UKF	23
4.2	Implementering av UKF i Simulink	26
5	Testing og tuning	29
5.1	Normal drift	30
5.1.1	P-matrise	32
5.1.2	Kalmanfilterforsterkning	34
5.2	Prosessforstyrrelse tank 1	35
5.2.1	P-matrise	36
5.2.2	Kalmanfilterforsterkning	37
5.3	Prosessforstyrrelse tank 2	38
5.3.1	P-matrise og Kalmanfilterforsterkning	40
5.4	Mer om valg av tuning-parametre	41
5.5	Sigmapunkter	42
 II Kalmanfilter for estimering av lekkasje i ventil og/eller tank		43
6	Modellering	45
6.1	Utvide modell fra del I	45
6.2	Differensialligninger	46
6.2.1	Differensialligning for nivå i tank 1	46
6.2.2	Differensialligning for nivå i tank 2	47
6.2.3	Differensialligning for lekkasje	47
6.3	Observerbarhet	47
6.3.1	Kontroll av observerbarhet	48
7	Implementering, testing og tuning	51
7.1	Implementering	51
7.2	Sammenligning av filtre uten lekkasje	54
7.3	Motvikring av estimeringsfeil p.g.a. dødtid	55
7.4	Testing ved lekkasje fra tank 1	57

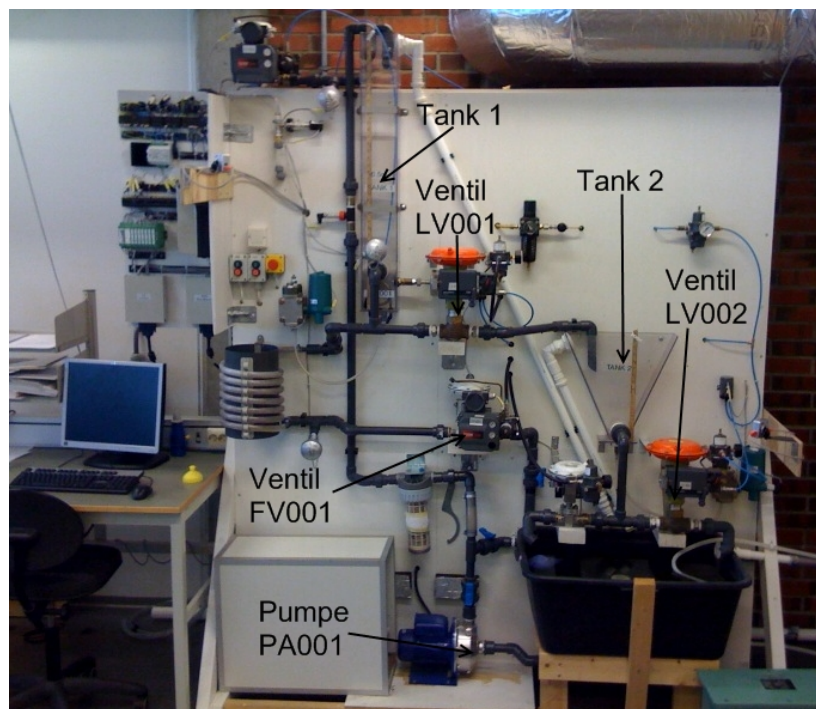
<i>INNHOLD</i>	vii
8 Konklusjon	59
8.1 Del I	59
8.2 Del II	59
Bibliografi	61
A UKF og EKF for nivåestimering	63
A.1 Modellverifisering	64
A.2 Matlab-kode for å finne transisjonsmatrisen	65
A.3 Implementering av EKF	67
A.4 Implementering av UKF	72
A.5 Offline kjøring	78
A.6 Modelldata	78
B EKF for estimering av lekkasje i ventil	83
B.1 Transisjonsmatrise	84
B.2 Observerbarhetsmatrise	84
B.3 Implementering av EKF	85

Kapittel 1

Innledning

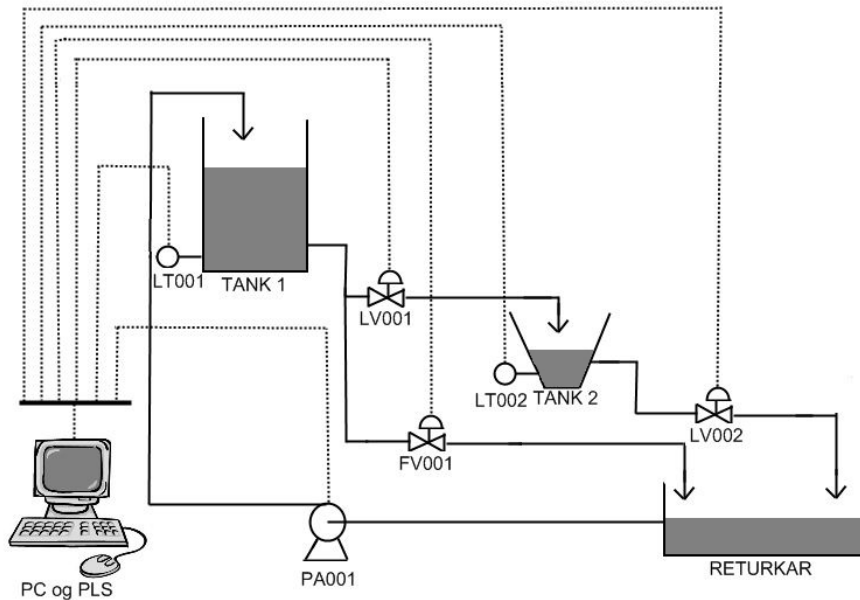
1.1 Bakgrunn for oppgaven

Faggruppa i kybernetikk ved Institutt for data og elektroteknikk på Universitetet i Stavanger har utviklet et lite prosessanlegg til bruk i undervisningen, se figur 1.1 og 1.2. Dette anlegget brukes for å la studentene teste ut teorien de lærer i praksis. Anlegget benyttes særlig av prosessrettede kybernetikkfag innen reguleringsteknikk og systemidentifikasjon. Flere kandidatoppgaver og hovedoppgaver har tidligere blitt utført på dette anlegget, blant annet [2, 7].



Figur 1.1: Bilde av prosessanlegg

Våren 2007 ble det skrevet en hovedoppgave [7] om bruk av utvidet Kalmanfilter (EKF) for å estimere nivå i den øverste tanken i anlegget, tank 1. I denne oppgaven ble signalet fra nivå-transmitter *LT002* i tank 2 benyttet som måling, og som pådrag ble signal ut til nivåkontrollventilene *LV001* og *LV002* i tillegg til pådragssignal til pumpen *PA001* brukt, se prinsippskisse i figur 1.2.



Figur 1.2: Prosesskisse av totankoppsett

I den siste tiden har det blitt mer og mer oppmerksomhet rundt en relativt ny variant av Kalmanfilteret, nemlig "Unscented" Kalmanfilter (UKF) [5, 8, 10, 12]. Dette filteret har vist seg å ha bedre egenskaper på en del ulineære systemer enn det utvidede Kalmanfilteret [10].

I del I av denne oppgaven er det implementert både et utvidet Kalmanfilter og et "Unscented" Kalmanfilter som blir sammenlignet m.h.t. ytelse.

I del II av oppgaven benyttes et Kalmanfilter for å estimere lekkasje i tank 1. Denne lekkasjen skjer gjennom å åpne ventilen *FV001* som fører vannet rett ned i returkaret.

1.2 Kalmanfilter

1.2.1 Kort historie

Kalmanfilteret er oppkalt etter Rudolf E. Kalman som har fått æren for utvikling av denne algoritmen, selv om flere andre har jobbet med lignende algoritmer

tidligere. De første artiklene skrevet av R. E. Kalman om denne algoritmen kom ut i 1960 [3]. Under ett besøk ved "NASA Ames Research Center" så han muligheten til å benytte den nye algoritmen ved baneestimering for Apollo-programmet. Dette ledet til at Kalmanfilter ble implementert i NASA sitt navigasjonssystem. I dag er Kalmanfilter i utstrakt bruk innenfor blant annet reguleringssystemer, maskinsyn og radarsystemer. [10]

Senere har flere varianter og forbedringer av det opprinnelige filteret blitt utviklet. To av de meste kjente er utvidet Kalmanfilter (EKF) [11] og "Unscented" Kalmanfilter (UKF)[5] som er benyttet i denne oppgaven. Mer om historien og teorien bak disse variantene er nevnt i kap. 3.1 og 4.1.

1.2.2 Teori

Kalmanfilter er en rekursiv algoritme for å estimere tilstander og/eller parametre i dynamiske systemer med støy. Tanken er at man ved å ha god kjennskap om egenskapene til systemet man jobber med skal kunne ha en optimal estimator for de tilstandene/parametrene man ikke kan måle direkte.

Filteret består av to deler; en foroverprediktor og en korreksjonsdel. Foroverprediktoren predikerer neste tilstand, $\bar{x}(k)$ ut fra forrige tilstandsestimat, $\hat{x}(k-1)$ og oppdaterte parametre. I korreksjonsdelen, se ligning (1.1), blir estimatet korrigert ved at man sammenligner estimatet fra foroverprediktoren med ny måling[10].

$$\hat{x}(k) = \bar{x}(k) + K(k) \cdot e(k) \quad (1.1)$$

Ligning (1.1) viser korreksjonen fra estimatet fra foroverprediktoren som gjentas for hvert tidssteg. $\bar{x}(k)$ blir korrigert til $\hat{x}(k)$ ved at man legger til $K(k) \cdot e(k)$ der $K(k)$ er en forsterkningsfaktor og $e(k)$ er differansen mellom målingen, $y(k)$, og forventet måling, $\bar{y}(k)$, ved tidssteg k . Faktoren K kalles ofte for Kalmanfilterforsterkning og avgjør hvor raskt filteret skal korrigere for avvik mellom estimat og måling. Størrelsen på K varierer mens filteret kjører, men bestemmes i stor grad av Kalmanfilterets tuning-parametre. Med tuning-parametre menes parametrene Q og R , der Q beskriver varians i prosesstøy og R beskriver varians i målestøy. Disse antas ofte konstante, men det er ingenting i veien for å la disse variere med tiden. For at Kalmanfilteret skal kunne fungere er det også et krav at systemet er observerbart. Observerbarhet blir beskrevet mer inngående i kap. 6.3.

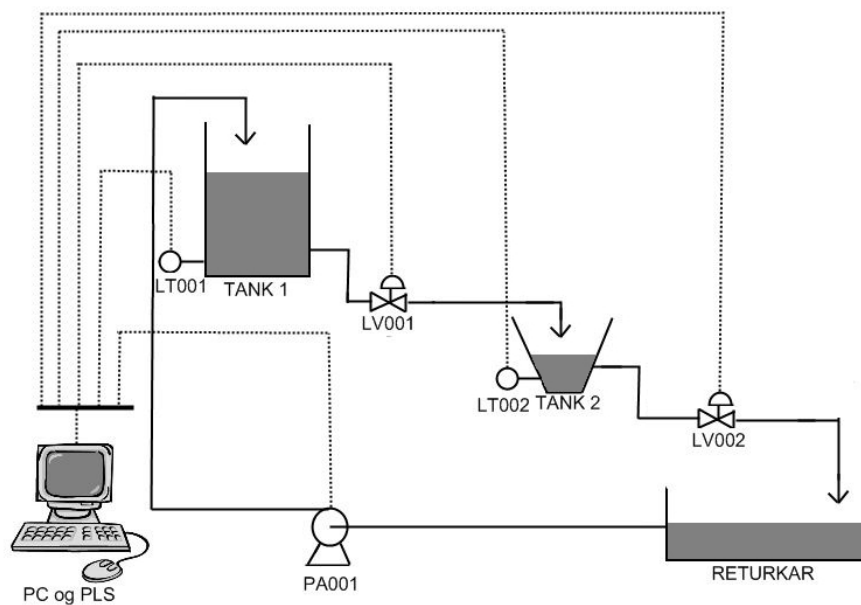
Del I

UKF og EKF for
nivåestimering

Kapittel 2

Modellering og modellverifisering

I dette kapitlet er den matematiske modellen for totankprosessen beskrevet og forklart. Først er differensialligningene for systemet satt opp, så er implementeringen av modellen i Simulink vist.



Figur 2.1: Prosesskisse av totankoppsett del I

Tabell 2.1 forklarer symbolene som benyttes videre i dette kapitlet

Tabell 2.1: Forklaring til symboler i differensialligningene

Variabel	Enhet	Beskrivelse
$h_1(t)$	m	Vannivå i tank 1 målt fra bunnen i tanken
$h_2(t)$	m	Vannivå i tank 2 målt fra bunnen i tanken
h_{LV001}	m	Høyde fra ventil $LV001$ opp til bunn i tank 1
h_{LV002}	m	Høyde fra ventil $LV002$ opp til bunn i tank 2
$q_{PA001}(t)$	$\frac{m^3}{s}$	Strømning av vann gjennom pumpe $PA001$
$q_{LV001}(t)$	$\frac{m^3}{s}$	Strømning av vann gjennom ventil $LV001$
$q_{LV002}(t)$	$\frac{m^3}{s}$	Strømning av vann gjennom ventil $LV002$
A_1	m^2	Areal av vannspeilet i tank 1
$A_2(t)$	m^2	Areal av vannspeilet i tank 2
K_{vLV001}	$\frac{m^3}{h \cdot \sqrt{bar}}$	Ventilkonstant for ventil $LV001$
K_{vLV002}	$\frac{m^3}{h \cdot \sqrt{bar}}$	Ventilkonstant for ventil $LV002$
$u_{PA001}(t)$	-	Pådrag på pumpe $PA001$
$u_{LV001}(t)$	-	Pådrag på ventil $LV001$
$u_{LV002}(t)$	-	Pådrag på ventil $LV002$
$f_1(u_{LV001}(t))$		Tabelloppslag av ventilkarakteristikk
$f_2(u_{LV002}(t))$		Tabelloppslag av ventilkarakteristikk
$f_3(u_{PA001}(t))$	$\frac{m^3}{s}$	Tabelloppslag av pumpekarakteristikk
ρ	$\frac{kg}{m^3}$	Tettheten til vann
g	$\frac{m}{s^2}$	Tyngdeakselerasjon
c_1		Samling av konstanter i differensialligning (2.2)
c_2		Samling av konstanter i differensialligning (2.5)

2.1 Differensialligninger

Differensialligningene (2.3) og (2.7) beskriver systemet representert med de to tilstandene $h_1(t)$ og $h_2(t)$ som er vannivå i de to tankene som funksjon av tiden. Disse er hentet fra [2].

Funksjonene $f_1(\cdot)$, $f_2(\cdot)$ og $f_3(\cdot)$ er henholdsvis ventilkarakteristikk for $LV001$, ventilkarakteristikk for $LV002$ og pumpekarakteristikk for $PA001$. I [2] er det lagt mye arbeid i å lage og verifisere disse funksjonene, derfor er disse hentet direkte fra den masteroppgaven.

2.1.1 Differensialligning for nivå i tank 1

$$\begin{aligned}
 \frac{dh_1(t)}{dt} &= \frac{1}{A_1} (q_{PA001}(t) - q_{LV001}(t)) \\
 &= \frac{1}{A_1} \left(f_3(u_{PA001}(t)) \right. \\
 &\quad \left. - \frac{K_{vLV001} \cdot f_1(u_{LV001}(t))}{3600} \cdot \sqrt{\frac{\rho \cdot g \cdot (h_1(t) + h_{LV001})}{100000}} \right) \quad (2.1)
 \end{aligned}$$

Ved å bruke

$$c_1 = \frac{Kv_{LV001}}{3600} \cdot \sqrt{\frac{\rho \cdot g}{100000}} \quad (2.2)$$

blir differensialligningen for høyden i tank 1:

$$\begin{aligned} \frac{dh_1(t)}{dt} &= \frac{1}{A_1} \left(f_3(u_{PA001}(t)) \right. \\ &\quad \left. - c_1 \cdot f_1(u_{LV001}(t)) \cdot \sqrt{h_1(t) + h_{LV001}} \right) \end{aligned} \quad (2.3)$$

2.1.2 Differensialligning for nivå i tank 2

$$\begin{aligned} \frac{dh_2(t)}{dt} &= \frac{1}{A_2(t)} (q_{LV001}(t) - q_{LV002}(t)) \\ &= \frac{1}{A_2(t)} \left(\frac{Kv_{LV001} \cdot f_1(u_{LV001}(t))}{3600} \cdot \sqrt{\frac{\rho \cdot g \cdot (h_1(t) + h_{LV001})}{100000}} \right. \\ &\quad \left. - \frac{Kv_{LV002} \cdot f_2(u_{LV002}(t))}{3600} \cdot \sqrt{\frac{\rho \cdot g \cdot (h_2(t) + h_{LV002})}{100000}} \right) \end{aligned} \quad (2.4)$$

Ved å bruke

$$c_2 = \frac{Kv_{LV002}}{3600} \cdot \sqrt{\frac{\rho \cdot g}{100000}} \quad (2.5)$$

og

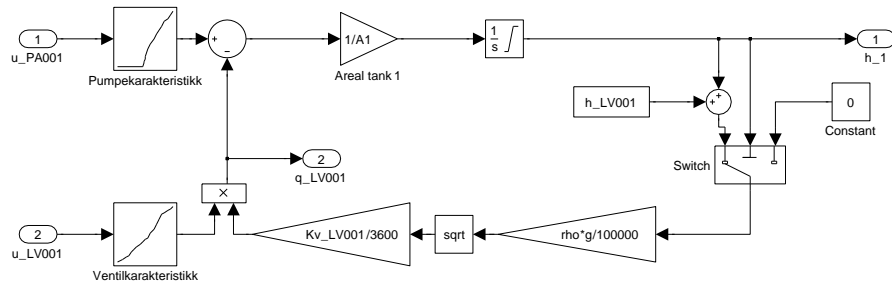
$$A_2(t) = 0.004 + 0.07 \cdot h_2(t) \quad (2.6)$$

blir differensialligningen for høyden i tank 2:

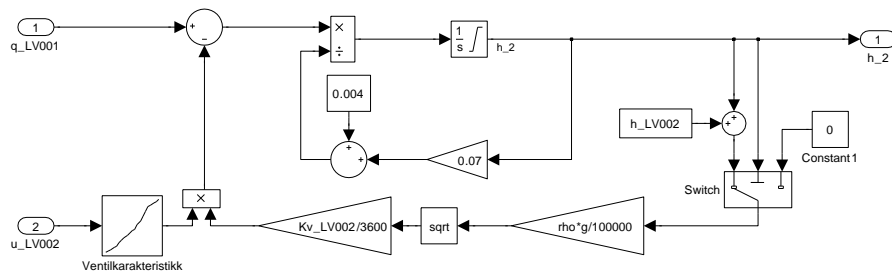
$$\begin{aligned} \frac{dh_2(t)}{dt} &= \frac{1}{A_2(t)} \left(c_1 \cdot f_1(u_{LV001}(t)) \cdot \sqrt{h_1(t) + h_{LV001}} \right. \\ &\quad \left. - c_2 \cdot f_2(u_{LV002}(t)) \cdot \sqrt{h_2(t) + h_{LV002}} \right) \end{aligned} \quad (2.7)$$

2.2 Implementering av totankmodell i Simulink

Ved å implementere differensialligningene fra kap. 2.1 i Simulink fås følgende systemblokker for tank 1 og tank 2:



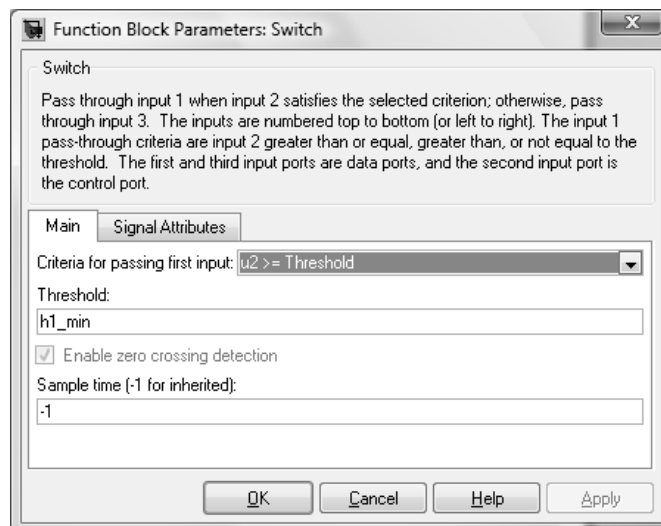
(a) Tank 1, ligning (2.3)



(b) Tank 2, ligning (2.7)

Figur 2.2: Implementering i Simulink

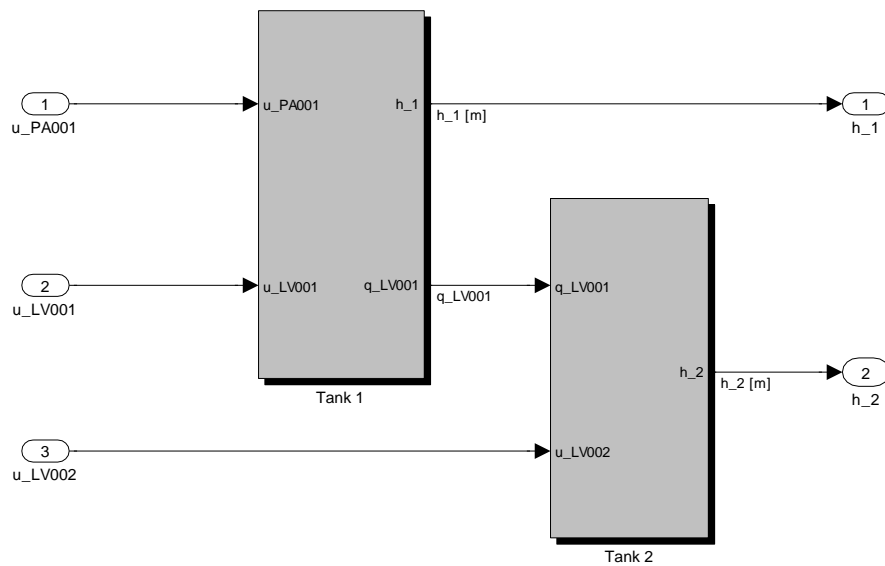
Som det går frem av figur 2.2 har systemblokkene for begge tankene blitt utstyrt med en "Switch". Denne switch-blokken måler inngangen i midten, og ut fra denne inngangens verdi og innstillinger som er lagt inn i switch-blokken, se figur 2.3, velges det hvilken inngang som skal slippe gjennom bryteren.



Figur 2.3: Switch-blokk

Som det går frem av innstillingene i switch-blokken vil første inngang slippes gjennom når målingen fra inngangen i midten er større enn eller lik terskelen $h2_min$. Ved å legge inn dette i modellen hindrer man at modellen tillater at vann renner ut fra tanken når nivået er lavere enn $h2_min$. Figur 2.3 viser innstillingene i switch-blokken for tank 2. Switch-blokken for tank 1 er satt opp på samme måte.

De to systemblokkene over blir koblet sammen til totalsystemet som vist i figur 2.4. Det er strømmingen fra tank 1 til tank 2 gjennom $LV001$ som kobler de to tankene/delsystemene sammen.

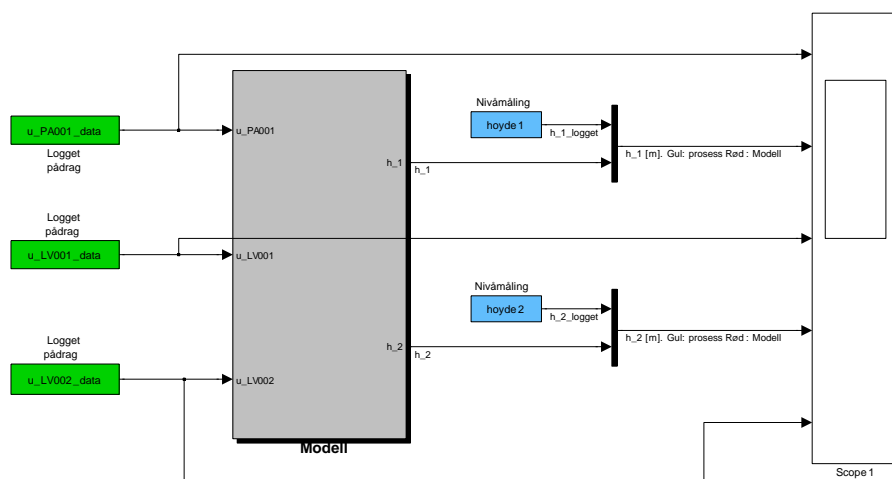


Figur 2.4: Totankmodell i Simulink

2.3 Modellverifisering

For å verifisere modellen i kap. 2.2 har totankanlegget blitt kjørt flere ganger mens pådrag og nivåmålinger har blitt logget. Deretter kjøres modellen mot de loggede pådragene, og modellutgangene sammenlignes med de loggede nivåmålingene.

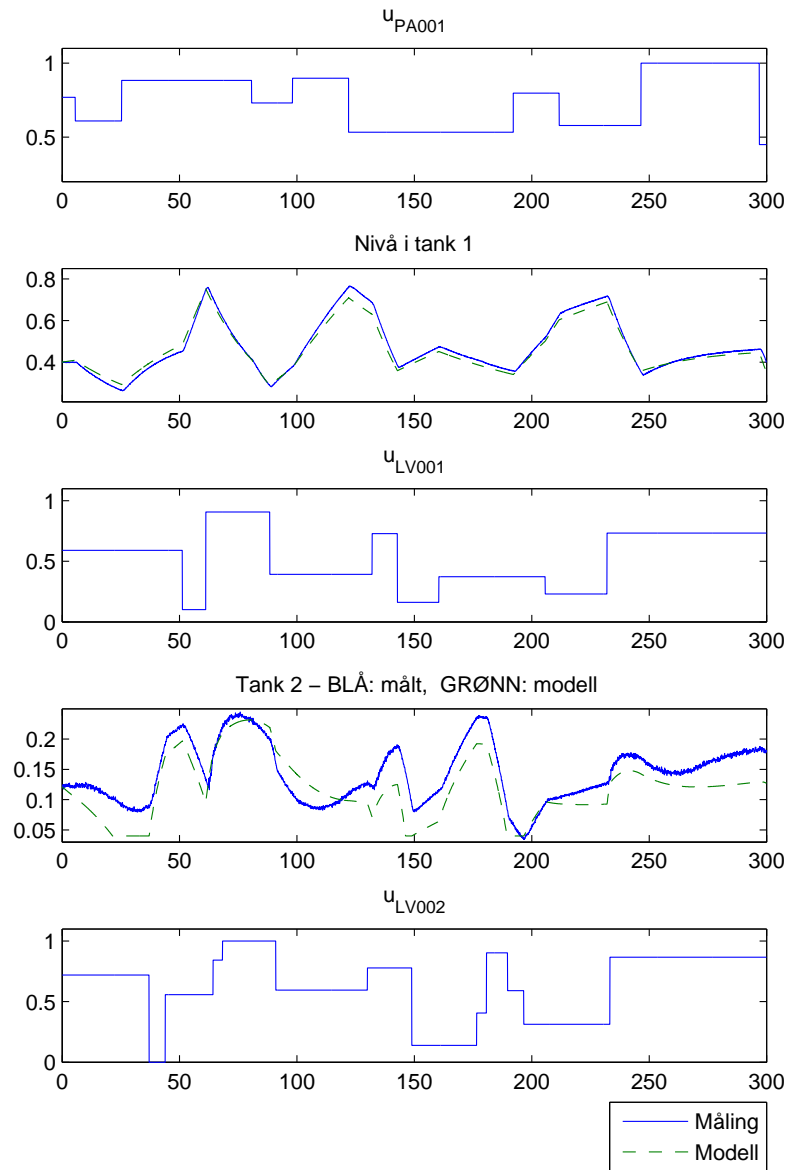
Figur 2.5 viser Simulink-implementering som benyttes til modellverifisering:



Figur 2.5: Modellverifisering Simulink

Etter mange kjøringar har det vist seg at "dagsformen" på totankanlegget har en del å si for resultatet. Noen ganger stemmer modellen veldig bra med prosessen, mens andre ganger er det større avvik. Selv om "dagsformen" varierer, har en ganske god modell for vannivåene blitt funnet.

Figur 2.6 viser modellen verifisert mot prosessen.



Figur 2.6: Modellverifisering

Som det går frem av figur 2.6, stemmer modellen godt med virkeligheten i dette forsøket. Se også figur A.1 i vedlegg A.1 som viser at den samme modellen også gir gode resultater ved et annet forsøk.

Kapittel 3

Implementering av Utvidet Kalman-filter (EKF)

3.1 Teori og historie

Utvidet Kalmanfilter ble utviklet av Stanley Schmidt i 1967 [10] for å kunne bruke Kalmanfilter på ulineære navigasjonssystem for romfartøy. Motivasjonen for å utvikle dette filteret var at det vanlige Kalmanfilteret bare virket godt på lineære prosesser eller prosesser som var tilnærmet lineære rundt et fast arbeidspunkt. Måten det utvidede Kalmanfilteret løser dette på er ved å linearisere modellen rundt et arbeidspunkt for hvert tidssteg. Dette var en stor forbedring i forhold til det opprinnelige Kalmanfilteret, men et symbolsk uttrykk for den lineære transisjonsmatrisen, Φ , må finnes. Dette kan være svært krevende for en komplisert prosess.

3.1.1 Ligningene for EKF

Det utvidede Kalmanfilteret er basert på et sett ligninger som gjentas for hvert tidssteg. Se algoritme 3.1. For hvert tidssteg benyttes informasjon fra forrige tidssteg, som gjør at algoritmen klassifiseres som rekursiv.

Algoritmen tar utgangspunkt i den ulineære modellen:

$$x(k+1) = x(k) + f(x(k), u(k)) \quad (3.1)$$

$$y(k) = g(x(k)) \quad (3.2)$$

der $x(k+1)$ er systemets diskrete differensligninger og $y(k)$ er den diskrete måleligningen.

Algoritme 3.1 Ligningene for EKF

$$\Phi(k) = \left. \frac{\partial f(\cdot)}{\partial x} \right|_{\hat{x}(k-1), u(k-1)} \quad (3.3)$$

$$\bar{x}(k) = f(\hat{x}(k-1), u(k-1)) \quad (3.4)$$

$$\bar{P}(k) = \Phi(k) \cdot \hat{P}(k-1) \cdot \Phi^T(k) + Q(k) \quad (3.5)$$

$$D(k) = \left. \frac{\partial g(\cdot)}{\partial x} \right|_{\bar{x}(k)} \quad (3.6)$$

$$K(k) = \bar{P}(k) \cdot D^T(k) \cdot (D(k) \cdot \bar{P}(k) \cdot D^T(k) + R)^{-1} \quad (3.7)$$

$$\bar{y}(k) = g(\bar{x}(k)) \quad (3.8)$$

$$\hat{x}(k) = \bar{x}(k) + K(k) \cdot (y(k) - \bar{y}(k)) \quad (3.9)$$

$$\hat{P}(k) = (I - K(k) \cdot D(k)) \cdot \bar{P}(k) \quad (3.10)$$

Forklaring til ligningene i EKF- algoritmen:

- Ligning (3.3) lineariserer modellen i arbeidspunkt gitt av aposteriori tilstandsestimat og pådrag i forrige tidssteg. Dette blir utfyllende beskrevet i kap. 3.1.2.
- Ligning (3.4) beregner apriori tilstandsestimat.
- Ligning (3.5) beregner apriori kovariansmatrise.
- Ligning (3.6) beregner målematrisen. For totankprosessen blir denne konstant lik $[0 \ 1]$ siden nivå i tank 2 måles direkte.
- Ligning (3.7) beregner Kalmanfilterforsterkningen.
- Ligning (3.8) beregner prediktert måling.
- Ligning (3.9) beregner aposteriori tilstandsestimat, som blir utgangen fra Kalmanfilteret. Dette tilstandsestimatet skal også brukes i ligning (3.3) og (3.4) ved neste tidssteg.
- Ligning (3.10) oppdaterer kovariansmatrisen som skal brukes i ligning (3.5) ved neste tidssteg.

3.1.2 Linearisering

Ligning (3.3) beregner som sagt den lineariserte transisjonsmatrisen for systemet. For å få til dette i implementeringen av EKF må man ha et symbolsk uttrykk for den diskrete transisjonsmatrisen på formen.

$$\Phi(k) = \begin{bmatrix} \frac{\partial g_1(\cdot)}{\partial x_1} & \frac{\partial g_1(\cdot)}{\partial x_2} \\ \frac{\partial g_2(\cdot)}{\partial x_1} & \frac{\partial g_2(\cdot)}{\partial x_2} \end{bmatrix} \quad (3.11)$$

der $g_1(\cdot)$ og $g_2(\cdot)$ er systemets diskrete differensligninger.

For å finne et symbolsk uttrykk for transisjonsmatrisen, må differensligningene fra kap. 2.1 først diskretiseres.

Differensligningene (2.3) og (2.7) diskretiseres ved hjelp av Eulers forovermetode

$$x(k+1) = x(k) + T \cdot \dot{x}(k) \quad (3.12)$$

der T er sample-tiden. Definerer:

$$x_1(k) = h_1(k) \quad (3.13)$$

og får:

$$\begin{aligned} x_1(k+1) &= x_1(k) + T \cdot \dot{x}_1(k) \\ &= x_1(k) + \frac{T}{A_1} \left(f_3(u_{PA001}(k)) \right. \\ &\quad \left. - c_1 \cdot f_1(u_{LV001}(k)) \cdot \sqrt{x_1(k) + h_{LV001}} \right) \\ &= g_1(\cdot) \end{aligned} \quad (3.14)$$

Definerer

$$x_2(k) = h_2(k) \quad (3.15)$$

og får:

$$\begin{aligned} x_2(k+1) &= x_2(k) + T \cdot \dot{x}_2(k) \\ &= x_2(k) + \frac{T}{A_2(k)} \left(c_1 \cdot f_1(u_{LV001}(k)) \cdot \sqrt{x_1(k) + h_{LV001}} \right. \\ &\quad \left. - c_2 \cdot f_2(u_{LV002}(k)) \cdot \sqrt{x_2(k) + h_{LV002}} \right) \\ &= g_2(\cdot) \end{aligned} \quad (3.16)$$

Ved å benytte symbolsk derivasjon i Maple (og Matlab), se filen "transisjonsmatrise.m" i vedlegg A.2 finnes $\Phi(k)$ til:

$$\Phi(k) = \begin{bmatrix} \phi_{11}(k) & \phi_{12}(k) \\ \phi_{21}(k) & \phi_{22}(k) \end{bmatrix} \quad (3.17)$$

Der $\phi_{11}(k)$, $\phi_{12}(k)$, $\phi_{21}(k)$ og $\phi_{22}(k)$ er gitt som:

$$\phi_{11}(k) = 1 - \frac{T \cdot c_1 \cdot f_1(k)}{2 \cdot A_1 \cdot \sqrt{x_1(k) + h_{LV001}}} \quad (3.18)$$

$$\phi_{12}(k) = 0 \quad (3.19)$$

$$\phi_{21}(k) = \frac{T \cdot c_1 \cdot f_1(k)}{2 \cdot A_2(k) \cdot \sqrt{x_1(k) + h_{LV001}}} \quad (3.20)$$

$$\phi_{22}(k) = 1 - \frac{0.07 \cdot T}{A_2^2(k)} \cdot \left(c_1 \cdot f_1(k) \cdot \sqrt{x_1(k) + h_{LV001}} - c_2 \cdot f_2(k) \cdot \sqrt{x_2(k) + h_{LV002}} \right) - \frac{T \cdot c_2 \cdot f_2(k)}{2 \cdot A_2(k) \cdot \sqrt{x_2(k) + h_{LV002}}} \quad (3.21)$$

Ved å velge

$$q_{LV001}(k) = c_1 \cdot f_1(k) \cdot \sqrt{x_1(k) + h_{LV001}} \quad (3.22)$$

$$q_{LV002}(k) = c_2 \cdot f_2(k) \cdot \sqrt{x_2(k) + h_{LV002}} \quad (3.23)$$

der $q_{LV001}(k)$ og $q_{LV002}(k)$ er strømmingen gjennom henholdsvis $LV001$ og $LV002$, og la disse bli beregnet før $\Phi(k)$ for hvert tidssteg blir elementene i transisjonsmatrisen ytterligere forenklet til:

$$\phi_{11}(k) = 1 - \frac{T \cdot q_{LV001}(k)}{2 \cdot A_1 \cdot (x_1(k) + h_{LV001})} \quad (3.24)$$

$$\phi_{12}(k) = 0 \quad (3.25)$$

$$\phi_{21}(k) = \frac{T \cdot q_{LV001}(k)}{2 \cdot A_2(k) \cdot (x_1(k) + h_{LV001})} \quad (3.26)$$

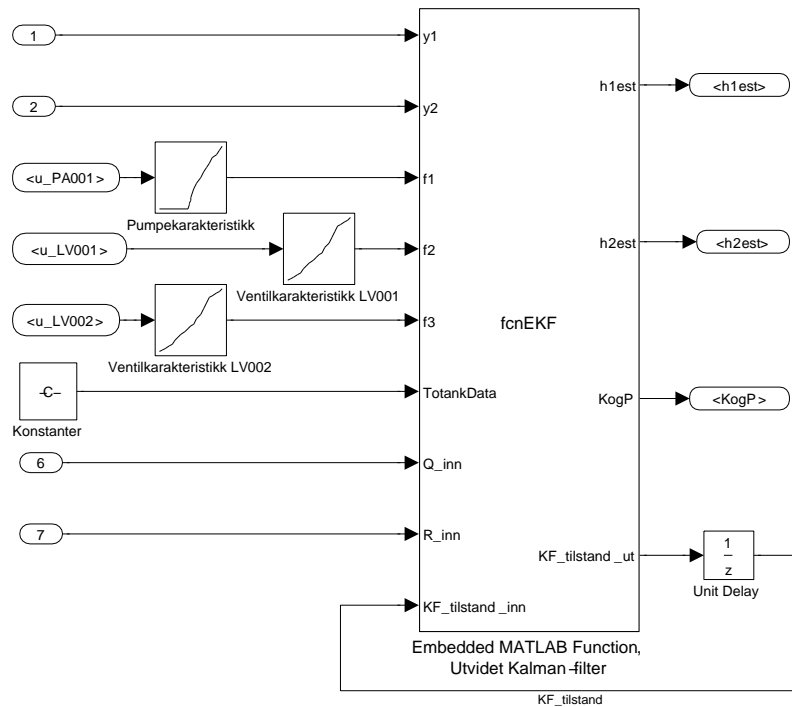
$$\phi_{22}(k) = 1 - \frac{0.07 \cdot T}{A_2^2(k)} \cdot \left(q_{LV001}(k) - q_{LV002}(k) \right) - \frac{q_{LV002}(k)}{2 \cdot A_2(k) \cdot (x_2(k) + h_{LV002})} \quad (3.27)$$

Siden transisjonsmatrisen skal beregnes ved hvert tidssteg er det viktig at beregningene i algoritmen ikke er mer ressurskrevende enn det som er nødvendig. Forenklingen ved å regne ut $q_{LV001}(k)$ og $q_{LV002}(k)$ før transisjonsmatrisen beregnes, gjør at Simulink også klarer å kjøre EKF filteret i sanntid mot prosessen. Innføringen av $q_{LV001}(k)$ og $q_{LV002}(k)$ gjør også uttrykket lettere å lese og forstå.

3.2 Implementering av EKF i Simulink

Det utvidede Kalmanfilteret er implementert som en funksjonsblokk av typen "Embedded MATLAB Function" i Simulink. Funksjonsblokken inneholder Matlab-kode for EKF-ligningene (3.3) til (3.10). Se vedlegg A.3 for utskrift av kode i funksjonsblokken.

Figur 3.1 viser funksjonsblokken med innganger og utganger som beskrevet i tabell 3.1.



Figur 3.1: EKF Simulink

Tabell 3.1: Innganger og utganger EKF

Inn-/utgang	Forklaring
y1	Måling av nivå i tank 1. Benyttes kun ved initialisering
y2	Måling av nivå i tank 2
f1	Tabelloppslag for pumpekarakteristikk for pumpe PA001
f2	Tabelloppslag for ventilkarakteristikk for ventil LV001
f3	Tabelloppslag for ventilkarakteristikk for ventil LV002
TotankData	Henter inn konstanter fra en fil som inneholder alle konstante parametre i totank-systemet
Q_inn	Tuning-matrise Q som brukes i ligning (3.5)
R_inn	Tuning-matrise R som brukes i ligning (3.7)
KF_tilstand_inn	Verdier (apriori) fra forrige tilstand som brukes i ligning (3.3), (3.4) og (3.5)
h1est	Estimat av høyde i tank 1
h2est	Estimat av høyde i tank 2
KogP	Kalmanfilterforsterkningsmatrise og P-matrise
KF_tilstand_ut	Verdier (aposteriori) som skal brukes ved neste tilstand i ligning (3.3), (3.4) og (3.5)

Kapittel 4

Implementering av ”Unscented” Kalmanfilter (UKF)

4.1 Teori og historie

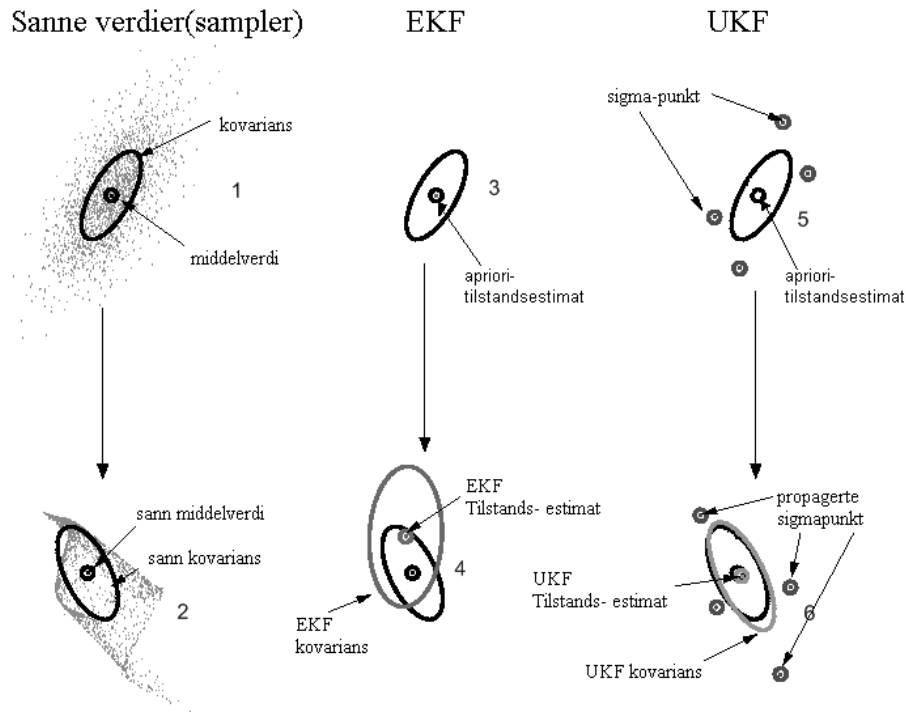
I 1997 ble den første artikkelen om ”Unscented” Kalmanfilter (UKF) utgitt av Simon J. Julier og Jeffrey K. Uhlmann [5]. Her peker de på to viktige begrensninger ved bruk av EKF:

1. Lineariseringen kan gi svært ustabile filter i systemer der den antatte betingelsen om linearitet rundt arbeidspunktet ikke er oppfylt.
2. Derivasjonene som må gjøres for å finne et uttrykk for den lineære transisjonsmatrisen (kap. 3.1.2) kan være veldig vanskelige/ ressurskrevende å finne. Uttrykkene man kommer fram til kan bli veldig store og komplekse, og siden disse må beregnes for hvert tidssteg kan anvendelsen av filteret blir svært ressurskrevende.

Det at UKF- algoritmen løser disse to problemene er den store fordelen til UKF fremfor EKF.

Ved å benytte EKF, lineariseres modellen i et arbeidspunkt bestemt av tilstands-estimat og pådrag i forrige tidssteg. Hvis man antar at estimatet er slik at like mange estimat er for høye som for lave kan man allikevel få uriktige estimater. Ved stor ulinearitet kan et avvik på den ene siden av sann tilstand gi et mye større utslag enn et avvik på den andre siden. Dette kan medføre estimater som ikke er forventningsrette ved bruk av EKF på svært ulineære prosesser. Dette problemet løses i UKF ved å plukke ut noen punkter rundt estimert tilstand, såkalte sigmapunkter, og propagere dem gjennom den ulineære funksjonen. De propagerede sigmapunktene vektet og benyttes til å finne et apriori tilstands-estimat som tar hensyn til ulineariteten. Siden disse sigmapunktene propageres

gjennom den ulineære modellen unngår man lineariseringen som må gjøres i EKF.



Figur 4.1: Sammenligning av EKF og UKF fra [6]

Figur 4.1 illustrerer problemet med EKF ved store ulineariteter og hvordan dette kan forbedres ved å bruke UKF. Figuren øverst til venstre, ①, viser sann middelværdi og kovarians i et tidssteg. ② viser hvordan middelværdien og formen på kovariansen endrer seg når punktene rundt middelværdien blir propagert gjennom ulineariteten.

For EKF lineariseres modellen i arbeidspunkt gitt av tilstandsestimat, ③, og pådrag i forrige tidssteg (ligning (3.3)):

$$\Phi(k) = \left. \frac{\partial f(\cdot)}{\partial x} \right|_{\hat{x}(k-1), u(k-1)}$$

④ viser hva som skjer når apriori tilstandsestimat beregnes (ligning (3.4)):

$$\bar{x}(k) = f(\hat{x}(k-1), u(k-1))$$

Siden bare ett punkt, altså det overnevnet arbeidspunktet behandles, blir ikke informasjon om den propagerte kovariansen fanget opp. Det er da sannsynlig at estimatet vil være feil.

⑤ viser hvordan UKF benytter sigmapunkter, $\chi(k-1)$, plukket ut rundt apriori tilstandsestimatet (ligning (4.10)):

$$\chi_i(k-1) = \hat{x}(k-1) \pm \Gamma \sqrt{\hat{P}(k-1)}$$

Det er størrelsen på Γ som bestemmer hvor mye sigmapunktene spres rundt $\hat{x}(k-1)$. Ved så å propagere sigmapunktene gjennom ulineariteten ender man opp i \textcircled{C} (ligning (4.11)):

$$\chi_i(k) = f(\chi_i(k-1), u(k-1))$$

Nytt apriori tilstandsestimat blir da gitt som en vektet sum av de propagerede sigmapunktene (ligning (4.12)):

$$\bar{x}(k) = \sum_{i=1}^{2n} W_i^x \cdot \chi_i(k)$$

Ligningene for UKF er beskrevet mer utfyllende i kap. 4.1.1.

Jo større ulinearitet man har rundt målepunktet desto større utbytte vil man kunne ha av å bruke UKF.

Det finnes flere forskjellige implementeringer av UKF. Forskjellen ligger ofte i hvordan sigmapunktene plukkes ut og hvordan de vektet, men prinsippet er det samme. I denne oppgaven er det valgt å bruke en algoritme foreslått av Dan Simon i 2006 [8, 10] som er beregnet på systemer med additiv støy. At støyen er additiv innebærer at prosesstøy og målestøy blir lagt til prosess og måling direkte og er uavhengig av parametre som pådrag og vannivå i tankene.

4.1.1 Ligningene for UKF

Før ligningene for UKF presenteres må noen vekter velges. I ligningene for disse vektene inngår følgende konstanter:

- n , som er antall tilstander, som for totankprosessen er 2 (nivå i tank 1 og nivå i tank 2).
- α , som bestemmer spredningen av sigmapunktene rundt apriori tilstandsestimat fra forrige tidssteg $\hat{x}(k-1)$ som er det beste tilgjengelige estimatet når sigmapunktene skal beregnes. Se algoritme 4.1. α velges i intervallet $0 \leq \alpha \leq 1$ [12].
- β , som brukes for å inkorporer kjennskap til støyens fordeling. β velges i intervallet $0 \leq \beta \leq \infty$ [12]. For Gaussisk støy er $\beta = 2$ optimalt [6, 10]. Støyen i totankanlegget antas Gaussisk.
- κ , som er en skaleringsfaktor velges i intervallet $0 \leq \kappa \leq \infty$ for å garantere positiv semidefinit ¹ kovariansmatrise, $\hat{P}(k)$ [12]. Dette er en forutsetning for at matrisekvadratrotten i ligning (4.10) skal kunne regnes ut. Størrelsen på κ kan som α benyttes til å påvirke spredningen av sigmapunktene. Vanligvis settes κ til $\kappa = 0$. [12]

Fra punktene over velges $n = 2$, $\alpha = 1$, $\beta = 2$ og $\kappa = 0$. Disse valgene av α , β og κ er også foreslått i [10, 12].

¹ $\hat{P}(k)$ er positiv semidefinit hvis og bare hvis alle egenverdiene til $\hat{P}(k)$ er ikke-negative.

λ samler noen av konstantene over og er gitt ved:

$$\lambda = \alpha^2(n + \kappa) - n \quad (4.1)$$

I en generell UKF algoritme defineres følgende vektorer:

$$W_0^x = \frac{\lambda}{n + \lambda} \quad (4.2)$$

$$W_0^c = \frac{\lambda}{n + \lambda} + (1 + \alpha + \beta) \quad (4.3)$$

$$W_i^x = W_i^c = \frac{1}{2(n + \lambda)} \quad (4.4)$$

$$\Gamma = \sqrt{n + \lambda} \quad (4.5)$$

der λ er gitt ved:

$$\lambda = \alpha^2(n + \kappa) - n \quad (4.6)$$

hvor W_0^x og W_0^c er vektorer som brukes i algoritmer der forrige tilstandsestimat, $\hat{x}(k-1)$, tas med som ett sigmapunkt. W_i^x og W_i^c er vektene som brukes for sigmapunktene som finnes rundt $\hat{x}(k-1)$. I algoritme 4.1, som benyttes i denne oppgaven er ikke $\hat{x}(k-1)$ ett sigmapunkt, så vektene W_0^x og W_0^c blir ikke brukt videre. Γ er en skaleringsfaktor for sigmapunktene. Ved å sette inn tallene fås følgende vektorer og skaleringsfaktor:

$$\lambda = \alpha^2(n + \kappa) - n = 1^2(2 + 0) - 2 = 0 \quad (4.7)$$

$$W_i^x = W_i^c = \frac{1}{2(n + \lambda)} = \frac{1}{2(2 + 0)} = \frac{1}{4} \quad (4.8)$$

$$\Gamma = \sqrt{n + \lambda} = \sqrt{2 + 0} = \sqrt{2} \quad (4.9)$$

Algoritme 4.1 viser ligningene som definerer UKF- filteret som er brukt i denne oppgaven.

Algoritme 4.1 Ligningene for UKF

$$\chi_i(k-1) = \hat{x}(k-1) \pm \Gamma \sqrt{\hat{P}(k-1)} \quad (4.10)$$

$$\chi_i(k) = f(\chi_i(k-1), u(k-1)) \quad (4.11)$$

$$\bar{x}(k) = \sum_{i=1}^{2n} W_i^x \cdot \chi_i(k) \quad (4.12)$$

$$\bar{P}(k) = \sum_{i=1}^{2n} W_i^c (\chi_i(k) - \bar{x}(k)) (\chi_i(k) - \bar{x}(k))^T + Q \quad (4.13)$$

$$\gamma_i(k) = h(\chi_i(k)) \quad (4.14)$$

$$\hat{y}(k) = \sum_{i=1}^{2n} W_i^x \cdot \gamma_i(k) \quad (4.15)$$

$$P_{yy}(k) = \sum_{i=1}^{2n} W_i^c (\gamma_i(k) - \hat{y}(k)) (\gamma_i(k) - \hat{y}(k))^T + R \quad (4.16)$$

$$P_{xy}(k) = \sum_{i=1}^{2n} W_i^c (\chi_i(k) - \bar{x}(k)) (\gamma_i(k) - \hat{y}(k))^T \quad (4.17)$$

$$K(k) = P_{xy}(k) \cdot P_{yy}^{-1}(k) \quad (4.18)$$

$$\hat{x}(k) = \bar{x}(k) + K(k) (y(k) - \hat{y}(k)) \quad (4.19)$$

$$\hat{P}(k) = \bar{P}(k) - K(k) \cdot P_{yy}(k) \cdot K^T(k) \quad (4.20)$$

Forklaring til ligningene i UKF- algoritmen:

- Ligning (4.10) beregner sigmapunktene. Notasjonen $\hat{x}(k-1) \pm \Gamma \sqrt{\hat{P}(k-1)}$ ² betyr her at $\hat{x}(k-1)$ adderes/subtraheres til hver kolonne i $\Gamma \sqrt{\hat{P}(k-1)}$. Dette gir $2n$ sigmapunkt, der n er antall tilstander i prosessen. For totankprosessen blir da $\chi(k-1)$ en 2×4 matrise med ett sigmapunkt i hver kolonne:

$$\chi(k-1) = \begin{bmatrix} \chi_{x11} & \chi_{x12} & \chi_{x13} & \chi_{x14} \\ \chi_{x21} & \chi_{x22} & \chi_{x23} & \chi_{x24} \end{bmatrix} \quad (4.21)$$

- Ligning (4.11) propagerer sigmapunktene gjennom de ulineære prosessligningene som tilsvarer å beregne apriori tilstandsestimat som for EKF (ligning (3.4)) for hvert sigmapunkt.
- Ligning (4.12) beregner apriori tilstandsestimat ved å vekte $\chi_i(k)$ med W_i^x og deretter summere. Siden man har $2n = 4$ sigmapunkter, og ligning (4.4) gir $W_i^x = \frac{1}{4}$ tilsvarer dette gjennomsnitt av sigmapunktene:

$$\bar{x}(k) = \frac{1}{4} \cdot \chi_1 + \frac{1}{4} \cdot \chi_2 + \frac{1}{4} \cdot \chi_3 + \frac{1}{4} \cdot \chi_4 \quad (4.22)$$

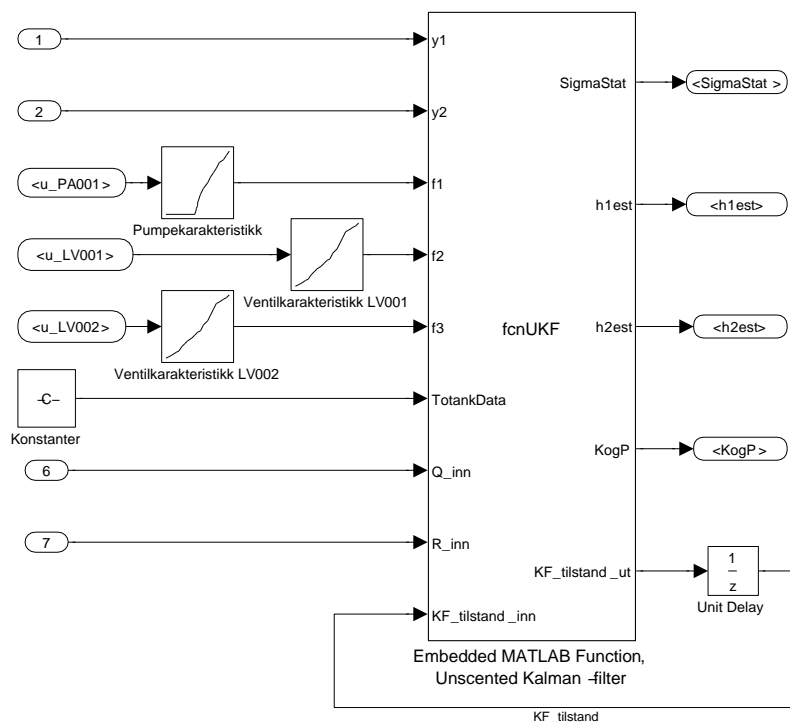
²Matrisekvadratrotten, R , av en matrise, A , er definert ved at $A = R \cdot R^T$

- Ligning (4.13) beregner apriori kovariansmatrise.
- Ligning (4.14) propagerer sigmapunktene gjennom måleligningen som kan være ulineære. På totankprosessen måles nivået i tank 2 direkte. $\gamma_i(k)$ blir da en vektor med sigmapunktverdiene for x_2 .
- Ligning (4.15) beregner predikert måling ved å vekte $\gamma_i(k)$ med W_i^x og deretter summere.
- Ligning (4.16) og (4.17) beregner målekovariansmatrisa, som her blir en skalar, og kryss-kovariansen mellom tilstand og måling, som blir en 2×1 kolonnevektor.
- Ligning (4.18) beregner Kalmanfilterforsterkningen.
- Ligning (4.19) beregner aposteriori tilstandsestimat, som blir utgangen fra Kalmanfilteret. Dette tilstandsestimatet skal også brukes i ligning (4.10) ved neste tidssteg.
- Ligning (4.20) oppdaterer kovariansmatrisen som skal brukes i ligning (4.10) ved neste tidssteg.

4.2 Implementering av UKF i Simulink

"Unscented" Kalmanfilteret implementeres som en funksjonsblokk av typen "Embedded MATLAB Function" i Simulink. Funksjonsblokken inneholder Matlab-kode som implementer UKF-ligningene fra ligning (4.10) til ligning (4.20). Se vedlegg A.4 for utskrift av kode i funksjonsblokk.

Figur 4.2 viser funksjonsblokken med innganger og utganger som beskrevet i tabell 4.1.



Figur 4.2: UKF Simulink

Tabell 4.1: Innganger og utganger UKF

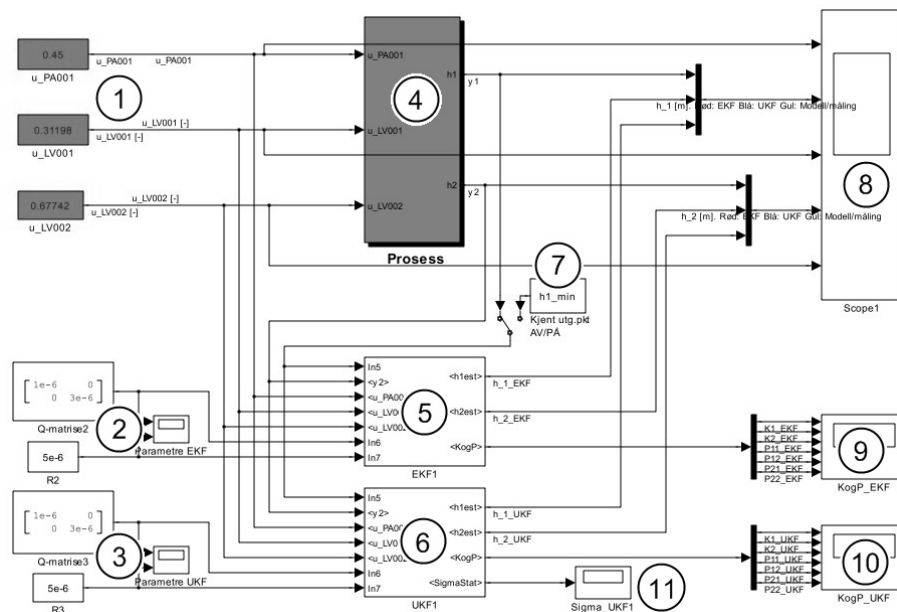
Inn-/utgang	Forklaring
y1	Måling av nivå i tank 1. Benyttes kun ved initialisering
y2	Måling av nivå i tank 2
f1	Tabelloppslag for pumpekararakteristikk for pumpe PA001
f2	Tabelloppslag for ventilkarakteristikk for ventil LV001
f3	Tabelloppslag for ventilkarakteristikk for ventil LV002
TotankData	Henter inn konstanter fra to_tank_data.m
Q_inn	Tuning-matrise Q som brukes i ligning (4.13).
R_inn	Tuning-matrise R som brukes i ligning (4.16).
KF_tilstand_inn	Verdier fra forrige tilstand som brukes i ligning (3.3), (3.4) og (3.5)
SigmaStat	Sigma punkt og propagerte sigma punkt hentes ut for hvert tidssteg.
h1est	Estimat av høyde i tank 1
h2est	Estimat av høyde i tank 2
KogP	Kalmanfilterforsterkning og P -matrise
KF_tilstand_ut	Verdier som skal brukes ved neste tilstand i ligning (3.3), (3.4) og (3.5)

Kapittel 5

Testing og tuning

I dette kapittelet blir EKF og UKF utviklet i kap. 3 og 4 testet mot totankprosessen. Filtrene testes også mot hverandre ved ulike driftssituasjoner for å studere hvorvidt ytelsen er forskjellig.

Figur 5.1 viser hvordan denne sammelingen mot totankprosessen er implementert i Simulink.



Figur 5.1: Kalmanfilter mot prosess

Hver hoveddel av implementeringen i figur 5.1 er nummerert, og med referanse til dette nummeret blir implementeringen forklart mer detaljert.

1. Innganger til prosessen; pådragssignal til pumpe *PA001* og de to ventilene *LV001* og *LV002*. (Styres manuelt fra PC).

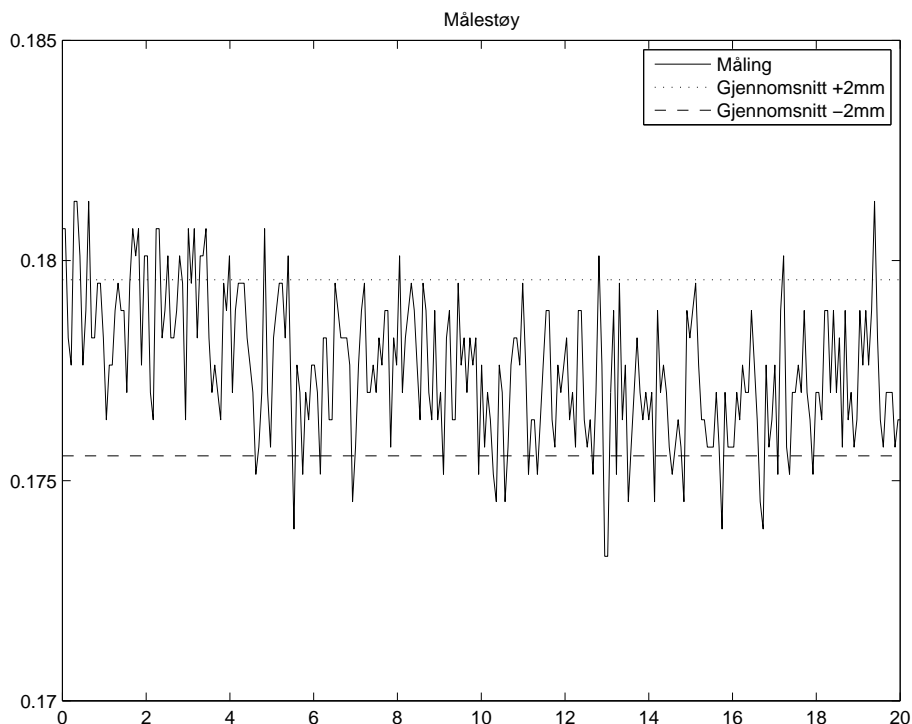
2. Tuning-matriser, Q og R , og logging av disse for EKF.
3. Tuning-matriser, Q og R , og logging av disse for UKF.
4. Totankprosessen.
5. Funksjonsblokk EKF, figur 3.1.
6. Funksjonsblokk UKF, figur 4.2.
7. Bryter for å velge om initielt nivå i tank 1 skal være kjent.
8. Logging av prosessinn ganger, nivåmålinger og estimat fra UKF og EKF
9. Logging av Kalmanfilterforsterkning og P -matrise for EKF.
10. Logging av Kalmanfilterforsterkning og P -matrise for UKF.
11. Logging av sigmapunkter fra UKF.

Under disse forsøkene ønsker man å sammenligne ytelsen til filtrene ved forskjellige innstillinger av tuning-matrisene. For enkelt å kunne sammenligne virkningen av ulike innstillinger, logges innganger og utganger ved kjøring av totankprosessen. Deretter kan filtrene kjøres mot de loggede inngangene og utgangene fra prosessen. Simulink-implementeringen for dette er vist i vedlegg A.5.

5.1 Normal drift

Fra teorien om Kalmanfilter er det en tommelfingerregel å sette R tilnærmet lik variansen til målestøyen i systemet [4], for deretter å bruke diagonalelementene i matrisen Q som tuning-parametre. Diagonalelementene i Q som representere varians til prosesstøyen kan det være vanskelig å vite noe om på forhånd. Vanligvis benytter man seg av prøving og feiling for å stille inn elementene i Q [1].

For å ha et utgangspunkt for R viser figur 5.2 nivåmåling i tank 2 over en 20 sekunds periode. Målinger viser at målestøyen gir et utslag i størrelsesorden 1-3 mm avhengig av dagsform og nivå i tanken. Ved å anta at støyen er Gaussisk fordelt, har man at ca 68% av observasjonene vil ligge innenfor ett standardavvik fra gjennomsnittet [9].



Figur 5.2: Målestøy

Figuren viser også linjene for gjennomsnitt $\pm 2mm$. Disse er valgt ut fra at flere forsøk har vist at det meste av målestøyen ligger innenfor disse linjene.

Fra figur 5.2 og teorien over fås

$$R = \sigma^2 \approx (2 \cdot 10^{-3})^2 = 4 \cdot 10^{-6} \quad (5.1)$$

Et eksakt tall for målestøyen lar seg ikke finne da denne varierer over tid og etter væsknivå. Det er heller ikke viktig å ha dette nøyaktig, da tuningparametrene uansett kan fin-tunes under drift. Fra en tidligere hovedoppgave [7] er $R = 5 \cdot 10^{-6}$ benyttet, og siden anslaget fra figur 5.2 bekrefter størrelsesordenen benyttes også $R = 5 \cdot 10^{-6}$ som utgangspunkt her.

Ved å anta at nivået i begge tankene kan variere med ca 1 mm per tidssteg, velger $Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 1 \cdot 10^{-6} \end{bmatrix}$ som et utgangspunkt.

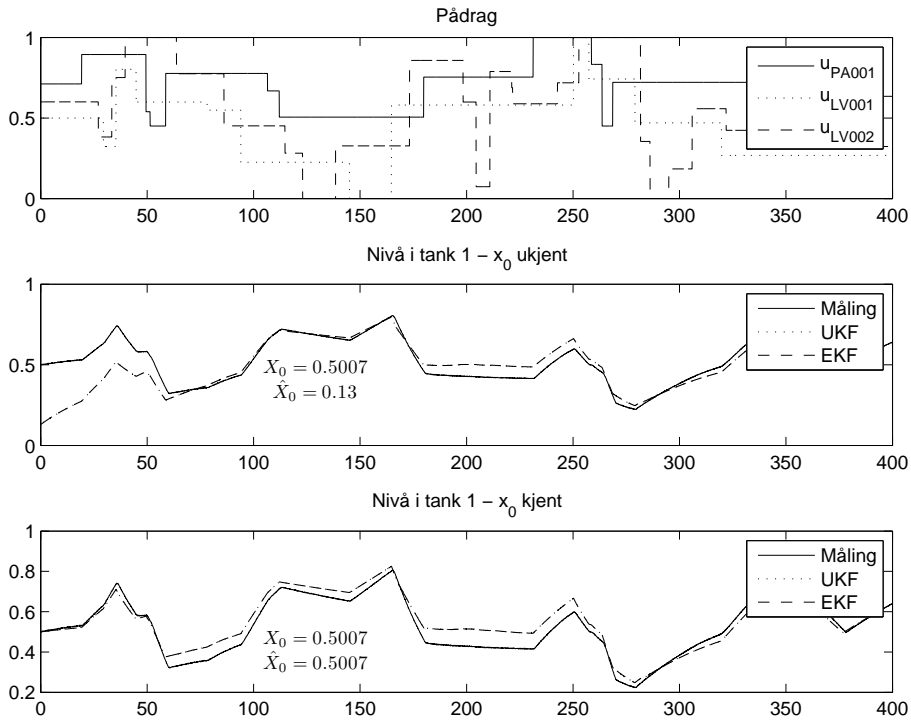
Etter å ha testet filtrene mot totankprosessen flere ganger med forskjellige verdier for Q viser det seg at $Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 3 \cdot 10^{-6} \end{bmatrix}$ gir gode resultat for både EKF og UKF. Resultatene i figur 5.3 viser filtrenes prestasjon med $R = 5 \cdot 10^{-6}$ og $Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 3 \cdot 10^{-6} \end{bmatrix}$.

Fra forsøkene over er følgende tuningparametre valgt:

$$R = 5 \cdot 10^{-6} \quad (5.2)$$

$$Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 3 \cdot 10^{-6} \end{bmatrix} \quad (5.3)$$

En av forutsetningene for Kalmanfilteret er at alle initielle tilstander er kjent. Figur 5.3 viser hvordan filtrene estimerer seg inn ved ukjente startverdier, sammenlignet med når startverdiene er kjent. For forsøket med kjente initialtilstander måles nivået i begge tankene ved det første tidssteget.



Figur 5.3: Innsvingning ved ukjent/kjent initialtilstand

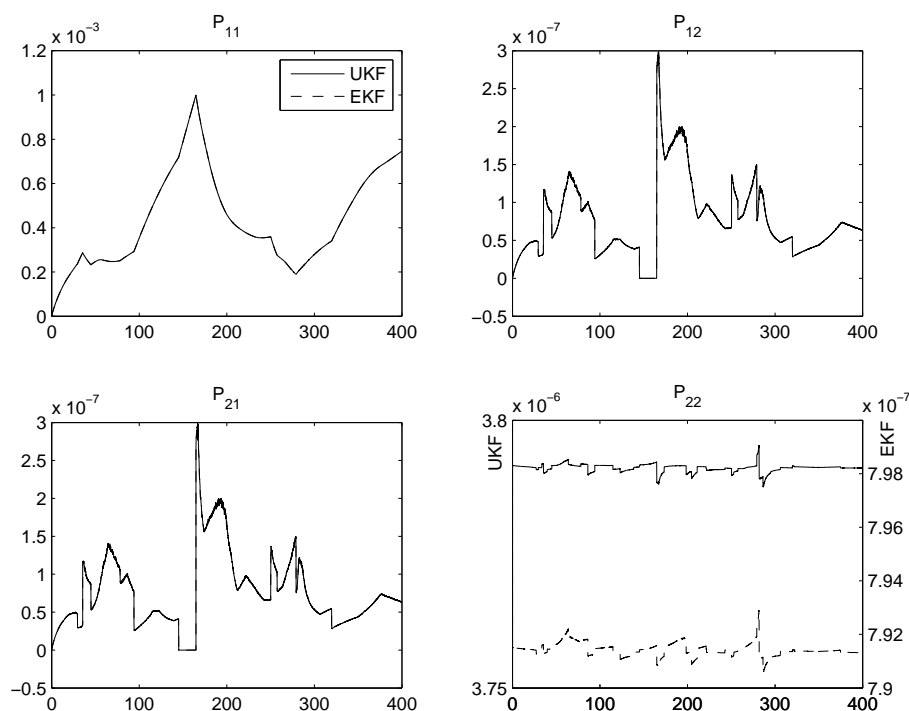
Figur 5.3 viser at de valgte parametrene gir god ytelse både når initielt nivå i tank 1 er kjent og ikke. Filtrene svinger seg inn til rett verdi i løp av ca ett minutt når initialtilstanden er ukjent. Sann tilstand, X_0 , og estimert tilstand \hat{X}_0 i oppstartstispunktet er også vist i figuren over. At filtrene svinger seg inn såpass raskt, og følger målingen så bra, tyder på at både modellen funnet i kap. 2 og tuning-parametrene funnet her passer godt.

5.1.1 P-matrise

Kovariansmatrisen for estimeringsavvikene, også kalt P-matrisen, gir informasjon om hvordan usikkerheten i tilstandsestimatene er. P-matrisen fra forrige

tidssteg benyttes til å beregne Kalmanfilterforsterkning i ligning (3.7) for EKF og ligning (4.18) for UKF. P-matrisen oppdateres altså for hvert tidssteg ut fra hvordan avviket mellom estimat og måling utvikler seg.

Figur 5.4 viser hvordan P-matrisen på formen $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$ utvikler seg ved forsøket i figur 5.3 for EKF og UKF.

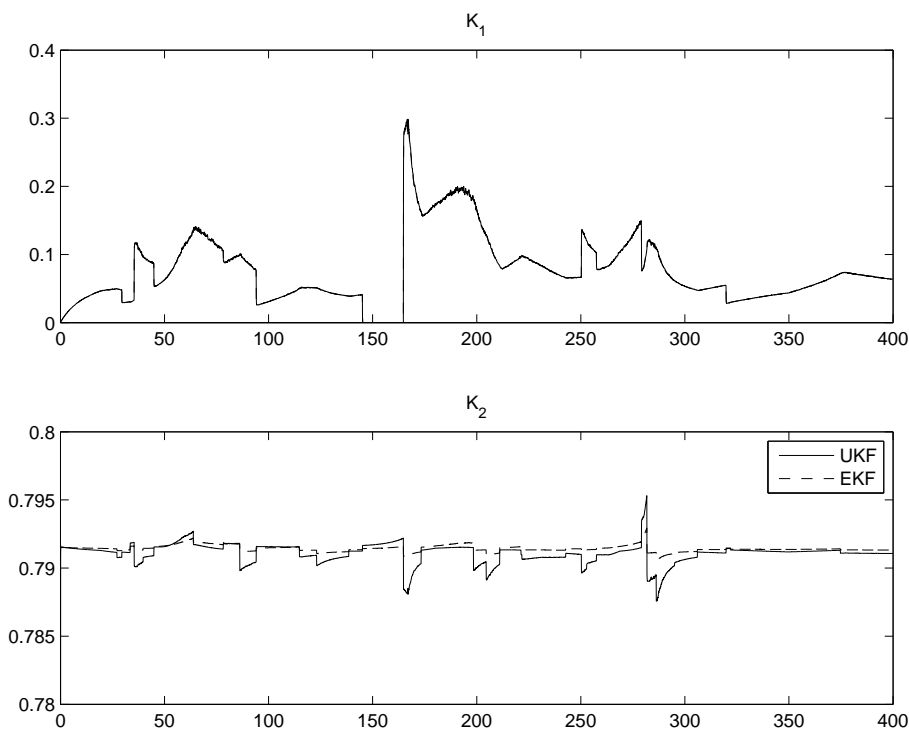


Figur 5.4: P-matrise

Det er verdiene til P_{11} og P_{22} som er mest intuitive siden disse gir informasjon om estimeringsavvikene for nivået i henholdsvis tank 1 og tank 2. Som det går fram av figuren over, variere P_{11} mye i løp av kjøringen. Dette er også rimelig siden man her ikke måler nivået i tank 1. Som det går frem av figuren er det en markant forskjell i P_{22} mellom EKF og UKF. Det har ikke lyktes å finne ut hva som forårsaker denne forskjellen, men det antas at det har med hvordan P -matrisen beregnes i de to algoritmene 3.1 og 4.1 å gjøre, og som det blir vist i neste kapittel gir ikke denne forskjellen i P_{22} et betydelig utslag i Kalmanfilterforsterkningen for de to filtrene. P_{22} er meget lavt og tilnærmet konstant for begge filtrene. Dette er som forventet, siden man måler nivå i tank 2 i hvert tidssteg, og derfor hele tiden har et godt estimat for nivået. Elementene i P -matrisen er alltid ≥ 0 . Som det går frem av figuren er både P_{12} og P_{21} null i et lite intervall. Ved å studere figur 5.3 og sammenligne med figur 5.4 ser man at dette er i det samme intervallet hvor ventil $LV001$ er stengt. Når ventilen mellom de to tankene er stengt mister man koblingen mellom tankene, og siden P_{12} og P_{21} sier noe om denne koblingen det er da logisk at disse blir null.

5.1.2 Kalmanfilterforsterkning

P-matrisen fra kap. 5.1.1 benyttes til å beregne Kalmanfilterforsterkningen, K . Denne forsterkningsfaktoren benyttes av både EKF og UKF til å bestemme hvor raskt filtrene skal korrigere for avvik mellom apriori tilstandsestimat og måling. Jo mer man kan stole på estimatet av hver tilstand, dess større vil forsterkningsfaktorene bli. Figur 5.5 viser hvordan K_1 og K_2 utvikler seg ved forsøket i kap. 5.1 for EKF og UKF.



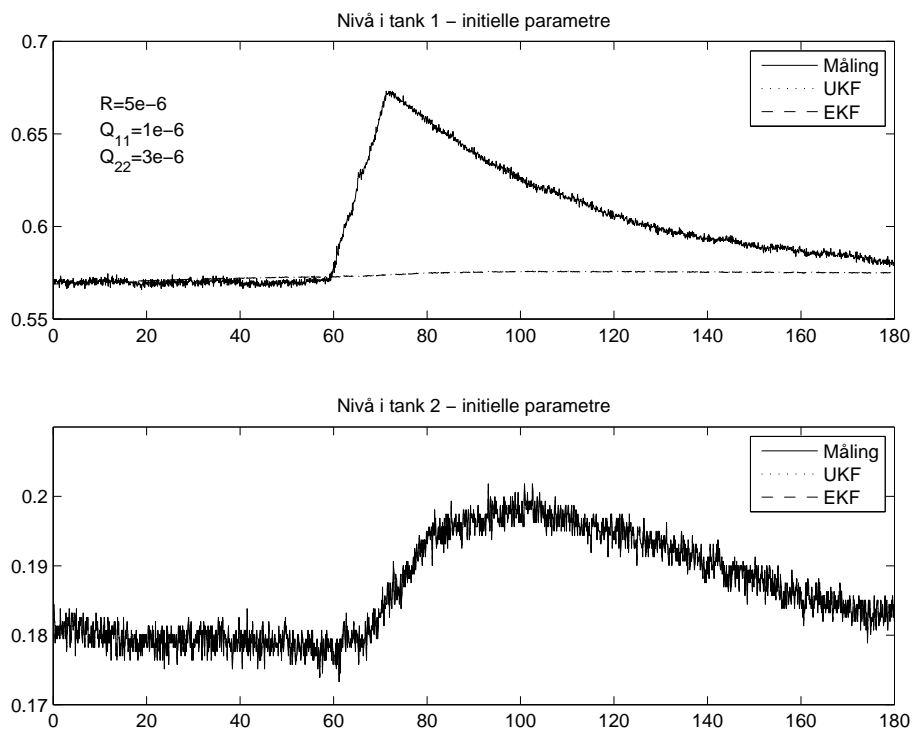
Figur 5.5: Kalmanfilterforsterkningen

Som vist i figur 5.5 er forsterkningsfaktoren, K_1 , for tank 1 mye mindre enn K_2 , for tank 2, og dette er som forventet siden man måler nivået i tank 2 direkte og dermed har et bedre grunnlag for raskt å korrigere for estimeringsavvik for denne tanken. Som for P_{12} og P_{21} i P -matrisen ser man her at også K_1 blir null i intervallet hvor ventilen mellom de to tankene er stengt. Dette betyr at nivåestimatet for tank 1 ikke blir justert ut fra endringer i tank 2 når ventilen mellom tankene er steng. Dette er fornuftig da stengt ventil betyr at endringer i nivå i tank 2 ikke kan skylles endring av nivå i tank 1. Figur 5.5 viser også at UKF og EKF gir tilnærmet like forsterkningsfaktorer. At K_2 varierer litt mer for UKF enn for EKF skyldes at P_{21} er litt større for UKF enn EKF, ref figur 5.4.

Kalmanfiltrene med parametre som i ligning (5.2) og (5.3) har vist seg å fungere godt for å estimere nivå i tank 1 under normal drift¹. Det kan tenkes situasjoner hvor prosessen kan bli påvirket eksternt, for eksempel at vann blir tilført eller tappet ut av en av tankene eller at fremmedlegemer legger seg i pumpen eller i en av ventilene og påvirker strømming gjennom disse. For å sjekke hvor "robuste" Kalmanfiltrene er mot ekstern påvirkning blir det her vist hvordan estimatet av høyden i tank 1 påvirkes når systemet utsettes for forstyrrelser. Under disse testene holdes pådragene til pumpen og ventilene konstant mens nivåene i tank 1 og tank 2 manipuleres.

5.2 Prosessforstyrrelse tank 1

Nivået i tank 1 utsettes for en ekstra forstyrrelse ved at en liter ekstra vann tilføres tanken. Initielt benyttes tuning-parametrene funnet i ligning (5.2) og (5.3).



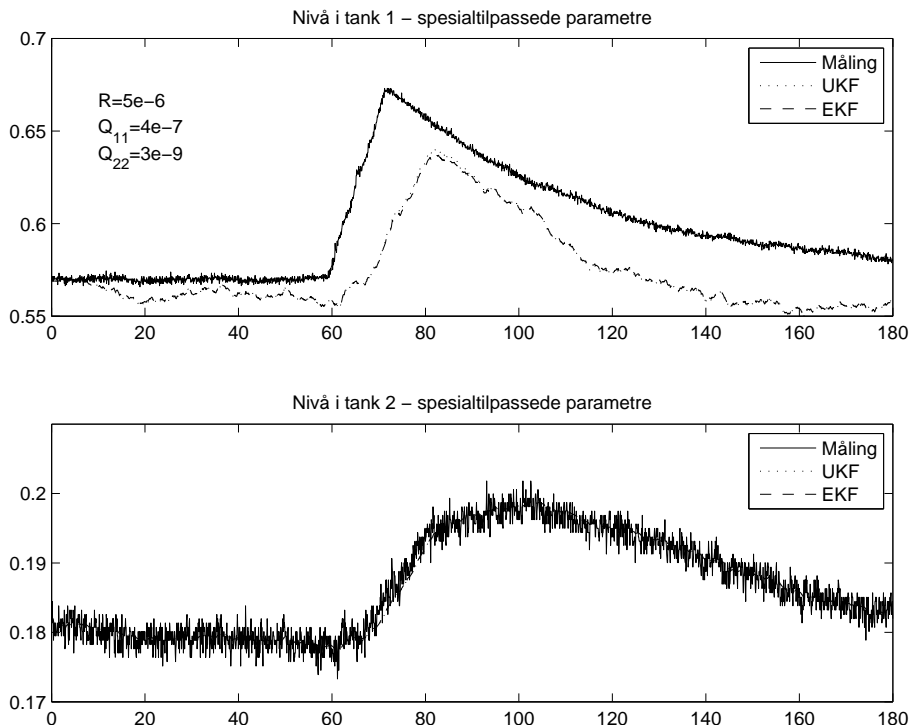
Figur 5.6: Forstyrrelse i tank 1, initielle tuning-parametre

Som det går fram av figur 5.6 klarer ikke filtrene å estimere denne eksterne påvirkningen. Etter flere forsøk har det vist seg at man ved å endre Q til

¹Med "normal drift" menes her at prosessen ikke blir utsatt for ekstern påvirkning

$$Q = \begin{bmatrix} 4 \cdot 10^{-7} & 0 \\ 0 & 3 \cdot 10^{-9} \end{bmatrix} \quad (5.4)$$

får et Kalmanfilter som håndterer forstyrrelser i tank 1 mye bedre, se figur 5.7.

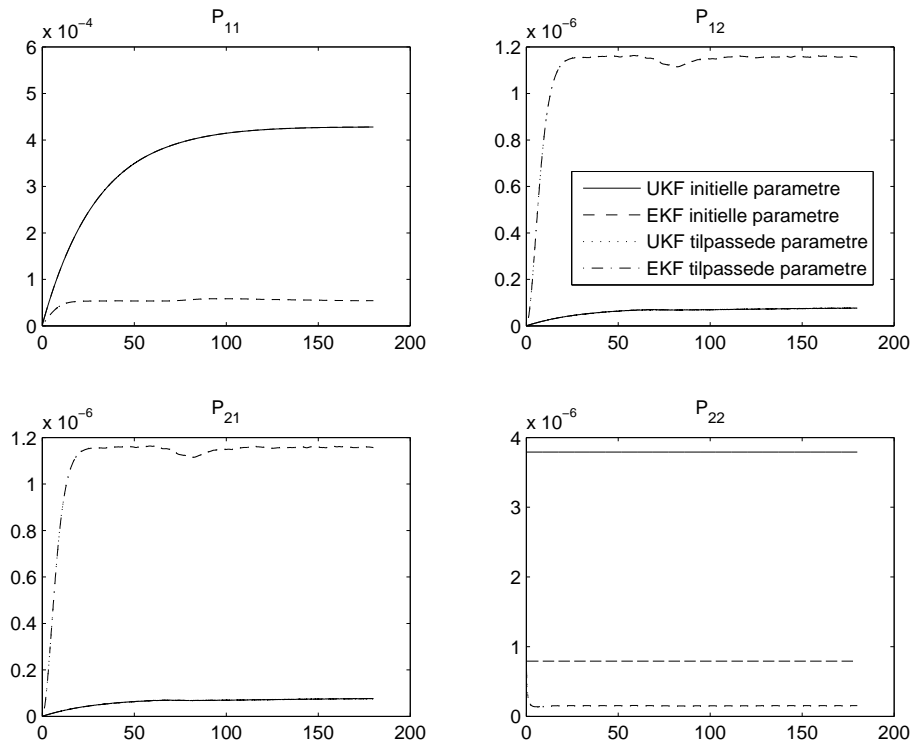


Figur 5.7: Forstyrrelse i tank 1, tilpassede tuning-parametre

I den nye Q -matrisen er prosesstøykovariansen for tank 2 forholdsvis mye mindre enn for tank 1. Dette fører til at Kalmanfiltrene i større grad tolker endringen av målt nivå i tank 2 som en konsekvens av forstyrrelse i tank 1 enn som forstyrrelse i tank 2. Som det går frem av figur 5.7 blir estimatene for nivået i tank 1 også mer støyfulle som en følge av de nye parametrene. Dette skyldes at den nye Q -matrisen fører til at en del av prosesstøyen som er i tank 2 blir tolket som forstyrrelse i tank 1.

5.2.1 P-matrise

Figur 5.8 viser hvordan elementene i P -matrisen utvikler seg i løp av forsøket med forstyrrelse i tank 1. Det går klart fram av figuren at kovariansen til estimeringsavvikene for nivået i begge tankene er lavere med de nye tuning-parametrene.



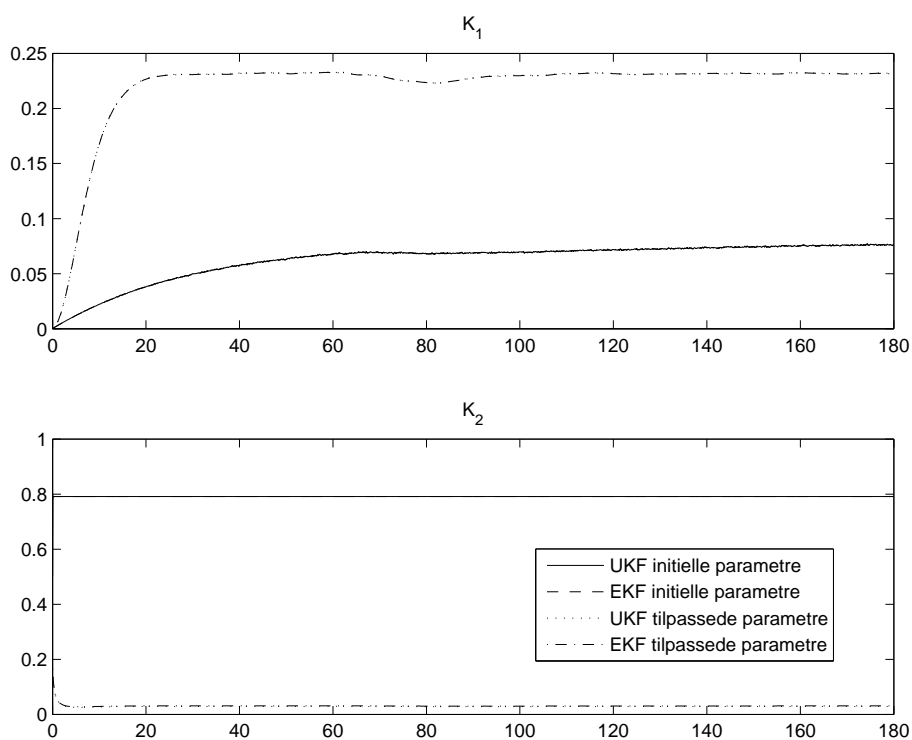
Figur 5.8: P-matrise, forstyrrelse i tank 1

Av figuren kan det se ut som det bare er to kurver i hver graf. Grunnen til dette er at UKF og EKF presterer så likt at kurvene for UKF og EKF legger seg over hverandre, unntatt for P_{22} .

Elementene i P -matrisen er også mye glattere enn i figur 5.4 siden alle pådrag her er faste.

5.2.2 Kalmanfilterforsterkning

Som forklart i forbindelse med figur 5.6 gjør de nye tuning-parametrene at endringer i nivå i tank 2 blir tolket som at det har vært en endring av nivå i tank 1. Dette viser i figur 5.9 hvor K_1 som bestemmer hvor raskt nivåestimatet i tank 1 skal oppdateres er mye større med de nye tuning-parametrene, mens K_2 blir mye mindre.

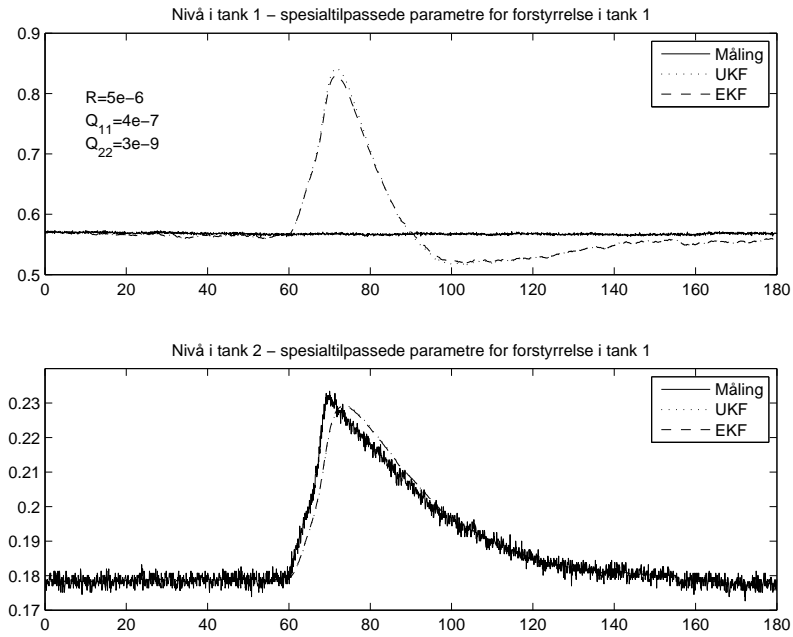
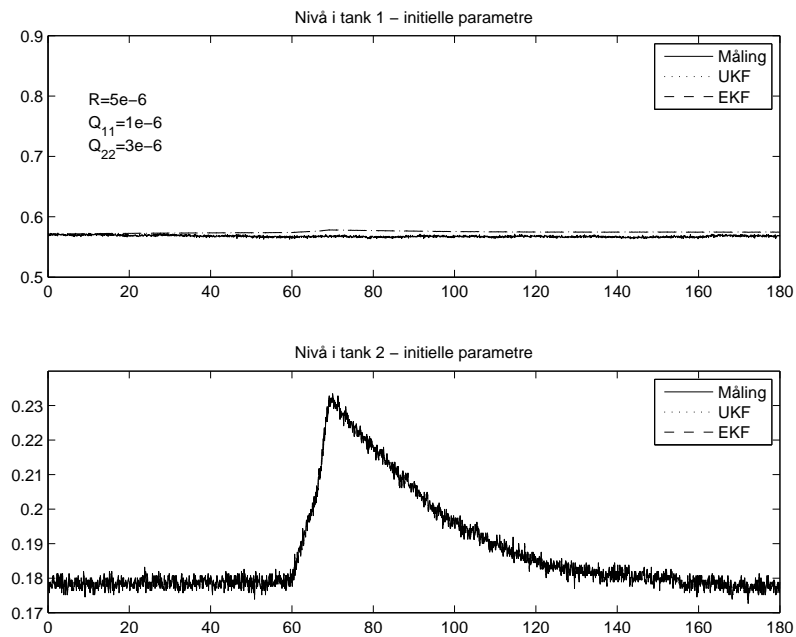


Figur 5.9: Kalmanfilterforsterkning, forstyrrelse i tank 1

5.3 Prosessforstyrrelse tank 2

I dette forsøket manipuleres nivået i tank 2 på samme måte som for tank 1 i kap. 5.2, ved at en liter ekstra vann tilsettes. Først sjekkes det hvordan filtrene fungerer med parametre funnet i ligning (5.4), som raskt fanget opp endring i nivået i tank 1.

Som det går fram av figur 5.10 gjør disse tuning-parametrene at filtrene gir store utslag i estimatet for nivå i tank 1 ved forstyrrelse i tank 2. Dette skyldes at filtrene som forklart i kap. 5.2 tolker nivåendringen i tank 2 som et resultat av en nivåendring i tank 1. For å teste hvorvidt tuning-parametrene funnet i ligning (5.2) og (5.3) gir bedre prestasjon utføres forsøket på nytt med disse.

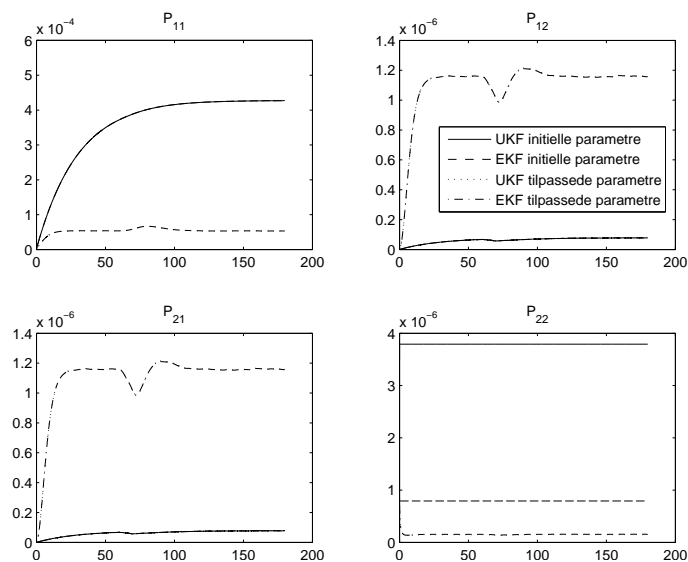
Figur 5.10: Forstyrrelse i tank 2, Q som funnet i ligning (5.4)

Figur 5.11: Forstyrrelse i tank 2, initielle tuning-parametre

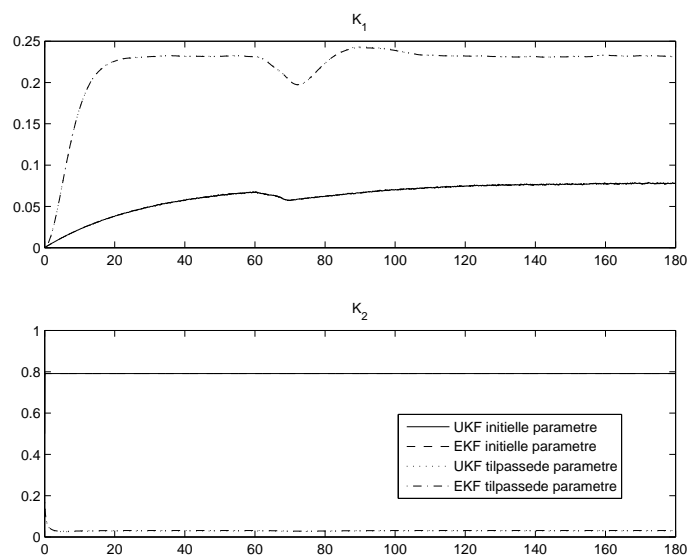
Figur 5.11 viser at forstyrrelsen bare gir et minimalt bidrag til estimatet av nivået i tank 1 med disse parametrene.

5.3.1 P-matrise og Kalmanfilterforsterkning

P-matrisen i figur 5.12 og Kalmanfilterforsterkningen i figur 5.13 blir her omtrent som de ble vist i kap. 5.2.



Figur 5.12: P-matrise, forstyrrelse i tank 2



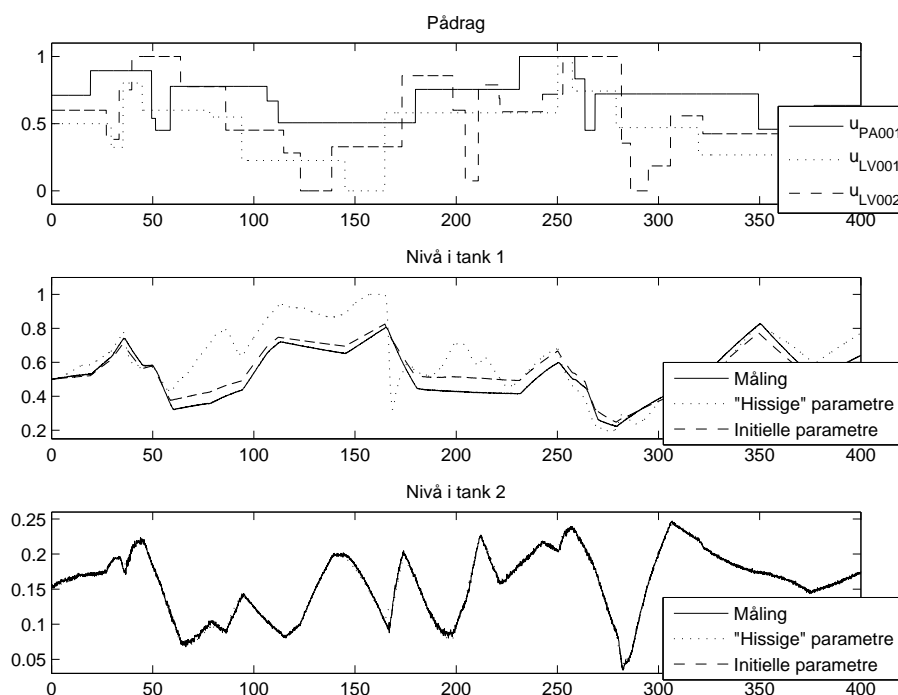
Figur 5.13: Kalmanfilterforsterkning, forstyrrelse i tank 2

At disse er så like figurene for P -matrise, figur 5.8, og Kalmanfilterforsterkningen, figur 5.9, fra kap. 5.2 er rimelig siden det er de samme tuning-parametrene som er benyttet.

5.4 Mer om valg av tuning-parametre

Som det framgår av de foregående kapitlene kan man ikke få i pose og sekk. Tuning-parametre som raskt oppdager forstyrrelse i tank 1 vil reagere altfor kraftig på forstyrrelser i tank 2. Og motsatt vil tuning-parametrene som kompenserer for forstyrrelser i tank 2, gjør at filtrene blir veldig trege til å oppdage forstyrrelse i tank 1.

For å designe et best mulig Kalmanfilter må man derfor ha kjennskap til prosessen og vite noe om hva slags forstyrrelser en kan vente seg. For totankanlegget ser en at under vanlig drift fungerer begge filtrene relativt bra med tuning-parametrene funnet i kap. 5.1. Ønsker man at filteret skal bli raskere til å oppdage forstyrrelser på nivået i tank 1 må man gjøre noe med tuning-parametrene. En kan for eksempel prøve med parametrene mer i retning av de som ble funnet i kap. 5.2. Figur 5.14 viser hvordan forskjellige tuning-parametre fungerer på den loggede kjøringen fra kap. 5.1.



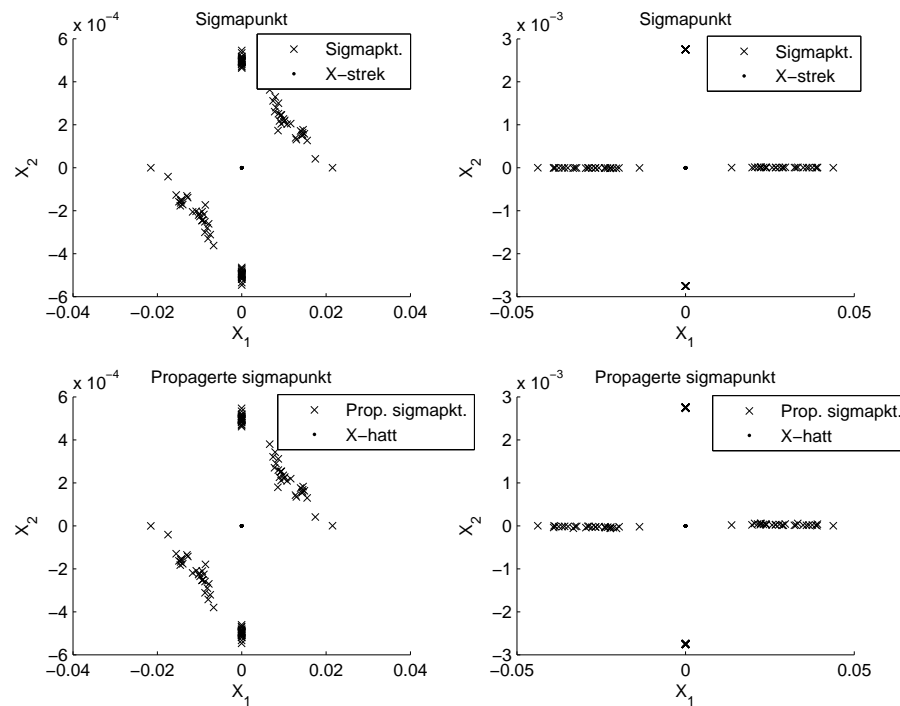
Figur 5.14: Forskjellige tuning-parametre

Som det går frem av figur 5.14 vil $R = 5 \cdot 10^{-6}$ og $Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 3 \cdot 10^{-6} \end{bmatrix}$ (vist som "Initielle parametre" i figur 5.14) som ble funnet i kap. 5.1 for vanlige situasjoner fungere mye bedre enn $R = 5 \cdot 10^{-6}$ og $Q = \begin{bmatrix} 4 \cdot 10^{-7} & 0 \\ 0 & 3 \cdot 10^{-9} \end{bmatrix}$ (vist som "Hissige parametre" i figur 5.14) som i kap. 5.2 ble valgt for å raskt oppdage ekstern påvirkning på høyden i tank 1. Man må vurdere hvor ofte

forstyrrelser på nivået i tank 1 kan oppstå, og hvor viktig man anser det å oppdage dette raskt. Hvis man ønsker å kunne oppdage dette raskere enn det de opprinnelige parametrene fra kap. 5.1 gjør, kan man velge en Q -matrise et sted mellom de som er foreslått over.

5.5 Sigmapunkter

Som det framgår av figurene i foregående kapitler, så har resultatene fra EKF og UKF vært svært like på totankprosessen. Som vist i figur 4.1 vil fordelingen til sigmapunktene endre seg når sigmapunktene blir propagert gjennom ulineariteten. Jo større ulinearitet, dess større vil forskjellen mellom fordelingen til sigmapunktene og de propagerte sigmapunktene være. Figuren under viser fordelingen til sigmapunkter og propagerte sigmapunkter ved forskjellige tuning-parametre ved samme kjøring. Overgangen fra sigmapunktene til de propagerte sigmapunktene er beskrevet i kap. 4.1.



Figur 5.15: Sigmapunkter ved forskjellige tuning-parametre

Som figuren viser har de propagerte sigmapunktene så godt som identisk fordeling som de opprinnelige sigmapunktene plukket rundt apriori tilstandsestimat. Ved å se dette i sammenheng med figur 4.1 ser en at det ikke er så mye ulinearitet rundt apriori tilstandsestimat. Dette forklarer hvorfor EKF og UKF har gitt så like estimater.

Del II

Kalmanfilter for estimering av lekkasje i ventil og/eller tank

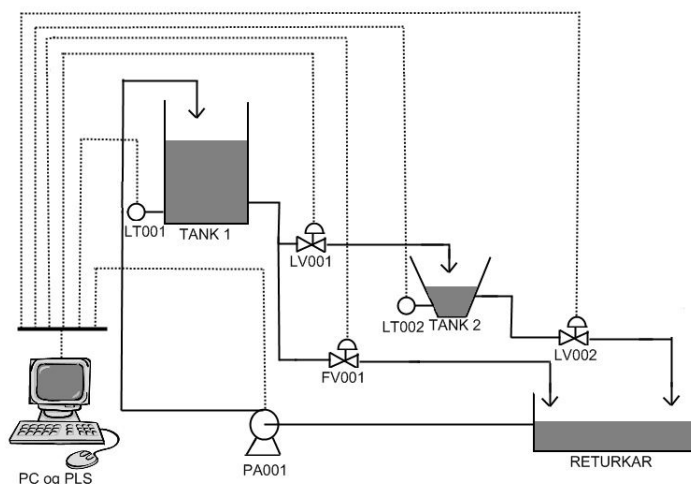
Kapittel 6

Modellering

Her utvides modellen fra del I for å kunne brukes til lekkasjeestimering. Målet er å utvikle en metode for å oppdage lekkasje fra tank 1 ut fra kjenneskap til prosessen og måling av nivå i tank 2. I industriell sammenheng kan det være flere anvendelser for denne teknologien, feks for å detektere uønsket varmetap, lekkasje eller inntrengning av uønsket medie i en prosess.

6.1 Utvide modell fra del I

Modellen for totankprosessen som ble utviklet i kap. 2 under del I hadde to tilstander som skulle estimeres, høyde i tank 1 og i tank 2. I del II skal modellen utvides for å inkludere lekkasje fra tank 1, Q_{lekk} , gjennom ventil $FV001$. Se prosesskisse i figur 6.1 og tabell 6.1 som forklarer symbolene som benyttes videre i dette kapittelet



Figur 6.1: Prosesskisse av totankoppsett del II

Tabell 6.1: Forklaring til symboler i differensialligningene

Variabel	Enhet	Beskrivelse
$h_1(t)$	m	Vannnivå i tank 1 målt fra bunnen i tanken
$h_2(t)$	m	Vannnivå i tank 2 målt fra bunnen i tanken
h_{LV001}	m	Høyde fra ventil $LV001$ opp til bunn i tank 1
h_{LV002}	m	Høyde fra ventil $LV002$ opp til bunn i tank 2
$Q_{lekk}(t)$	$\frac{m^3}{s}$	Lekkasje fra tank 1 til oppsamlingskar gjennom ventil $FV001$
$q_{PA001}(t)$	$\frac{m^3}{s}$	Strømning av vann gjennom pumpe $PA001$
$q_{LV001}(t)$	$\frac{m^3}{s}$	Strømning av vann gjennom ventil $LV001$
$q_{LV002}(t)$	$\frac{m^3}{s}$	Strømning av vann gjennom ventil $LV002$
A_1	m^2	Areal av vannspeilet i tank 1
$A_2(t)$	m^2	Areal av vannspeilet i tank 2
Kv_{LV001}	$\frac{m^3}{h \cdot \sqrt{bar}}$	Ventilkonstant for ventil $LV001$
Kv_{LV002}	$\frac{m^3}{h \cdot \sqrt{bar}}$	Ventilkonstant for ventil $LV002$
$u_{PA001}(t)$	-	Pådrag på pumpe $PA001$
$u_{LV001}(t)$	-	Pådrag på ventil $LV001$
$u_{LV002}(t)$	-	Pådrag på ventil $LV002$
$f_1(u_{LV001}(t))$		Tabelloppslag av ventilkarakteristikk
$f_2(u_{LV002}(t))$		Tabelloppslag av ventilkarakteristikk
$f_3(u_{PA001}(t))$	$\frac{m^3}{s}$	Tabelloppslag av pumpekarakteristikk
ρ	$\frac{kg}{m^3}$	Tettheten til vann
g	$\frac{m}{s^2}$	Tyngdeakselerasjon
c_1		Samling av konstanter i differensialligning 2.2
c_2		Samling av konstanter i differensialligning 2.5

6.2 Differensialligninger

6.2.1 Differensialligning for nivå i tank 1

Differensialligningen (2.3) for nivået i tank 1 som ble funnet i kap. 2.1 utvides ved at lekkasjen $Q_{lekk}(t)$ tas med:

$$\frac{dh_1(t)}{dt} = \frac{1}{A_1} \left(f_3(u_{PA001}(t)) - c_1 \cdot f_1(u_{LV001}(t)) \cdot \sqrt{h_1(t) + h_{LV001}} - Q_{lekk}(t) \right) \quad (6.1)$$

der c_1 er gitt ved:

$$c_1 = \frac{Kv_{LV001}}{3600} \cdot \sqrt{\frac{\rho \cdot g}{100000}} \quad (6.2)$$

6.2.2 Differensialligning for nivå i tank 2

Siden det ikke gjøres endringer i hva som strømmer inn og ut av tank 2 blir differensialligningen (2.7) for nivået i tank 2 som ble funnet i kap. 2.1 som før:

$$\frac{dh_2(t)}{dt} = \frac{1}{A_2(t)} \left(c_1 \cdot f_1(u_{LV001}(t)) \cdot \sqrt{h_1(t) + h_{LV001}} - c_2 \cdot f_2(u_{LV002}(t)) \cdot \sqrt{h_2(t) + h_{LV002}} \right) \quad (6.3)$$

der c_2 og $A_2(t)$ er gitt ved:

$$c_2 = \frac{K v_{LV002}}{3600} \cdot \sqrt{\frac{\rho \cdot g}{100000}} \quad (6.4)$$

$$A_2(t) = 0.004 + 0.07 \cdot h_2(t) \quad (6.5)$$

6.2.3 Differensialligning for lekkasje

Lekkasjen $Q_{lekk}(t)$ som inngår i den nye differensialligningen (6.1) for nivå i tank 1 defineres som en egen tilstand. Siden lekkasjen ikke er avhengig av noen kjente parametre, modelleres den som en funksjon av hvit støy [1]. Da støy ikke tas med i differensialligningene fås:

$$\frac{dQ_{lekk}(t)}{dt} = 0 \quad (6.6)$$

Differensialligningen (6.6) diskretiseres ved hjelp av Eulers forovermetode (3.12):

Definerer:

$$x_3(k) = h_1(k) \quad (6.7)$$

og får:

$$x_3(k+1) = x_3(k) \quad (6.8)$$

6.3 Observerbarhet

En forutsetning for at Kalmanfilteret for estimering av lekkasje fra tank 1 skal fungere er som nevnt i kap. 1.2.2 at systemet er observerbart. I dette kapitlet blir observerbarhet undersøkt for systemet vist i figur 6.1. Før det kan avgjøres om systemet er observerbart må den nye transisjonsmatrisen $\Phi(k)$ finnes.

Definerer:

$$x_1(k) = h_1(k) \quad (6.9)$$

$$x_2(k) = h_2(k) \quad (6.10)$$

$$x_3(k) = Q_{lekk}(k) \quad (6.11)$$

og lineariserer systemet ved å benytte Eulers forovermetode, og får:

$$\begin{aligned}x_1(k+1) &= x_1(k) + T \cdot \dot{x}_1(k) = g_1 \\x_2(k+1) &= x_2(k) + T \cdot \dot{x}_2(k) = g_2 \\x_3(k+1) &= x_3(k) + T \cdot \dot{x}_3(k) = g_3\end{aligned}\quad (6.12)$$

Transisjonsmatrisen finnes ved uttrykket:

$$\Phi(k) = \begin{bmatrix} \frac{\partial g_1(\cdot)}{\partial x_1} & \frac{\partial g_1(\cdot)}{\partial x_2} & \frac{\partial g_1(\cdot)}{\partial x_3} \\ \frac{\partial g_2(\cdot)}{\partial x_1} & \frac{\partial g_2(\cdot)}{\partial x_2} & \frac{\partial g_2(\cdot)}{\partial x_3} \\ \frac{\partial g_3(\cdot)}{\partial x_1} & \frac{\partial g_3(\cdot)}{\partial x_2} & \frac{\partial g_3(\cdot)}{\partial x_3} \end{bmatrix}\quad (6.13)$$

på samme måte som vist i kap. 3.1.2. Som vist i vedlegg B.1 finnes den nye transisjonsmatrisen på formen:

$$\Phi(k) = \begin{bmatrix} \phi_{11} & 0 & \phi_{13} \\ \phi_{21} & \phi_{22} & 0 \\ 0 & 0 & \phi_{33} \end{bmatrix}\quad (6.14)$$

hvor

$$\begin{aligned}\phi_{11} &= 1 - \frac{T \cdot q_{LV001}}{2 \cdot A_1 \cdot (x_1 + h_{LV001})} \\ \phi_{13} &= -\frac{T}{A_1} \\ \phi_{21} &= \frac{T \cdot q_{LV001}}{2 \cdot A_2 \cdot (x_1 + h_{LV001})} \\ \phi_{22} &= 1 - \frac{0.07 \cdot T}{A_2^2} \cdot (q_{LV001} - q_{LV002}) - \frac{q_{LV002}}{2 \cdot A_2 \cdot (x_2 + h_{LV002})} \\ \phi_{33} &= 1\end{aligned}$$

6.3.1 Kontroll av observerbarhet

Systemet er observerbart hvis, og bare hvis observerbarhetsmatrisen

$$Q_o = \begin{bmatrix} D \\ D \cdot \Phi \\ \vdots \\ D \cdot \Phi^{n-1} \end{bmatrix}\quad (6.15)$$

har rang lik n , der n er systemets orden (antall tilstandsvariable) [1].

For modellen i ligning 6.12 er $n = 3$ og målematrisen D er gitt ved:

$$D = [0 \quad 1 \quad 0] \quad (6.16)$$

siden det er nivå i tank 2 som måles.

Ved å benytte transisjonsmatrisen fra ligning 6.14 fås observerbarhetsmatrisen (se vedlegg B.2 for utregning):

$$Q_o = \begin{bmatrix} 0 & 1 & 0 \\ \phi_{21} & \phi_{22} & 0 \\ \phi_{21} \cdot \phi_{11} + \phi_{22} \cdot \phi_{21} & \phi_{22}^2 & \phi_{21} \cdot \phi_{13} \end{bmatrix} \quad (6.17)$$

Som har $rang = 3 = n$ og systemet er observerbart.

Kapittel 7

Implementering, testing og tuning

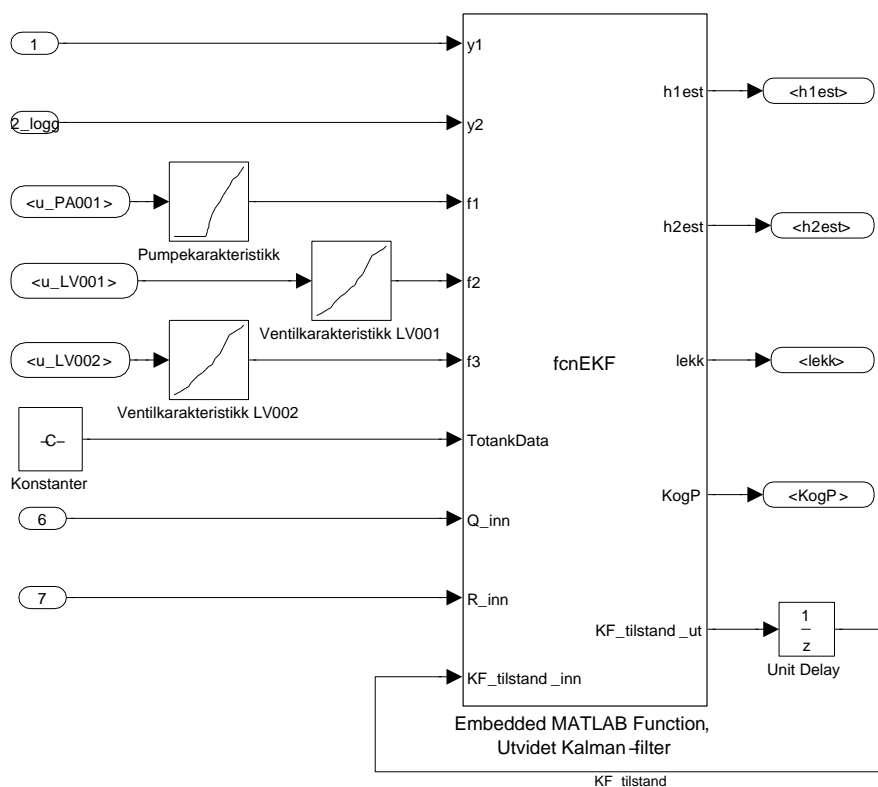
I dette kapitlet skal EKF for lekkasjeestimering utvikles og testes mot totankprosessen. Som en del av testingen for å se hvor godt det nye Kalmanfilteret fungerer, sammenlignes det også med det utvidede Kalmanfilteret funnet i kap. 3. Filtrene testes mot hverandre ved ulike driftssituasjoner for å studere hvorvidt ytelsen er forskjellig, og hvorvidt det nye Kalmanfilteret gir et godt estimat for lekkasje.

7.1 Implementering

I kap. 3.2 ble et utvidet Kalmanfilter for totankprosessen i del I utviklet. Ved å oppgradere dette med den nye modellen og den nye transisjonsmatrisen funnet i kap. 6.3, kan dette filteret brukes videre i denne delen.

Figur 7.1 viser funksjonsblokken med innganger og utganger som beskrevet i tabell 7.1. Den synlige forskjellen mellom denne nye blokken og blokken fra del I (figur 3.1) er utgangen ”lekk”.

Se vedlegg B.3 for utskrift av kode i funksjonsblokken.

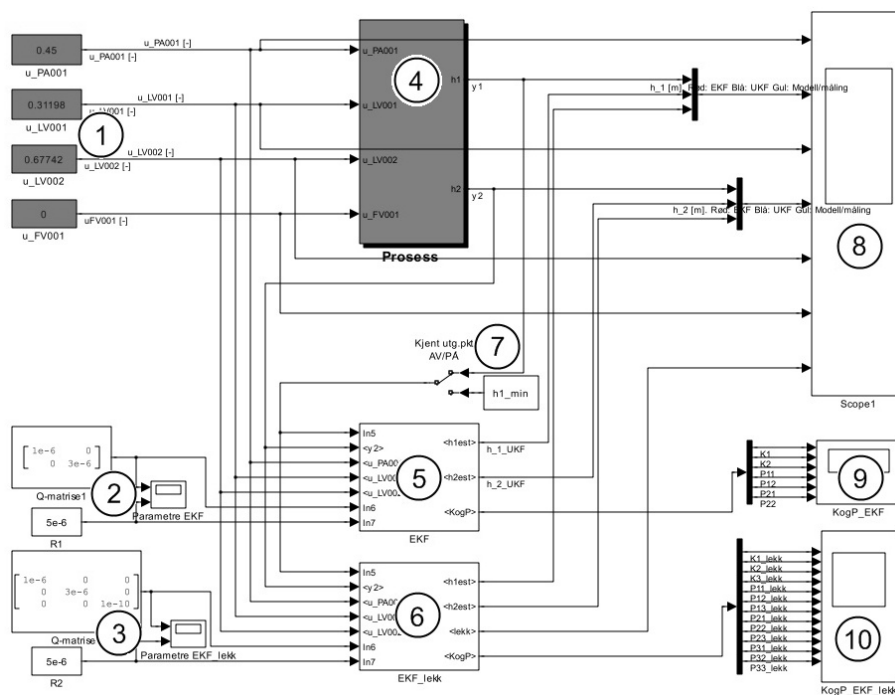


Figur 7.1: EKF Simulink

Tabell 7.1: Innganger og utganger EKF

Inn-/utgang	Forklaring
y1	Måling av nivå i tank 1. Benyttes kun ved initialisering
y2	Måling av nivå i tank 2
f1	Tabelloppslag for pumpekararakteristikk for pumpe PA001
f2	Tabelloppslag for ventilkararakteristikk for ventil LV001
f3	Tabelloppslag for ventilkararakteristikk for ventil LV002
TotankData	Henter inn konstanter fra to_tank_data.m
Q_inn	Tuning-parameter Q som brukes i ligning (3.5)
R_inn	Tuning-parameter R som brukes i ligning (3.7)
KF_tilstand_inn	Verdier fra forrige tilstand som brukes i ligning (3.3), (3.4) og (3.5)
h1est	Estimat av høyde i tank 1
h2est	Estimat av høyde i tank 2
lekk	Estimat av lekkasje gjennom ventil FV001
KogP	Kalmanfilterforsterkning og P-matrise
KF_tilstand_ut	Verdier som skal brukes ved neste tilstand i ligning (3.3), (3.4) og (3.5)

Figur 7.2 viser hvordan kjøring mot loggede prosessdata er implementert i Simulink. Dette tilsvarer implementeringen i figur 5.1 i del I.



Figur 7.2: Kalmanfilter mot loggede data

1. Innganger til prosessen; pådragssignal til pumpe *PA001* og de tre ventilene *LV001*, *LV002* og *FV001*
2. Tuning-parametre og logging av disse for EKF fra del I
3. Tuning-parametre og logging av disse for EKF fra del II
4. Prosessen
5. EKF fra del I
6. EKF fra del II
7. Bryter for å velge om initielt nivå i tank 1 skal være kjent.
8. Logging av prosessinnnganger, nivåmålinger og estimat fra begge filtrene
9. Logging av Kalmanfilterforsterkning og P-matrise for EKF fra del I
10. Logging av Kalmanfilterforsterkning og P-matrise for EKF fra del II

7.2 Sammenligning av filtre uten lekkasje

Før testene med lekkasje i systemet er det greit å verifisere at filteret fungerer godt når det ikke er lekkasje. Det vil si at det nye Kalmanfilteret klarer å estimere nivået i tank 1 og at estimert lekkasje hele tiden er tilnærmet lik null. I figur 7.3 er filtrene kjørt mot totankprosessen uten lekkasje med følgende tuning-parametre:

Tuning-parametre for EKF uten lekkasjeestimering:

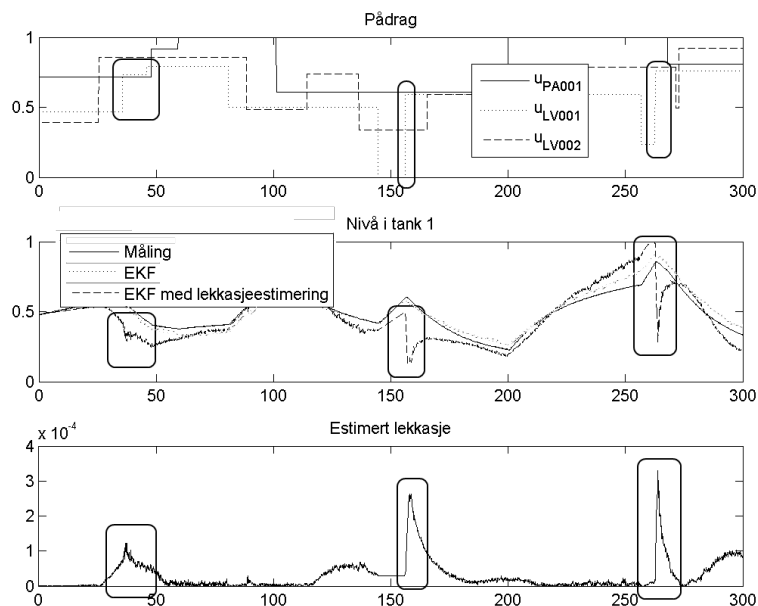
$$R = 5 \cdot 10^{-6} \quad (7.1)$$

$$Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 3 \cdot 10^{-6} \end{bmatrix} \quad (7.2)$$

Tuning-parametre for EKF med lekkasjeestimering:

$$R = 5 \cdot 10^{-6} \quad (7.3)$$

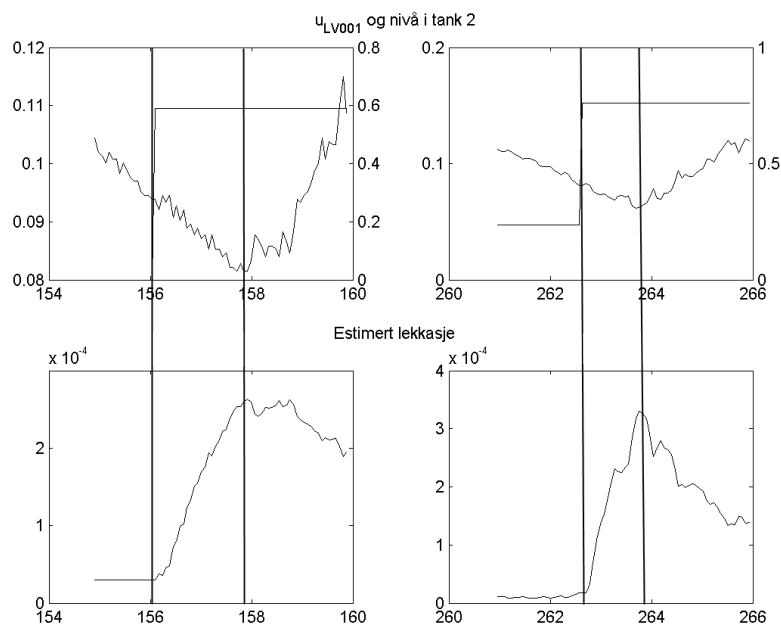
$$Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 & 0 \\ 0 & 3 \cdot 10^{-6} & 0 \\ 0 & 0 & 3 \cdot 10^{-11} \end{bmatrix} \quad (7.4)$$



Figur 7.3: Ingen lekkasje

Som det går frem av figuren følger begge filtrene målt nivå i tank 1 relativt bra når systemet ikke er utsatt for lekkasje, utenom i de tre markerte tidspunktene. I disse tidspunktene ser man en kraftig økning i estimert lekkasje. Som det vises i den midterste grafen går også estimert nivå i tank 1 noe ned i disse tidspunktene. Dette er fornuftig, da en lekkasje fra tank 1 vil føre til at nivået avtar eller stiger mindre enn for et system uten lekkasje. Ved å se tidspunktene for de estimerte lekkasjene i sammenheng med pådragskurvene ser man at det er en sammenheng mellom estimert lekkasje og økning i pådrag til ventil *LV001*.

Dette er vist i større detalj i figur 7.4.



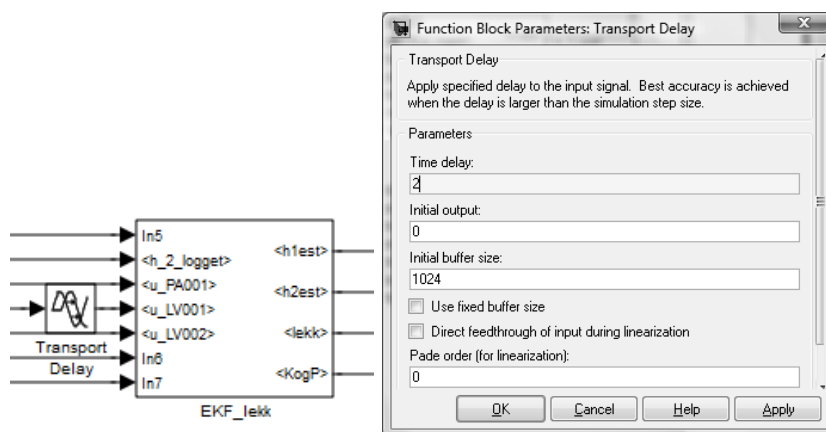
Figur 7.4: Feilestimert lekkasje

Kalmanfiltret skal ved hjelp av måling av nivå i tank 2 estimere nivå i tank 1 og en eventuell lekkasje. I de to øverste grafene i figur 7.4 ser man at det er en viss tidsforsinkelse fra man gir signal om å åpne *LV001* til nivået i tank 2 begynner å stige. Dette kan skyldes dynamikk i ventil og/eller transporttid i rørene. I denne tidsperioden hvor nivået ikke endres som det i følge den matematiske modellen burde, vil Kalmanfilteret tolke dette som en lekkasje. Dette viser i de to nederste grafene i figur 7.4.

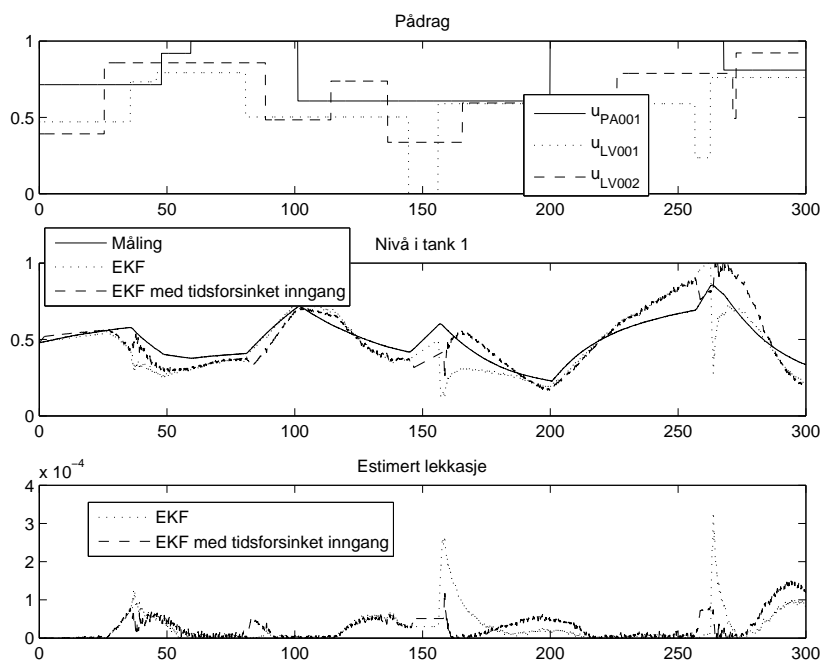
7.3 Motvikring av estimeringsfeil p.g.a. dødtid

Fra figur 7.4 ser man at tidsforsinkelsen fra pådrag på ventil *LV001* blir endret til nivået i tank 2 blir påvirket er ca 1.5 til 2 sekunder. For å redusere problemet denne forsinkelsen medfører med tanke på feil lekkasjeestimat, gjøres en

enkel endring på modellen i Simulink. Ved å sette en "Transport-Delay"-blokk mellom pådraget til LV001 og inngangen på Kalmanfilterblokken (se figur 7.5) vil filteret registrere endring i pådrag til LV001 omtrent samtidig med at pådragsendringen gir utslag på nivå i tank 2.



Figur 7.5: Forsinkelse av ventiltpådrag

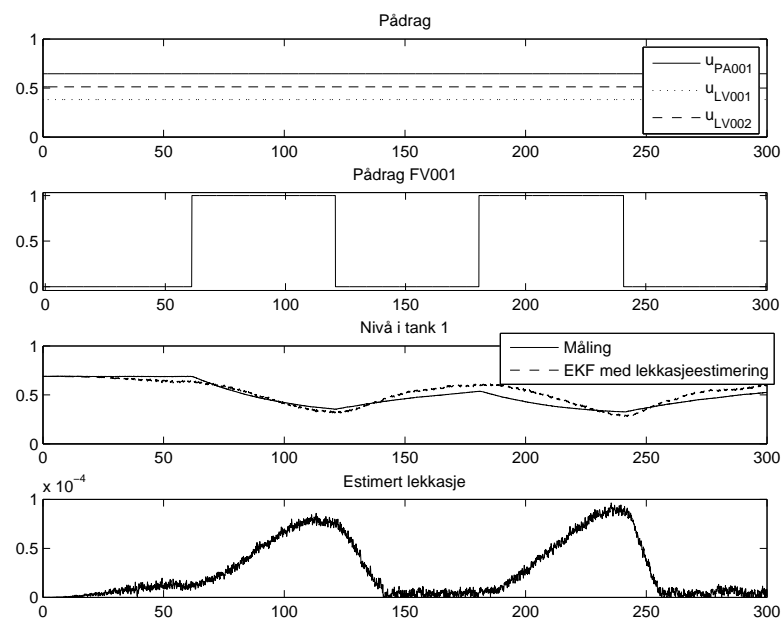


Figur 7.6: Effekt av pådragsforsinkelse

Figur 7.6 viser at den innførte forsinkelen fører til at de brå endringene i estimert nivå i tank 1 og estimert lekkasje i sammenheng med endring av ventiltpådrag til LV001 er betydelig redusert. Dette er benyttet videre i oppgaven.

7.4 Testing ved lekkasje fra tank 1

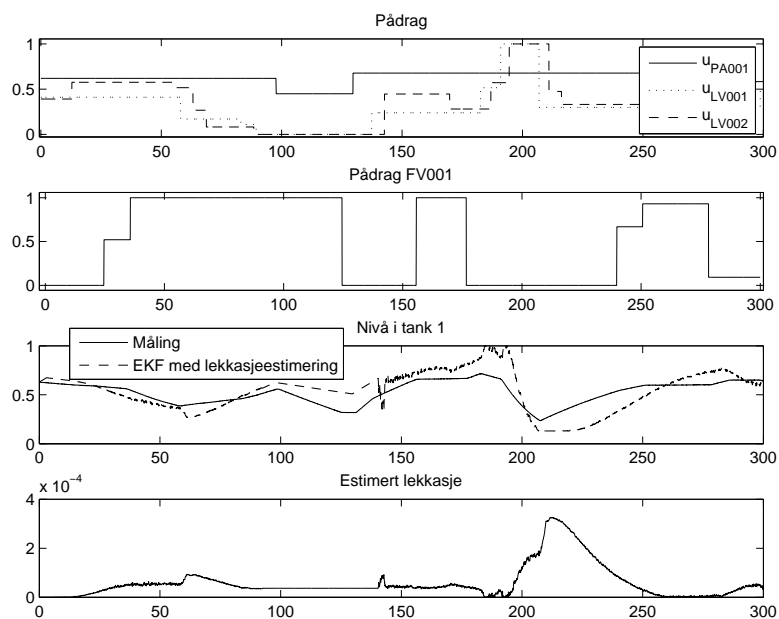
For å undersøke hvor egnet Kalmanfilteret er til å detektere lekkasje og estimere rett nivå i tank 2 under en situasjon med lekkasje, testes filteret ved en situasjon hvor alle pådrag er konstante unntatt pådraget til *FV001* som pulseres med periode på 2 minutt. Figur 7.7 viser dette forsøket med de samme tuningparametrene som før, ligning (7.3) og (7.4).



Figur 7.7: Pulset lekkasje og konstante pådrag

Som det går frem av figuren gir Kalmanfilteret et tydelig utslag for lekkasjen samtidig som det estimerte nivået i tank 1 stemmer godt med målingen.

I figur 7.8 ser man på filterets ytelse i en situasjon hvor både pådrag og lekkasje varierer.



Figur 7.8: Alle pådrag og lekkasje varierer

I figur 7.8 viser det tydelig at Kalmanfilteret ikke gir et brukbart estimat for lekkasje i systemet når pådrag varierer som sprang. Ut fra de viste forsøkene kan man konkludere med at modellen og Kalmanfilteret som her er brukt ikke er anvendelig for lekkasjeestimering når pådragene i systemet variere mye. Lekkasjeestimatene som følge av modellfeil som vist i figur 7.8 er faktisk større enn lekkasjeestimatene i forsøket med pulset lekkasje og konstante pådrag ellers, figur 7.7. Dersom pådragene endres sakte eller holdes konstante som i figur 7.7, gir Kalmanfilteret et godt estimat for lekkasjen som kan brukes for å gi en alarm. Hvis man har en situasjon hvor systemet for det meste har stabile pådrag, men det kan forekomme raske endringer i pådrag noen ganger, kan man unngå "falsk" alarm ved å programmere inn noen betingelser. Slike betingelser kan for eksempel være at alle lekkasjeestimat i en viss tidsperiode etter en brå pådragsendring ignoreres eller at det skal være estimert en viss lekkasje i en minimumstid før alarmen går.

Kapittel 8

Konklusjon

8.1 Del I

Basert på resultatene er det så godt som ingen forskjell på prestasjonen til EKF og UKF. Teorien i kap. 4.1 sier at UKF presterer bedre enn EKF i ”systemer der den antatte betingelsen om linearitet rundt arbeidspunktet ikke er oppfylt”[5]. Årsaken til at UKF ikke gir et mye bedre estimat enn EKF i denne oppgaven er, som nevnt i kap. 5.5, at prosessen ikke er mer ulineær enn at man kan anta linearitet rundt arbeidspunktet for hvert tidssteg. Så når det kommer til filterets ytelse til denne applikasjonen, er det ikke noe som taler for å velge det ene filteret foran det andre.

Siden ytelsen ikke gir noe svar på hvilket filter som er å foretrekke, kan man se på andre faktorer som kan avgjøre hva som bør velges. Her er det særlig en viktig ting som skiller de to algoritmene, nemlig at EKF krever et generelt uttrykk for transisjonsmatrisen til systemet linearisert i et arbeidspunkt, ref ligning 3.3 og kapittel 3.1.2. Akkurat dette gjorde at det tok lengre tid å få utviklet et EKF som virket enn det som ble brukt på å få til UKF. Dette til tross for at EKF var kjent stoff når oppgaven ble påbegynt. I dette tilfellet er ikke systemet spesielt komplekst, men det er allikevel ganske krevende å finne dette generelle uttrykket.

Med bakgrunn i punktene over er det mye som tyder på at UKF i de fleste sammenhenger kan være å foretrekke fremfor EKF, og da spesielt for mer komplekse systemer der lineariseringen kan bli veldig krevende, både når det gjelder forarbeid og i forhold til behov for regnekraft i selve filteret. Og siden UKF i følge teorien (kap. 4.1) også presterer bedre for svært ulineære systemer bør man nok vurdere å bruke UKF oftere enn det som er vanlig idag.

8.2 Del II

Det har vist seg at Kalmanfilteret for lekkasjeestimering kan gi en god indikasjon på lekkasje i systemet hvis pådragene til systemet ikke endres mye og

raskt. Ved rask økning i pådraget på ventil *LV001* ga filteret uriktige lekkasje-estimat. Som vist i figur 7.4 skyltes dette bl.a. en tidsforsinkelse i systemet. Ved å forsinke den ene inngangen til Kalmanfilteret fikk man redusert problemet denne tidsforsinkelen medførte noe, men det var fortsatt store feilestimerte lekkasjer i systemet. Hvis dette filterte hadde blitt tatt i bruk i en industriell applikasjon kunne man løst dette problemet ved å legge betingelser for når lekkasjealarmen skulle gå. På denne måten kunne man fått filtrert bort de fleste falske alarmene. Hadde dette vært et kritisk system som krevde rask deteksjon av lekkasje burde man finne en bedre modell som inkluderte for denne dødtiden.

Bibliografi

- [1] Finn Haugen. *Regulering av dynamiske systemer*. Tapir Akademisk Forlag, 2001.
- [2] Jarle Haugstad. Reguleringsstrategier anvendt på totank laboratorieanlegg. Masteroppgave, Universitetet i Stavanger, 2008.
- [3] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [4] Greg Welch og Gary Bishop. An introduction to the kalman filter. Technical report, Department of Computer Science University of North Carolina at Chapel Hill, 2006.
- [5] Simon J. Julier og Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. Technical report, The Robotics Research Group, Department of Engineering Science, The University of Oxford, 1997.
- [6] Eric A. Wan og Rudolph van der Merwe. The unscented kalman filter for nonlinear estimation. Technical report, Mathematics Department, Macquarie University, Sydney, 2000.
- [7] Jan Erik Qvam. Kalmanfilter anvendt på prosessanlegg. Masteroppgave, Universitetet i Stavanger, 2007.
- [8] D. Simon. Optimal state estimation. 2006.
- [9] Morten Simonsen. Hypotesetest med standard normalfordeling og t-fordeling. Technical report, Høgskulen i Sogn og Fjordane, 2006.
- [10] T.S. Schei S.Kolås, B. A.Foss. Constrained nonlinear state estimation based on the ukf approach. Technical report, Department of Engineering Cybernetics, NTNU, 2008.
- [11] Karl Skretting. Utledning av kalman-filteret. Technical report, Institutt for data- og elektroteknikk (IDE), Universitetet i Stavanger (UiS), 2008.
- [12] Rudolph van der Merwe. Sigma-point kalman filters for probabilistic inference in dynamic state-space models. Technical report, OGI School of Science & Engineering ved Oregon Health & Science University, 2004.

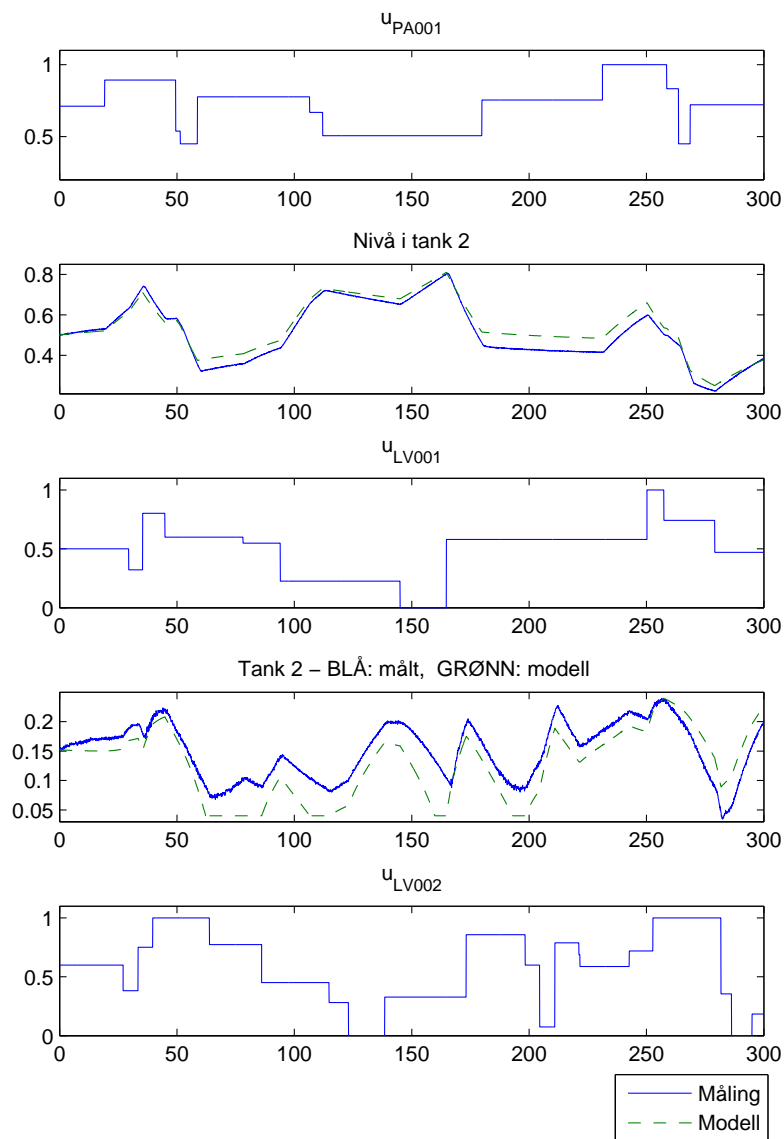
Mye informasjon om forskning og historie relatert til Kalmanfilteret kan også finnes på <http://www.cs.unc.edu/~welch/kalman/>. Denne siden er laget av Greg Welch og Gary Bishop, fra "Department of Computer Science at the University of North Carolina at Chapel Hill".

Tillegg A

UKF og EKF for
nivåestimering

A.1 Modellverifisering

Figur A.1 viser modellen verifisert ved ett annet forsøk enn i kap.2.3



Figur A.1: Modellverifisering 2

A.2 Matlab-kode for å finne transisjonsmatrisen

Matlab-kode for å finne lineær transisjonsmatrise som brukes trengs ved implementering av 3.3

```
%% transisjonsmatrise.m finner phi-matrisen som skal brukes ved EKF
syms x1 x2 u1 u2 u3 c1 c2 h_LV001 h_LV002 T A1

%% Differanseligningene
g1 = x1 + T/A1*(u3-c1*u1*sqrt(x1+h_LV001));
g2 = x2 + T/(0.004 + 0.07*x2)*(c1*u1*sqrt(x1 + h_LV001) - c2*u2*sqrt(x2 + h_LV002));

%% Partiellderiverer
dg1_dx1 = diff(g1, x1);
dg1_dx2 = diff(g1, x2);
dg2_dx1 = diff(g2, x1);
dg2_dx2 = diff(g2, x2);

%% Lager matrisen phi
phi = [dg1_dx1, dg1_dx2;...
       dg2_dx1, dg2_dx2];
```

A.3 Implementering av EKF

Koden fra EKF-blokken i Simulink.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               fcnEKF                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Utvidet Kalmanfilter for å estimere nivå i tank 1 i
totankanlegget ut fra
% kunskap om prosessen, pådragsdata og måling av nivå i tank 2

function [hlest, h2est, KogP, KF_tilstand_ut] = fcnEKF(y1,y2,f1,
f2,f3,TotankData,Q_inn,R_inn,KF_tilstand_inn)

% Innganger:
% y1 - Måling av høyde i tank 1
% y2 - Måling av høyde i tank 2
% f1 - Tabelloppslag for pumpepådrag til Pumpe PA001
% f2 - Tabelloppslag for ventilpådrag til ventil FV001
% f3 - Tabelloppslag for ventilpådrag til ventil FV002
% TotankData - Vektor som inneholder følgende data fra filen
to_tank_data.m:
% [rho,g,c_p,Kv_LV001,h_LV001,h1_max,h1_min,A1,Kv_LV002,
h_LV002,h2_max,h2_min]
% KF_tilstand_inn - Vektor med data fra forrige tidssteg

% Utganger:
% hlest - Estimat av høyde i tank 1
% h2est - Estimat av høyde i tank 2
% KogP - Verdi av K og P som skal logges
% KF_tilstand_ut - Her leges data som blir "KF_tilstand_inn" ved
neste tidssteg.

% Konstanter fra filen to_tank_data.m
rho = TotankData(1);
g = TotankData(2);
Kv_LV001 = TotankData(4);
h_LV001 = TotankData(5);
h1_max = TotankData(6);
h1_min = TotankData(7);
A1 = TotankData(8);
Kv_LV002 = TotankData(9);
h_LV002 = TotankData(10);
h2_max = TotankData(11);
h2_min = TotankData(12);
```



```

% Tuningparametre
Q = Q_inn;
R = R_inn;

% Tidssteg
T = 0.07;

% Konstanter for å forenkle utregninger lenger nede
c1 = Kv_LV001*sqrt(rho*g/100000)/3600;
c2 = Kv_LV002*sqrt(rho*g/100000)/3600;

% Data fra forrige tilstand
x1 = KF_tilstand_inn(2);           % Tilstandsestimat x1_hatt
fra forrige tidssteg
x2 = KF_tilstand_inn(3);           % Tilstandsestimat x2_hatt
fra forrige tidssteg
P = [KF_tilstand_inn(4), KF_tilstand_inn(6); ...
      KF_tilstand_inn(5), KF_tilstand_inn(7)]; % P_hatt fra
forrige tiddsteg
u1 = KF_tilstand_inn(8);           % f1 i fra forrige tidssteg
u2 = KF_tilstand_inn(9);           % f2 i fra forrige tidssteg
u3 = KF_tilstand_inn(10);          % f3 i fra forrige tidssteg

% Initielle verdier for høyden i tank 1 og tank 2.
if KF_tilstand_inn(1) <= 1
    x1 = y1;
    x2 = y2;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stegene for EKF

% Steg 1: Phi-matrisen.
% A1 = 0.01;           % Areal av vannspeilet i tank 1 ved
forrige tidssteg. (Konstant, fra to_tank_data.m)
A2 = 0.004 + 0.07*x2; % Areal av vannspeilet i tank 2 ved
forrige tidssteg.
q1 = c1 * u2 * sqrt(x1+h_LV001); % Strømning ut av tank 1
q2 = c2 * u3 * sqrt(x2+h_LV002); % Strømning ut av tank 2

Phi = [1 - T*q1/(2*A1*(x1 + h_LV001)) , 0 ; ...
        T*q1/(2*A1*(x1 + h_LV001))      , 1 - T*(0.07/(A2^2))*(q1 -

```

```

q2) - q2/(2*A2*(x2 + h_LV002))]);

% Steg 2: Beregner x_strek
x1 = x1 + T/A1*(u1-q1);           % x1_strek
x2 = x2 + T/A2*(q1-q2);         % x2_strek

% Steg 3: Beregner P_strek
P = Phi*P*Phi' + Q;

% Steg 4: D-matrisen
D = [0,1];           % måler x2

% Steg 5: Beregner K
K = P*D' / (D*P*D' + R);

% Steg 6: y_strek
y_strek = x2; %Måler høyden i tankt 2

% Steg 7: Beregner x_hatt
x1 = x1 + K(1)*(y2-y_strek);
x2 = x2 + K(2)*(y2-y_strek);

% Steg 8: Beregner P_hatt
P = (eye(2) - K*D)*P;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Begrenser estimatene til rimelige verdier:
if (x1 < h1_min); x1 = h1_min; end;
if (x1 > h1_max); x1 = h1_max; end;
if (x2 < h2_min); x2 = h2_min; end;
if (x2 > h2_max); x2 = h2_max; end;

% Legger ut verdier som skal brukes ved neste steg:
KF_tilstand_ut = zeros(1,10);           % Oppretter en tom
vektor
KF_tilstand_ut(1) = KF_tilstand_inn(1)+1; % Antall steg
totalt, inkludert dette
KF_tilstand_ut(2) = x1;                 % x1_hatt
KF_tilstand_ut(3) = x2;                 % x2_hatt
KF_tilstand_ut(4:7) = P(:)';           % P_hatt
KF_tilstand_ut(8) = f1;                 % f1 - Tabelloppslag

```

```
for pumpepådrag til pumpe PA001
KF_tilstand_ut(9) = f2;           % f2 - Tabelloppslag✓
for ventilpådrag til ventil FV001
KF_tilstand_ut(10) = f3;        % f3 - Tabelloppslag✓
for ventilpådrag til ventil FV002

% Setter de nye estimatene på utgangen fra funksjonsblokken:
h1est = x1;
h2est = x2;

% Verdi av K og P som skal logges:
KogP = [K(1,1) K(2,1) P(1,1) P(1,2) P(2,1) P(2,2)];
```

A.4 Implementering av UKF

Koden fra UKF-blokken i Simulink.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               fcnUKF                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Unscented Kalmanfilter for å estimere nivå i tank 1 i
totankanlegget ut fra
% kunskap om prosessen, pådragsdata og måling av nivå i tank 2

function [SigmaStat, hlest, h2est, KogP, KF_tilstand_ut] =
fcnUKF(y1,y2,f1,f2,f3,TotankData,Q_inn,R_inn,KF_tilstand_inn)

% Innganger:
% y1 - Måling av høyde i tank 1
% y2 - Måling av høyde i tank 2
% f1 - Tabelloppslag for pumpepådrag til pumpe PA001
% f2 - Tabelloppslag for ventilpådrag til ventil FV001
% f3 - Tabelloppslag for ventilpådrag til ventil FV002
% TotankData - Vektor som inneholder følgende data fra filen
to_tank_data.m:
% [rho,g,c_p,Kv_LV001,h_LV001,h1_max,h1_min,A1,Kv_LV002,
h_LV002,h2_max,h2_min]
% KF_tilstand_inn - Vektor med data fra forrige tidssteg

% Utganger:
% SigmaStat - Sigmapunktene som skal logges
% hlest - Estimat av høyde i tank 1
% h2est - Estimat av høyde i tank 2
% KogP - Verdi av K og P som skal logges
% KF_tilstand_ut - Her leges data som blir "KF_tilstand_inn" ved
neste tidssteg.

% Konstanter fra filen to_tank_data.m
rho = TotankData(1);
g = TotankData(2);
Kv_LV001 = TotankData(4);
h_LV001 = TotankData(5);
h1_max = TotankData(6);
h1_min = TotankData(7);
A1 = TotankData(8);
Kv_LV002 = TotankData(9);
h_LV002 = TotankData(10);
h2_max = TotankData(11);
h2_min = TotankData(12);
```

```
% Tuningparametre
Q = Q_inn;
R = R_inn;

n = 2;
kappa = 0;
alfa = 1;
%beta = 2; % ikke i bruk

lambda = alfa^2*(n+kappa)-n;
%Wc0 = lambda/(n+lambda)+(1-alfa^2-beta); % ikke i bruk
%Wx0 = lambda/(n+lambda); % ikke i bruk
Wxi = 1/(2*(n+lambda));
Wci = Wxi;
Gamma = sqrt(lambda + n);

% Tidssteg
T = 0.07;

% Konstanter for å forenkle utregninger lenger nede
c1 = Kv_LV001*sqrt(rho*g/100000)/3600;
c2 = Kv_LV002*sqrt(rho*g/100000)/3600;

% Data fra forrige tilstand
x1 = KF_tilstand_inn(2); % Tilstandsestimat x1_hatt↵
fra forrige tidssteg
x2 = KF_tilstand_inn(3); % Tilstandsestimat x2_hatt↵
fra forrige tidssteg
P = [KF_tilstand_inn(4), KF_tilstand_inn(6); ...
     KF_tilstand_inn(5), KF_tilstand_inn(7)]; % P_hatt fra↵
forrige tiddsteg
u1 = KF_tilstand_inn(8); % f1 i fra forrige tidssteg
u2 = KF_tilstand_inn(9); % f2 i fra forrige tidssteg
u3 = KF_tilstand_inn(10); % f3 i fra forrige tidssteg

% Initielle verdier for høyden i tank 1 og tank 2.
if KF_tilstand_inn(1) <= 1
    x1 = y1;
    x2 = y2;
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stegene for UKF

% Steg 1: Finne sigmapunkter
A = Gamma*chol(P)';
X = [x1;x2]; % X_hatt forrige tigssteg
Xa = X(:,ones(1,numel(X))); % Matrise med X_hatt forrige
tidssteg i alle kolonnene
Sigma = [Xa+A Xa-A]; % Matrise med sigmapunkter i
kolonnene

% Steg 2: Propagerer sigmapunkt
i=size(Sigma,2); %antall sigmapunkt
Sigma_x=zeros(2,i); %matrise som skal inneholde propagerte
sigmapunkt
while i>=1
    A2 = 0.004 + 0.07*Sigma(2,i); % Areal av
vannspeilet i tank 2 i sigmapunktet
    q1 = c1 * u2 * sqrt(Sigma(1,i)+h_LV001); % Strømning ut av
tank 1
    q2 = c2 * u3 * sqrt(Sigma(2,i)+h_LV002); % Strømning ut av
tank 2
    Sigma_x(1,i) = Sigma(1,i) + T/A1*(u1-q1); % Propagerte
sigmapunkt
    Sigma_x(2,i) = Sigma(2,i) + T/A2*(q1-q2);
    i = i-1;
end

% Steg 3: Beregne x_strek
X = sum (Sigma_x,2)*Wxi;

% Steg 4: Beregner P_strek
i = size(Sigma_x,2);
a1 = [0,0;0,0];
while i>=1
    a1 = a1 + Wci*([Sigma_x(1,i); Sigma_x(2,i)]-X)*([Sigma_x(1,
i); Sigma_x(2,i)]-X)';
    i = i-1;
end
P = a1 + Q;

% Steg 5: Beregner sigma-punkt utganger

```

```
D = [0,1];
Sigma_y = D*Sigma_x;

% Steg 6: Beregner y_hatt
Y = sum (Sigma_y)*Wxi;

% Steg 7: Beregner utgangskovarians Pyy
i = size(Sigma_y,2);
a2 = 0;
while i>=1
    a2 = a2 + Wci*(Sigma_y(i)-Y)^2;
    i = i-1;
end
Pyy = a2 + R;

% Steg 8: Beregner krysskovarians mellom måling og tilstand Pxy
i = size(Sigma_y,2);
a3 = [0;0];
while i>=1
    a3 = a3 + Wci*([Sigma_x(1,i); Sigma_x(2,i)]-X)*(Sigma_y(i)-Y);
    i = i-1;
end
Pxy = a3;

% Steg 9: Beregner K
K = Pxy * Pyy^(-1);

% Steg 10: Beregner x_hatt
X = X + K*(y2-Y);

% Steg 12: Bergner P_hatt
P = P - K*Pyy*K';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Begrenser estimatene til rimelige verdier:
x1 = X(1,1);
x2 = X(2,1);
if (x1 < h1_min); x1 = h1_min; end;
if (x1 > h1_max); x1 = h1_max; end;
if (x2 < h2_min); x2 = h2_min; end;
```



```
if (x2 > h2_max); x2 = h2_max; end;

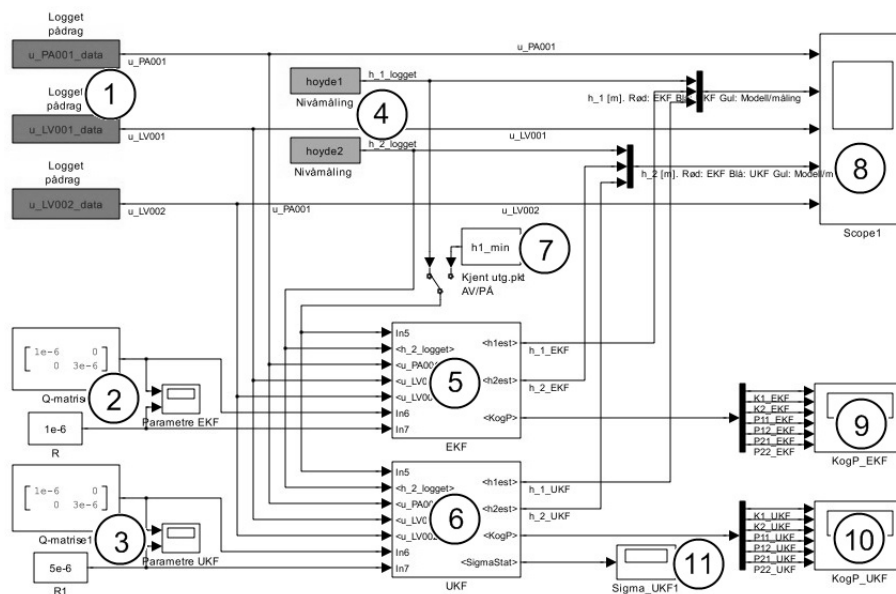
% Legger ut verdier som skal brukes ved neste steg:
KF_tilstand_ut = zeros(1,10);           % Oppretter en tom vektor
KF_tilstand_ut(1) = KF_tilstand_inn(1)+1; % Antall steg totalt, inkludert dette
KF_tilstand_ut(2) = x1;                 % x1_hatt
KF_tilstand_ut(3) = x2;                 % x2_hatt
KF_tilstand_ut(4:7) = P(:)';           % P_hatt
KF_tilstand_ut(8) = f1;                 % f1 - Tabelloppslag
for pumpep drag til pumpe PA001
KF_tilstand_ut(9) = f2;                 % f2 - Tabelloppslag
for ventilp drag til ventil FV001
KF_tilstand_ut(10) = f3;                % f3 - Tabelloppslag
for ventilp drag til ventil FV002

% Setter de nye estimatene p  utgangen fra funksjonsblokken:
hlest = x1;
h2est = x2;

% Verdi av K og P som skal logges:
KogP = [K(1,1) K(2,1) P(1,1) P(1,2) P(2,1) P(2,2)];

% Sigmapunkter som skal logges:
SigmaStat = [Sigma, Sigma_x];
```

A.5 Offline kjøring



Figur A.2: Kalmanfilter mot loggede data

Forskjellen fra implementeringen i figur 5.1 er at her er (1) loggede pådrag og (4) loggede målinger, ikke online målinger fra prosessen.

A.6 Modelldata

Utkrift av filen `to_tank_data.m` som er hentet fra [2] og modifisert noe. Denne filen benyttes i både del I og del II.

```
%% to_tank_data.m inneholder konstanter og data som benyttes av
i modellen av totankprosessen
clear all
close all

%% data om vann og tyngdekraft
rho = 1000;           % tetthet vann           [kg/m^3]
g = 9.81;            % tyngdens akselerasjon  [m/s^2]
c_p = 4200;         % varmekapasitet vann     [j/kg*K]

%% Tank 1
Kv_LV001 = 11.25*0.65; % ventilkonstant LV001 [m^3/h/sqrt
(bar)]
h_LV001 = 0.05;       % høyde til LV001 [m]
h1_max = 1;          % maks høyde tank 1 [m]
h1_min = 0.13;       % min høyde tank 1 [m]
A1 = 0.01;          % areal tank 1 [m^2]

%% Tank 2
Kv_LV002 = 11.25*0.60; %695; % ventilkonstant LV002
[m3/h/sqrt(bar)]
h_LV002 = 0.25;       % høyde fra bunn av tank 2 til LV002
h2_max = 0.35;       % maks høyde tank 2 [m]
h2_min = 0.04;       % min høyde tank 2 [m]

%% Pumpekarakteristikk
u_PA001_vektor = [0.00 0.45 0.46 0.47 0.48 0.49 0.50
0.55...
0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95
1.00];
q_PA001_vektor = [0.00 0.00 0.00 1.30 1.90 2.84 3.93
6.79...
8.81 10.86 12.74 14.31 16.1 17.7 19.1 20.6 21.88];

q_PA001_vektor = q_PA001_vektor/60000*0.93; % liter/min -> m3/s

f3 = polyfit(u_PA001_vektor(2:17),q_PA001_vektor(2:17),10);

if (0)
figure
plot(u_PA001_vektor, q_PA001_vektor, '*-')
hold on
plot([0.45:0.01:1],polyval(f3,[0.45:0.01:1]),'-r')
```

```
title('Pumpekarakteristikk')
xlabel('Pådrag u_{PA001}(t) til pumpe PA001')
ylabel('Volumstrøm q_{PA001}(t) gjennom PA001 [m^3/s]')
end

%% Ventilkarakteristikk
u_LV001_vektor = 0:0.05:1;
f_LV001_vektor = [0 0.0192 0.0384 0.0823 0.1152 0.1782 0.2248
0.2742...
0.3236 0.3647 0.4031 0.4388 0.5210 0.5896 0.6883 0.7705
0.8528...
0.8912 0.9269 0.9625 1.00];

u_LV002_vektor = u_LV001_vektor;
f_LV002_vektor = f_LV001_vektor;

f1 = polyfit(u_LV001_vektor,f_LV001_vektor,10);
f1_inv = polyfit(f_LV001_vektor,u_LV001_vektor,10);

f2 = polyfit(u_LV002_vektor,f_LV002_vektor,10);
f2_inv = polyfit(f_LV002_vektor,u_LV002_vektor,10);

if (0)
figure
plot(u_LV001_vektor,f_LV001_vektor,'*-')
hold on
plot([0:0.01:1],polyval(f1,[0:0.01:1]),'-r')
set(gcf,'Position',[829 496 560 420])
title('Ventilkarakteristikk for LV001 og LV002')
xlabel('Ventilpådrag u_{LV001}(t)')
ylabel('f(u_{LV001}(t))')
end

%% Last inn pådrag og målinger

% Del 1 - EKF vs UKF
%load tankData
%load tankData2
%load tankData3
%load tankData4
%load tankData5
%load tankData_kodd_mer_T1
```

```
%load tankData_kodd_mer_T2
%load tankData_kodd_mindre_T1

% Del 2 - lekkasjeestimering
load tankData_uten_lekk_300.mat
%load tankData_med_lekk_konstant_300.mat
%load tankData_med_lekk_puls_300.mat
%load tankData_med_lekk_300.mat

%% Initiale tilstander (for offline testing)
h1_initial = hoyde1.signals.values(1);
h2_initial = hoyde2.signals.values(1);
```


Tillegg B

EKF for estimering av lekkasje i ventil

B.1 Transisjonsmatrise

Ved å benytte symbolsk derivasjon i Maple (og Matlab), finnes $\Phi(k)$ til:

$$\begin{aligned} \Phi(k) &= \begin{bmatrix} \frac{\partial g_1(\cdot)}{\partial x_1} & \frac{\partial g_1(\cdot)}{\partial x_2} & \frac{\partial g_1(\cdot)}{\partial x_3} \\ \frac{\partial g_2(\cdot)}{\partial x_1} & \frac{\partial g_2(\cdot)}{\partial x_2} & \frac{\partial g_2(\cdot)}{\partial x_3} \\ \frac{\partial g_3(\cdot)}{\partial x_1} & \frac{\partial g_3(\cdot)}{\partial x_2} & \frac{\partial g_3(\cdot)}{\partial x_3} \end{bmatrix} \\ &= \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix} \end{aligned}$$

hvor

$$\begin{aligned} \phi_{11} &= 1 - \frac{T \cdot q_{LV001}}{2 \cdot A_1 \cdot (x_1 + h_{LV001})} \\ \phi_{12} &= 0 \\ \phi_{13} &= -\frac{T}{A_1} \\ \phi_{21} &= \frac{T \cdot q_{LV001}}{2 \cdot A_2 \cdot (x_1 + h_{LV001})} \\ \phi_{22} &= 1 - \frac{0.07 \cdot T}{A_2^2} \cdot (q_{LV001} - q_{LV002}) - \frac{q_{LV002}}{2 \cdot A_2 \cdot (x_2 + h_{LV002})} \\ \phi_{23} &= 0 \\ \phi_{31} &= 0 \\ \phi_{32} &= 0 \\ \phi_{33} &= 1 \end{aligned}$$

B.2 Observerbarhetsmatrise

Observerbarhetsmatrisen finnes på formen:

$$Q_o = \begin{bmatrix} D \\ D \cdot \Phi \\ D \cdot \Phi^2 \end{bmatrix} \quad (\text{B.1})$$

hvor transisjonsmatrisen fra ligning 6.14 er gitt ved:

$$\Phi(k) = \begin{bmatrix} \phi_{11} & 0 & \phi_{13} \\ \phi_{21} & \phi_{22} & 0 \\ 0 & 0 & \phi_{33} \end{bmatrix} \quad (\text{B.2})$$

og målematrisen fra ligning ?? er gitt som:

$$D = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \quad (\text{B.3})$$

Beregner så elementene i observerbarhetsmatrisen:

$$\begin{aligned} D \cdot \Phi &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \phi_{11} & 0 & \phi_{13} \\ \phi_{21} & \phi_{22} & 0 \\ 0 & 0 & \phi_{33} \end{bmatrix} \\ &= \begin{bmatrix} \phi_{21} & \phi_{22} & 0 \end{bmatrix} \end{aligned} \quad (\text{B.4})$$

$$\begin{aligned} D \cdot \Phi^2 &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \phi_{11} & 0 & \phi_{13} \\ \phi_{21} & \phi_{22} & 0 \\ 0 & 0 & \phi_{33} \end{bmatrix} \cdot \begin{bmatrix} \phi_{11} & 0 & \phi_{13} \\ \phi_{21} & \phi_{22} & 0 \\ 0 & 0 & \phi_{33} \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \phi_{11}^2 & 0 & \phi_{11} \cdot \phi_{13} + \phi_{13} \cdot \phi_{33} \\ \phi_{21} \cdot \phi_{11} + \phi_{22} \cdot \phi_{21} & \phi_{22}^2 & \phi_{21} \cdot \phi_{13} \\ 0 & 0 & \phi_{33}^2 \end{bmatrix} \\ &= \begin{bmatrix} \phi_{21} \cdot \phi_{11} + \phi_{22} \cdot \phi_{21} & \phi_{22}^2 & \phi_{21} \cdot \phi_{13} \end{bmatrix} \end{aligned} \quad (\text{B.5})$$

og får observerbarhetsmatrisen:

$$Q_o = \begin{bmatrix} 0 & 1 & 0 \\ \phi_{21} & \phi_{22} & 0 \\ \phi_{21} \cdot \phi_{11} + \phi_{22} \cdot \phi_{21} & \phi_{22}^2 & \phi_{21} \cdot \phi_{13} \end{bmatrix} \quad (\text{B.6})$$

B.3 Implementering av EKF

Koden fra EKF-blokken i Simulink.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               fcnEKF2                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Utvidet Kalmanfilter for å estimere nivå i tank 1 og lekkasje
% fra tank 1
% i totankanlegget ut fra kunnskap om prosessen, pådragsdata
% og måling av nivå i tank 2

function [hlest, h2est, lekk, KogP, KF_tilstand_ut] = fcnEKF2(
(y1,y2,f1,f2,f3,TotankData,Q_inn,R_inn,KF_tilstand_inn)

% Innganger:
% y1 - Måling av høyde i tank 1
% y2 - Måling av høyde i tank 2
% f1 - Tabelloppslag for pumpepådrag til pumpe PA001
% f2 - Tabelloppslag for ventilpådrag til ventil FV001
% f3 - Tabelloppslag for ventilpådrag til ventil FV002
% TotankData - Vektor som inneholder følgende data fra filen
to_tank_data.m:
% [rho,g,c_p,Kv_LV001,h_LV001,h1_max,h1_min,A1,Kv_LV002,
h_LV002,h2_max,h2_min]
% KF_tilstand_inn - Vektor med data fra forrige tidssteg

% Utganger:
% hlest - Estimat av høyde i tank 1
% h2est - Estimat av høyde i tank 2
% lekk - Estimert lekkasje fra tank 1
% KogP - Verdi av K og P som skal logges
% KF_tilstand_ut - Her leges data som blir "KF_tilstand_inn" ved
neste tidssteg.

% Konstanter fra filen to_tank_data.m
rho = TotankData(1);
g = TotankData(2);
Kv_LV001 = TotankData(4);
h_LV001 = TotankData(5);
h1_max = TotankData(6);
h1_min = TotankData(7);
A1 = TotankData(8);
Kv_LV002 = TotankData(9);
h_LV002 = TotankData(10);
h2_max = TotankData(11);
```

```
h2_min = TotankData(12);

% Tuningparametre
Q = Q_inn;
R = R_inn;

% Tidssteg
T = 0.07;

% Konstanter for å forenkle utregninger lenger nede
c1 = Kv_LV001*sqrt(rho*g/100000)/3600;
c2 = Kv_LV002*sqrt(rho*g/100000)/3600;

% Data fra forrige tilstand
x1 = KF_tilstand_inn(2);           % Tilstandsestimat x1_hatt
fra forrige tidssteg
x2 = KF_tilstand_inn(3);           % Tilstandsestimat x2_hatt
fra forrige tidssteg
x3 = KF_tilstand_inn(4);
P = [KF_tilstand_inn(5), KF_tilstand_inn(8), KF_tilstand_inn
(11); ...
      KF_tilstand_inn(6), KF_tilstand_inn(9), KF_tilstand_inn
(12); ...
      KF_tilstand_inn(7), KF_tilstand_inn(10), KF_tilstand_inn
(13)]; % P_hatt fra forrige tiddsteg
u1 = KF_tilstand_inn(14);           % f1 i fra forrige tidssteg
u2 = KF_tilstand_inn(15);           % f2 i fra forrige tidssteg
u3 = KF_tilstand_inn(16);           % f3 i fra forrige tidssteg

% Initielle verdier for høyden i tank 1 og tank 2 og lekkasje
fra tank 1.
if KF_tilstand_inn(1) <= 1
    x1 = y1;
    x2 = y2;
    x3 = 0; % Antar null lekkasje initielt
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stegene for EKF

% Steg 1: Phi-matrisen.
% Her legges arbeidspunktverdiene inn i det generelle uttrykket
```



```
% Begrenser estimatene til rimelige verdier:
if (x1 < h1_min); x1 = h1_min; end;
if (x1 > h1_max); x1 = h1_max; end;
if (x2 < h2_min); x2 = h2_min; end;
if (x2 > h2_max); x2 = h2_max; end;
if (x3 < 0); x3 = 0; end; % Antar at vann kun lekker ut fra
tanken, ikke inn.

% Legger ut verdier som skal brukes ved neste steg:
KF_tilstand_ut = zeros(1,16);
KF_tilstand_ut(1) = KF_tilstand_inn(1)+1; % antall steg
totalt, inkludert dette
KF_tilstand_ut(2) = x1; % x1_hatt
KF_tilstand_ut(3) = x2; % x2_hatt
KF_tilstand_ut(4) = x3; % x3_hatt
KF_tilstand_ut(5:13) = P(:)'; % P_hatt
KF_tilstand_ut(14) = f1; % f1 - Tabelloppslag
for pumpepådrag til pumpe PA001
KF_tilstand_ut(15) = f2; % f2 - Tabelloppslag
for ventilpådrag til ventil FV001
KF_tilstand_ut(16) = f3; % f3 - Tabelloppslag
for ventilpådrag til ventil FV002

% Setter de nye estimatene på utgangen fra funksjonsblokken:
hlest = x1;
h2est = x2;
lekk = x3;

% Verdi av K og P som skal logges:
KogP = [K(1,1) K(2,1) K(3,1) P(1,1) P(1,2) P(1,3) P(2,1) P(2,2)
P(2,3) P(3,1) P(3,2) P(3,3)];
```