



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Automatisering og signalbehandling	Vårsemesteret, 2017 Åpen / Konfidensiell
Forfatter: Duy Viet Nguyen (signatur forfatter)
Fagansvarlig: Karl Skretting og Tormod Drengstig Veileder(e): Karl Skretting og Tormod Drengstig	
Tittel på masteroppgaven: Kalmanfilter anvendt på totankprosess for temperatur estimering Engelsk tittel: Kalmanfilter applied on two tank process for temperature estimation	
Studiepoeng: 30	
Emneord: Modellering og verifisering Utvidet Kalmanfilter, EKF Tilstandsestimering Simulering Tilsatsestimering	Sidetall: 63 +vedlegg/annet: 27 Stavanger, 15.6.2017 dato/år



Universitetet
i Stavanger

Kalmanfilter anvendt på totankprosess for temperatur estimering

Duy Viet Nguyen

15.juni 2017

Forord

Denne hovedoppgaven er en avsluttende oppgave på min masterutdanning i automatisering og signalbehandling ved institutt for data- og elektroteknikk på Universitet i Stavanger, våren 2017.

Jeg vil takke mine veiledere Karl Skretting og Tormod Drengstig for god veiledning og verdifulle tilbakemeldinger.

Stavanger, Juni 2017

Duy Viet Nguyen

Sammendrag

Denne hovedoppgaven beskriver hvordan et Kalmanfilter kan brukes for estimering av temperatur på et prosessanlegg ved Universitet i Stavanger (UiS).

Først lages en matematisk modell av den fysiske prosessen ved å benytte masse- og energibalanse. Deretter testes modellen i Simulink og verifiseres mot virkelig prosess. Videre vises implementering av et utvidet Kalmanfilter (EKF) for å estimere temperaturen på vannet ved innløpet av tanken. Estimaten baseres på temperatur- og nivåmåling i tanken og kjennskap til prosessen. For dette tilfellet vurderes hvordan EKF presterer under forskjellige driftssituasjoner, blant annet når prosessen blir utsatt av ekstern tilsats av vann. En utvidet modell av EKF blir laget for å estimere denne eksterne tilsatsen. Resultatene fra forsøkene viser at EKF gir et godt estimat av temperaturen ved innløpet når varmeelementet ikke er med å påvirke prosessen. På grunn av dynamikk i varmeelementet, har det ført til merkbart avvik når varmeelementet slår seg på eller av. Kalmanfilter for tilsatsestimering ga en god indikasjon på når prosessen ble utsatt for ekstern tilsats, og samtidig et tilfredsstillende estimat for temperaturen ved innløpet.

Oppgaven tar videre for seg å implementere et EKF for å estimere temperaturen i tanken. Estimaten baseres nå kun på nivåmåling i tanken og temperaturen på vannet inn antas å være kjent. Systemet i denne delen er ikke observerbart. Ved å anta at EKF har tilgang til måling av temperaturen i tanken en gang i blant, gir dette et system som er observerbart ved dette tidssteget. Hensikten er å studere hvordan EKF opererer i intervallene hvor det ikke blir tatt temperaturmåling. Resultatene viser at EKF ikke klarer å beregne Kalmanfilterforsterkningen i disse intervallene. Dette Kalmanfilteret er med andre ord kun basert på det apriori tilstandsestimatet.

Innhold

1	Innledning	1
1.1	Bakgrunn for oppgaven	1
1.2	Oppgavebeskrivelse.....	3
1.3	Kalmanfilter.....	3
1.3.1	Teori	3
1.4	Rapportens innhold og struktur	4
2	Prosessanlegg.....	5
2.1	Tank.....	5
2.2	Skalering av temperatur transmittere	6
2.2.1	Skalering av temperatur transmitter TT001	6
2.2.2	Skalering av temperatur transmitter TT003	7
2.3	Varmeelement HE001	9
2.4	Ventiler.....	10
2.4.1	Verifisering av ventilkarakteristikk LV003.....	10
3	Modellering og modellverifisering av prosessmodell	13
3.1	Matematisk modell av prosessanlegg.....	14
3.1.1	Differensialligning for nivå i tanken	14
3.1.2	Differensialligning for temperatur i tanken.....	16
3.2	Implementering av prosessmodell i Simulink.....	17
3.3	Verifisering av modell.....	20
4	Implementering av utvidet Kalmanfilter.....	22
4.1	Teori	23
4.1.1	EKF ligninger	23
4.2	Linearisering og diskretisering av modellen	25
4.3	Implementering av utvidet Kalmanfilter i Simulink	28
5	Testing og tuning	30

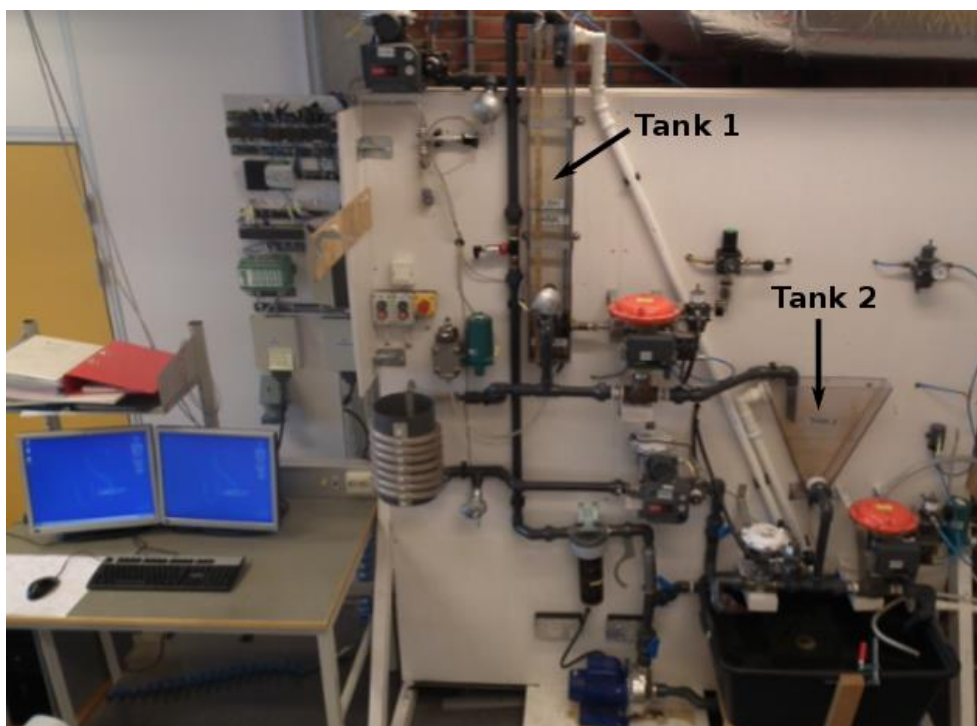
5.1	Drift av EKF.....	31
5.1.1	P-matrise.....	36
5.1.2	Kalmanfilter forsterkning.....	37
5.1.3	Test av EKF med ukjent initialtilstand.....	38
5.1.4	Test av EKF med prosessforstyrrelse.....	39
6	Kalmanfilter for estimering av tilsats.....	42
6.1	Matematisk modell av prosessen.....	42
6.1.1	Differensialligning for nivå i tanken.....	42
6.1.2	Differensialligning for temperatur i tanken.....	42
6.1.3	Differensialligning for temperatur fra vannbatteri.....	43
6.1.4	Differensialligning for inntrengning av vann.....	43
6.2	Linearisering og diskretisering av modellen.....	43
6.3	Observerbart system.....	44
6.4	Implementering i Simulink.....	45
6.5	Test av EKF med tilsatsestimering.....	46
7	EKF for estimering av temperatur i tanken.....	52
7.1	Implementering av EKF.....	53
7.2	Test av EKF.....	54
7.2.1	P-matrise.....	56
7.2.2	Kalmanfilterforsterkning.....	57
7.3	Test av EKF med ekstra tilsats.....	59
8	Konklusjon.....	61
	Referanser.....	62
	Vedlegg.....	64
A	64
A.2	Test av EKF uten blanding.....	66
A.3	P-matrise.....	67
A.4	Kalmanfilter forsterkning.....	68
B	69
B.1	Transisjonsmatrise (kapittel 6).....	69
B.2	Transisjonsmatrise (kapittel 7.1).....	70
B.3	Transisjonsmatrise (kapittel 7.3).....	71
C	– MATLAB kode.....	72

C.1 Transisjonsmatrise	72
C.2 Observerbarhet.....	75
C.3 Estimering av temperatur inn (kapittel 4.3)	76
C.4 Estimering av ekstern tilsats (kapittel 6.4)	79
C.5 Estimering av temperatur i tanken (kapittel 7.1)	82
C.6 Estimering av ekstern tilsats (kapittel 7.3)	86
C.7 Totankanlegg data.....	90

1 Innledning

1.1 Bakgrunn for oppgaven

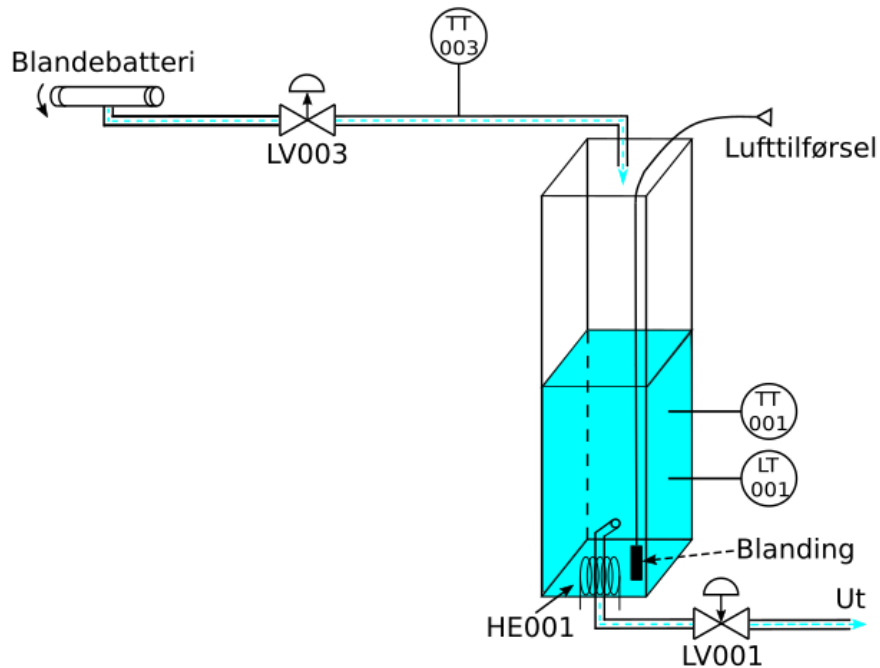
Institutt for data- og elektroteknikk ved Universitetet i Stavanger har laget et prosessanlegg. Dette brukes i undervisninger der studentene kan teste ut teoriene om å styre en dynamisk prosess i virkeligheten, se figur 1.1. Totankanlegget er utstyrt med mye instrumentering, slik at flere problemstillinger kan testes ut og dermed har flere hovedoppgaver blitt utført på dette anlegget.



Figur 1.1: Totankanlegg på laboratoriet, E-459

Tidligere ble det skrevet hovedoppgaver [1, 2], hvor et utvidet Kalmanfilter (EKF) har blitt brukt for å estimere nivå i tank 1 basert på nivåmåling i tank 2.

I denne oppgaven skal et utvidet Kalmanfilter anvendes for estimering av temperatur. Totankanlegget består av to tanker, men her blir bare tank 1 og tilhørende instrumentering rundt den benyttes, se figur 1.2.



Figur 1.2: Skisse av prosessen

Prosessen består av en tank med ett inn- og utløp. Vannet i tanken hentes fra blandebatteriet via ventil LV003. Temperaturen på vannet som strømmer inn i tanken kan justeres med å vri på blandebatteriet og måles av temperaturtransmitter TT003. Nivå og temperatur i tanken måles av nivåtransmitter LV001 og temperaturtransmitter TT001 henholdsvis. På bunnen av tanken er det et varmeelement HE001 og et lufrør som bidrar til en bedre blanding i tanken. Vannstrømning ut av tanken skjer via ventil LV001.

1.2 Oppgavebeskrivelse

Denne oppgaven omfatter følgende:

1. Verifisering av komponentene: TT001, TT003 og LV003.
2. Setter opp matematisk modell av den fysiske prosessen, modellerer prosessen i Simulink og verifisere mot virkelig prosess.
3. Konstruere et EKF for å estimere vanntemperatur ved innløpet av tanken, hvor systemet er observerbart.
4. Utvide punkt 2 og 3 til å estimere ekstern tilsats av vann, hvor den nye tilstanden estimeres samtidig med de opprinnelige tilstandene. Tilsats av vann blir manuelt tømt oppi tanken.
5. Konstruere et EKF for estimering av temperatur i tanken. Systemet som er gitt, er ikke observerbart. Ved å anta at det er mulig å få tilgang til måling av temperatur i tanken en gang i blant, gir dette et observerbart system ved dette tidssteget. Hensikten er å se hvordan EKF opererer i intervallene hvor måling ikke er tilgjengelig.

1.3 Kalmanfilter

Det klassiske Kalmanfilteret er en rekursiv algoritme som ble publisert av Rudolf E. Kalman i 1960 [3]. Kalmanfilteret er essensielt et sett av matematiske ligninger som estimerer tilstanden i en prosess på en effektiv måte, og var opprinnelig utviklet for lineære systemer.

Etter hvert har andre metoder blitt utviklet for ulineære systemer, blant annet utvidet Kalmanfilter og Unscented Kalmanfilter (UKF) [4]. UKF er ikke vurdert i denne oppgaven, men kan være et alternativ dersom EKF ikke gir et tilfredsstillende resultat. Mer om teorien bak EKF blir beskrevet i kapittel 4.1.

1.3.1 Teori

Algoritmen består av et prediksjonsteg og et korreksjonssteg. Prediksjonsteget bruker forrige tidssteg tilstandsestimat, $\hat{x}(k-1)$ til å predikere neste tilstand, $\bar{x}(k)$. I korreksjonssteget blir estimatet korrigert ved

å sammenligne tilstandsestimatet fra prediksjonsteget med ny måling [5], se ligningen under.

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

Estimatet $\hat{x}(k)$ betegnes som aposteriori tilstandsestimat, siden estimatet er beregnet etter nåværende måling er tatt. $e(k)$ er differansen mellom måling $y(k)$ og forventet måling $\bar{y}(k)$. $K(k)$ er Kalmanfilter forsterkning. Forsterkningen forteller hvor raskt Kalmanfilteret skal korrigere for avviket. Hvor stor Kalmanfilter forsterkningen vil være, er avhengig av størrelsen på tuningsparametre Q og R. Q og R er autokovariansmatriser for prosesstøy og målestøy.

1.4 Rapportens innhold og struktur

Kapittel 2. Prosessanlegg: Beskrivelse av den delen av prosessen som benyttes, og verifisering av komponenter.

Kapittel 3. Modellering og modellverifisering av prosessmodell: Massebalanse, energibalanse, implementering og verifisering av modell.

Kapittel 4. Implementering av EKF: Kort teori om EKF, finne uttrykk for transisjonsmatrise og implementasjon av EKF.

Kapittel 5. Testing og tuning: Valg av tuningsparameter og test av EKF ved ulike driftssituasjoner.

Kapittel 6. Kalmanfilter for estimering av tilsats av vann: Utvide modell for estimering av ekstern tilsats, implementere og test av Kalmanfilteret.

Kapittel 7. EKF for estimering av temperatur i tanken: Implementere og test av EKF.

Kapittel 8. Konklusjon

2 Prosessanlegg

For å få en bra estimering av tilstanden er det viktig å ha en god modell av den fysiske prosessen. Hensikten med kapittelet er å finne ut hvordan komponentene som inngår i prosessen best kan modelleres.

2.1 Tank

Dataene i dette delkapittelet er hentet fra [6].

Tanken som er installert på totankanlegget er laget av polykarbonat og er rektangulært formet med en høyde på 1.0 meter. Tanken har et konstant innvendig areal lik 0.0096 m^2 . I teorien vil tanken ha et maksimalt volum:

$$V = A * h = 0.0096 \text{ m}^2 \cdot 1 \text{ m} = 0.0096 \text{ m}^3 = 9.6 \text{ liter}$$

men det virkelige volumet er noe mindre. Dette skyldes på et varmeelement som er montert i bunnen av tanken, et luftrør som brukes til å få en bedre blanding og diverse skru koplinger som tar plass. I praksis kan tanken ikke bli helt fullt. Dette er på grunn av at det hvite utløpsrøret på toppen av tanken sluker vannet direkte ned i oppsamlingskaret når nivået nærmer seg full. Tanken kan heller ikke bli helt tomt, siden det svarte utløpsrøret er montert et lite stykke over bunnen av tanken, se figur 2.1.



Figur 2.1: Bilde av tanken med tilhørende utstyr

2.2 Skalering av temperatur transmittere

Ved skalering av TT001 og TT003 er det gjort flere målinger på den virkelige prosessen, og det er antatt en lineær sammenheng mellom avlest temperatur og avlest spenning. Dette ble gjort basert på måling av spenning på I/O-kortet, PCI-6024 fra National Instruments (NI), og måling av temperatur med en referansemåler, FLUKE 179. De lineære funksjonene som blir funnet i delkapittel 2.2.1 og 2.2.2 blir implementert i Simulink for å konvertere spenning til temperatur, se vedlegg A.1.

2.2.1 Skalering av temperatur transmitter TT001

Ved skalering av TT001 er det benyttet varmeelement HE001 til å varme opp vannet i tanken, hvor referansemåleren er plassert direkte oppi tanken.

Tabell 2.1 viser målingene som ble funnet.

Tabell 2.1: Målt resultat på temperatur i tanken

Målt spenning [V] på I/O-kortet	Målt temperatur [°C]
6.77	24.8
6.92	26.00
7.25	28.00
7.58	30.00
7.93	32.00
8.24	34.00
8.56	36.00
8.96	38.00
9.22	40.00
9.58	42.00
9.86	44.00
10.00	44.5

Som det går frem i tabell 2.1, begynner temperaturen i litt underkant av 25 grader celsius. Dette skyldes på at vannbatteriet ikke klarer å levere kaldere vann til prosessen.

For å finne ett fullskala sammenheng mellom spenning og temperatur for TT001, er det nødvendig å finne skaleringskoeffisientene.

Dette er gjort ved å bruke minste kvadraters metode av målt data fra tabell 2.1, og uttrykker den lineære sammenheng på formen:

$$y = ax + b \quad (2.1)$$

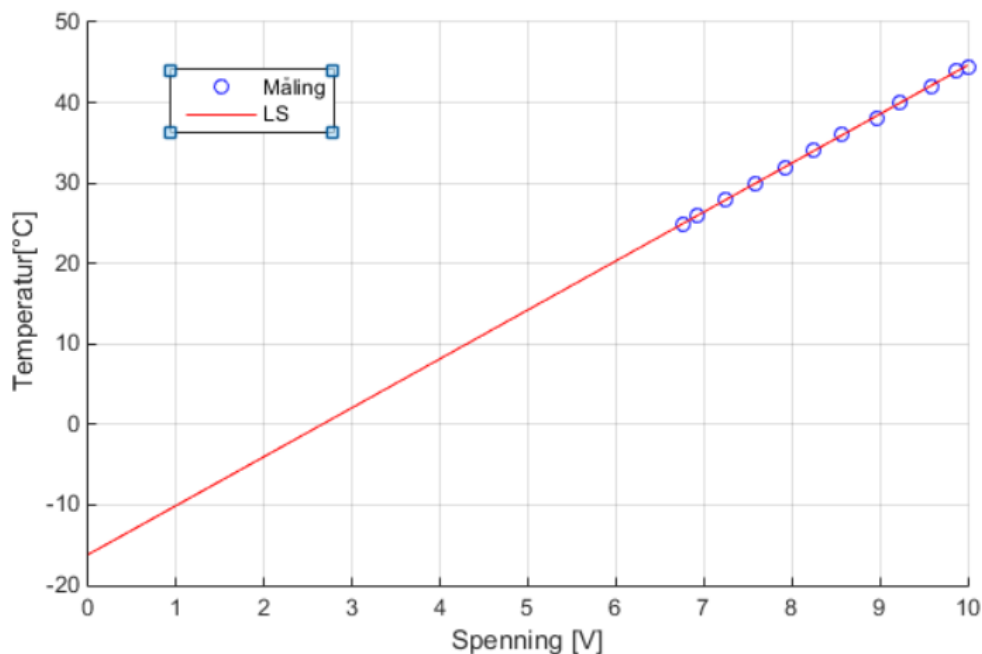
hvor

$$a = 6.08 \quad (2.2)$$

$$b = -16.15 \quad (2.3)$$

der konstanten a har måleenhet, grader celsius per volt [$^{\circ}\text{C}/\text{V}$].

Figur 2.2 viser den lineære funksjonen og målte verdier.



Figur 2.2: Fullskala sammenheng mellom spenning og temperatur i tank

2.2.2 Skalering av temperatur transmitter TT003

I denne oppgaven er det antatt at temperaturen ved innløpet av tanken, har den samme temperaturen fra vannbatteriet.

Skalering av TT003 er gjort ved å plassere referansemåleren ved innløpet av tanken og temperatur justeringen skjer via vannbatteriet.

Tabell 2.2 viser måleresultatet som ble funnet.

Tabell 2.2: Målt resultat på temperatur på innløpet

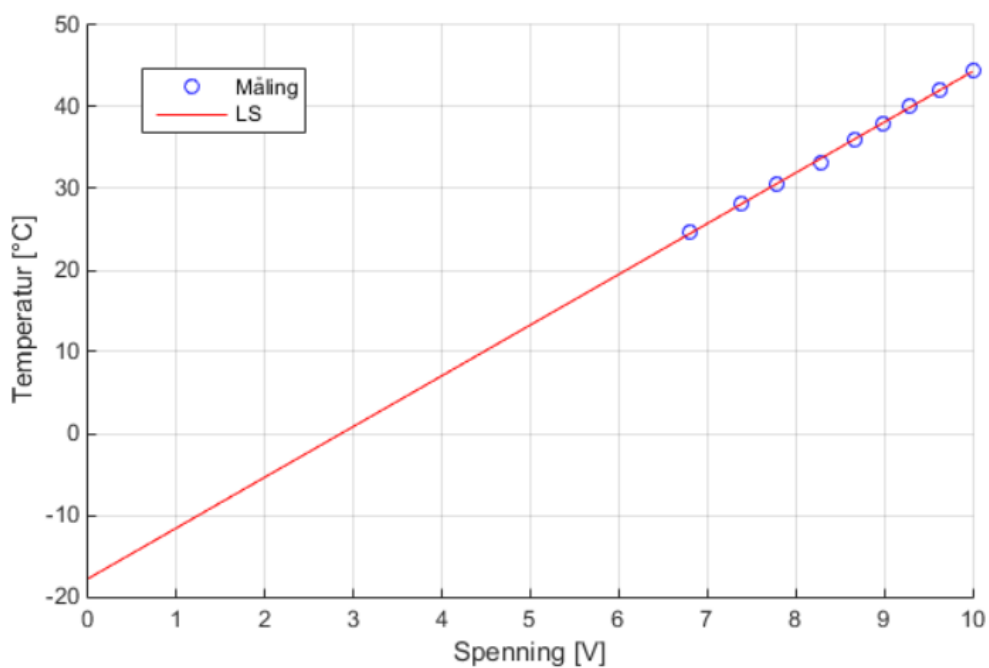
Spenning [V] på I/O-kortet	Målt temperatur [°C]
6.8	24.6
7.38	28.2
7.78	30.6
8.28	33.2
8.66	36.0
8.98	38.0
9.28	40.0
9.62	42.1
10.00	44.4

For å finne skaleringskoeffisientene til TT003, ble det gjort på tilsvarende måte som for TT001 i forrige delkapittel. Det gir:

$$a = 6.21 \quad (2.4)$$

$$b = -17.76 \quad (2.5)$$

Figur 2.3 viser den lineære sammenhengen og målte verdier.

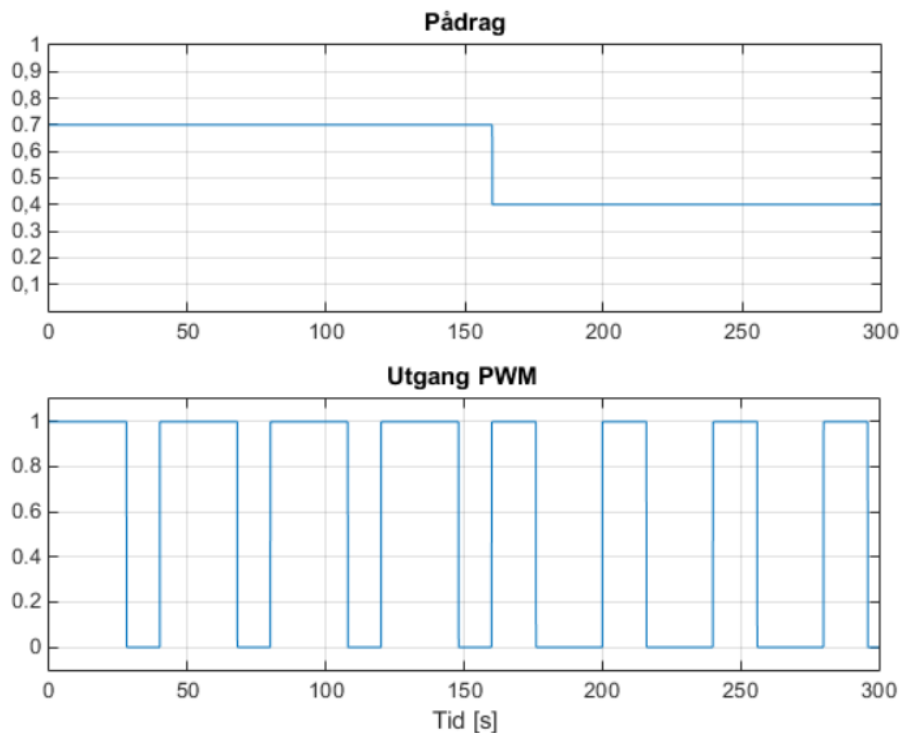


Figur 2.3: Fullskala sammenheng mellom spenning og temperatur på innløpet

2.3 Varmeelement HE001

I [7] er effekten til varmeelementet estimert til være på 1725W. Denne verdien vil bli brukt i denne oppgaven som et utgangspunkt.

Varmeelementet har et styre signal mellom 0 og 1 i Simulink. Signalet blir deretter skalert til 0-10 Volt ut fra I/O-kortet, PCI-6703, som igjen går inn til effektregulator, EFM 9161. Effektregulatoren omsetter et 0-10V signal til et puls/pauseforhold av tilkoblet effekt, hvor en periode har en varighet på 40 sekunder. Et pådrag på 0.7 fra Simulink, vil varmeelementet være innkoplet i 70% av periodetiden, dvs. elementet tilfører 1725W i 28 sekunder og slår seg av i de påfølgende 12 sekundene. Figur 2.4 viser et eksempel på responsen til varmeelementet.



Figur 2.4: Figuren øverst viser pådrag fra Simulink og nederst viser firkant puls signal som beskriver av/på tilstanden til varmeelementet

2.4 Ventiler

Dette delkapittel tar for seg å verifisere ventil LV003 for bruk i Simulink, som et tabelloppslag av ventilkarakteristikken. I [8] er det lagt mye arbeid på å verifisere ventilkarakteristikken til LV001, dermed er tabelloppslaget av ventilkarakteristikken hentet derifra.

Disse reguleringsventilene er av type seteventiler. Ventilene kommer i to typer; air to open (ATO) og air to close (ATC). ATO ventiler er normalt lukket, som betyr at det trengs luftrykk for å åpne ventilene, mens ATC ventiler står normalt åpen og trenger trykkluft for å lukke. Forskjellen er at en ATC har lufttilførselen inn på oversiden av membranen, mens ATO har på undersiden av membranen. Figur 2.5 viser bildene av LV001 og LV003.



Figur 2.5: Venstre, LV001 og høyre, LV003

Disse ventilene blir styrt av et signal mellom 0 og 1 i Simulink og blir skalert til 0-10 volt ut fra I/O-kortet, PCI-6703.

2.4.1 Verifisering av ventilkarakteristikk LV003

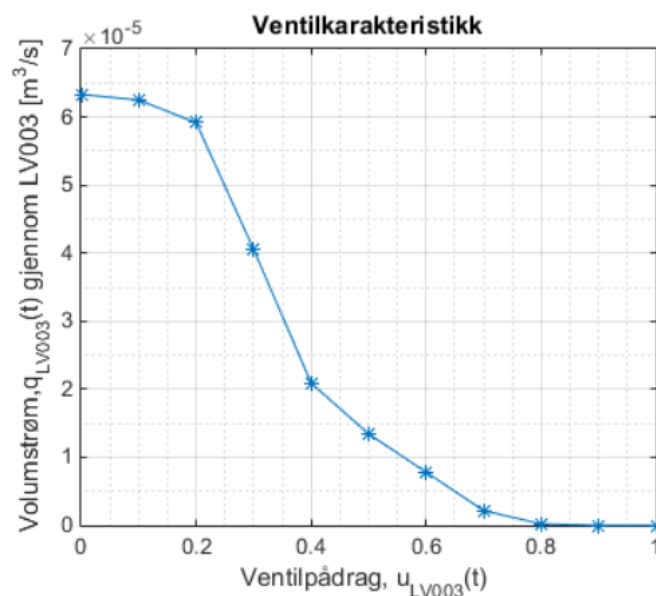
Kartlegging av ventilkarakteristikk for LV003 er basert på noen antagelser.

Slik som prosessanlegget er i dag, er det ikke anledning til å finne ut hvilke trykkfall som ligger over ventilen, altså trykket fra vannbatteriet. Siden trykkfallet er ukjent, kan ikke ventilkonstanten beregnes.

For å finne en form for ventilkarakteristikken for LV003, ble valgt å la vannbatteriet være helt åpen, og antar at trykket som ligger over ventilen er konstant, selv om vanntrykket i bygget ved Universitet i Stavanger kan variere over tid. Deretter ble det tatt tiden på å fylle opp en bestemt vannmengde i tanken med forskjellige ventilpådrag. For hver 0.1 pådrag ble det beregnet en strømningsverdi, og verdier mellom disse pådragene ble satt opp ved hjelp av lineær interpolering. Tabell 2.3 og figur 2.6 viser sammenhengen mellom ventilpådrag og volumstrøm gjennom LV003.

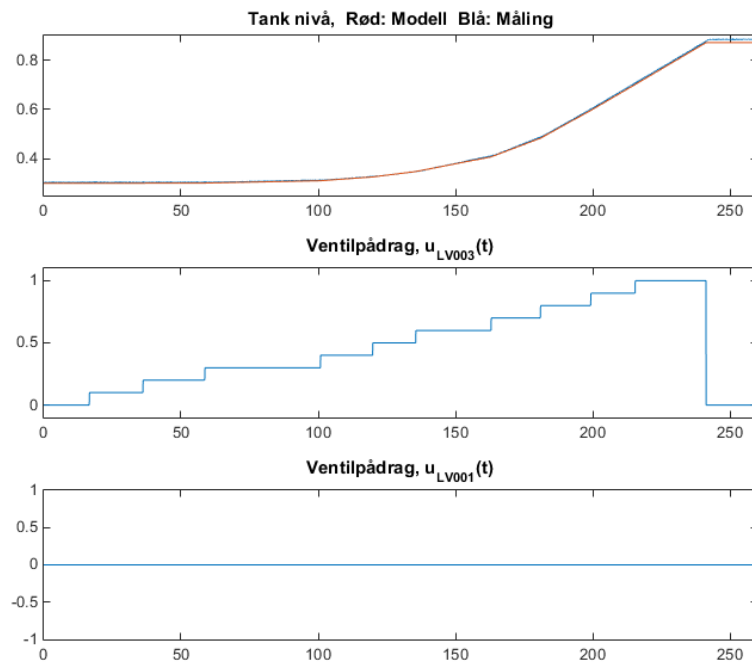
Tabell 2.3: Ventilpådrag og tilhørende volumstrøm gjennom LV003

Ventilpådrag	Volumstrøm [l/min]
1	0.00
0.9	0.00
0.8	0.012
0.7	0.129
0.6	0.469
0.5	0.806
0.4	1.254
0.3	2.432
0.2	3.582
0.1	3.749
0	3.799



Figur 2.6: Ventilkararakteristikk for LV003

Det er her gjort en test for å verifisere om ventilkarakteristikken funnet ovenfor stemmer bra med virkelig væskestrøm som strømmer gjennom LV003. Dette ble gjennomført ved å stenge utløpsventil LV001, og åpne innløpsventil LV003 for vann stegvis. Se de to nederste grafene på figur 2.7. Øverste grafen viser to kurver av nivået i tanken. Den blå kurven presenterer virkelig nivå, mens den røde er en tilnæringsmodell av nivået ut i fra ventilkarakteristikken.

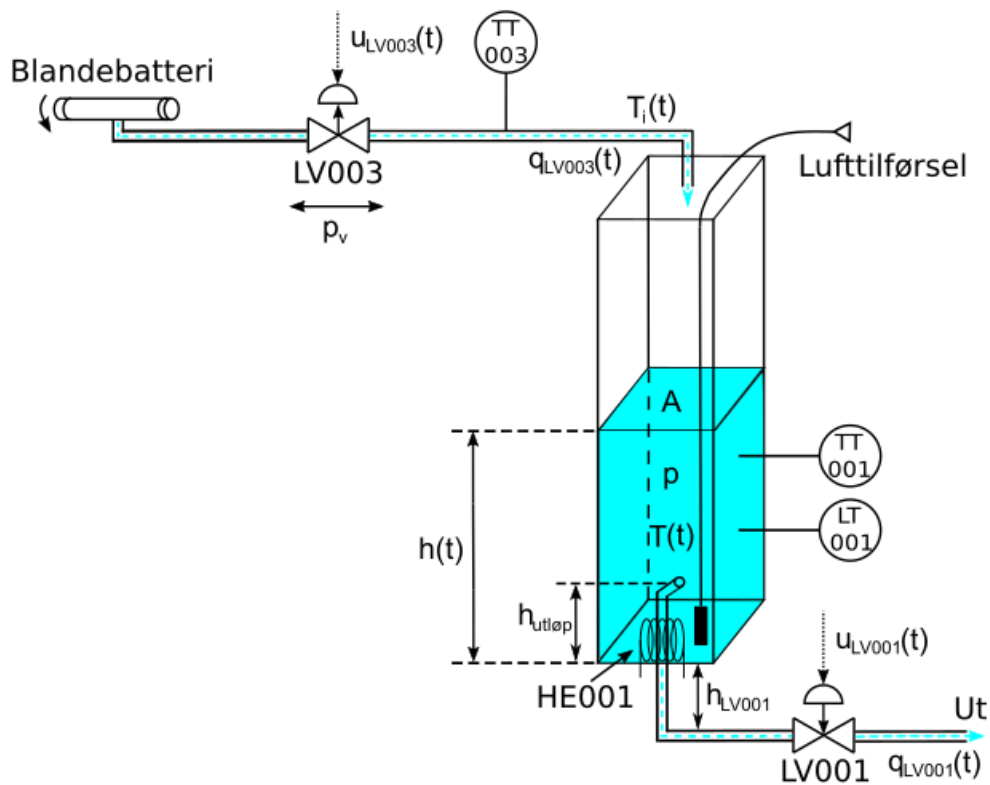


Figur 2.7: Test av LV003

Testen viser at kalkulert væskestrøm og virkelig væskestrøm gjennom ventil LV003 er rimelig nøyaktig.

3 Modellering og modellverifisering av prosessmodell

Kapittelet tar for seg å sette opp en matematisk modell av den fysiske prosessen, implementere i Simulink og verifisere modellen mot den virkelige prosessen. Figur 3.1 viser en skjematisk skisse med variabler som beskriver prosessmodellen. Navn på variabler er hentet fra [6].



Figur 3.1 Skjematisk skisse av prosessen

Med utgangspunkt i figur 3.1, er det laget en tabell som en oppsummering av notasjoner for denne prosessmodellen, se tabell 3.1.

Tabell 3.1: Forklaring av notasjoner for prosessmodellen

Variabel	Beskrivelse	Enhet
ρ	Massetetthet til vann	kg/m ³
g	Tyngdeakselerasjon	m/s ²
A	Arealet til tank	m ²
ρ_v	Trykkfall over ventil LV003, fra blandebatteri	bar
c_p	Spesifikk varmekapasitet	J/(kg*°C)
$K_{v,LV001}$	Ventilkonstant til ventil LV001	$\frac{m^3}{time * \sqrt{bar}}$
h_{LV001}	Høyde fra ventil LV001 opp til bunn i tank	m
$h_{utløp}$	Høyde fra bunn i tank opp til tankutløp	m
$h(t)$	Tilstand av vannivået i tanken	m
$q_{LV001}(t)$	Volumstrømning gjennom ventil LV001	m ³ /s
$q_{LV003}(t)$	Volumstrømning gjennom ventil LV003	m ³ /s
$\omega_{inn}(t)$	Massestrøm av vann inn i tanken	kg/s
$\omega_{ut}(t)$	Massestrøm av vann ut av tanken	kg/s
$u_{LV001}(t)$	Pådrag til ventil LV001	-
$u_{LV003}(t)$	Pådrag til ventil LV003	-
$f_1(u_{LV001}(t))$	Tabelloppslag av ventilkarakteristikk for LV001	-
$f_3(u_{LV003}(t))$	Tabelloppslag av ventilkarakteristikk for LV003	-
$T_i(t)$	Vanntemperatur ved innløpet av tanken	°C
$T(t)$	Vanntemperatur i tanken	°C
$P(t)$	Tilført/Generert effekt til varmeelement i tanken	W

3.1 Matematisk modell av prosessanlegg

Dette delkapittelet er basert på [9, 6]

3.1.1 Differensialligning for nivå i tanken

Massebalansen kan uttrykkes ved [9]:

$$\frac{dm(t)}{dt} = \sum \omega_{inn}(t) - \sum \omega_{ut}(t) + \sum \omega_{generert}(t) \quad (3.1)$$

Når massebalansen settes opp er det antatt at vannet ikke blir fordampet og ikke noe andre kilder som generer masse til prosessen. Siste leddet i ligning (3.1) blir derfor neglisjert. Uttrykket blir nå:

$$\frac{dm(t)}{dt} = \rho \cdot q_{LV003}(t) - \rho \cdot q_{LV001}(t) \quad (3.2)$$

Massen i tanken er gitt ved:

$$m(t) = \rho \cdot A \cdot h(t) \quad (3.3)$$

Videre er det antatt at arealet til tanken og massetetthet til vannet er konstant, slik at A og ρ settes utenfor brøken. Setter (3.3) inn i (3.2), og får da:

$$\rho \cdot A \cdot \frac{dh(t)}{dt} = \rho \cdot q_{LV003}(t) - \rho \cdot q_{LV001}(t) \quad (3.4)$$

Vannetettheten er med i alle ledd, og kan derfor kortes vekk. Løser (3.4) m.h.t til $\frac{dh(t)}{dt}$:

$$\frac{dh(t)}{dt} = \frac{1}{A} \cdot (q_{LV003}(t) - q_{LV001}(t)) \quad (3.5)$$

hvor $q_{LV001}(t)$ kan uttrykkes som følgende [6]:

$$q_{LV001}(t) = \frac{K_{v,LV001}}{3600} \cdot \sqrt{\frac{\rho g (h(t) + h_{LV001})}{100000}} \cdot f_1(u_{LV001}(t)) \quad (3.6)$$

og får

$$\frac{dh(t)}{dt} = \frac{1}{A} (q_{LV003}(t) - \frac{K_{v,LV001}}{3600} \sqrt{\frac{\rho g (h(t) + h_{LV001})}{100000}} \cdot f_1(u_{LV001}(t))) \quad (3.7)$$

For å gjøre det enklere for senere beregninger, definerer:

$$c = \frac{K_{v,LV001}}{3600} \cdot \sqrt{\frac{\rho \cdot g}{100000}} \quad (3.8)$$

og differensialligningen for høyde i tanken blir:

$$\frac{dh(t)}{dt} = \frac{1}{A} (q_{LV003}(t) - c \cdot \sqrt{h(t) + h_{LV001}} \cdot f_1(u_{LV001}(t))) \quad (3.9)$$

3.1.2 Differensialligning for temperatur i tanken

Energibalansen kan uttrykkes ved [9]:

$$\frac{dE(t)}{dt} = \sum Q_{inn}(t) - \sum Q_{ut}(t) + \sum Q_{generert}(t) \quad (3.10)$$

I denne delen er det antatt ideell blanding i tanken, null varmetap til omgivelsen og konstant varmespesifikk. Får da:

$$\frac{dE(t)}{dt} = c_p \omega_{inn}(t) \cdot T_i(t) - c_p \omega_{ut}(t) \cdot T(t) + P(t) \quad (3.11)$$

Energien, $E(t)$ kan uttrykkes ved:

$$\frac{dE(t)}{dt} = m(t) \cdot c_p \cdot T(t) \quad (3.12)$$

Setter nå (3.12) inn i (3.11) og har da:

$$c_p \cdot \frac{d m(t) \cdot T(t)}{dt} = c_p \omega_{inn}(t) \cdot T_i(t) - c_p \omega_{ut}(t) \cdot T(t) + P(t) \quad (3.13)$$

Fjerner c_p fra venstre siden ved å dividere på begge sider av (3.13), får da:

$$\frac{dm(t) \cdot T(t)}{dt} = \omega_{inn}(t) \cdot T_i(t) - \omega_{ut}(t) \cdot T(t) + \frac{P(t)}{c_p} \quad (3.14)$$

Venstre siden består nå av den deriverte av to tidsvarierende variabler. Fra matematikken kan dette løses ved hjelp av produktregelen, som gir følgende uttrykk:

$$\frac{dm(t)}{dt} \cdot T(t) + \frac{dT(t)}{dt} \cdot m(t) = \omega_{inn}(t) T_i(t) - \omega_{ut}(t) T(t) + \frac{P(t)}{c_p} \quad (3.15)$$

hvor den deriverte av $m(t)$ er massebalansen i ligning (3.2). Flytter den over på høyre siden og erstatter med massebalanse ligning (3.2), gir da:

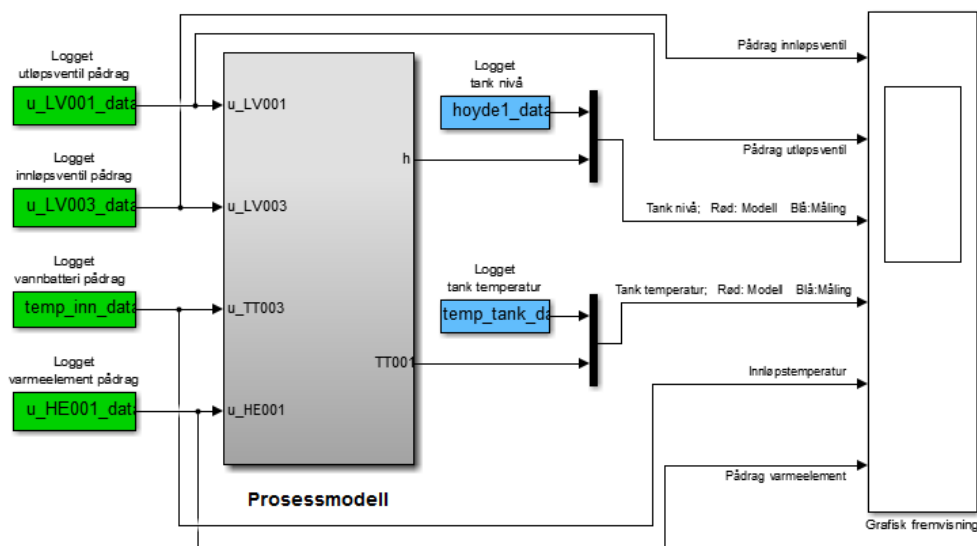
$$\begin{aligned} \frac{dT(t)}{dt} \cdot m(t) &= \omega_{inn}(t) \cdot T_i(t) - \omega_{ut}(t) \cdot T(t) + \frac{P(t)}{c_p} - \omega_{inn}(t) \\ &\quad \cdot T(t) + \omega_{ut}(t) \cdot T(t) \end{aligned} \quad (3.16)$$

Løser (3.16) v.h.t den deriverte av $T(t)$, trekker sammen og erstatter $m(t)$ med (3.3). Dermed blir differensialligning for temperaturen i tanken følgende:

$$\begin{aligned} \frac{dT(t)}{dt} &= \frac{1}{m(t)} \cdot \left(\omega_{\text{inn}}(t) \cdot (T_i(t) - T(t)) + \frac{P(t)}{c_p} \right) \\ &= \frac{1}{\rho \cdot A \cdot h(t)} \cdot \left(\rho \cdot q_{LV003}(t) \cdot (T_i(t) - T(t)) + \frac{P(t)}{c_p} \right) \quad (3.17) \\ &= \frac{1}{A \cdot h(t)} \cdot \left(q_{LV003}(t) \cdot (T_i(t) - T(t)) + \frac{P(t)}{c_p \rho} \right) \end{aligned}$$

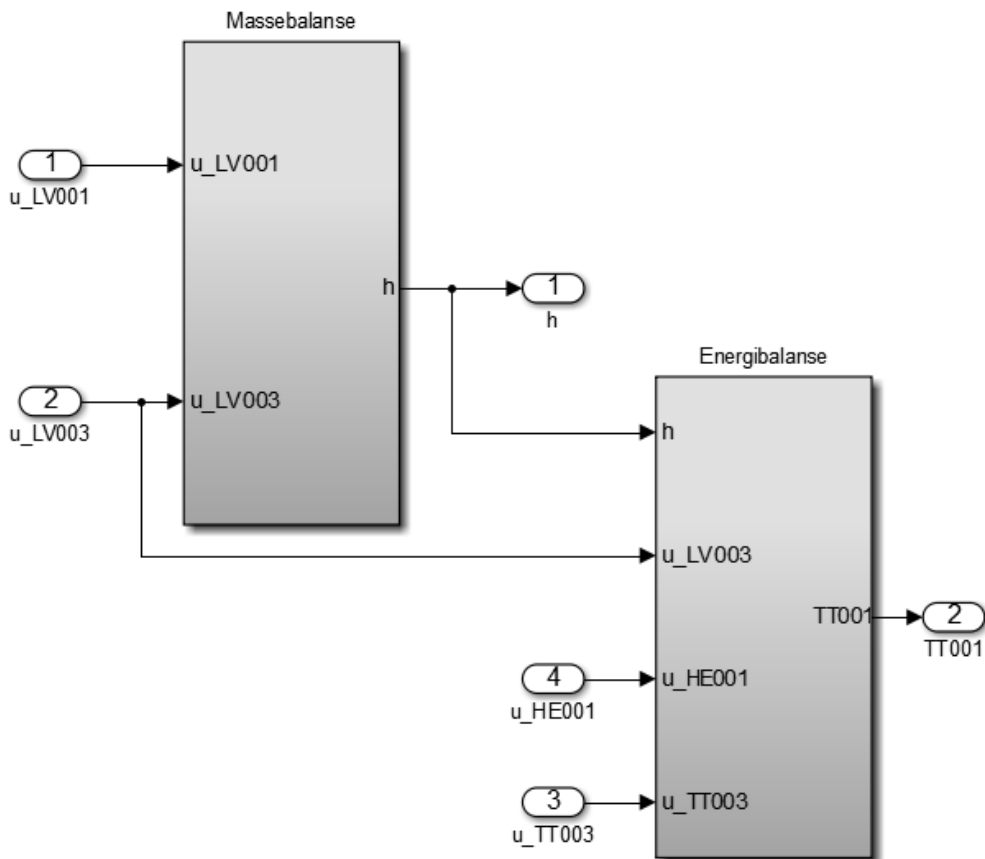
3.2 Implementering av prosessmodell i Simulink

Dette delkapittelet tar for seg å implementere modellen i Simulink. Modellen består av differensialligningene (3.9) og (3.17). Figur 3.2 viser modellen som ble implementert for å teste mot loggede data fra den virkelige prosessen.



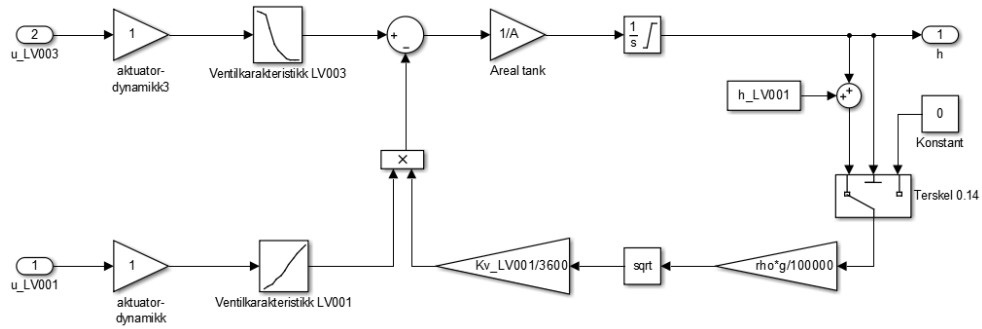
Figur 3.2 Modell implementert i Simulink

De grønne blokkene er inngangsverdiene til prosessmodellen av de ulike pådragene til ventiler, varmeelement og temperatur fra vannbatteriet, mens de blå blokkene er nivå- og temperaturmåling som blir sammenlignet med prosessmodellens utganger. Den store lysegrå blokken representerer differensialligningene, kalt Prosessmodell. Ved å dobbelklikke på den dukker det opp to mindre blokker, hvor differensialligningene befinner seg, se figur 3.3.

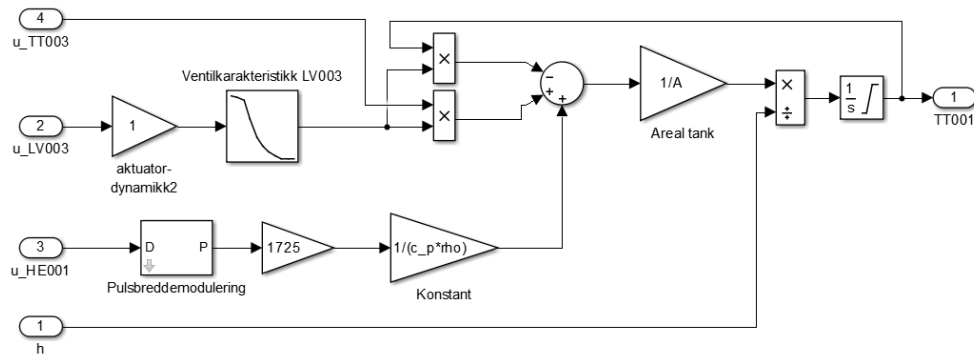


Figur 3.3: Subsystem av prosessmodell for nivå og temperatur i tanken

Figur 3.4 og 3.5 viser differensialligningene (3.9) og (3.17) for nivå og temperatur i tanken, modellert i form av blokkdiagram.



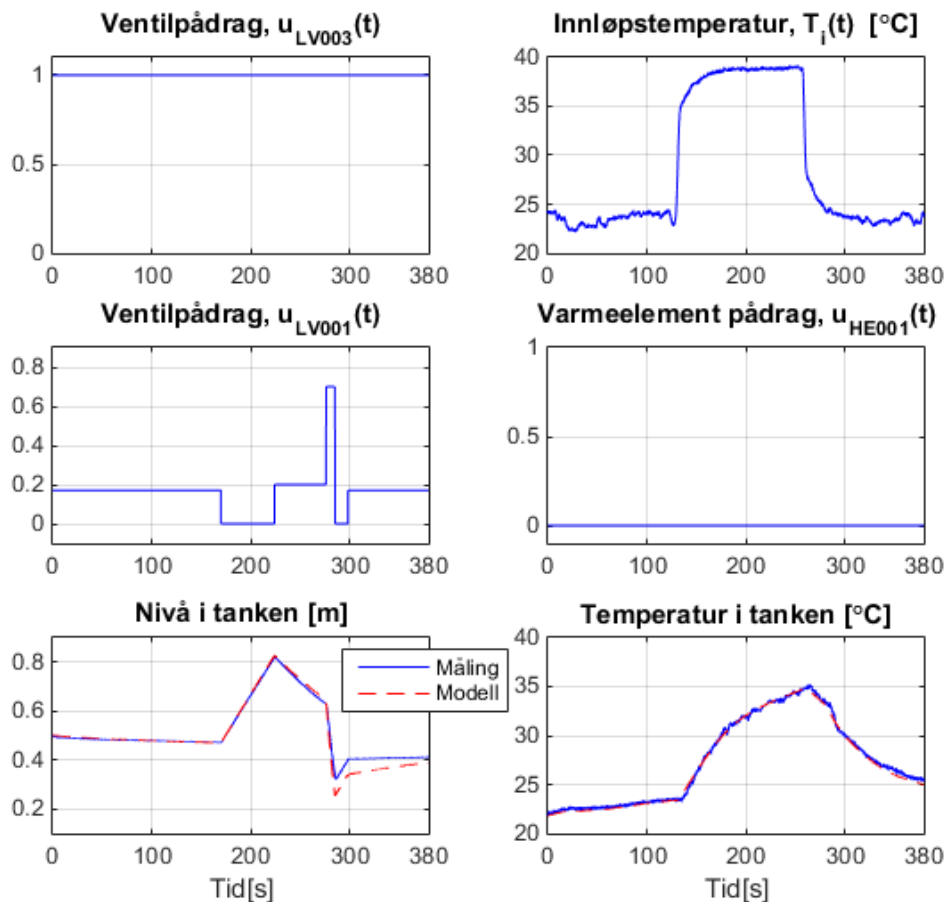
Figur 3.4: Blokkmodell for nivå i tanken basert på differensialligning (3.9)



Figur 3.5: Blokkmodell for temperatur i tanken basert på differensialligning (3.17)

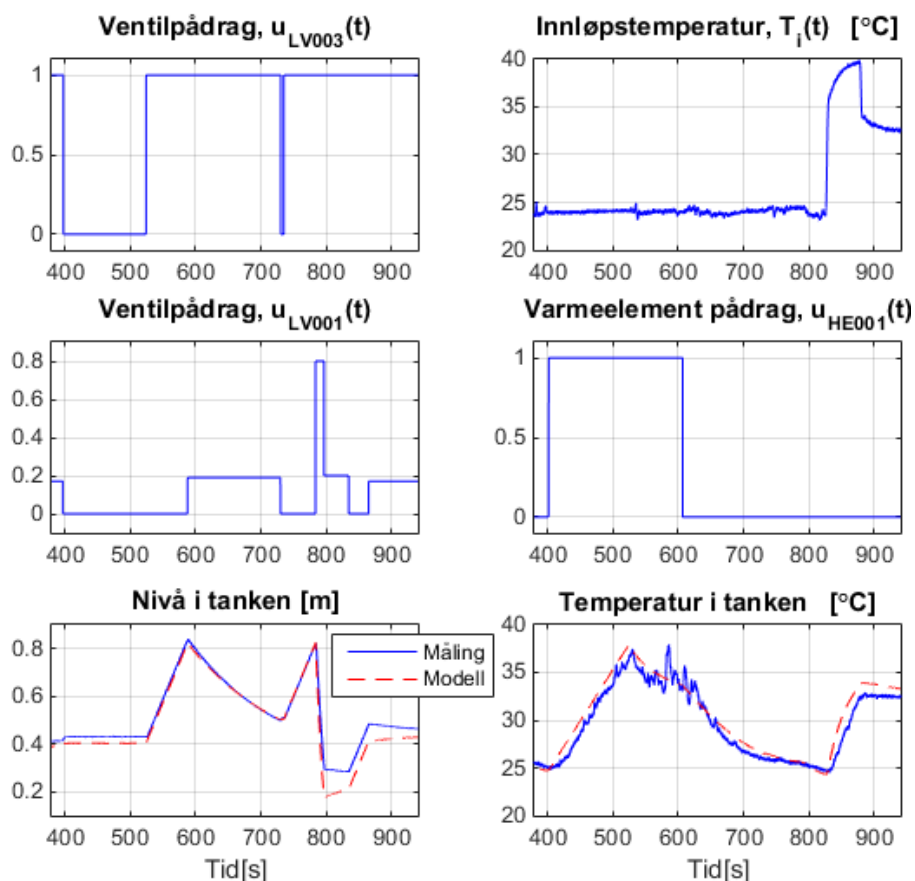
3.3 Verifisering av modell

For å kunne benytte modellen til estimering av tilstander for en fysisk prosess, er det nødvendig å verifisere modellen for å se om den gir et riktig bilde av prosessen. Ved verifisering har den virkelige prosessen blitt kjørt flere ganger, mens målingene og pådragene er blitt logget. Figur 3.6 og 3.7 viser et forsøk hvor modellen testes mot prosessen over en tidsperiode på 940 sekunder. Det er delt opp i to figurer for oversiktens skyld og for at grafene skal bli mer detaljert.



Figur 3.6 Figuren viser responsen fra 0–380 sek. Øverst viser innløpsventil pådrag, samt temperaturen ved innløpet av tanken. I midten vises utløpsventil pådrag og pådrag til varmeelement. Nederst er estimert nivå og temperatur i tanken

Teoretisk vil ikke en slik modell stemme perfekt med den virkelige prosessen på grunn av antagelser og tilnæringer. Figur 3.6 viser at modellens dynamikk stemmer bra med virkelig prosess over denne tidsperioden. Dette gjelder både nivå og temperatur i tanken.

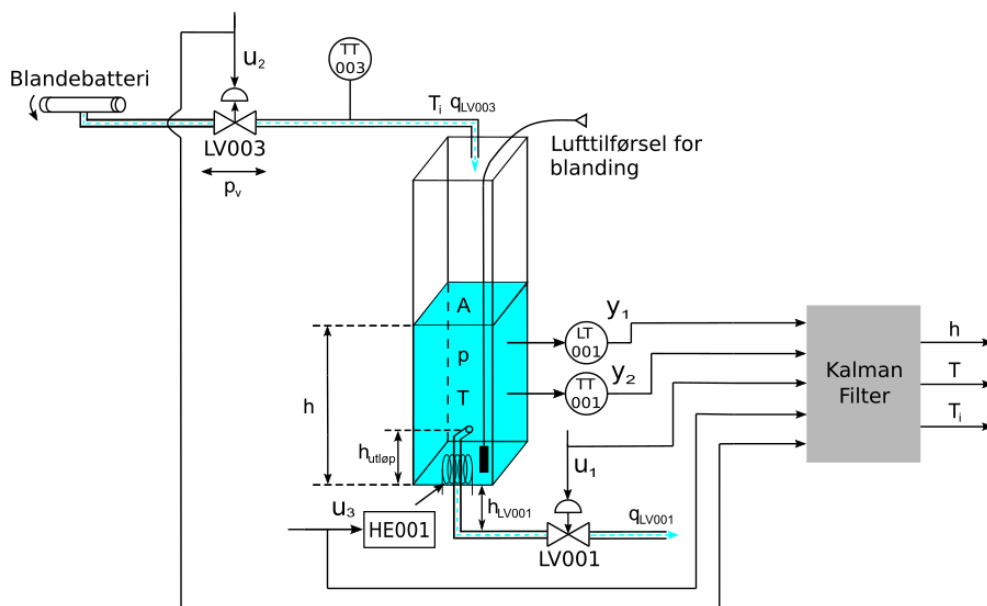


Figur 3.7: Figuren viser responsen fra 380–940 sek. Øverst viser innløpsventil pådrag, samt temperaturen på innløpet til tanken. I midten vises utløpsventil pådrag og pådrag til varmeelement. Nederst er estimert nivå og temperatur i tanken

Figur 3.7 viser en tidsperiode fra $t = 380$ til $t = 940$. En ser at temperaturen i tanken ikke stemmer like bra som på figur 3.6. Fra $t = 400$ til $t = 600$, er varmeelement på. I denne perioden går frem at modellens respons reagerer med engang, mens det tar litt tid før det skjer noe temperatur endring i den virkelige prosessen. Dette skyldes på at varmeelementet i virkeligheten bruker litt tid for å varme seg opp, mens i modellen gir den maks effekt med engang. Rundt $t = 600$, er det registrert mye forstyrrelser av temperatur i tanken. Dette kan skyldes på avstanden mellom varmeelementet og temperaturmåleren TT001 er for kort, og nivået i tanken i samme tidsperiode er nesten helt full. Dermed bruker vannet som strømmer oppi tanken litt tid for å synke nedover, slik at varmeelementet dominerer nedenfor og skaper denne temperaturforstyrrelsen. Fra $t = 820$ viser det at modellresponsen har en brattere kurve enn det som er målt i virkelig. Årsaken er at modellrespons for nivået i samme tidsperiode har et mye mindre volum, som fører til et større effektutslag på modellens estimat av temperaturen i tanken. Selv om det er litt avvik, så stemmer også dynamikken bra med virkelig prosess i denne perioden også.

4 Implementering av utvidet Kalmanfilter

Kapittelet tar for seg å konstruere et EKF for å estimere tilstanden på vanntemperatur $T_i(t)$ som strømmer oppi tanken. For å gjøre dette ble signalet fra nivåtransmitter LT001 og temperaturtransmitter TT001 i tanken brukt som måling. Disse er også tilstander i systemet. Signalet til varmelement HE001 og ventilene LV001 og LV003 er brukt som pådrag, se figur 4.1. EKF fra dette kapittelet blir testet i kapittel 5.



Figur 4.1: Enkel prinsippskisse (uten støy)

4.1 Teori

Med hensyn til ulineære systemer, ble det klassiske Kalmanfilteret utvidet. Det utvidede Kalmanfilteret ble utviklet av Stanley Schmidh i 1967 [10]. EKF løser dette ved å linearisere modellen omkring et arbeidspunkt. Ved linearisering finnes det et uttrykk for den lineære transisjonsmatrisen, Φ . Derivasjonen som må utføres for å finne Φ kan være vanskelig, og det er ressurskrevende siden Φ må beregnes for hvert tidssteg [4, 11].

4.1.1 EKF ligninger

Algoritmen for EKF tar i utgangspunktet i den ulineære system modellen, gitt ved [12]:

$$x(k+1) = f(x(k), u(k)) + v(k) \quad (4.1)$$

$$y(k) = g(x(k)) + w(k) \quad (4.2)$$

Variabler og funksjoner i modellen ovenfor er som følgende:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_l \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_s \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_l \end{bmatrix} \quad (4.3)$$

hvor x er tilstandsvektor med n tilstander, y er målevektor med l målinger, u er pådragsvektor med s pådrag, f er systemets funksjonsvektor med n funksjoner og g er målevektor funksjon. Støyvektorene, v og w , er stokastisk hvitt prosess- og målestøy henholdsvis.

Algoritmen er hentet fra [12].

Algoritme for EKF

$$\bar{x}(k) = f(\hat{x}(k-1), u(k-1)) \quad (4.4)$$

$$\Phi(k) = \left. \frac{\partial f(\cdot)}{\partial x} \right|_{\hat{x}(k-1), u(k-1)} \quad (4.5)$$

$$\bar{P}(k) = \Phi(k) \cdot \hat{P}(k-1) \cdot \Phi^T(k) + Q(k) \quad (4.6)$$

$$D(k) = \left. \frac{\partial g(\cdot)}{\partial x} \right|_{\bar{x}(k)} \quad (4.7)$$

$$K(k) = \bar{P}(k) \cdot D^T(k) \cdot (D(k) \cdot \bar{P}(k) \cdot D^T(k) + R)^{-1} \quad (4.8)$$

$$\bar{y}(k) = g(\bar{x}(k)) \quad (4.9)$$

$$\hat{x}(k) = \bar{x}(k) + K(k) \cdot (y(k) - \bar{y}(k)) \quad (4.10)$$

$$\hat{P}(k) = (I - K(k) \cdot D(k)) \cdot \bar{P}(k) \quad (4.11)$$

Presisering og forklaring av ligningene i EKF algoritmen:

- Ligning (4.4) beregner apriori tilstandsestimat. Tilstandsestimatet beregnes ut i fra forrige tidssteg av aposteriori tilstandsestimat og pådrag.
- Ligning (4.5) beregner den lineariserte transisjonsmatrisen til modellen omkring et arbeidspunkt ved bruk av aposteriori tilstandsestimat og pådrag i forrige tidssteg.
- Ligning (4.6) beregner apriori kovariansmatrise.
- Ligning (4.7) beregner målematrise.
- Ligning (4.8) beregner Kalmanfilterforsterkning.
- Ligning (4.9) beregner prediktert måling.
- Ligning (4.10) beregner aposteriori tilstandsestimat. Estimaten blir benyttet i både ligning (4.4) og (4.5).
- Ligning (4.11) beregner aposteriori kovariansmatrise, som blir benyttet i ligning (4.6) ved neste tidssteg.

4.2 Linearisering og diskretisering av modellen

Det ulineære systemet i dette kapitlet kan tilnærmes med et lineært system omkring et arbeidspunkt. Ligning (4.5) i EKF algoritmen beregner den lineariserte transisjonsmatrisen. Dimensjonen på transisjonsmatrisen er $n \times n$, hvor $n = 3$ tilstander i dette tilfellet.

For å få dette implementert i EKF må den diskrete transisjonsmatrisen uttrykkes følgende:

$$\begin{aligned} \Phi(k) &= \frac{\begin{bmatrix} \partial f_1(\cdot) \\ \partial f_2(\cdot) \\ \partial f_3(\cdot) \end{bmatrix}}{\begin{bmatrix} \partial x_1 & \partial x_2 & \partial x_3 \end{bmatrix}} = \begin{bmatrix} \frac{\partial f_1(\cdot)}{\partial x_1} & \frac{\partial f_1(\cdot)}{\partial x_2} & \frac{\partial f_1(\cdot)}{\partial x_3} \\ \frac{\partial f_2(\cdot)}{\partial x_1} & \frac{\partial f_2(\cdot)}{\partial x_2} & \frac{\partial f_2(\cdot)}{\partial x_3} \\ \frac{\partial f_3(\cdot)}{\partial x_1} & \frac{\partial f_3(\cdot)}{\partial x_2} & \frac{\partial f_3(\cdot)}{\partial x_3} \end{bmatrix} \\ &= \begin{bmatrix} \Phi_{11}(k) & \Phi_{12}(k) & \Phi_{13}(k) \\ \Phi_{21}(k) & \Phi_{22}(k) & \Phi_{23}(k) \\ \Phi_{31}(k) & \Phi_{32}(k) & \Phi_{33}(k) \end{bmatrix} \end{aligned} \quad (4.12)$$

hvor $f_1(\cdot)$ og $f_2(\cdot)$ er systemets differensialligninger (3.9) og (3.17) som ble utledet i kapittel 3.1, mens $f_3(\cdot)$ er den ukjente vanntemperatur tilstanden, $T_i(t)$, som kommer fra vannbatteriet. For å gjøre dette, må $f_3(\cdot)$ modelleres slik at den er i form av en differensialligning, gitt ved [5];

$$\frac{dT_i(t)}{dt} = 0 \quad (4.13)$$

For å sette opp transisjonsmatrisen, $\Phi(k)$ må ligningene (3.9), (3.17) og (4.13) diskretiseres. Den mest vanlige metoden er å bruke Eulers forovermetode (EF).

Diskretisering av differensialligningene kan gjøres direkte med Eulers forovermetode [13]:

$$x(k+1) = x(k) + T_s \dot{x}(k) \quad (4.14)$$

hvor T_s er samplingstiden og $\dot{x}(k)$ settes til $f_1(\cdot)$, $f_2(\cdot)$ og $f_3(\cdot)$

Det er her praktisk å bytte navn på tilstandsvariablene.

Definerer:

$$x_1(k) = h(k) \quad (4.15)$$

$$x_2(k) = T(k) \quad (4.16)$$

$$x_3(k) = T_i(k) \quad (4.17)$$

Diskretisering av differensialligning (3.9) med Eulers forovermetode:

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s \dot{x}_1(k) \\ &= x_1(k) + \frac{T_s}{A} \cdot (q_{LV003}(k) \\ &\quad - c \cdot f_1(u_{LV001}(k)) \cdot \sqrt{x_1(k) + h_{LV001}}) \\ &= f_1(\cdot) \end{aligned} \quad (4.18)$$

For differensialligning (3.17) gir da:

$$\begin{aligned} x_2(k+1) &= x_2(k) + T_s \dot{x}_2(k) \\ &= x_2(k) + \frac{T_s}{A \cdot x_1(k)} (q_{LV003}(k) \cdot (x_3(k) \\ &\quad - x_2(k)) + \frac{P(k)}{c_p \rho}) \\ &= f_2(\cdot) \end{aligned} \quad (4.19)$$

og differensialligning (4.13):

$$\begin{aligned} x_3(k+1) &= x_3(k) + T_s \dot{x}_3(k) \\ &= x_3(k) + T_s \cdot 0 = x_3(k) \\ &= f_3(\cdot) \end{aligned} \quad (4.20)$$

For å derivere elementene i transisjonsmatrisen, $\Phi(k)$ er det benyttet MATLAB, se vedlegg C.1 for kode. Elementene i transisjonsmatrisen, $\Phi(k)$ ble følgende:

$$\Phi_{11}(k) = 1 - c \cdot f_1(u_{LV001}(k)) \cdot \frac{T_s}{2A \cdot \sqrt{x_1(k) + h_{LV001}}} \quad (4.21)$$

$$\Phi_{12}(k) = 0 \quad (4.22)$$

$$\Phi_{13}(k) = 0 \quad (4.23)$$

$$\Phi_{21}(k) = \frac{T_s}{Ax_1^2(k)} \cdot \left(q_{LV003}(k)(x_2(k) - x_3(k)) - \frac{P(k)}{c_p \rho} \right) \quad (4.24)$$

$$\Phi_{22}(k) = 1 - \frac{T_s \cdot q_{LV003}(k)}{A \cdot x_1(k)} \quad (4.25)$$

$$\Phi_{23}(k) = \frac{T_s \cdot q_{LV003}(k)}{A \cdot x_1(k)} \quad (4.26)$$

$$\Phi_{31}(k) = 0 \quad (4.27)$$

$$\Phi_{32}(k) = 0 \quad (4.28)$$

$$\Phi_{33}(k) = 1 \quad (4.29)$$

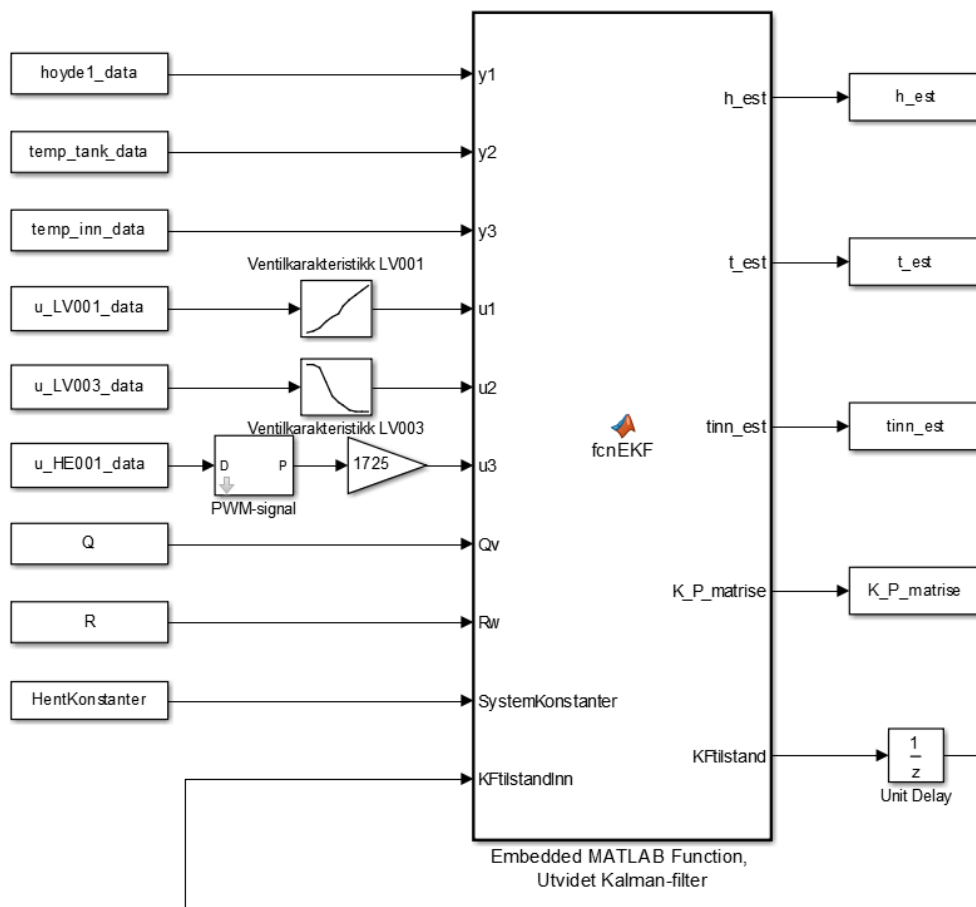
Dette gir en transisjonsmatrise på formen:

$$\Phi(k) = \begin{bmatrix} \Phi_{11}(k) & 0 & 0 \\ \Phi_{21}(k) & \Phi_{22}(k) & \Phi_{23}(k) \\ 0 & 0 & \Phi_{33}(k) \end{bmatrix} \quad (4.30)$$

4.3 Implementering av utvidet Kalmanfilter i Simulink

Koden for EKF algoritmen er implementert i Embedded MATLAB Function, se figur 4.2 funksjonsblokken, fcnEKF. Denne funksjonsblokken er nyttig for koding algoritmer som er bedre angitt i tekstspråket MATLAB enn i det grafiske språket Simulink. Utskrift av koden i funksjonsblokken finnes på vedlegg C.3.

Figur 4.2 viser funksjonsblokken til Kalmanfilteret med inn- og utganger.



Figur 4.2: EKF funksjonsblokk implementert i Simulink

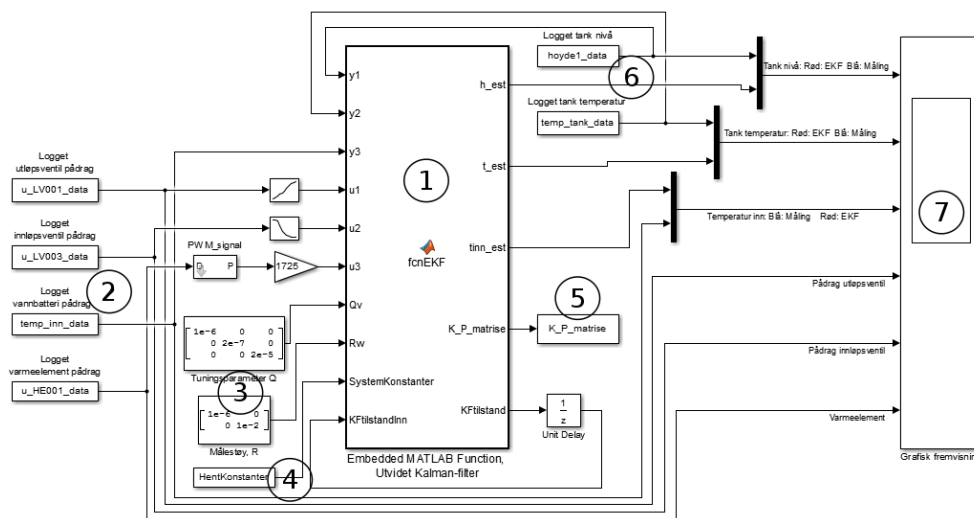
Med utgangspunkt i figur 4.2, er det laget en tabell med oversikt og forklaring over notasjonen brukt her.

Tabell 4.1: Inn- og utganger av EKF funksjonsblokk

Inngang/utgang	Beskrivelse
y1	Måling av nivå i tanken.
y2	Måling av temperatur i tanken.
y3	Måling av temperatur fra vannbatteriet. Denne målingen brukes normalt ikke. Brukes kun som initialtilstand.
u1	Tabelloppslag av ventilkarakteristikk for ventil LV001, hvor det er ventilåpning som er inngangen her og ikke pådraget til ventilen.
u2	Tabelloppslag av ventilkarakteristikk for ventil LV003, hvor det er flyt i ventil LV003 som er inngang.
u3	PWM-signal av effekten som inngang, og ikke selve pådraget til varmeelementet.
Qv	Tuningsparameter, Q.
Rw	Tuningsparameter, R.
KFtilstandInn	Forrige tilstandsverdier av Kalmanfilteret.
h_est	Estimat av nivå i tanken (tilstand).
t_est	Estimat av temperatur i tanken (tilstand).
tinn_est	Estimat av temperatur inn i tanken (tilstand).
K_P_matrise	Verdier av Kalmanfilterforsterkningsmatrise og kovariansmatrise for estimeringsavvikene.
SystemKonstanter	Henter konstanter fra MATLAB fil.
KFtilstand	Tilstanden for Kalmanfilteret som er lagre fra et steg til neste.

5 Testing og tuning

I dette kapittelet blir EKF utviklet i kapittel 4 testet mot totankanlegget. For at det skal være enkel å gjøre forskjellige innstillinger av tuningsmatrisene, logges det innganger og utganger av totankanlegget ved kjøring. Vedlegg A.1 viser Simulink modell for logging av disse verdiene. I alle forsøk er det tatt i bruk luftblanding i tanken. Figur 5.1 viser hvordan EKF kjøres mot loggede data.



Figur 5.1: Test av Kalmanfilter mot virkelig loggede data

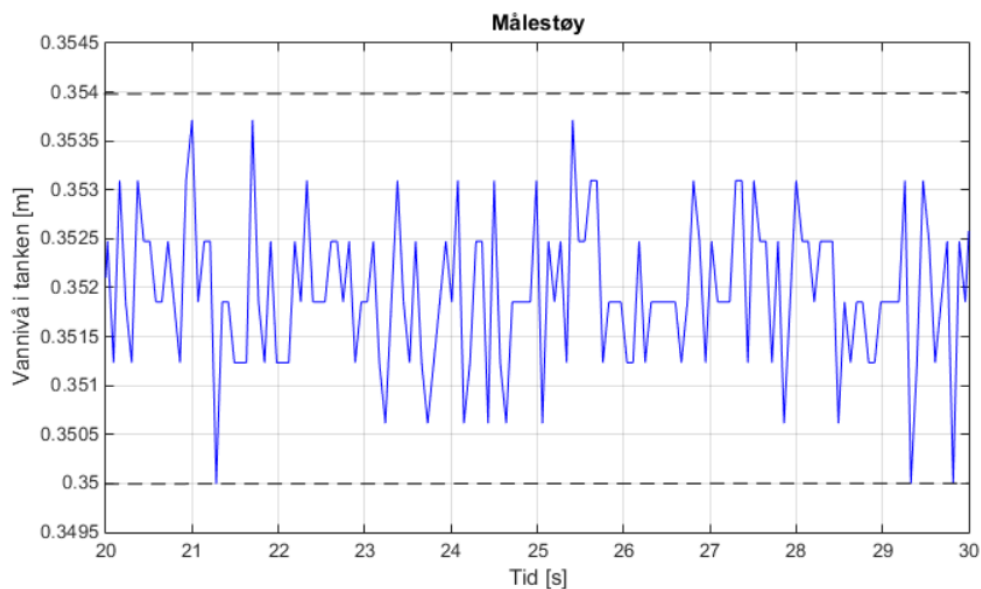
Nedenfor viser det en mer detaljert forklaring av de nummererte delene i figur 5.1.

1. EKF-funksjonsblokk som ble implementert i kapittel 4.
2. Logget pådragssignal til varmeelement HE001 og ventilene LV001 og LV003. Logget måling av temperaturen ved innløpet som brukes til initialtilstand og sammenlignes med EKF estimat.
3. Tuningsparametre, Q og R .
4. Totankanlegg konstanter som hentes fra TotankData.m, vedlegg C.7.
5. Kovariansmatrise og Kalmanfilter forsterkningsmatrise skrives til "workspace".
6. Logget nivå- og temperaturmåling fra virkelig prosess for sammenligning.
7. Fremvisning og sammenligning av EKF estimatene med virkelige målinger.

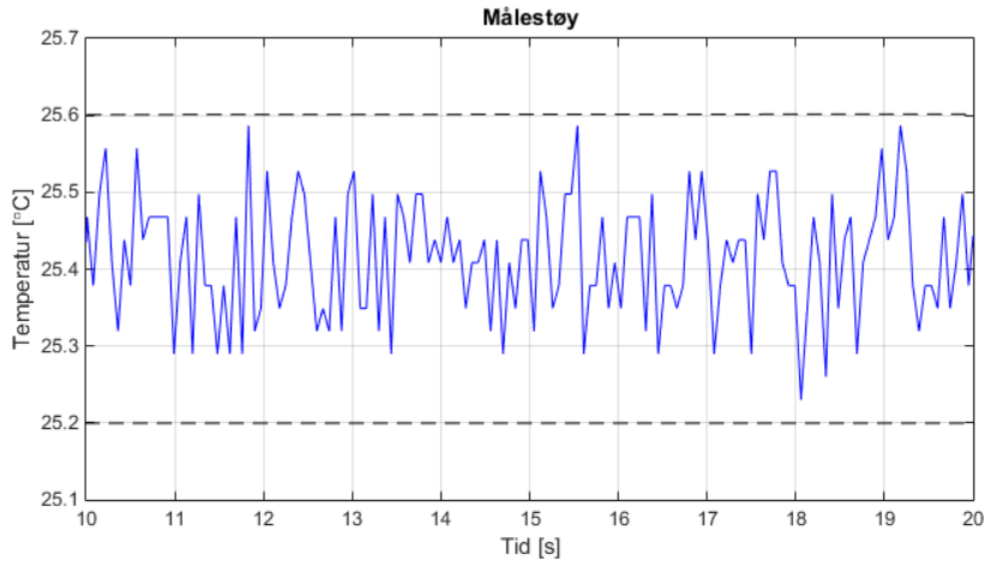
5.1 Drift av EKF

For at Kalmanfilteret skal kunne estimere gode estimater er det viktig at valg av tuningsparametre Q og R , er omtrent riktig. Siden forholdet mellom Q og R er det vesentlige så velges en verdi for R og Q inneholder tuningsparametrene, som vanlig er det kun elementene i Q som brukes. Valg av målestøy autokovarians matrisen R , tar en i utgangspunktet å sette den tilnærmet lik variansen til målestøyen i systemet [14]. Normalt sett kan en prøver seg frem med litt prøving og feiling metoden for å stille inn elementene i Q [5].

I dette systemet er det nivå og temperatur i tanken som måles, dermed er det rimelig å ta i utgangspunktet i målestøyen til disse målingene. Figur 5.2 og 5.3 viser en sekvens av nivå- og temperaturmåling i tanken.



Figur 5.2: En tidsserie av nivåmåling i tanken



Figur 5.3: En tidsserie av temperaturmåling i tanken

Som det går frem i figur 5.2 og figur 5.3 ligger målingene mellom de stiplede linjene for gjennomsnitt $\pm 2\text{mm}$ og $\pm 0.2^\circ\text{C}$. Ut i fra flere forsøk har det vist at målingene ligger innenfor disse linjene.

Fra teorien er støyen antatt til å være Gaussisk fordelt [14], dermed har det at 95% av målingene er innenfor to standardavvik. Fra statistikkberregning blir standardavviket for nivåmåling lik:

$$\sigma_{11} = \frac{1}{2} \cdot 2[\text{mm}] = 1[\text{mm}] \quad (5.1)$$

og for temperaturmåling:

$$\sigma_{22} = \frac{1}{2} \cdot 0.2[^\circ\text{C}] = 0.1[^\circ\text{C}] \quad (5.2)$$

Dette gir autokovarians matrise for målestøy, R:

$$R = \begin{bmatrix} \sigma_{11}^2 & 0 \\ 0 & \sigma_{22}^2 \end{bmatrix} = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 0.01 \end{bmatrix} \quad (5.3)$$

og denne målestøymatrisen er et utgangspunkt i denne oppgaven.

Bestemmelsen av diagonalelementene i Q er noe vanskeligere enn for R . Generelt er det slik at store verdier i Q fører til at Kalmanfilteret antar at det er mye feil/støy i modellen. Dette gir parameterestimat som legger stor vekt på målingen og endres fort, men samtidig er det mye virring på estimatet. Små verdier i Q betyr god modell med lite feil/støy, dermed legges det mer vekt forrige tilstandsestimat enn målingen. Dette kan en se ut fra ligning (5.4) som beregner det aposteriori tilstandsestimatet:

$$\hat{x}(k) = \bar{x}(k) + K(k) \cdot (y(k) - \bar{y}(k)) \quad (5.4)$$

Dersom en i tillegg studerer ligningene (4.6) og (4.8) i EKF algoritmen, viser det at dersom R er liten og Q er stor, vil $K(k)$ bli stor. Når $K(k)$ er stor, fører det til at ligning (5.4) legger stor vekt på feilen.

Siden hver av prosessforstyrrelse vanligvis antas å være uavhengig hverandre, kan Q uttrykkes som en diagonalmatrise [5]:

$$Q = \begin{bmatrix} Q_{11} & 0 & 0 \\ 0 & Q_{22} & 0 \\ 0 & 0 & Q_{33} \end{bmatrix} \quad (5.5)$$

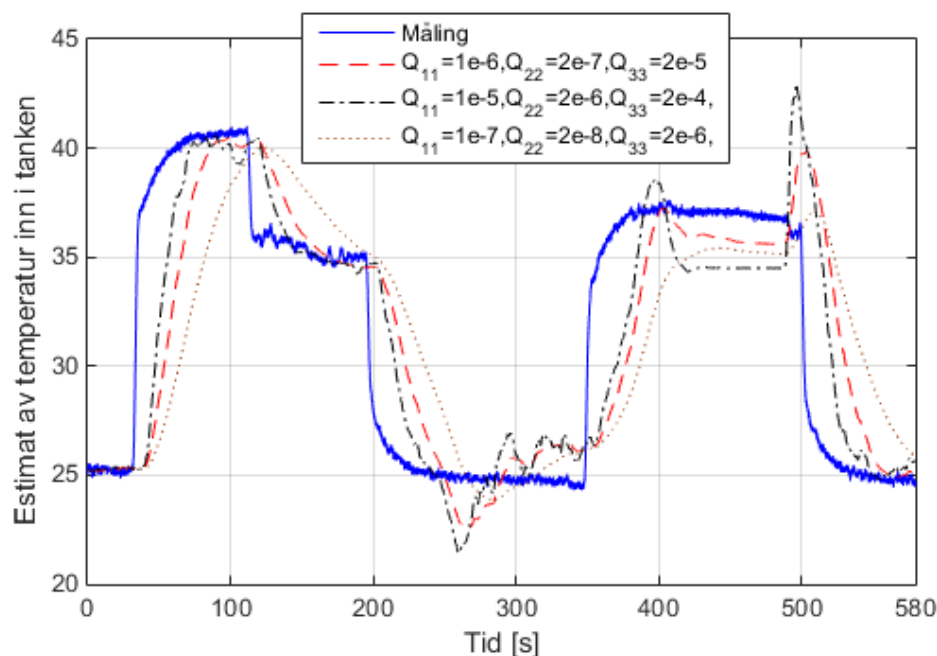
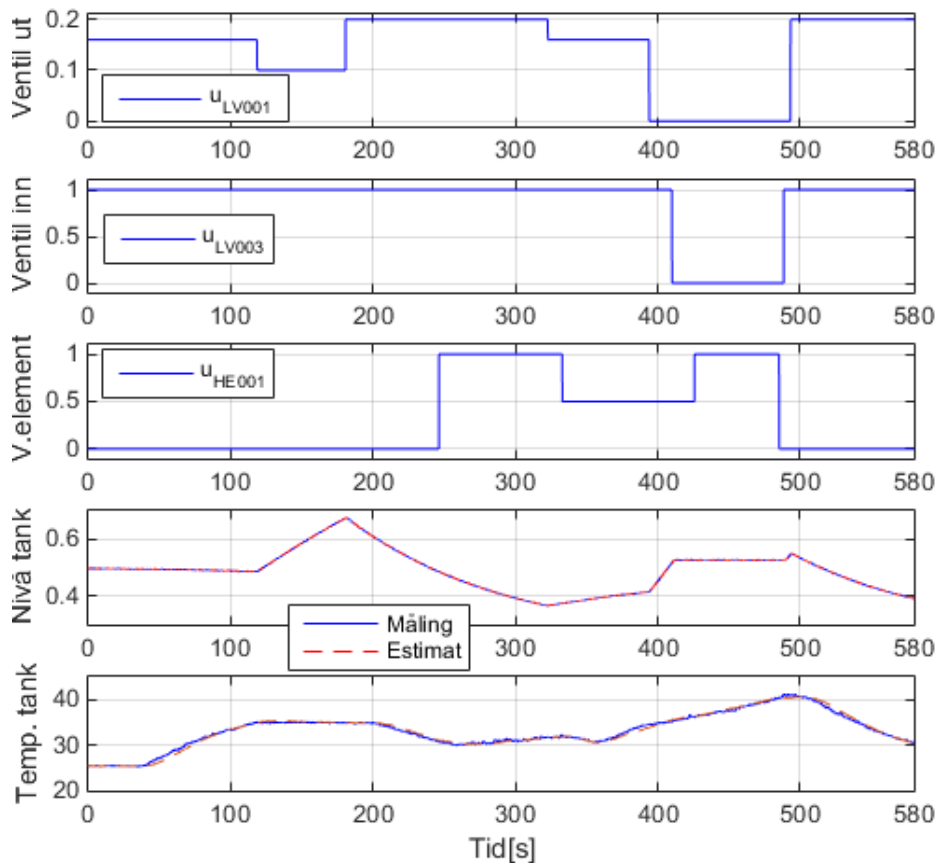
der hvert element er variansen til prosessstøy og de kan bli justert uavhengig av hverandre.

For å ha et utgangspunkt for Q , er det en tommelfingerregel med å starte og velge Q så stort som mulig uten at tilstandsestimatet blir for mye støy(virring) [5]. Etter å ha utført flere tester av Kalmanfilteret mot virkelig prosess med forskjellige Q verdier, viser det seg at Q lik:

$$Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 & 0 \\ 0 & 2 \cdot 10^{-7} & 0 \\ 0 & 0 & 2 \cdot 10^{-5} \end{bmatrix} \quad (5.6)$$

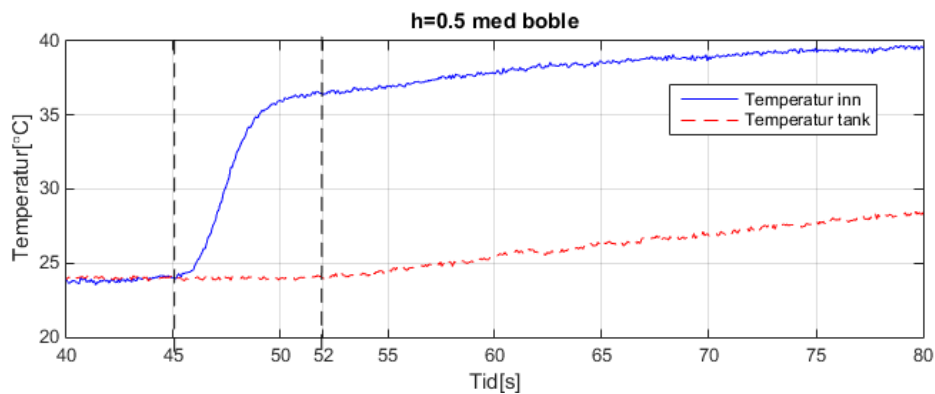
gir et godt resultat. Q_{11} og Q_{22} er prosessstøykovarians for nivå og temperatur i tanken henholdsvis, mens Q_{33} er prosessstøykovarians for temperatur fra vannbatteriet.

Figur 5.4 viser et forsøk av Kalmanfilteret med Q verdier som ble funnet ovenfor, samt et sett med større og mindre verdier av Q . Forsøket er tatt i utgangspunktet i et logget data ved kjøring av totankanlegget, og det er antatt at initialtilstandene er kjente.



Figur 5.4: Øverste delen av figuren viser pådrag til ventilene og varmeelement, samt måling og EKF estimat av både nivå og temperatur i tanken. Nederst viser måling og EKF estimat av temperatur inn i tanken med forskjellige verdier av tuningsparameter Q

Figur 5.4 viser ved stor verdi av Q , gir det et estimat som er raskere, og til gjengjeld fører det til at estimatet gir et større avvik enkelte steder. Mens liten verdi av Q , gir et estimat som er glatt, men tregt. Det går også frem at estimatene reagerer litt etter den virkelige temperaturmålingen inn i tanken. Dette skyldes på at temperaturmåleren er plassert for langt ned i tanken, slik at det går litt tid før måleren registrerer en temperaturendring, som gjør at estimatene er forsinket. Dette er fornuftig, siden Kalmanfilteret skal ved hjelp av måling av temperatur i tanken estimere temperaturen inn. Figur 5.5 viser et eksempel på at det tar rundt 7 sekunder fra det skjer en endring i temperaturen inn til at temperaturtransmitteren i tanken registrerer en temperaturendring ved et nivå på 0.5 meter. Denne tiden er også avhengig av hvor høyt nivået er i tanken.



Figur 5.5: Måling av temperatur inn og temperatur i tanken

Årsaken kan også skyldes på at temperaturtransmitter ved innløpet er plassert et lite stykke i fra innløpet av tanken. Der en får litt transport tid i røret og den tiden vannet bruker å falle ned i tanken, og ikke minst blandingsprosessen i selve tanken.

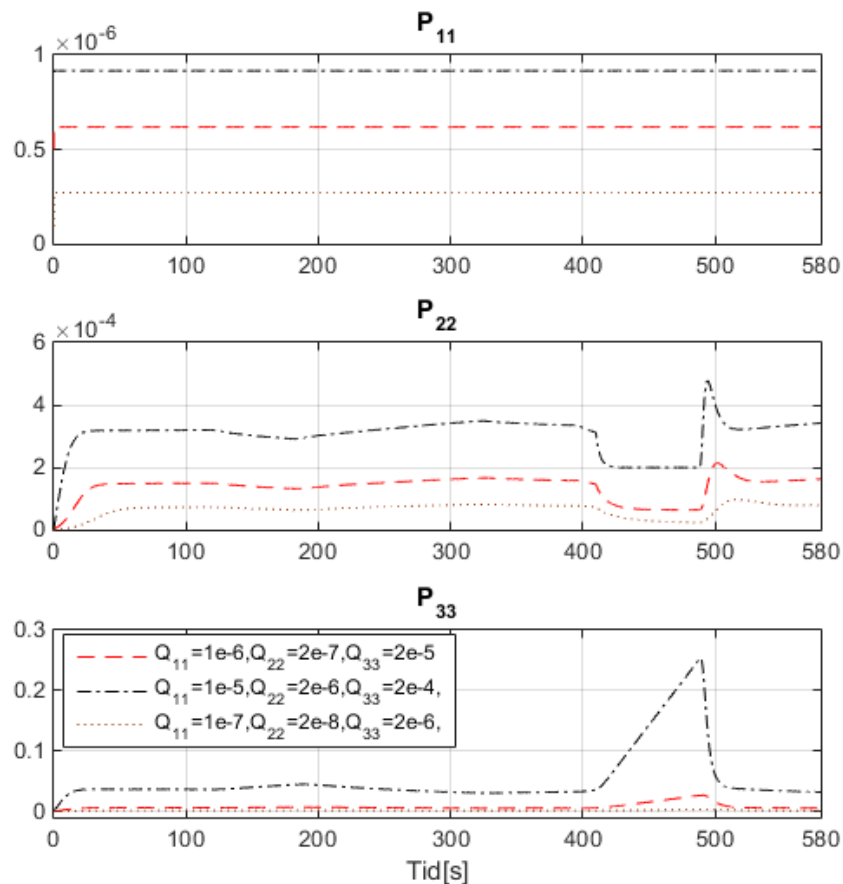
Fra figur 5.4 kan en registrere at det er en dupp ved ca. $t = 250$. Dette skjer når varmeelementet slås på. Som nevnt i kapittel 3.3 bruker varmeelementet i virkeligheten litt tid til å varme seg opp, som førte til at vannet i tanken ikke har den temperaturen som den matematiske modellen forventer, og dermed tolker Kalmanfilteret at det er inn temperaturestimatet som har blitt kaldere. Ved ca. $t = 490$ er det et betydelig hopp på inn temperaturestimatet. Dette er fordi temperaturen i tanken fortsetter å øke etter at elementet er slått av. Dette skyldes på at elementet i virkeligheten er fremdeles varm og fortsetter å varme opp vannet, som forårsaker til denne temperaturøkningen i tanken. Og med engang ventil LV003 åpner ved dette tidspunktet, tolker Kalmanfilteret at det er en stor økning i temperaturestimat inn.

Ved dette forsøket viser det at Kalmanfilteret svinger seg inn til riktig verdi rimelig kjapp og har et godt estimat når systemet ikke er påvirket av varmeelementet i de to første hundre sekundene. På grunn av dynamikken i varmeelementet gjør det vanskeligere for Kalmanfilteret.

På vedlegg A.2 viser det et annet forsøk hvor luftblanding ikke er tatt i bruk.

5.1.1 P-matrise

Kovariansmatrisen for estimeringsavvikene gir et bilde om usikkerheten i tilstandsestimatene. Matrisen blir også kalt for P-matrise. P-matrisen er tidsvarierende, dvs. at den oppdateres for hvert tidssteg basert på utviklingen av avviket mellom måling og estimat. Siden P-matrisen har størrelse 3×3 , tas ikke alle elementene med her, se vedlegg A.3 for en fullstendig oversikt over alle elementene i P-matrisen. Det er spesielt diagonalelementene i P-matrisen som er interessant i seg selv, siden det er disse som gir et estimat over estimeringsavvikene for temperatur og nivå i tanken, og temperatur inn. Figur 5.6 viser hvordan diagonalelementene i P-matrisen utvikler seg over tid ved forsøket i figur 5.4.

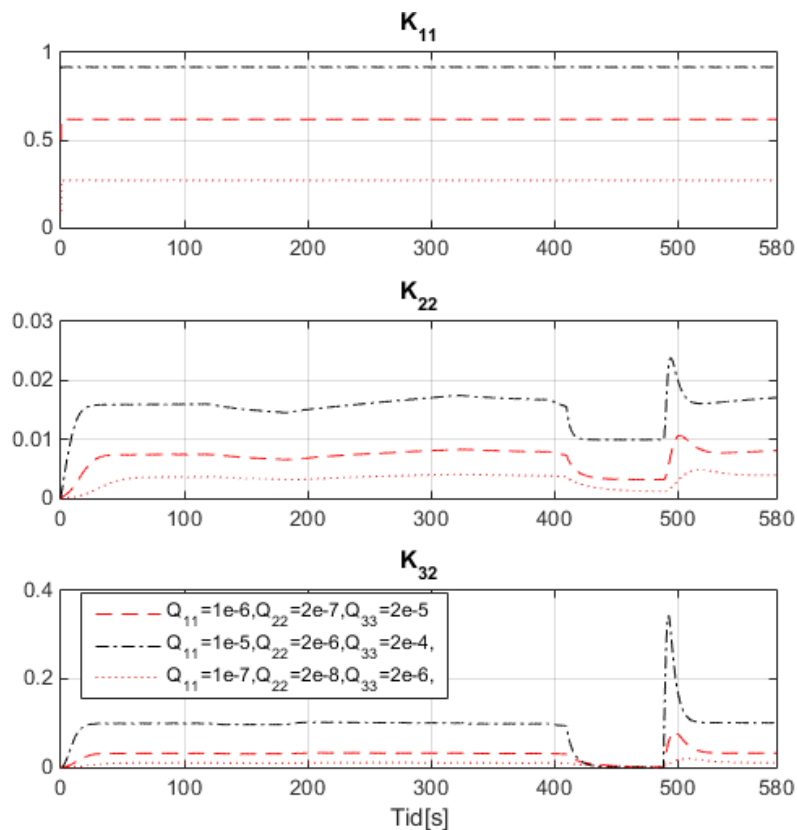


Figur 5.6: Diagonalelement til kovariansmatrise for aposteriori estimeringsavviket

Som det viser på figuren, er P_{11} tilnærmet lik konstant og samtidig liten. Dette er ganske rimelig siden det er nivå i tanken som måles direkte, og dermed har Kalmanfilteret alltid et godt estimat for nivået. For P_{22} ser en at diagonalelementet har en større verdi enn P_{11} selv om temperatur i tanken måles direkte. Forklaringen til dette er at målestøyen funnet tidligere i kapitlet for temperaturtransmitteren, har en større verdi enn for nivåmåleren. Det viser også at P_{22} har en dupp mellom tidspunktet $t = 400$ og $t = 490$. Ved å se på figur 5.4 og sammenligne med figur 5.6 ser en at i samme tidsintervallet er ventil LV003 stengt. Når et ledd i differensialligningene blir borte, er det rimelig at usikkerheten for estimeringsavviket blir mindre. Når det gjelder P_{33} har den en betydelig større verdi enn de to andre diagonalelementene, fordi temperaturen på vannet inn måles ikke direkte her, og har dermed alltid et dårligere estimat.

5.1.2 Kalmanfilter forsterkning

Ved beregning av Kalmanfilter forsterkningen, K er det benyttet P -matrisen fra forrige delkapittel. Se vedlegg A.4 for en fullstendig oversikt over alle elementene i K -matrisen.

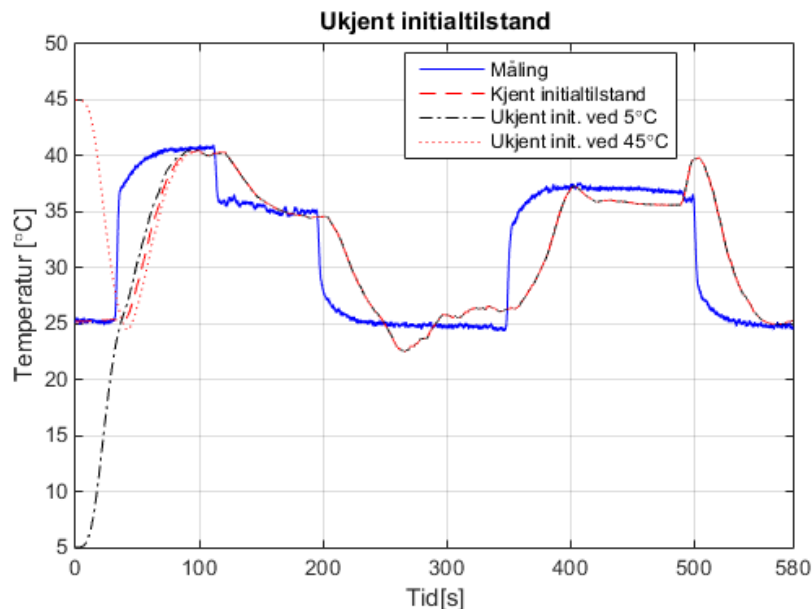


Figur 5.7: Kalmanfilter forsterkning

Verdien for K_{11} i figur 5.7 er stor, og det er ikke overraskende siden nivået i tanken måles direkte og at målestøyen for nivåmåleren er liten. K_{22} er lav siden målestøy for termperaturmåleren er ganske stor, dermed legges det lite vekt på målingen. Ved å se på intervallet mellom $t = 400$ og $t = 490$ for K_{32} , hvor forsterkningen er meget lav. Dette er ikke overraskende, siden stengt ventil LV003, betyr temperaturendring av vanntemperatur i tanken i denne perioden, kan ikke skyldes på temperaturen ved innløpet.

5.1.3 Test av EKF med ukjent initialtilstand

I dette delkapittelet ønskes det å teste ut hvordan Kalmanfilteret estimerer seg inn med forskjellige startverdier. Testen er basert på tuningsparametre (5.3) og (5.6).



Figur 5.8: Forsøk ved ukjent initialtilstand

Testen viser at tilstandsestimatene ved både kjent og ukjent startverdier henter seg inn rett før $t = 100$, og gir et ganske så likt estimat ved $t = 100$ og utover.

5.1.4 Test av EKF med prosessforstyrrelse

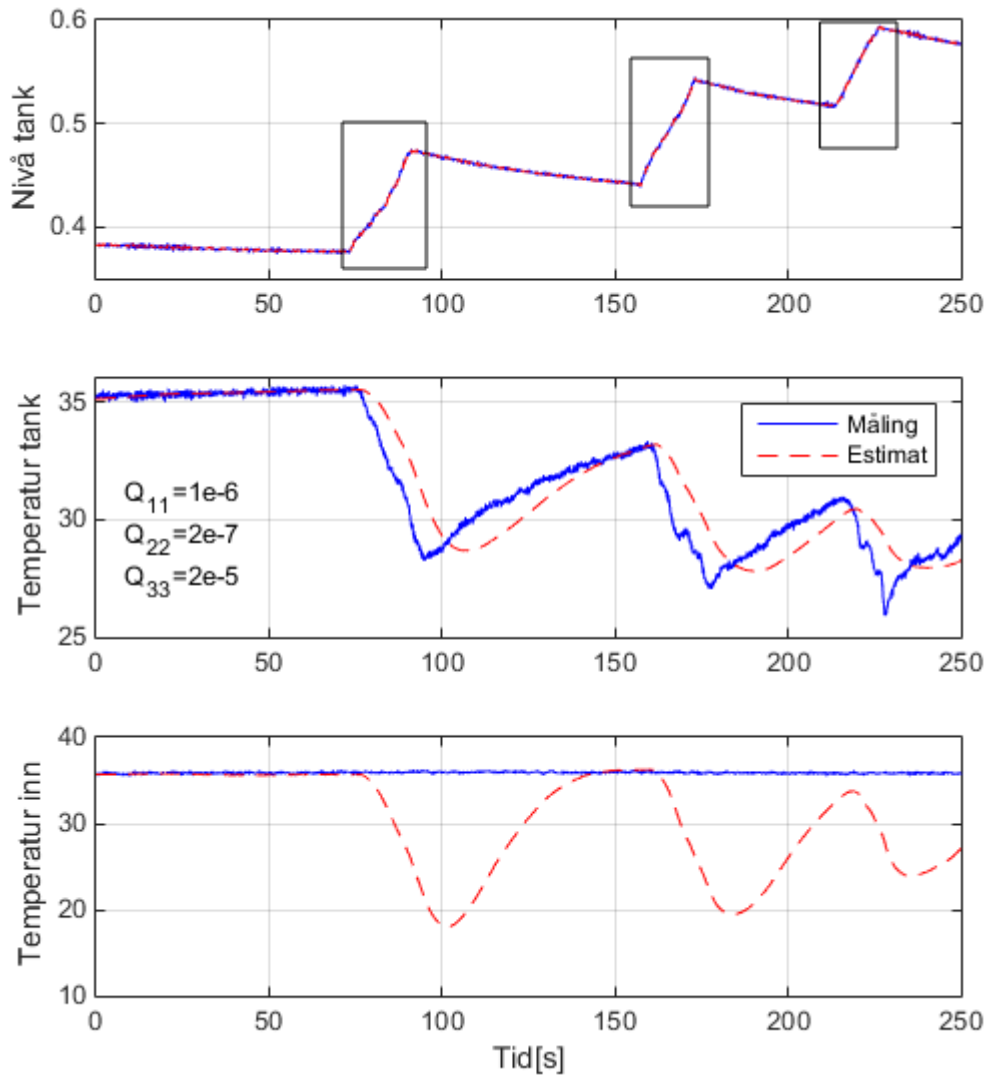
I dette delkapittelet blir prosessen utsatt av prosessforstyrrelse ved å tilsette en liter med iskaldt vann direkte oppi tanken. Figur 5.9 viser målt temperatur på vannet som tilsettes.



Figur 5.9: Vannet som tilføres tanken

Testen tar i utgangspunktet å holde temperaturen inn stabil og pådraget til ventil LV001 og LV003 konstant, slik at utskiftning av vann i tanken skjer kontinuerlig. Varmeelementet tas ikke med i denne testen.

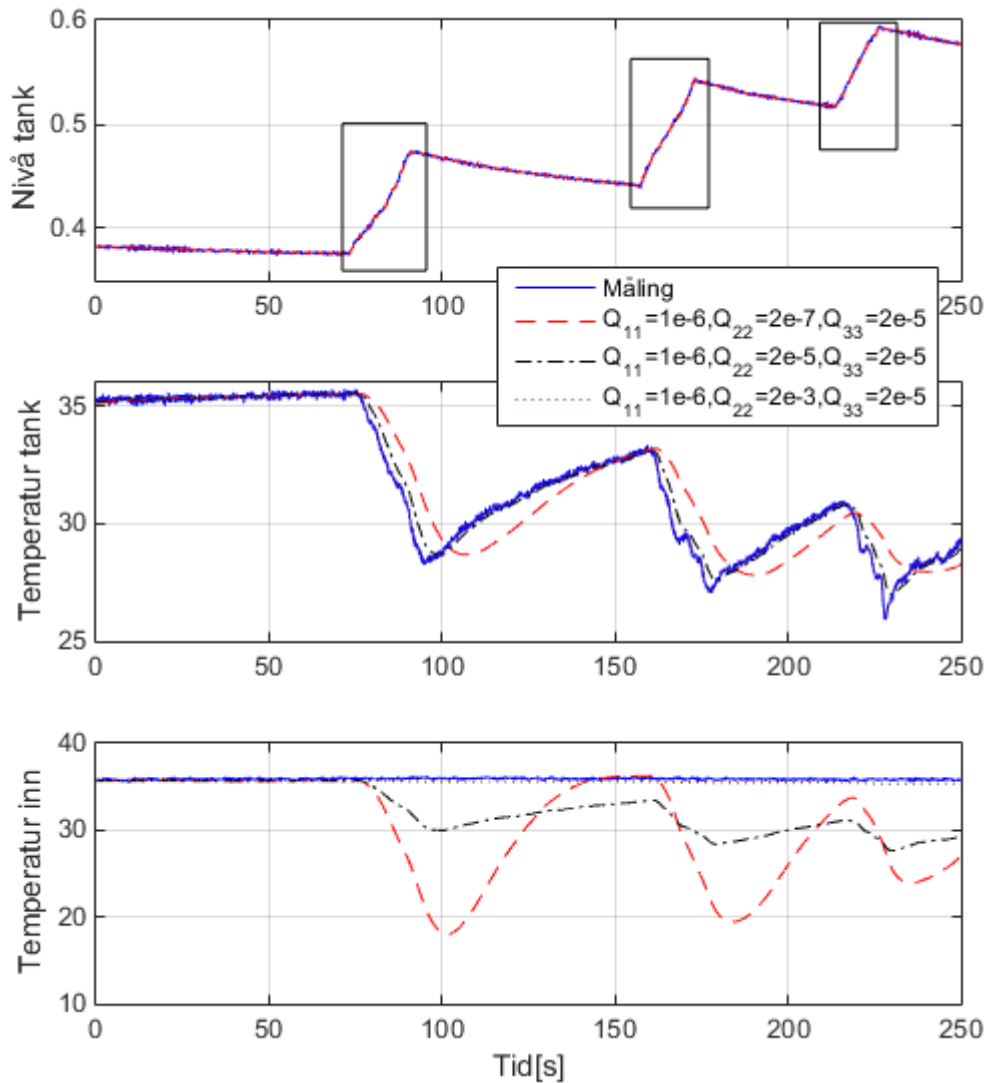
Figur 5.10 viser et resultat på hvordan Kalmanfilteret håndterer denne prosessforstyrrelsen med de opprinnelige tuningsparametre (5.3) og (5.6).



Figur 5.10: Prosessforstyrrelse med opprinnelig tuningsparamter

Som det viser på figur 5.10 har ikke Kalmanfilteret evnen til å fange opp denne forstyrrelsen i de tre markerte tidspunktene. Kalmanfilteret tolket det slik at temperaturen inn som har gjort denne temperaturendringen i tanken.

Figur 5.11 viser den samme testen når Q_{22} øker og samtidig beholde elementene Q_{11} og Q_{33} uendret.



Figur 5.11: Prosessforstyrrelse med variert tuningsparamter

Figur 5.11 viser ved større verdier av tuningsparameter Q_{22} , som betyr at prosessstøykovanansen for temperatur i tanken er stor. Kalmanfilteret vil da tolke den eksterne prosessforstyrrelsen som om at det er en forstyrrelse som har skjedd i tanken, og har derfor mindre påvirkning i tilstandsestimat for temperaturen inn ettersom Q_{22} blir større.

6 Kalmanfilter for estimering av tilsats

Kapittelet tar for seg å utvide modellen som ble utledet i kapittel 3 til å kunne oppdage ekstern tilsats av vann. Dermed vil systemet bli utvidet med en tilstand i forhold til systemet i kapittel 4 og 5. Dette gjør at det er nødvendig å beregne ny transisjonsmatrise. Kapittelet tar også for seg å implementere og teste det nye Kalmanfilteret.

6.1 Matematisk modell av prosessen

I dette delkapittelet utvides modellen fra kapittel 3.

6.1.1 Differensialligning for nivå i tanken

Differensialligning (3.9) for nivå i tanken som ble utledet i kapittel 3.1.1 blir her utvidet med et ekstra ledd, $q_t(t)$, siden det genereres ekstra masse i tanken. Dette gir da:

$$\frac{dh(t)}{dt} = \frac{1}{A} \left(q_{LV003}(t) - c \cdot \sqrt{h(t) + h_{LV001}} \cdot f_1(u_{LV001}(t)) + q_t(t) \right) \quad (6.1)$$

6.1.2 Differensialligning for temperatur i tanken

Differensialligning (3.17) for temperatur i tanken gjøres på samme måte som ble utledet i kapittel 3.1.2, og får et ekstra ledd.

Differensialligningen ble dermed:

$$\frac{dT(t)}{dt} = \frac{1}{Ah(t)} \cdot (q_{LV003}(t) \cdot (T_i(t) - T(t)) + \frac{P(t)}{c_p \rho} - q_t(t)(T(t) - T_t(t))) \quad (6.2)$$

hvor $T_t(t)$ er temperaturen på vannet som tilsettes. Denne antas å være kjent i forsøkene i dette kapittelet.

6.1.3 Differensialligning for temperatur fra vannbatteri

Differensialligningen (4.12) blir uendret:

$$\frac{dT_i(t)}{dt} = 0 \quad (6.3)$$

6.1.4 Differensialligning for inntrengning av vann

Den eksterne tilsatsen vil bli modellert som en egen tilstand i form av en differensialligning. For å gjøre dette er det forutsatt at den er langsomt varierende, og er nesten konstant [5, 12].

Dette gir differensialligning:

$$\frac{dq_t(t)}{dt} = 0 \quad (6.4)$$

6.2 Linearisering og diskretisering av modellen

Dette gjøres på tilsvarende måte som i kapittel 4.1.2.

Det er praktisk å bytte navn på tilstandsvariablene, definerer:

$$x_1(k) = h(k) \quad (6.5)$$

$$x_2(k) = T(k) \quad (6.6)$$

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

$$x_4(k) = q_t(k) \quad (6.8)$$

Diskretisering av differensialligningene gjøres direkte med Eulers forover metode:

$$x(k+1) = x(k) + T_s \dot{x}(k) \quad (6.9)$$

og gir:

$$x_1(k+1) = x_1(k) + T_s \dot{x}_1(k) = f_1(\cdot) \quad (6.10)$$

$$x_2(k+1) = x_2(k) + T_s \dot{x}_2(k) = f_2(\cdot) \quad (6.11)$$

$$x_3(k+1) = x_3(k) + T_s \dot{x}_3(k) = f_3(\cdot) \quad (6.12)$$

$$x_4(k+1) = x_4(k) + T_s \dot{x}_4(k) = f_4(\cdot) \quad (6.13)$$

Derivasjonene for å finne transisjonsmatrisen er benyttet MATLAB, se kode på vedlegg C.1 Dette gir transisjonsmatrise:

$$\Phi(k) = \begin{bmatrix} \Phi_{11}(k) & 0 & 0 & \Phi_{14}(k) \\ \Phi_{21}(k) & \Phi_{22}(k) & \Phi_{23}(k) & \Phi_{24}(k) \\ 0 & 0 & \Phi_{33}(k) & 0 \\ 0 & 0 & 0 & \Phi_{44}(k) \end{bmatrix} \quad (6.14)$$

Uttrykket for elementene i transisjonsmatrisen finnes i vedlegg B.1.

6.3 Observerbart system

For at Kalmanfilteret skal klare å estimere denne tilsatsen, er det nødvendig å sjekke for observerbarheten. Et system er observerbart, dersom observerbarhetsmatrisen Q_{obs} gitt ved [5]:

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

har rang n , hvor n er antall tilstander (systemets orden).

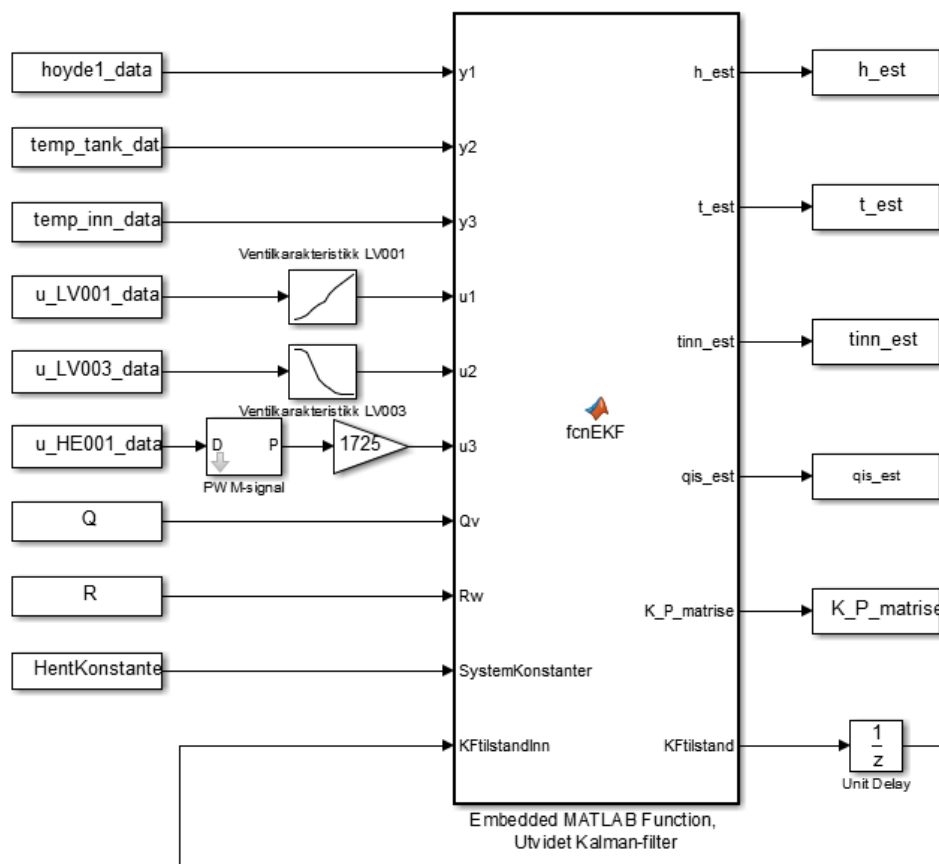
Som kjent har systemet i dette kapitlet $n = 4$ tilstander og målematrisen D har dimensjon $l \times n$, hvor l er antall målinger. Det er nivå og temperatur i tanken som måles, dermed vil $l = 2$. Målematrisen uttrykkes følgende:

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6.16)$$

Ved beregning av observerbarhetsmatrisen er det brukt MATLAB, siden observerbarhetsmatrisen har dimensjon 8×4 med en god del ledd i mange av elementene, som gjør det vanskelig å ha det med i rapporten. Koden for beregning av observerbarhetsmatrisen og rangen finnes på vedlegg C.2. Systemet har $rang = 4$, som vil si at systemet er observerbart.

6.4 Implementering i Simulink

Implementering av det nye Kalmanfilteret tar i utgangspunktet å bruke EKF som ble konstruert fra kapittel 4. ved å legge til den nye matematiske modellen, transisjonsmatrisen og utvide størrelsene til matrisene i EKF algoritmen. Utskrift av koden i funksjonsblokken finnes på vedlegg C.4. Figur 6.1 viser funksjonsblokken med pådrag og målinger som inngang, samt de opprinnelige estimatene og det nye estimatet som utgang.



Figur 6.1: EKF funksjonsblokk implementert i Simulink

Med utgangspunkt i figur 6.1 settes det opp en tabell med oversikt og forklaring over notasjonen brukt her, som i prinsippet er den samme tabellen som i kapittel 4.2 utenom den nye tilstanden.

Tabell 6.1: Inn- og utganger av EKF funksjonsblokk

Inngang/utgang	Beskrivelse
y1	Måling av nivå i tanken.
y2	Måling av temperatur i tanken.
y3	Måling av temperatur fra vannbatteriet. Denne målingen brukes normalt ikke. Brukes kun som initialtilstand.
u1	Tabelloppslag av ventilkarakteristikk for ventil LV001, hvor det er ventilåpning som er inngangen her og ikke pådraget til ventilen.
u2	Tabelloppslag av ventilkarakteristikk for ventil LV003, hvor det er flyt i ventil LV003 som er inngang.
u3	PWM-signal av effekten som inngang, og ikke selve pådraget til varmeelementet.
Qv	Tuningsparameter, Q.
Rw	Tuningsparameter, R.
KFtilstandInn	Forrige tilstandsverdier av Kalmanfilteret.
h_est	Estimat av nivå i tanken (tilstand).
t_est	Estimat av temperatur i tanken (tilstand).
tinn_est	Estimat av temperatur inn i tanken (tilstand).
qis_est	Estimat av tilsats vannvolumet (tilstand).
K_P_matrise	Verdier av Kalmanfilterforsterkningsmatrise og kovariansmatrise for estimeringsavvikene.
SystemKonstanter	Henter konstanter fra en fil.
KFtilstand	Tilstanden for Kalmanfilteret som er lagre fra et steg til neste.

6.5 Test av EKF med tilsatsestimering

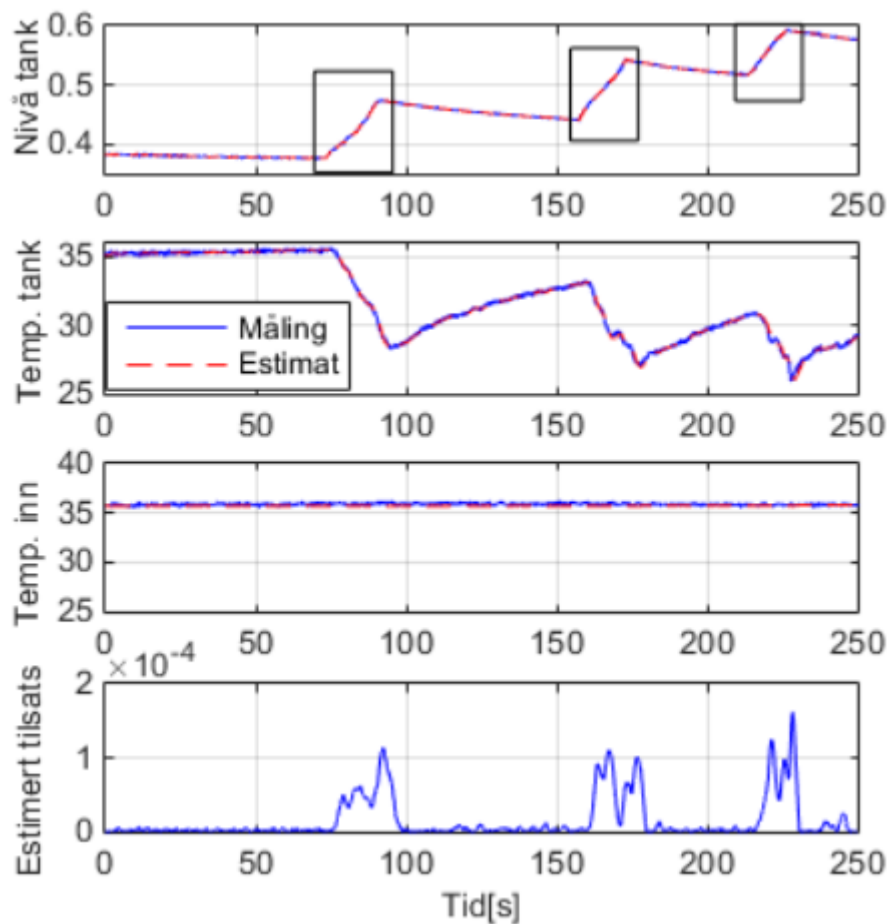
Testen som ble utført i kapittel 5 på figur 5.10, der EKF basert på feilaktig modell ikke virket når prosessen ble utsatt for ekstern tilsats av vann. Den samme testen skal nå testes med det nye Kalmanfilteret. Kalmanfilteret tar fortsatt i utgangspunktet i de samme tuningsparametre funnet i kapittel 5, men

tuningsparameter Q er utvidet med et diagonalelement. Tuningsparametre for forsøket i figur 6.2 er gitt følgende:

$$R = \begin{bmatrix} 1 \cdot 10^{-6} & 0 \\ 0 & 0.01 \end{bmatrix} \quad (6.9)$$

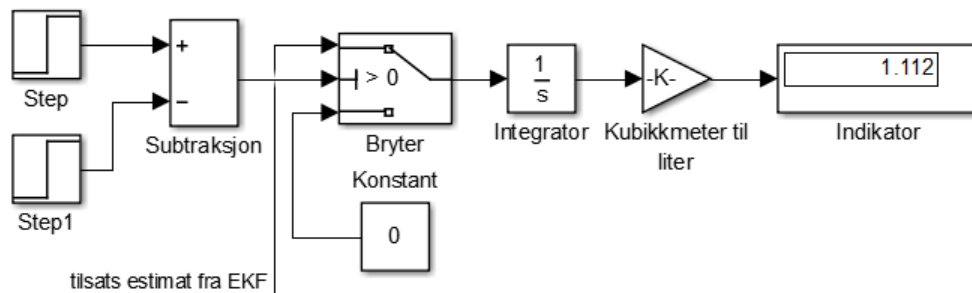
og

$$Q = \begin{bmatrix} 1 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 2 \cdot 10^{-7} & 0 & 0 \\ 0 & 0 & 2 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-12} \end{bmatrix} \quad (6.10)$$



Figur 6.2: Test av filter med tilsats av iskaldt vann, konstant temperatur inn og pådrag

Som det viser på figur 6.2 gir ikke Kalmanfilteret feil estimat av temperaturen inn i dette forsøket. Det går også frem at estimert tilsats har lave verdier i periodene hvor det ikke er forkommer ekstra tilsats, og viser et godt estimat når vannet blir tilført i de tre markerte tidspunktene. Figur 6.3 viser kalkulert tilsats i liter for det siste markerte tidspunktet, som også ble den høyeste estimerte verdien av de tre.



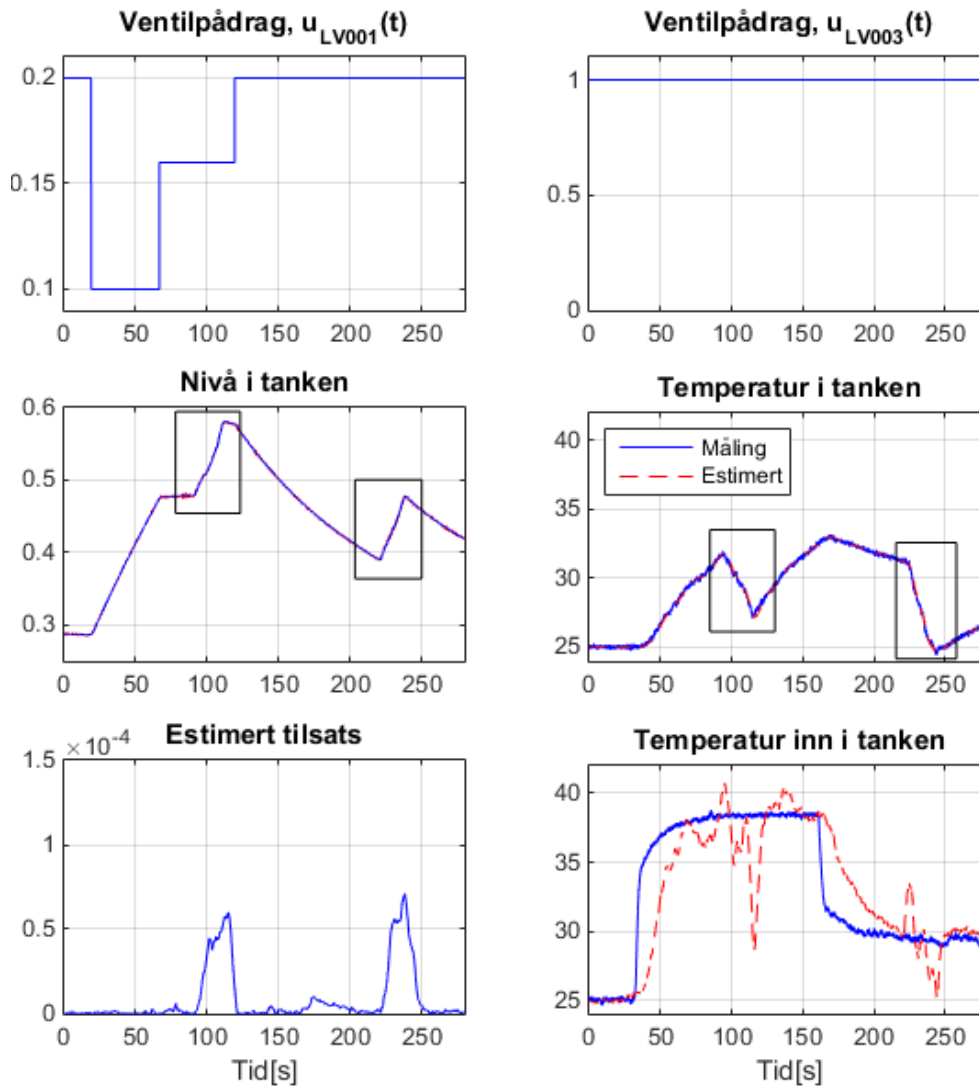
Figur 6.3: Beregner areal under grafen

Dette er gjort ved å kjøre estimert tilsatssignal fra Kalmanfilteret gjennom en integrator for beregning av areal under grafen. Bestemmelsen av når integratoren skal starte og stoppe integrasjonen er gjort ved å lese av start og stopp tidspunktene på den nederste grafen på figur 6.2. Verdiene legges manuelt inn i steg blokkene for triggering av start og stopp ved hjelp av en bryter. Det går frem på figur 6.3 at Kalmanfilteret estimerer litt mer enn det som er tilført i virkeligheten, som i grunn kan skyldes på modellfeil, men tilsatsestimateret er også avhengig av størrelsen på tuningsparametre Q og R.

Videre skal Kalmanfilteret testes mot den virkelige prosessen når både pådrag og temperaturen inn varierer, og samtidig tilføres det iskaldt vann i tanken. Etter å ha testet Kalmanfilteret med tuningsparameter i ligning (6.10), viser det seg at Kalmanfilteret ikke er egnet til denne bruken. Dermed ble andre tuningsparametre valgt. Autokovarians matrise for målestøy R er uendret, mens Q ble følgende:

$$Q = \begin{bmatrix} 1 \cdot 10^{-7} & 0 & 0 & 0 \\ 0 & 2 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 5 \cdot 10^{-3} & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-13} \end{bmatrix} \quad (6.11)$$

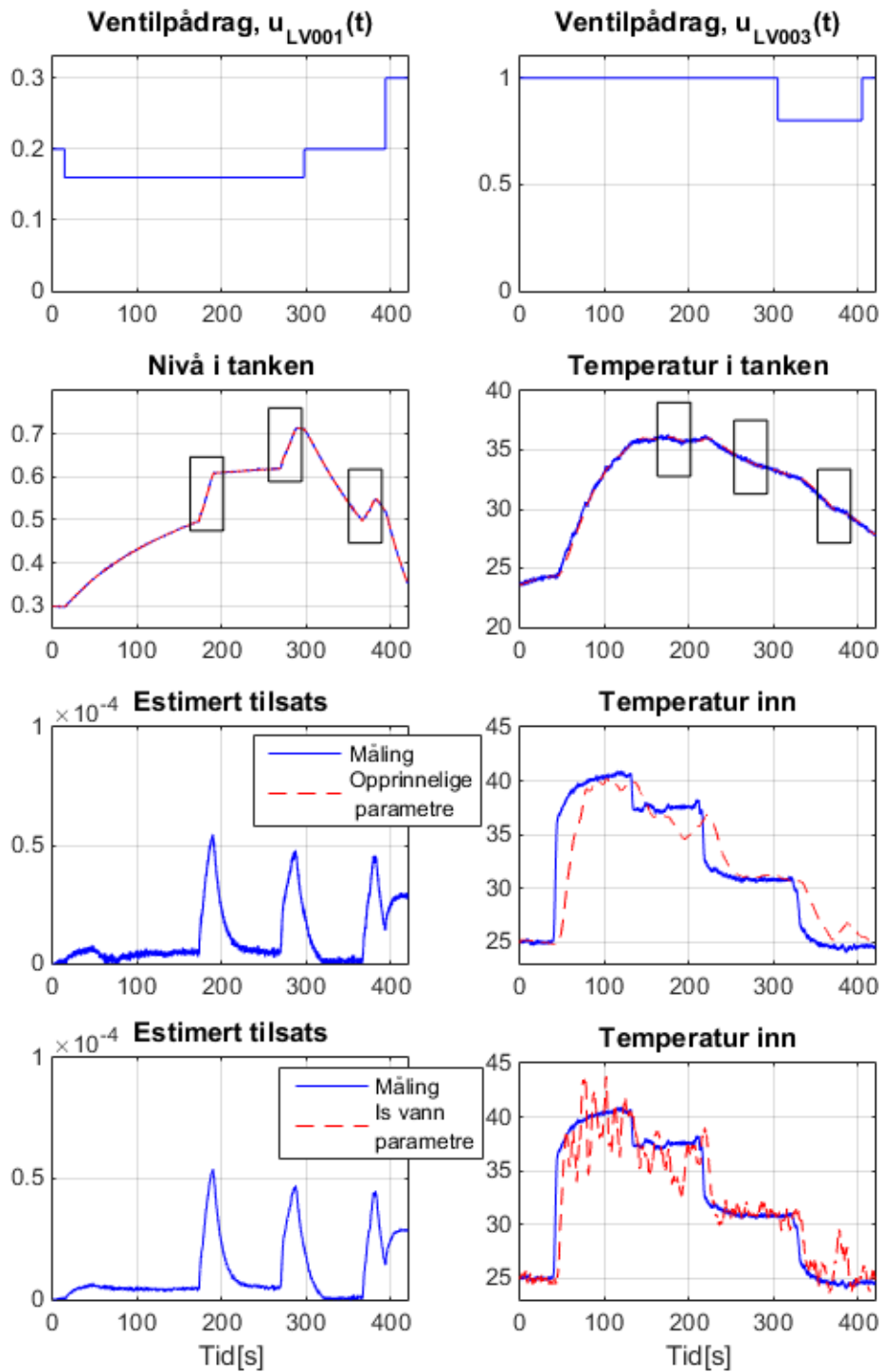
Figur 6.4 viser resultatet av denne testen med ny Q.



Figur 6.4: Test av filter med tilsats av iskaldt vann, variert temperatur inn og pådrag

Ut ifra figuren over gir Kalmanfilteret et relativt bra estimat av temperaturen inn i periodene når prosessen ikke utsatt for tilsats av iskaldt vann. I de to markerte tidspunktene viser det at estimatet har mye påvirkning av det som skjer i tanken, som gjør at estimatet får enten en kraftig økning på grunn av raskt nivå økning, eller en kraftig reduksjon på grunn av rask temperatur reduksjon i tanken. Det går også frem at Kalmanfilteret henter seg kjapp inn igjen etter tilføring av ekstra tilsats er sluttet.

Videre ønskes det å teste Kalmanfilteret med tilført vann som har samme temperatur som vannet i tanken. Dette er gjort ved å fylle opp en litermåler med vann som strømmer ut av ventil LV001. Figur 6.5 viser resultatet for forsøket med opprinnelige tuningsparametre (6.10) og den nye som ble valgt for iskaldt vann (6.11).

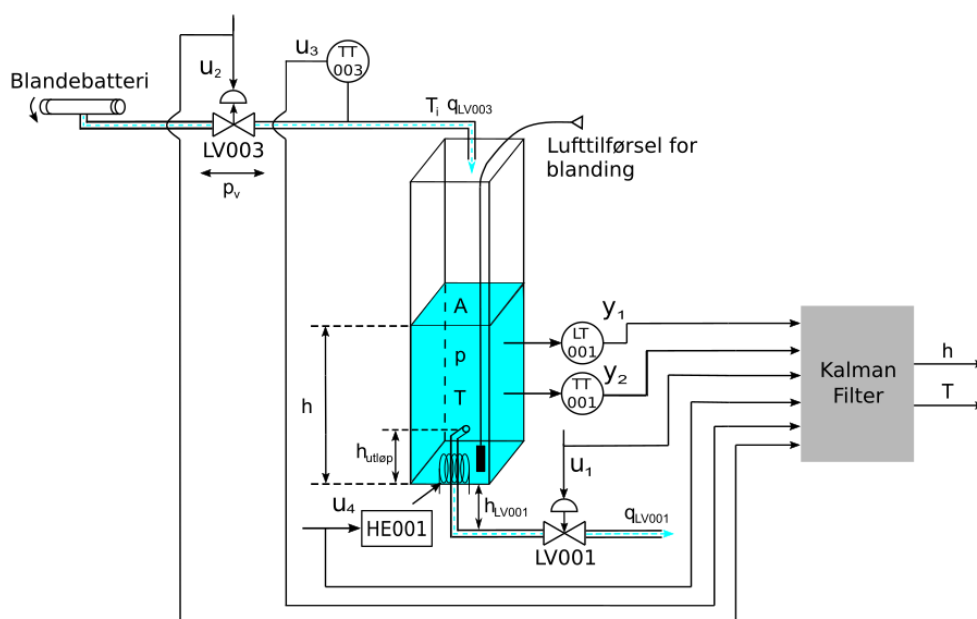


Figur 6.5: Forsøk hvor tilsats av vann har samme temperatur som vannet i tanken med både opprinnelige parametre og parametre som er egnet til is vann

Figur 6.5 viser at de opprinnelige tuningsparametre gir et bedre estimat enn tuningsparametre for iskaldt vann. Opprinnelig tuningsparametre gir et estimat som er glattere og mindre støyfull, men det tar til gjengjeld litt lengre tid til å svinge seg inn. Tuningsparametre for iskaldt vann oppdager temperaturendringen rimelig kjapp, men er altfor støyfullt. Dette er rimelig siden variansen Q_{22} og Q_{33} i ligning (6.11) er mye større enn ligning (6.10). Så valgt av tuningsparametre er en viktig ting. Et sett med tuningsparametre passer ikke til alle driftssituasjoner slik resultatene har vist seg. For å finne tuningsparametre som kan gi et greit resultat for begge forsøkene i figur 6.4 og 6.5, kan for eksempel Q-matrisen velges et sted mellom ligningene (6.10) og (6.11). Dette er opptil hvordan en ønsker at Kalmanfilteret skal oppføre seg.

7 EKF for estimering av temperatur i tanken

Kapittelet tar for seg å konstruere et EKF for å estimere temperaturen i tanken baseres på modellen i kapittel 3, implementere i Simulink og teste EKF mot den virkelige prosessen. Systemet som er gitt i dette kapittelet er ikke observerbart. Ved å anta at EKF har tilgang til måling av temperatur i tanken en gang i blant, gir dette et observerbart system ved dette tidssteget. Hensikten er å studere hvordan EKF fungerer i de tidsrommene hvor det ikke blir foretatt temperaturmåling.



Figur 7.1: Enkel prinsippskisse (uten støy)

Her estimeres temperatur i tanken, mens nivået i tanken måles direkte. Disse er tilstander i systemet. Pådragene blir som i kapittel 4, men nå blir vanntemperatur ved innløpet $T_i(t)$ også et pådrag.

7.1 Implementering av EKF

Implementering av Kalmanfilteret i dette delkapittelet er det tatt i bruk modellen i kapittel 3, ligning (3.9) og (3.17).

Transisjonsmatrisen for modellen har dimensjon 2×2 , siden det er nivå og temperatur i tanken som estimeres. Elementene i transisjonsmatrisen er i grunn funnet i kapittel 4.1.2, ligning (4.20), (4.21), (4.23) og (4.24). Transisjonsmatrisen for modellen er da gitt:

$$\Phi(k) = \begin{bmatrix} \Phi_{11}(k) & 0 \\ \Phi_{21}(k) & \Phi_{22}(k) \end{bmatrix} \quad (7.1)$$

Det er en endring i ligning (4.23), hvor temperaturen $T_i(t)$ er nå et pådrag og ikke en tilstand.

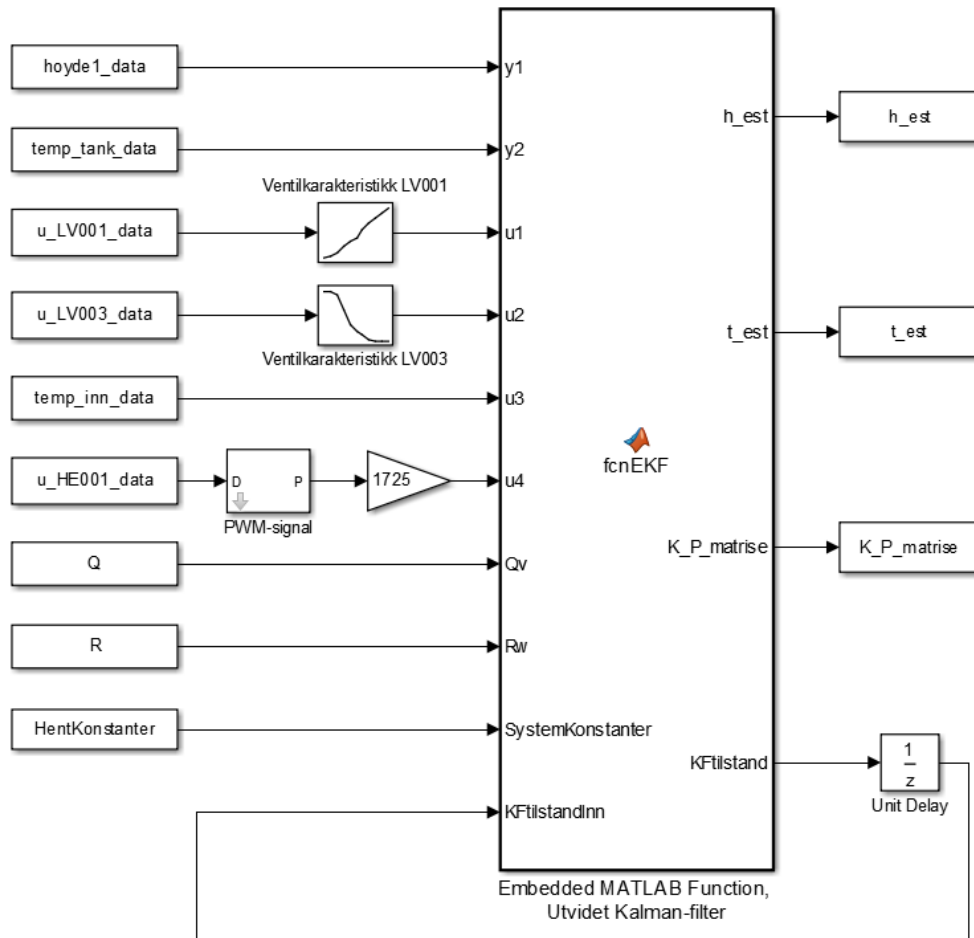
Målematrisen vil her være en 2×2 matrise og forveksler mellom to tilstander. I periodene hvor Kalmanfilteret ikke har tilgang til temperaturmåling i tanken er målematrisen gitt ved:

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (7.2)$$

mens ved tidsstegene hvor Kalmanfilteret har tilgang til temperaturmåling, blir målematrisen:

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (7.3)$$

Figur 7.2 viser funksjonsblokken av Kalmanfilteret med inn- og utganger. Utskrift av koden i funksjonsblokken finnes på vedlegg C.5.



Figur 7.2: EKF funksjonsblokk implementert i Simulink

7.2 Test av EKF

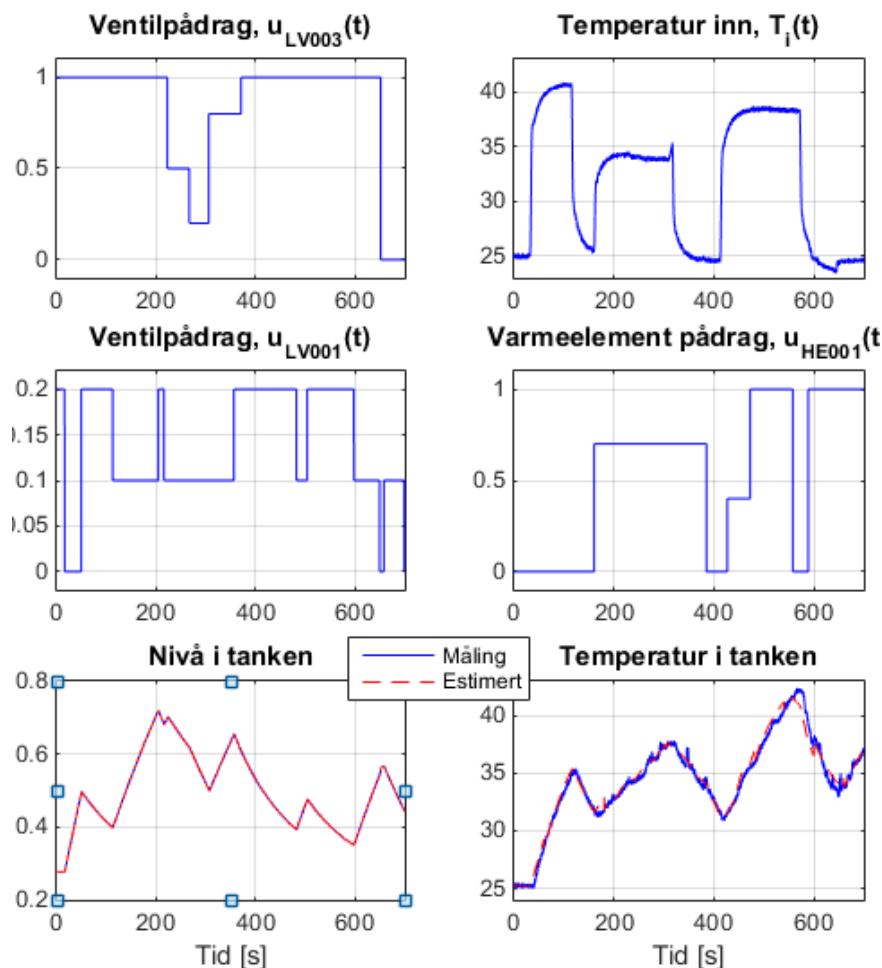
Forsøket i dette kapittelet er tatt utgangspunktet i målestøy kovariansen til nivåmåleren som ble funnet i kapittel 5.1, gitt ved:

$$R = \sigma^2 = (1 \cdot 10^{-3})^2 = 1 \cdot 10^{-6} \quad (7.4)$$

og prosessstøy kovariansmatrise

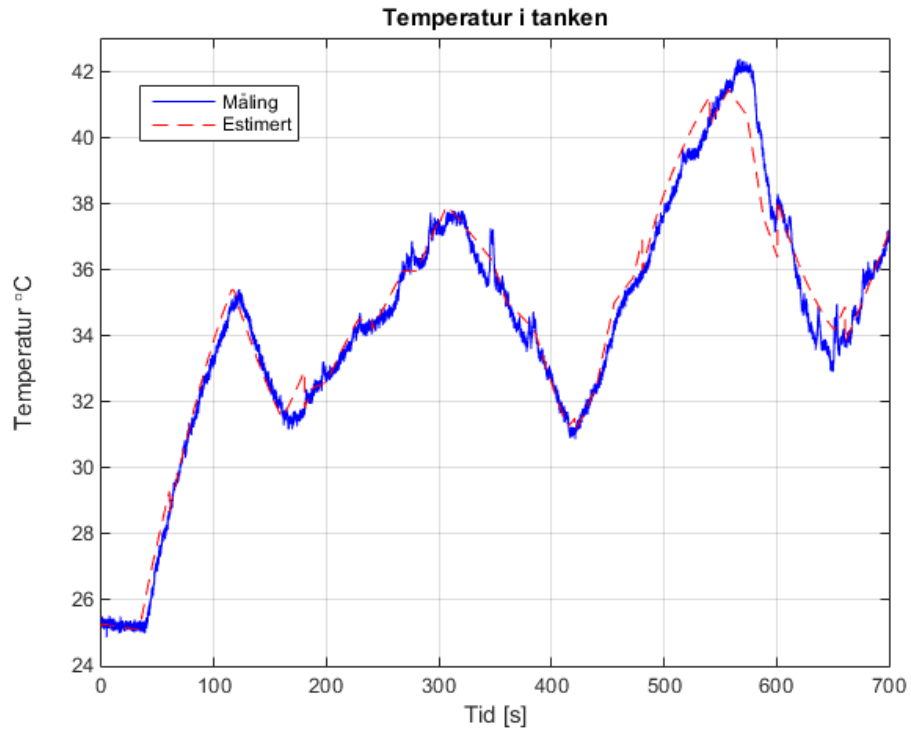
$$Q = \begin{bmatrix} 1 \cdot 10^{-5} & 0 \\ 0 & 1 \cdot 10^{-5} \end{bmatrix} \quad (7.5)$$

Figur 7.3 viser hvordan Kalmanfilteret svinger seg inn når startverdier er kjent, det vil si både temperatur og nivået i tanken måles ved første tidssteget. Forsøket er basert på at Kalmanfilteret har tilgang til en måling hvert minutt.



Figur 7.3: Innsvingning av Kalmanfilteret basert på en prøvemåling hvert minutt

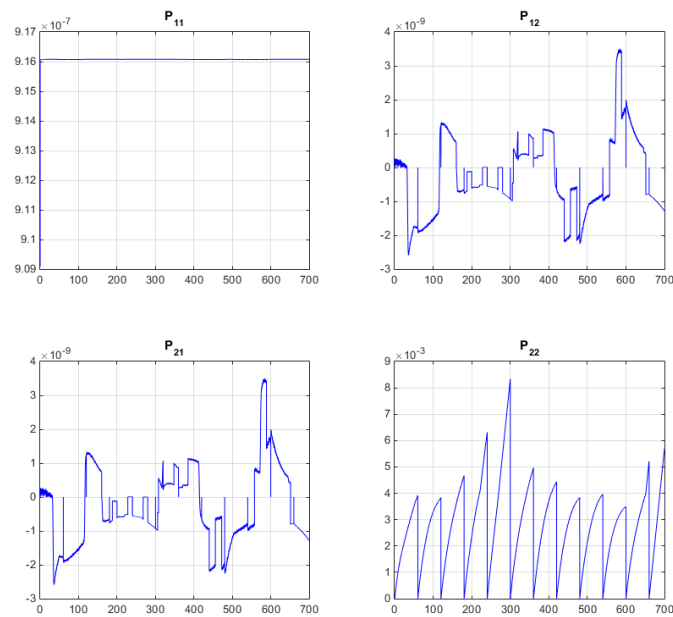
Som det går frem i figur 7.3 følger Kalmanfilteret temperaturmålingen ganske bra ved å ha tilgang til en prøvemåling hvert minutt. I tidsrommene hvor systemet ikke er observerbart, følger Kalmanfilteret målingen likevel bra. Dette tyder på at modellen som ble funnet i kapittel 3 er god. Figur 7.4 viser et mer detaljert av temperaturestimatet.



Figur 7.4: Estimert temperatur i tanken

7.2.1 P-matrise

Figur 7.5 viser hvordan diagonalelementene i P-matrisen utvikler seg over tid ved forsøket i figur 7.2.

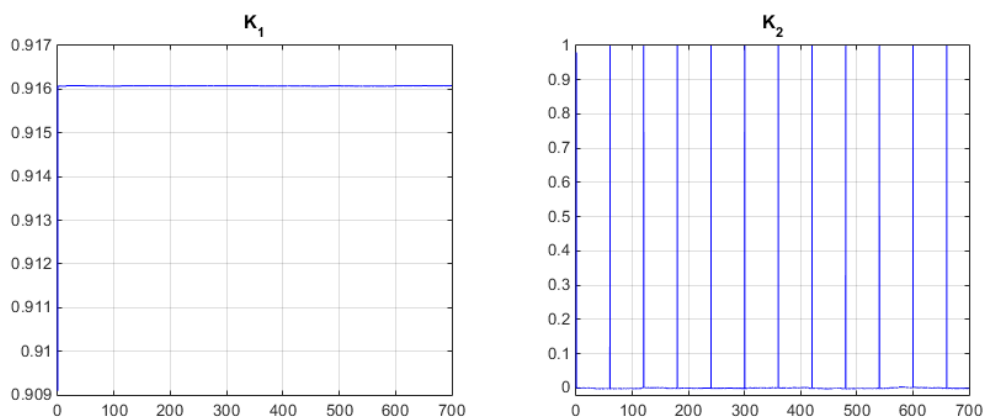


Figur 7.5: Kovariansmatrise for a posteriori estimeringsavvik

Som det viser på figuren, er P_{11} tilnærmet lik konstant og er rimelig lavt. Dette er ganske rimelig siden det er nivå i tanken som måles direkte. Det som gjør at P_{22} har verdi null innimellom er på grunn av at Kalmanfilteret får tilgang til målingen hvert minutt. Ved disse tidsstegene har temperaturestimatet den samme verdien som målingen, og etter hvert øker usikkerheten igjen.

7.2.2 Kalmanfilterforsterkning

Figur 7.6 viser utviklingen til Kalmanfilter forsterkningene, K_1 og K_2 .



Figur 7.6: Kalmanfilter-forsterkning

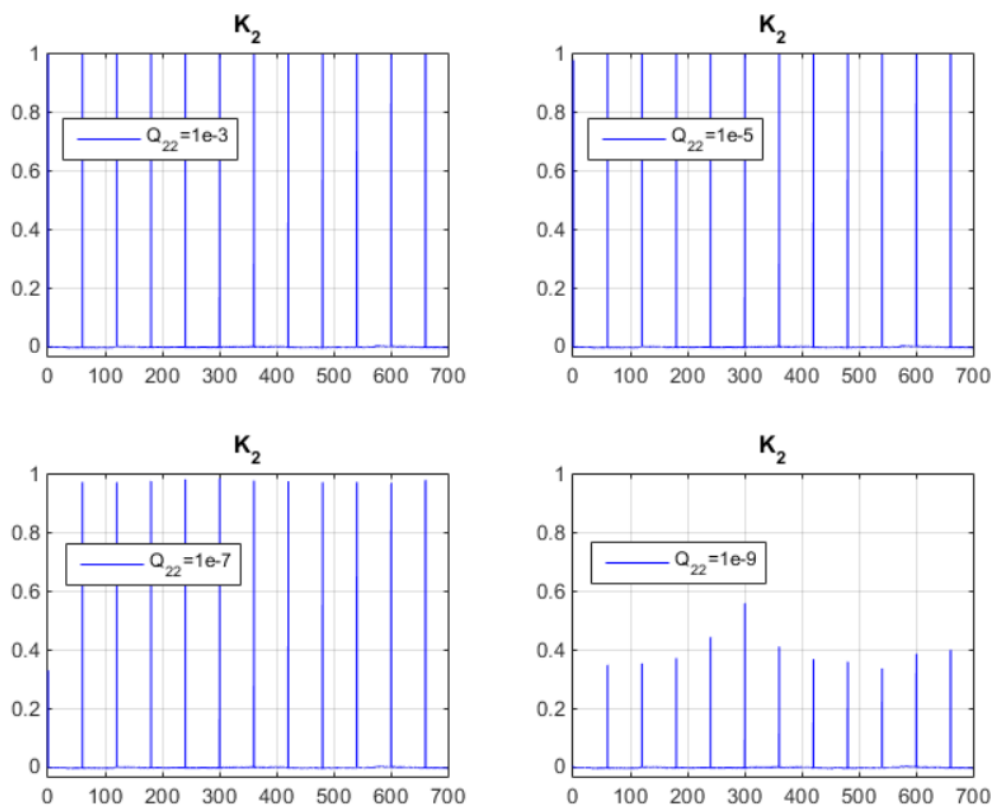
Figur 7.6 viser at Kalmanfilter forsterkningen, K_1 er stor og er tilnærmet konstant. Som sagt, tar Kalmanfilteret i bruk av en måling hvert minutt, som fører til at K_2 får maks utslag ved disse tidsstegene. I tidsrommene hvor Kalmanfilteret ikke måler temperaturen, er K_2 meget lavt. Dette skyldes på at systemet i disse tidsrommene ikke er observerbart. Når et system er ikke observerbart, gir det problemer med å beregne Kalmanfilter forsterkningen, som fører til at tilstandsestimatet ikke korrigerer for avviket mellom estimat og virkelig verdi [5].

Nå når K_2 er så lav og er tilnærmet null, vil ligningen for det aposteriori tilstandsestimatet i EKF-algoritmen:

$$\hat{x}(k) = \bar{x}(k) + K(k) \cdot (y(k) - \bar{y}(k)) \quad (7.6)$$

bare være basert på det apriori tilstandsestimatet $\bar{x}(k)$.

Figur 7.7 viser hvordan forsterkningen K_2 endre seg over tid ved forskjellige tuningsparameter Q_{22} .



Figur 7.7: Kalmanfilter-forsterkning

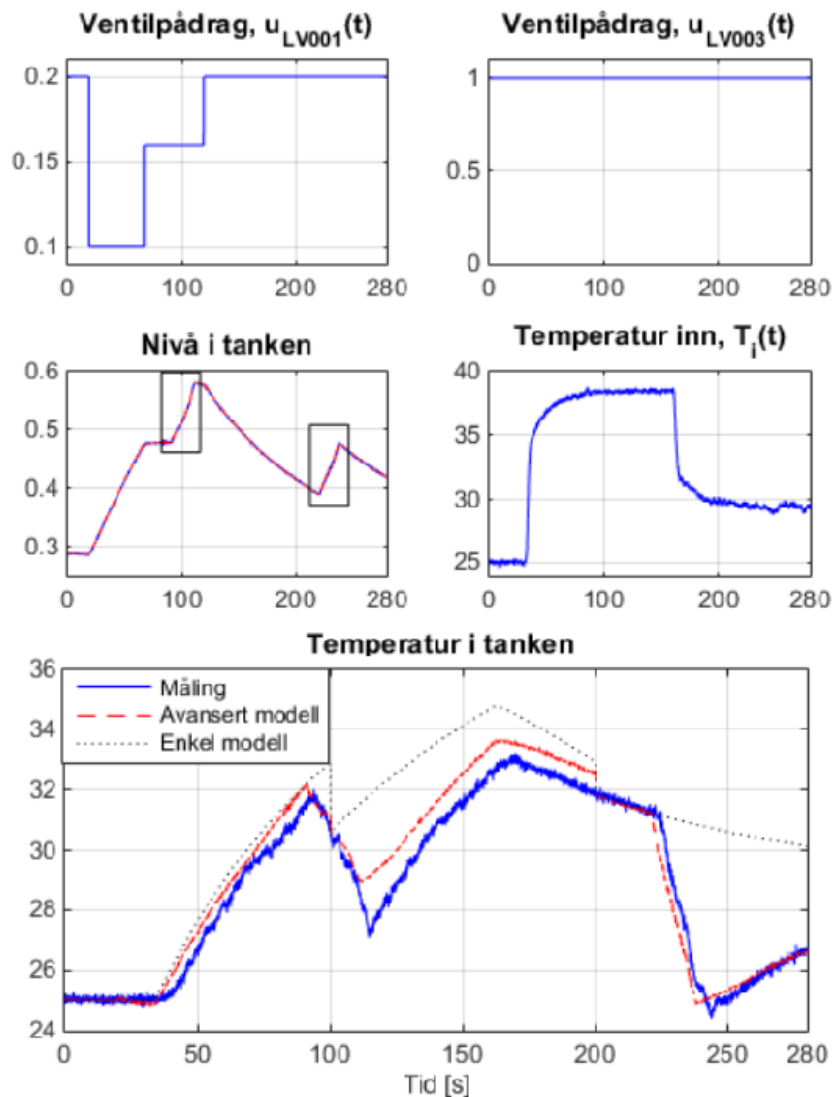
Som det går frem på figur 7.7, skjer det lite endringer i forsterkningen i disse tidsrommene selv om Q_{22} øker. En kan registrere at K_2 minker ved tidsstegene hvor det forekommer en måling når verdien for Q_{22} er mindre, og dette er som forventet. Kalmanfilter i disse tidsrommene kan uttrykkes som en ballistisk tilstandsestimator, dersom en antar at Kalmanfilter forsterkningen er null.

7.3 Test av EKF med ekstra tilsats

Videre ønskes det å teste Kalmanfilteret som ble implementert delkapittel 7.1 mot ekstra tilsats av iskaldt vann, slik det ble gjort i kapittel 6.5.

Kalmanfilteret blir her også utvidet for estimering av ekstra tilsats. Modelleringen gjøres på tilsvarende måte som i kapittel 6. Koden for det nye Kalmanfilteret finnes på vedlegg C.6 og transisjonsmatrisen på vedlegg B.3.

Figur 7.8 viser hvordan Kalmanfilteret i kapittel 7.1 og det nye Kalmanfilteret yter når prosessen er utsatt med ekstra tilsats av iskaldt vann. Testen er basert på at filtrene har tilgang til en måling hver 100 sekund, og tar i utgangspunktet i de samme loggede data av den virkelige prosessen som ble testet i kapittel 6.5 på figur 6.4.



Figur 7.8: Test av filtrene med ekstra tilsats

Figuren viser at filtrene har tilgang til målingen ved første tilsats, men ikke andre. Det går frem at det nye Kalmanfilteret gir et godt estimat over hele tidsperioden. Kalmanfilteret fra kapittel 7.1 gir kun et godt estimat før prosessen er utsatt for tilsatsen, og dette er som forventet siden Kalmanfilteret ikke er modellert slik at den skal ha evnen til å ta hensyn til denne tilsatsen.

8 Konklusjon

I kapittel 3 er en matematisk modell laget og implementert i Simulink for prosessanlegget på figur 1.2. Etter å ha verifisert modellen med virkelig data, viser resultatene at modellen passer bra med prosessen, selv om varmeelementet har bidratt til å skape temperaturforstyrrelse i tanken. Temperaturforstyrrelsene skyldte på dynamikken i varmeelementet og kort avstand av plassering av temperaturmåler og varmeelementet. Dette er noe som har medført til at avviket mellom EKF estimat for temperatur ved innløpet av tanken og sann verdi har fått merkbart økning når varmeelementet slår seg på eller av. Likevel viser resultatene at estimeringen med EKF fungerer bra med både kjent og ukjent initialverdi. Kalmanfilter for tilsatsestimering ga en god indikasjon på når prosessen ble utsatt for ekstern tilsats, og samtidig ga et tilfredsstillende estimat for temperaturen ved innløpet.

Resultatene fra kapittel 7 viser Kalmanfilter for det ikke observerbare systemet ikke klarer å beregne Kalmanfilter forsterkningen i intervallene hvor det ikke blir tatt temperaturmåling. Dermed korrigerer ikke Kalmanfilteret for avvik mellom måling og det apriori tilstandsestimatet. En kan konkludere med at Kalmanfilteret er da kun basert på det apriori tilstandsestimatet i disse intervallene.

Referanser

- [1] Jan Erik Qvann, «Kalmanfilter anvendt på prosessanlegg,» Masteroppgave, Universitet i Stavanger, 2007.
- [2] Atle Gjengedal, «Sammenligning av EKF- og UKF- kalmanfilter anvendt på totankprosess,» Masteroppgave, Universitet i Stavanger, 2009.
- [3] R. E. 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] 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.
- [5] Finn Haugen, «State estimation with Kalman Filter,» i *Advanced Dynamics and Control*, Skien, TechTeach, August, 2010, pp. 215-239.
- [6] Tormod Drengstig, «totank1 modellverifisering,» <http://www.ux.uis.no/~karlsk/ELE620/ELE620.html>, 2008.
- [7] Svein Erik Stokka og Rune Bjerland Risanger, «Regulering av totankprosessen,» Bacheloroppgave, Universitet i Stavanger, 2016.
- [8] Bjørnar Langeland, «Forholdsregulering av tilsetningsstoff i drivstoff,» Masteroppgave, Universitet i Stavanger, 2010.
- [9] Finn Haugen, *Reguleringsteknikk*, Tapir Akademisk Forlag, 2012.
- [10] S. Kolås, B.A Foss, T.S. Schei, «Constrained nonlinear state estimation based on the UKF approach,» *Computers and Chemical Engineering* 33 (2009) 1386–1401, vol. 33, p. 1386–1401, 2009.
- [11] Simon J. Julier og Jeffrey K. Uhlmann, «Unscented Filtering and Nonlinear Estimation,» *Proceedings of the IEEE*, vol. 92(3), Mars 2004.

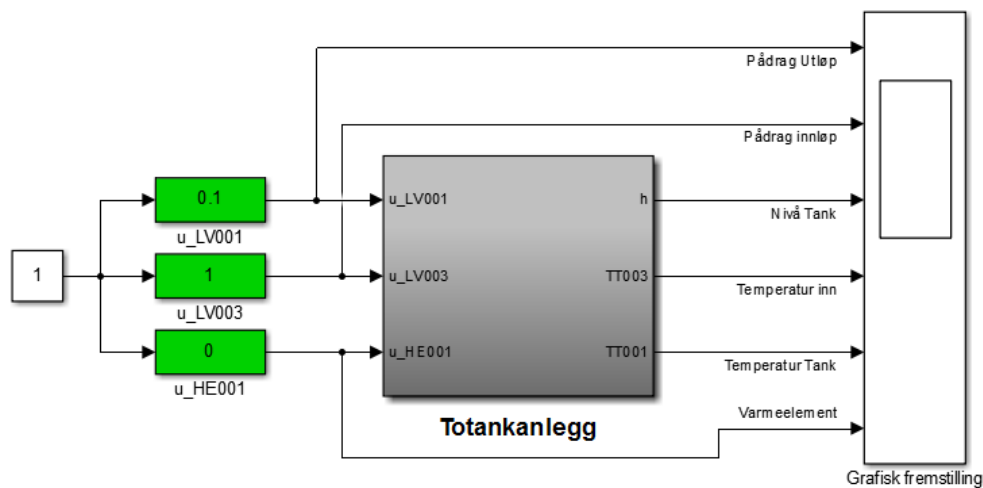
- [12] Karl Skretting, «notat3 - Systemidentifikasjon,»
<http://www.uv.uio.no/~karlsk/ELE620/ELE620.html>, 2016.
- [13] Karl Skretting, «notat1 - Systemidentifikasjon,»
<http://www.uv.uio.no/~karlsk/ELE620/ELE620.html>, 2016.
- [14] Greg Welch and Gary Bishop, «An introduction to the Kalman Filter,»
Technical report, Department of Computer Science University of North
Caroline at Chapel Hill, 2006.
- [15] Gordon N. J., Salmond D. J. og Smith A. F. M., «Novel approach to
nonlinear/non-Gaussian Bayesian state estimation,» IEE Proceedings F
- Radar and Signal Processing, vol. 140, pp. 107–113, April 1993.

Vedlegg

A

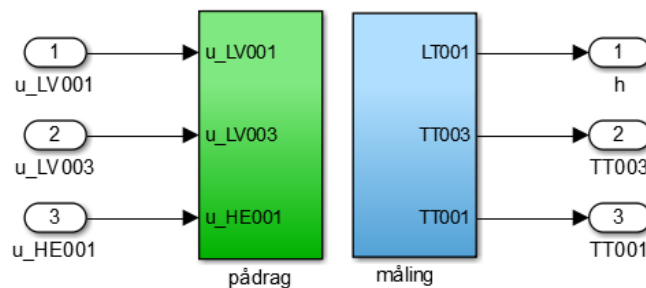
A.1 Kjøring av totankanlegget

Figur A.1.1 viser kjøring av totankanlegget for logging av data.



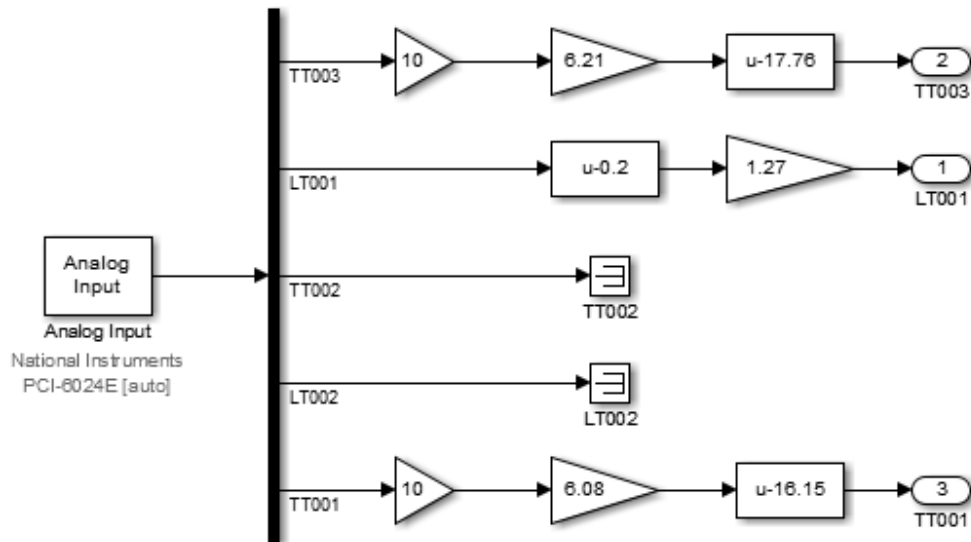
Figur A.1.1: Kjøring av totankanlegget

Figur A.1.2 viser inn- og utganger som er blitt benyttet. Ved å dobbeltklikke på den grønne eller den blåe blokken, dukker det opp figurene A.1.3 og A.1.4.



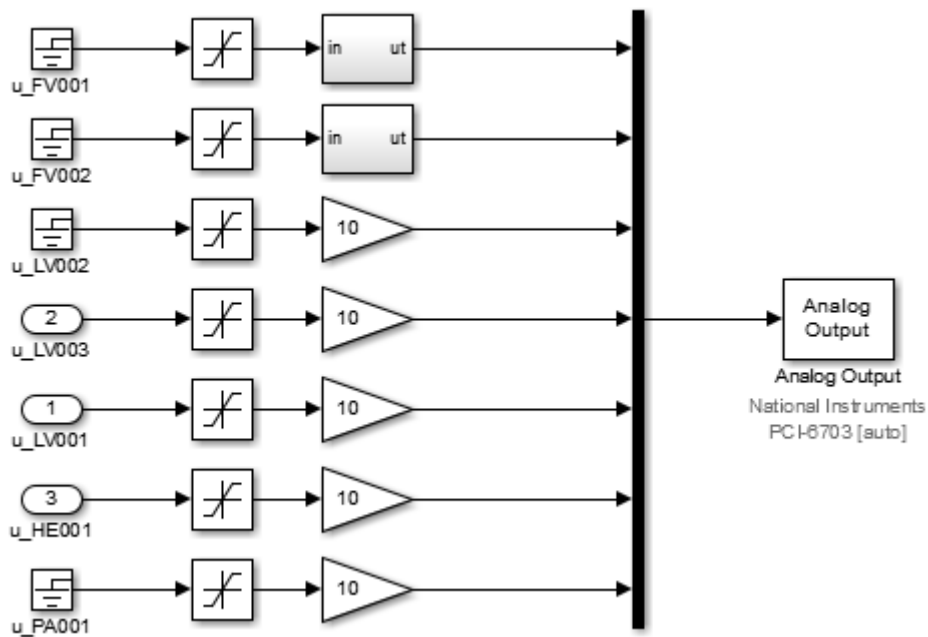
Figur A.1.2: Måling – analog inngang (PCI-6024E) og pådrag – analog utgang (PCI-6703)

Figur A.1.3 viser analoge signaler som blir konvertert fra spenning til nivå/temperatur.



Figur A.1.3: Analoge innganger

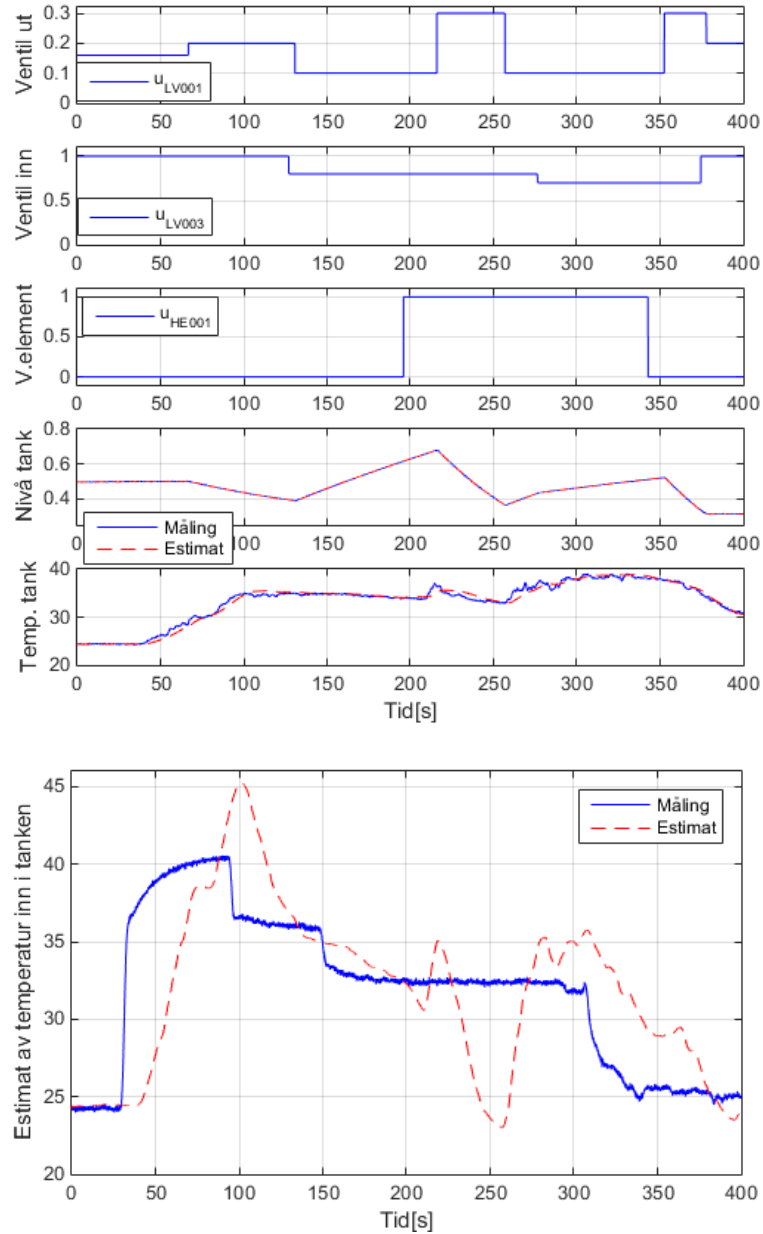
Figur A.1.4 viser konvertering av pådrag signal (0-1) til spenning (0-10).



Figur A.1.4: Analoge utganger

A.2 Test av EKF uten blanding

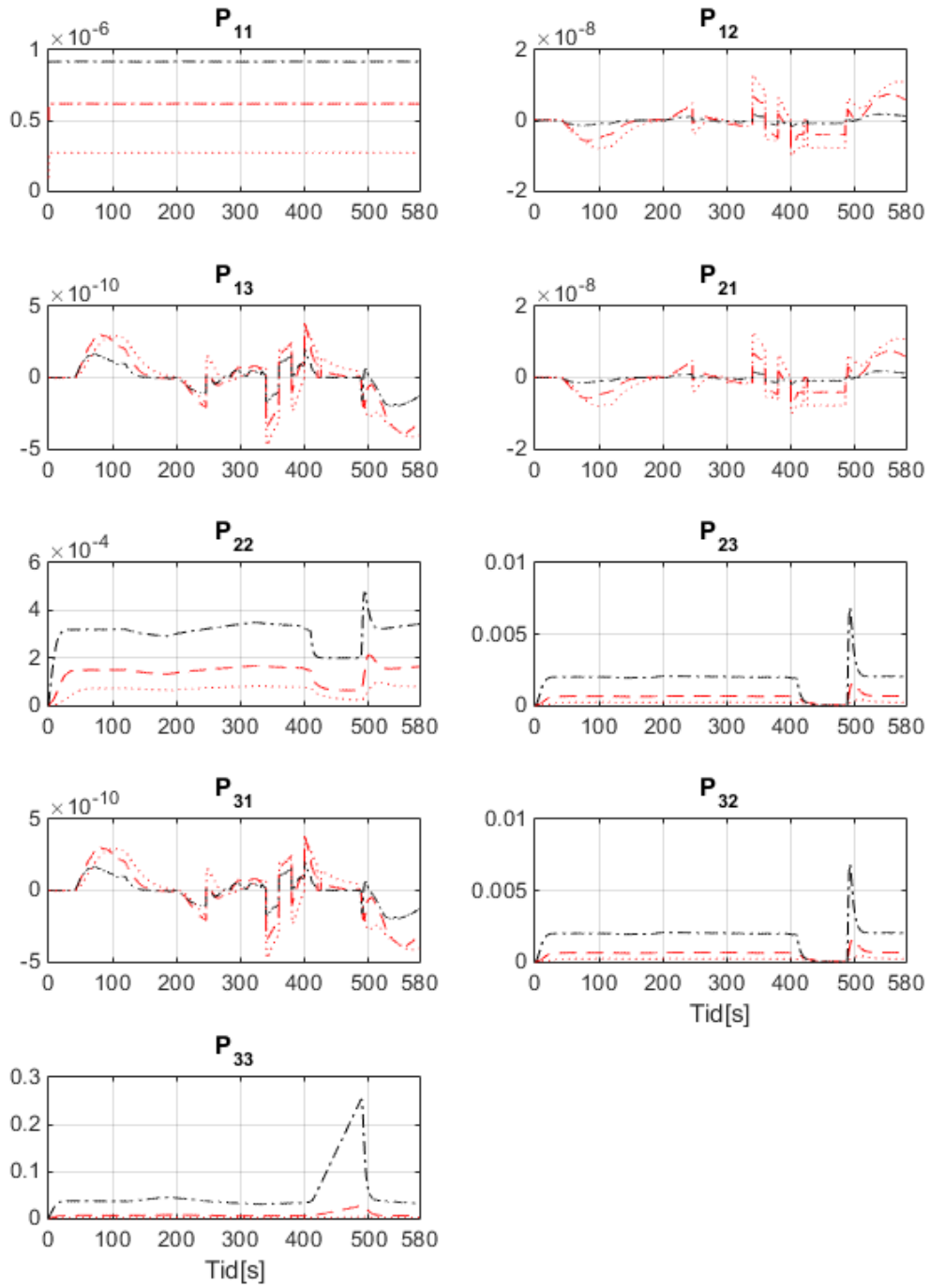
Figuren nedenfor viser et forsøk hvor det er ikke benyttet blanding i tanken.



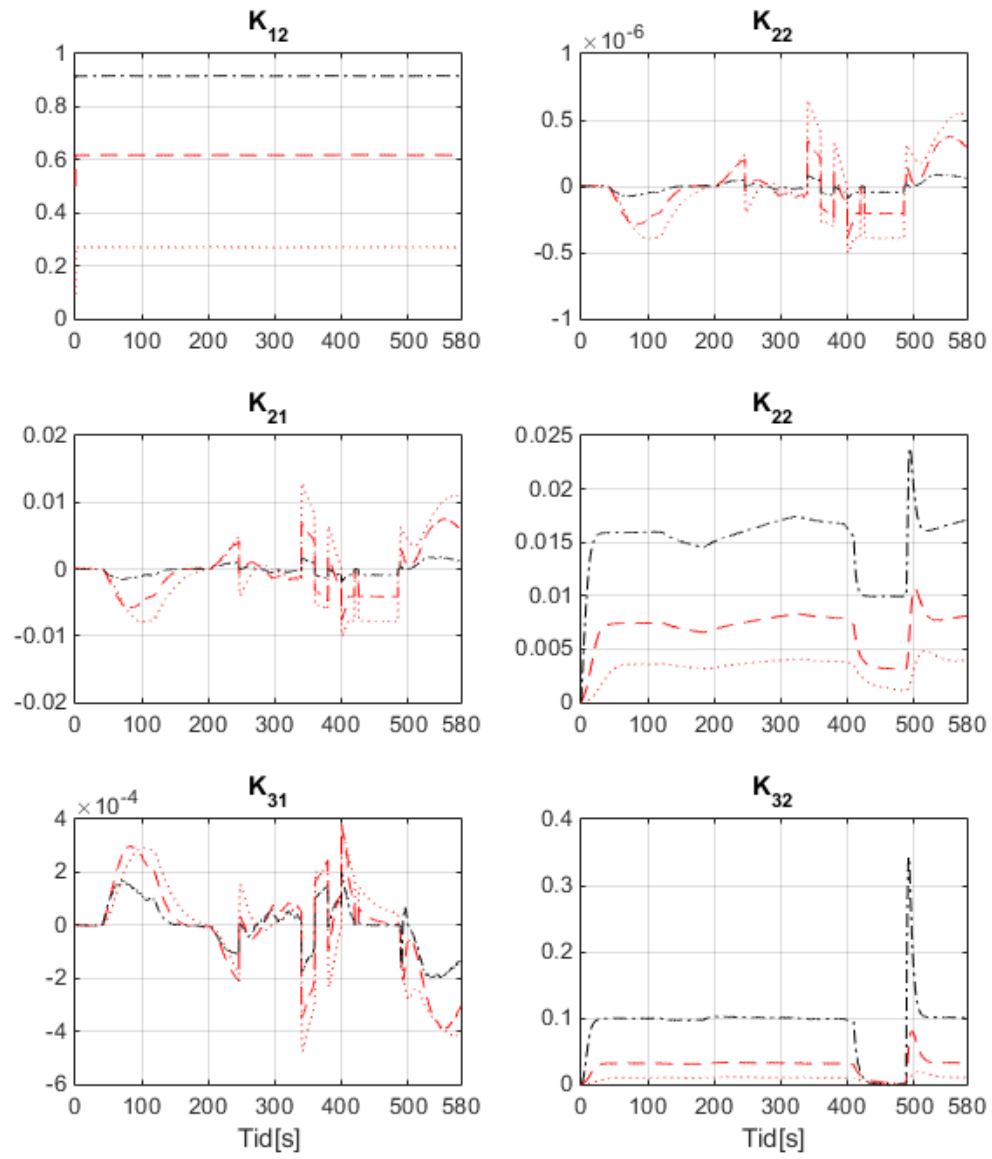
Figur A.2: Ingen blanding

Som det går frem på figuren, gir ikke EKF et brukende estimat.

A.3 P-matrise



A.4 Kalmanfilter forsterkning



B

B.1 Transisjonsmatrise (kapittel 6)

Transisjonsmatrisen uttrykkes følgende når det er fire tilstander som estimeres:

$$\Phi(k) = \begin{bmatrix} \frac{\partial f_1(\cdot)}{\partial x_1} & \frac{\partial f_1(\cdot)}{\partial x_2} & \frac{\partial f_1(\cdot)}{\partial x_3} & \frac{\partial f_1(\cdot)}{\partial x_4} \\ \frac{\partial f_2(\cdot)}{\partial x_1} & \frac{\partial f_2(\cdot)}{\partial x_2} & \frac{\partial f_2(\cdot)}{\partial x_3} & \frac{\partial f_2(\cdot)}{\partial x_4} \\ \frac{\partial f_3(\cdot)}{\partial x_1} & \frac{\partial f_3(\cdot)}{\partial x_2} & \frac{\partial f_3(\cdot)}{\partial x_3} & \frac{\partial f_3(\cdot)}{\partial x_4} \\ \frac{\partial f_4(\cdot)}{\partial x_1} & \frac{\partial f_4(\cdot)}{\partial x_2} & \frac{\partial f_4(\cdot)}{\partial x_3} & \frac{\partial f_4(\cdot)}{\partial x_4} \end{bmatrix} \quad (\text{B.1})$$

$$= \begin{bmatrix} \Phi_{11}(k) & \Phi_{12}(k) & \Phi_{13}(k) & \Phi_{14}(k) \\ \Phi_{21}(k) & \Phi_{22}(k) & \Phi_{23}(k) & \Phi_{24}(k) \\ \Phi_{31}(k) & \Phi_{32}(k) & \Phi_{33}(k) & \Phi_{34}(k) \\ \Phi_{41}(k) & \Phi_{42}(k) & \Phi_{43}(k) & \Phi_{44}(k) \end{bmatrix}$$

der elementene ble følgende etter partiell derivasjon med MATLAB:

$$\Phi_{11}(k) = 1 - c \cdot f_1(u_{LV001}(k)) \cdot \frac{T_s}{2A \cdot \sqrt{x_1(k) + h_{LV001}}} \quad (\text{B.2})$$

$$\Phi_{14}(k) = \frac{T_s}{A} \quad (\text{B.3})$$

$$\Phi_{21}(k) = \frac{T_s}{Ax_1^2(k)} \cdot \left(x_2(k)x_4(k) + q_{LV003}(k)(x_2(k) - x_3(k)) - \frac{P(k)}{c_p \rho} \right) \quad (\text{B.4})$$

$$\Phi_{22}(k) = 1 - \frac{T_s}{Ax_1(k)} \cdot (q_{LV003}(k) + x_4(k)) \quad (\text{B.5})$$

$$\Phi_{23}(k) = \frac{T_s \cdot q_{LV003}(k)}{A \cdot x_1(k)} \quad (\text{B.6})$$

$$\Phi_{24}(k) = -\frac{T_s \cdot x_2(k)}{A \cdot x_1(k)} \quad (\text{B.7})$$

$$\Phi_{33}(k) = 1 \quad (\text{B.8})$$

$$\Phi_{44}(k) = 1 \quad (\text{B.9})$$

B.2 Transisjonsmatrise (kapittel 7.1)

Transisjonsmatrisen uttrykkes følgende når det er to tilstander som estimeres:

$$\Phi(k) = \frac{\begin{bmatrix} \partial f_1(\cdot) \\ \partial f_2(\cdot) \end{bmatrix}}{\begin{bmatrix} \partial x_1 & \partial x_2 \end{bmatrix}} = \begin{bmatrix} \frac{\partial f_1(\cdot)}{\partial x_1} & \frac{\partial f_1(\cdot)}{\partial x_2} \\ \frac{\partial f_2(\cdot)}{\partial x_1} & \frac{\partial f_2(\cdot)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \Phi_{11}(k) & \Phi_{12}(k) \\ \Phi_{21}(k) & \Phi_{22}(k) \end{bmatrix} \quad (\text{B.10})$$

der elementene ble følgende etter partiell derivasjon med MATLAB:

$$\Phi_{11}(k) = 1 - c \cdot f_1(u_{LV001}(k)) \cdot \frac{T}{2A \cdot \sqrt{x_1(k) + h_{LV001}}} \quad (\text{B.11})$$

$$\Phi_{12}(k) = 0 \quad (\text{B.12})$$

$$\Phi_{21}(k) = -\frac{T}{Ax_1^2(k)} \cdot \left(q_{LV003}(k)(T_i(k) - x_2(k)) + \frac{P(k)}{c_p \rho} \right) \quad (\text{B.13})$$

$$\Phi_{22}(k) = 1 - \frac{T}{Ax_1(k)} \cdot q_{LV003}(k) \quad (\text{B.14})$$

B.3 Transisjonsmatrise (kapittel 7.3)

Transisjonsmatrisen uttrykkes følgende når det er to tilstander som estimeres:

$$\Phi(k) = \begin{bmatrix} \frac{\partial f_1(\cdot)}{\partial x_1} & \frac{\partial f_1(\cdot)}{\partial x_2} & \frac{\partial f_1(\cdot)}{\partial x_3} \\ \frac{\partial f_2(\cdot)}{\partial x_1} & \frac{\partial f_2(\cdot)}{\partial x_2} & \frac{\partial f_2(\cdot)}{\partial x_3} \\ \frac{\partial f_3(\cdot)}{\partial x_1} & \frac{\partial f_3(\cdot)}{\partial x_2} & \frac{\partial f_3(\cdot)}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \Phi_{11}(k) & \Phi_{12}(k) & \Phi_{13}(k) \\ \Phi_{21}(k) & \Phi_{22}(k) & \Phi_{23}(k) \\ \Phi_{31}(k) & \Phi_{32}(k) & \Phi_{33}(k) \end{bmatrix} \quad (\text{B.15})$$

$$\Phi(k) = \begin{bmatrix} \Phi_{11}(k) & 0 & \Phi_{13}(k) \\ \Phi_{21}(k) & \Phi_{22}(k) & \Phi_{23}(k) \\ 0 & 0 & \Phi_{33}(k) \end{bmatrix} \quad (\text{B.16})$$

$$\Phi_{11}(k) = 1 - c \cdot f_1(u_{LV001}(k)) \cdot \frac{T}{2A \cdot \sqrt{x_1(k) + h_{LV001}}} \quad (\text{B.17})$$

$$\Phi_{13}(k) = \frac{T}{A} \quad (\text{B.18})$$

$$\Phi_{21}(k) = -\frac{T}{Ax_1^2(k)} (q_{LV003}(k)(T_i(k) - x_2(k)) + \frac{P(k)}{c_p \rho} - x_2(k)x_3(k)) \quad (\text{B.19})$$

$$\Phi_{22}(k) = 1 - \frac{T}{Ax_1(k)} \cdot q_{LV003}(k) - x_3(k) \quad (\text{B.20})$$

$$\Phi_{23}(k) = -\frac{T}{Ax_1(k)} \cdot x_2(k) \quad (\text{B.21})$$

$$\Phi_{33}(k) = 1 \quad (\text{B.22})$$

C – MATLAB kode

C.1 Transisjonsmatrise

Koden finnes i filen "transisjonsmatrise.m"

```
% Beregning av transisjonsmatrise for kapittel 4, 6 og 7.
syms x1 x2 x3 x4 u1 u2 u3 u4 c h_LV001 Ts Tis A c_p rho

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Beregning av transisjonsmatrisen for estimering av temperaturen
inn
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Systemets differensialligning
f1 = x1 + (Ts/A)*(u2-c*u1*sqrt(x1+h_LV001));
f2 = x2 + (Ts/(A*x1))*(u2*(x3-x2)+(u3/(c_p*rho)));
f3 = x3;

% Partiell derivasjon
df1_dx1 = diff(f1, x1);
df1_dx2 = diff(f1, x2);
df1_dx3 = diff(f1, x3);

df2_dx1 = diff(f2, x1);
df2_dx2 = diff(f2, x2);
df2_dx3 = diff(f2, x3);

df3_dx1 = diff(f3, x1);
df3_dx2 = diff(f3, x2);
df3_dx3 = diff(f3, x3);

% Transisjonsmatrise
phi = [df1_dx1, df1_dx2, df1_dx3;...
       df2_dx1, df2_dx2, df2_dx3;...
       df3_dx1, df3_dx2, df3_dx3];
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Beregning av transisjonsmatrisen for estimering av ukjent tilsats,
% samt estimering av temperaturen inn
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Systemets differensialligning
f1 = x1 + (Ts/A)*(u2-c*u1*sqrt(x1+h_LV001)+x4);
f2 = x2 + (Ts/(A*x1))*(u2*(x3-x2)+(u3/(c_p*rho))-x4*x2+x4*Tt);
f3 = x3;
f4 = x4;

% Partiell derivasjon
df1_dx1 = diff(f1, x1);
df1_dx2 = diff(f1, x2);
df1_dx3 = diff(f1, x3);
df1_dx4 = diff(f1, x4);

df2_dx1 = diff(f2, x1);
df2_dx2 = diff(f2, x2);
df2_dx3 = diff(f2, x3);
df2_dx4 = diff(f2, x4);

df3_dx1 = diff(f3, x1);
df3_dx2 = diff(f3, x2);
df3_dx3 = diff(f3, x3);
df3_dx4 = diff(f3, x4);

df4_dx1 = diff(f4, x1);
df4_dx2 = diff(f4, x2);
df4_dx3 = diff(f4, x3);
df4_dx4 = diff(f4, x4);

% Transisjonsmatrise
phi1 = [df1_dx1, df1_dx2, df1_dx3, df1_dx4;...
        df2_dx1, df2_dx2, df2_dx3, df2_dx4;...
        df3_dx1, df3_dx2, df3_dx3, df3_dx4;...
        df4_dx1, df4_dx2, df4_dx3, df4_dx4];

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Beregning av transisjonsmatrisen for estimering av temperatur i
tanken
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Systemets differensialligning
f1 = x1 + (Ts/A)*(u2-c*u1*sqrt(x1+h_LV001));
f2 = x2 + (Ts/(A*x1))*(u2*(u3-x2)+(u4/(c_p*rho)));

% Partiell derivasjon
df1_dx1 = diff(f1, x1);
df1_dx2 = diff(f1, x2);
df2_dx1 = diff(f2, x1);
df2_dx2 = diff(f2, x2);

% Transisjonsmatrise
phi2 = [df1_dx1, df1_dx2;...
        df2_dx1, df2_dx2];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Beregning av transisjonsmatrisen for av ukjent tilsats, samt
estimering av temperaturen i tanken
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Systemets differensialligning
f1 = x1 + (Ts/A)*(u2-c*u1*sqrt(x1+h_LV001)+x3);
f2 = x2 + (Ts/(A*x1))*(u2*(u3-x2)+(u4/(c_p*rho))+x3*Tt);
f3 = x3;

% Partiell derivasjon
df1_dx1 = diff(f1, x1);
df1_dx2 = diff(f1, x2);
df1_dx3 = diff(f1, x3);
df2_dx1 = diff(f2, x1);
df2_dx2 = diff(f2, x2);
df2_dx3 = diff(f2, x3);
df3_dx1 = diff(f3, x1);
df3_dx2 = diff(f3, x2);
df3_dx3 = diff(f3, x3);

% Transisjonsmatrise
phi3 = [df1_dx1, df1_dx2, df1_dx3;...
        df2_dx1, df2_dx2, df2_dx3;...
        df3_dx1, df3_dx2, df3_dx3];

```

C.2 Observerbarhet

Koden finnes i filen "Observerbart.m"

```
% Observerbart.m finner rangen til observasjonsmatrisen
syms phi11 phi14 phi21 phi22 phi23 phi24 phi33 phi44 phi55 phi25

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Kapittel 4 - Finner rangen på observerbarhetsmatrisen
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D = [1 0 0 ; 0 1 0];          % Målematrise
phi = [phi11 0 0;...        % Transisjonsmatrise
       phi21 phi22 phi23;...
       0 0 phi33];
Obs = [ D; D*phi; D*phi^2];  % Observerbarhetsmatrise
rang= rank(Obs);            % Beregner rangen til Obs

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Kapittel 6 - Finner rangen på observerbarhetsmatrisen
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D1 = [1 0 0 0; 0 1 0 0];
phi1 = [phi11 0 0 phi14;...
        phi21 phi22 phi23 phi24;...
        0 0 phi33 0;...
        0 0 0 phi44];
Obs1 = [ D1; D1*phi1; D1*phi1^2; D1*phi1^3];
rang1= rank(Obs1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Kapittel 7 - Finner rangen på observerbarhetsmatrisen
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D2 = [1 0];
phi2 = [phi11 0;...
        phi21 phi22];
Obs2 = [ D2; D2*phi2];
rang2= rank(Obs2);
```

C.3 Estimering av temperatur inn (kapittel 4.3)

```
% EKF for estimering av temperatur i tanken ved hjelp av pådragsdata
og
% måling av nivå i tanken.
function [h_est,t_est,tinn_est,K_P_matrise,KFtilstand] =
fcnEKF(y1,...
y2,y3,u1,u2,u3,Qv,Rw,SystemKonstanter,KFtilstandInn)

% Innganger:
% y1 - Måling av høyde i tank
% y2 - Måling av temperatur i tank
% y3 - Måling av temperatur fra vannbatteriet (temperatur inn i
tanken)
% u1 - Tabelloppslag av ventilkarakteristikk for LV001
% u2 - Tabelloppslag av ventilkarakteristikk for LV003
% u3 - Pådrag fra varmeelement
% SystemKonstanter - En vektor fra workspace som inneholder
konstanter for vann, tyngdekraft og tank
% KFtilstandInn - Er en vektor med 16 element som er tilstanden til
KF fra forrige tidssteg

% Utganger:
% h_est - EKF estimat av nivå i tanken
% t_est - EKF estimat av temperatur i tanken
% tinn_est - er estimat av temperatur inn i tanken
% K_P_matrise - Kovariansmatrise og Kalman forsterkningsmatrise
% KFtilstand - Er en vektor med 16 element som inneholder tilstanden
til KF

% Data om vann, tyngdeakselerasjon og tank (hentes fra matlab fil)
rho = SystemKonstanter(1);          % Massetetthet vann [kg/m^3]
g    = SystemKonstanter(2);          % Tyngdeakselerasjon [m/s^2]
c_p  = SystemKonstanter(3);          % Varmekapasitet vann [j/kg*K]
A     = SystemKonstanter(4);         % Areal tank [m^2]
Kv_LV001 = SystemKonstanter(5);     % Ventilkonstant LV001
[m^3/h/sqrt(bar)]
h_LV001 = SystemKonstanter(6);      % Høyde til LV001 [m]
h_max   = SystemKonstanter(7);      % Maks nivå tanken [m]
h_min   = SystemKonstanter(8);      % Min nivå tanken [m]
Tmax    = SystemKonstanter(9);      % Maks temperatur i tanken [m]
Tmin    = SystemKonstanter(10);     % Min temperatur i tanken [m]

% Forenkling av uttrykk
c = Kv_LV001*sqrt(rho*g/100000)/3600;
```

```

% Samplingstid
Ts = 0.07;

% Tuningparametre
Q = Qv;      % Prosesstøy kovariansmatrise
R = Rw;      % Målestøy kovariansmatrise

% Data fra forrige tilstand
x1 = KFtilstandInn(2); % Aposteriori tilstandsestimat
x2 = KFtilstandInn(3); % Aposteriori tilstandsestimat
x3 = KFtilstandInn(4); % Aposteriori tilstandsestimat
P = [KFtilstandInn(5), KFtilstandInn(8), KFtilstandInn(11);
     KFtilstandInn(6), KFtilstandInn(9), KFtilstandInn(12);
     KFtilstandInn(7), KFtilstandInn(10), KFtilstandInn(13)];
% Kovariansmatrise til aposteriori estimeringsavvik
f1 = KFtilstandInn(14); % Pådrag fra forrige tidssteg
f2 = KFtilstandInn(15); % Pådrag fra forrige tidssteg
f3 = KFtilstandInn(16); % Pådrag fra forrige tidssteg

% Initielle verdier
if KFtilstandInn(1) <= 1
x1 = y1;
x2 = y2;
x3 = y3;
end

KFtilstand = zeros(1,16); % Oppretter en tom vektor
KFtilstand(1) = KFtilstandInn(1)+1; % Antall steg totalt, inkludert
dette

% EKF algoritmen
% Steg 1: Beregner x_
q1 = c * f1 * sqrt(x1+h_LV001); % Strømning ut av tanken
x1 = x1 + (Ts/A)*(f2-q1);
x2 = x2 + (Ts/(A*x1))*(f2*(x3-x2)+(f3/(c_p*rho)));

% Steg 2: Phi-matrisen
Phi = [1 - (Ts*q1)/(2*A*(h_LV001 + x1)), 0, 0;...
       (Ts*(f2*(x2-x3)-f3/(c_p*rho)))/(A*x1^2), 1-(Ts*f2)/(A*x1), ...
       (Ts*f2)/(A*x1); 0, 0, 1];

% Steg 3: Beregner P_
P = (Phi*P*Phi') + Q;

% Steg 4: Målematrise
D = [1,0,0;0,1,0]; % måler x1 og x2

```

```

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

% Steg 6: y_strek
y_strek = [x1;x2]; %Forventet nivå og temperatur i tanken

% Steg 7: Beregner aposteriori tilstandsestimat, x^
x = [x1;x2;x3] + K*([y1;y2]-y_strek);
x1 = x(1);
x2 = x(2);
x3 = x(3);

% Steg 8: Beregner P^
P = (eye(3) - K*D)*P;

% Holder estimatene innenfor enn gitt grense
if (x1 < h_min); x1 = h_min; end;
if (x1 > h_max); x1 = h_max; end;
if (x2 < Tmin); x2 = Tmin; end;
if (x2 > Tmax); x2 = Tmax; end;

% Gi ut verdier for neste tidssteg
KFtilstand(2) = x1;           % Aposteriori tilstandsestimat
KFtilstand(3) = x2;           % Aposteriori tilstandsestimat
KFtilstand(4) = x3;           % Aposteriori tilstandsestimat
KFtilstand(5:13) = P(:)';     % Aposteriori kovariansmatrise
KFtilstand(14) = u1;          % Tabelloppslag for ventilpådrag LV001
KFtilstand(15) = u2;          % Tabelloppslag for ventilpådrag LV003
KFtilstand(16) = u3;          % Pådrag til varmeelement HE001

% Nye tilstandestimater
h_est = x1;
t_est = x2;
tinn_est = x3;

% Tar ut noen data til workspace
K_P_matrise =
[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);K(1,1);K(1,2);K(2,1);K(2,2);K(3,1);K(3,2)];

```

C.4 Estimering av ekstern tilsats (kapittel 6.4)

```
% EKF for estimering av temperatur i tanken ved hjelp av pådragsdata og
% måling av nivå i tanken.
function[h_est,t_est,tinn_est,tils_est,K_P_matriise,KFtilstand]=...
fcnEKF(y1,y2,y3,u1,u2,u3,Qv,Rw,SystemKonstanter,KFtilstandInn)

% Innganger:
% y1 - Måling av høyde i tanken
% y2 - Måling av temperatur i tanken
% y3 - Måling av temperatur fra vannbatteriet (temperatur inn i tanken)
% u1 - Tabelloppslag av ventilkarakteristikk for LV001
% u2 - Tabelloppslag av ventilkarakteristikk for LV003
% u3 - Pådrag fra varmeelement
% SystemKonstanter - En vektor fra workspace som inneholder konstanter for
% vann, tyngdekraft og tank
% KFtilstandInn - Er en vektor med 24 element som er tilstanden til KF fra
% forrige tidssteg

% Utganger:
% h_est - Er estimat av nivå i tanken
% t_est - Er estimat av temperatur i tanken
% tinn_est - er estimat av temperatur inn i tanken
% tils_est - er estimat av ekstra tilsats som tilfører oppi tanken
% K_P_matriise - Kovariansmatrise og Kalman forsterkningsmatrise
% KFtilstand - Er en vektor med 24 element som inneholder tilstanden til KF

% Data om vann, tyngdeakselerasjon og tank
rho = SystemKonstanter(1);      % Massetetthet vann [kg/m^3]
g   = SystemKonstanter(2);      % Tyngdeakselerasjon [m/s^2]
c_p = SystemKonstanter(3);      % Varmekapasitet vann [j/kg*K]
A   = SystemKonstanter(4);      % Areal tank [m^2]
Kv_LV001 = SystemKonstanter(5); % Ventilkonstant LV001 [m^3/h/sqrt(bar)]
h_LV001 = SystemKonstanter(6);  % Høyde til LV001 [m]
h_max  = SystemKonstanter(7);   % Maks nivå tanken [m]
h_min  = SystemKonstanter(8);   % Min nivå tanken [m]
Tmax   = SystemKonstanter(9);   % Maks temperatur i tanken [m]
Tmin   = SystemKonstanter(10);  % Min temperatur i tanken [m]

% Konstant for å forenkle utregninger lenger nede
c = Kv_LV001*sqrt(rho*g/100000)/3600;

% Samplingstid
Ts = 0.07;
```

```

% Tuningparametre
Q = Qv;      % Prosesstøy kovariansmatrise
R = Rw;      % Målestøy kovariansmatrise

% Data fra forrige tilstand
x1 = KFtilstandInn(2); % Aposteriori tilstandsestimat
x2 = KFtilstandInn(3); % Aposteriori tilstandsestimat
x3 = KFtilstandInn(4); % Aposteriori tilstandsestimat
x4 = KFtilstandInn(5); % Aposteriori tilstandsestimat
P=[KFtilstandInn(6),KFtilstandInn(10),KFtilstandInn(14),KFtilstandInn(18);
   KFtilstandInn(7),KFtilstandInn(11),KFtilstandInn(15),KFtilstandInn(19);
   KFtilstandInn(8),KFtilstandInn(12),KFtilstandInn(16),KFtilstandInn(20);
   KFtilstandInn(9),KFtilstandInn(13),KFtilstandInn(17),KFtilstandInn(21)];
% Kovariansmatrise til aposteriori estimeringsavvik
f1 = KFtilstandInn(22); % Pådrag fra forrige tidssteg
f2 = KFtilstandInn(23); % Pådrag fra forrige tidssteg
f3 = KFtilstandInn(24); % Pådrag fra forrige tidssteg

KFtilstand = zeros(1,24); % Oppretter en tom vektor
KFtilstand(1) = KFtilstandInn(1)+1; % Antall steg totalt, inkludert dette

% Initielle verdier
if KFtilstandInn(1) <= 1
x1 = y1;
x2 = y2;
x3 = y3;
x4 = 0;
end

% EKF algoritmen
% Steg 1: Beregner x_
Tt = 0.8; % Tilsats temperatur
q1 = c * f1 * sqrt(x1+h_LV001); % Strømning ut av tanken
x1 = x1 + (Ts/A)*(f2-q1+x4);
x2 = x2 + (Ts/(A*x1))*(f2*(x3-x2)+(f3/(c_p*rho))-x4*x2+x4*Tt);

% Steg 2: Phi-matrisen
Phi = [1 - (Ts*q1)/(2*A*(h_LV001 + x1)), 0, 0, Ts/A;...
       (Ts/(A*x1^2))*(Tt*x4 - x2*x4-(f2*(x2-x3))-(f3/(c_p*rho))),...
       1-((Ts*(f2+x4))/(A*x1)), (Ts*f2)/(A*x1), -(Ts*(x2 - Tt))/(A*x1);...
       0, 0, 1, 0;...
       0, 0, 0, 1];

% Steg 3: Beregner P^
P = (Phi*P*Phi') + Q;

```

```

% Steg 4: Målematrise
D = [1 0 0 0;0 1 0 0]; % måler x1 og x2

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

% Steg 6: y_
y_strek = [x1;x2]; %Forventet nivå og temperatur i tanken

% Steg 7: Beregner aposteriori tilstandsestimat, x^
x = [x1;x2;x3;x4] + K*([y1;y2]-y_strek);
x1 = x(1);
x2 = x(2);
x3 = x(3);
x4 = x(4);

% Steg 8: Beregner P^
P = (eye(4) - K*D)*P;

% Holder estimatene innenfor gitt grense
if (x1 < h_min); x1 = h_min; end;
if (x1 > h_max); x1 = h_max; end;
if (x2 < Tmin); x2 = Tmin; end;
if (x2 > Tmax); x2 = Tmax; end;
if (x4 < 0); x4 = 0; end;

% Gi ut verdier for neste tidssteg
KFtilstand(2) = x1;           % Aposteriori tilstandsestimat
KFtilstand(3) = x2;           % Aposteriori tilstandsestimat
KFtilstand(4) = x3;           % Aposteriori tilstandsestimat
KFtilstand(5) = x4;           % Aposteriori tilstandsestimat
KFtilstand(6:21) = P(:)';    % Aposteriori kovariansmatrise
KFtilstand(22)= u1;           % Tabelloppslag for ventilpådrag LV001
KFtilstand(23)= u2;           % Tabelloppslag for ventilpådrag LV003
KFtilstand(24)= u3;           % Pådrag til varmeelement HE001

% Nye tilstandsestimater
h_est = x1;
t_est = x2;
tinn_est = x3;
tils_est = x4;

% Legger ut data som skal lagres
K_P_matrise = [K(1,1);K(2,2);K(3,1);K(3,2);K(4,1);P(1,1);...
               P(2,2);P(3,3);P(4,4)];

```


C.5 Estimering av temperatur i tanken (kapittel 7.1)

```
% Utvidet Kalmanfilter for estimering av temperatur i tanken
function [h_est, t_est, K_P_matrise, KFtilstand] = fcnEKF(y1, ...
y2, u1, u2, u3, u4, Qv, Rw, SystemKonstanter, KFtilstandInn)

% Innganger:
% y1 - Måling av høyde i tanken
% y2 - Måling av temperatur i tanken
% u1 - Tabelloppslag av ventilkarakteristikk for LV001
% u2 - Tabelloppslag av ventilkarakteristikk for LV003
% u3 - Pådrag fra vannbatteri (temperatur inn i tanken)
% u4 - Pådrag fra varmeelement
% SystemKonstanter - En vektor fra workspace som inneholder
konstanter for vann, tyngdekraft og tank
% KFtilstandInn - Er en vektor med 11 element som er tilstanden til
KF fra forrige tidssteg

% Utganger:
% h_est - er estimat av nivå i tanken
% t_est - er estimat av temperatur i tanken
% K_P_matrise - Kovariansmatrise og Kalman forsterkningsmatrise
% KFtilstand - Er en vektor med 11 element som inneholder tilstanden
til KF

% Data om vann, tyngdeakselerasjon og tank (hentes fra matlab fil)
rho = SystemKonstanter(1);           % Massetetthet vann [kg/m^3]
g    = SystemKonstanter(2);           % Tyngdeakselerasjon [m/s^2]
c_p  = SystemKonstanter(3);           % Varmekapasitet vann [j/kg*K]
A     = SystemKonstanter(4);           % Areal tank [m^2]
Kv_LV001 = SystemKonstanter(5);       % Ventilkonstant LV001
[m^3/h/sqrt(bar)]
h_LV001 = SystemKonstanter(6);        % Høyde til LV001 [m]
h_max   = SystemKonstanter(7);        % Maks nivå tanken [m]
h_min   = SystemKonstanter(8);        % Min nivå tanken [m]
Tmax    = SystemKonstanter(9);        % Maks temperatur i tanken [m]
Tmin    = SystemKonstanter(10);       % Min temperatur i tanken [m]

% Forenkling av uttrykk
c = Kv_LV001*sqrt(rho*g/100000)/3600;

% Samplingstid
T = 0.07;

% Tuningparametre
```

```

Q = Qv;      % Prosesstøy kovariansmatrise
R = Rw;      % Målestøy kovariansmatrise

% Forrige tidssteg data
x1 = KFtilstandInn(2); % Aposteriori tilstandsestimat
x2 = KFtilstandInn(3); % Aposteriori tilstandsestimat
P = [KFtilstandInn(4), KFtilstandInn(6);...
     KFtilstandInn(5), KFtilstandInn(7)];
     % Kovariansmatrise til aposteriori estimeringsavvik
f1 = KFtilstandInn(8); % Tabelloppslag fra forrige tidssteg
f2 = KFtilstandInn(9); % Tabelloppslag fra forrige tidssteg
f3 = KFtilstandInn(10); % Pådrag fra forrige tidssteg
f4 = KFtilstandInn(11); % Pådrag fra forrige tidssteg

KFtilstand = zeros(1,11); % Oppretter vektor med nuller
KFtilstand(1) = KFtilstandInn(1)+1; % Antall steg totalt, inkludert
dette

% Kjent start verdier for estimatene
if KFtilstand(1) <= 1
x1 = y1;
x2 = y2;
end

% EKF algoritmen
% Steg 1: Beregner apriori tilstandsestimat (x_strek)
q1 = c * f1 * sqrt(x1+h_LV001); % Strømning ut av tanken
x1 = x1 + (T/A)*(f2-q1); % x1_strek
x2 = x2 + (T/(A*x1))*(f2*(f3-x2) + f4/(c_p*rho)); % x2_strek

% Steg 2: Beregner transisjonsmatrisen (Phi-matrisen)
Phi = [1 - T*q1/(2*A*(x1 + h_LV001)) , 0 ;...
       -(T/(A*x1^2))*(f3*f2 - x2*f2 + f4/(c_p*rho)), 1-(T/(A*x1))*f2;];

% Steg 3: Beregner kovariansmatrise til apriori estimeringsavvik, P_
P = Phi*P*Phi' + Q;

% Måler tanken ved et gitt tidspunkt
if (mod(KFtilstand(1),858)==5) %(858*0.07s=60.06s)

% Steg 4: Målematrise
Dm = eye(2); % Her måles det både x1 og x2

% Steg 5: Beregner Kalmanforsterkning, K
Km = P*Dm' / (Dm*P*Dm' + R*eye(2));

% Steg 6: Måler nivå og temperatur

```

```

y_strek = [x1; x2];

% Steg 7: Beregner aposteriori tilstandsestimat, x_^
x = [x1;x2] + Km*([y1;y2]-y_strek);
x1 = x(1);
x2 = x(2);

% Steg 8: Beregner kovariansmatrise til aposteriori estimeringsavvik
(P^)
P = (eye(2) - Km*Dm)*P;

% Henter ut diagonal elementene i Km
K = Km;

else

% Steg 4: Målematrise
D = [1,0;0,0]; % Her måles det x1

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

% Steg 6: Måler nivå
y_strek = x1;

% Steg 7: Beregner aposteriori tilstandsestimatene, x^
x1 = x1 + K(1)*(y1-y_strek);
x2 = x2 + K(2)*(y1-y_strek);

% Steg 8: Beregner kovariansmatrise til aposteriori
estimeringsavvik, P^
P = (eye(2) - K*D)*P;
end

% Holder estimatene innenfor gitt grenser
if (x1 < h_min); x1 = h_min; end;
if (x1 > h_max); x1 = h_max; end;
if (x2 < Tmin); x2 = Tmin; end;
if (x2 > Tmax); x2 = Tmax; end;

% Gi ut verdier for neste tidssteg
KFtilstand(2) = x1; % Aposteriori tilstandsestimat
KFtilstand(3) = x2; % Aposteriori tilstandsestimat
KFtilstand(4:7) = P(:)'; % Aposteriori kovariansmatrise
KFtilstand(8) = u1; % Tabelloppslag for ventilpådrag LV001
KFtilstand(9) = u2; % Tabelloppslag for ventilpådrag LV003
KFtilstand(10) = u3; % Temperatur fra vannbatteri (pådrag)

```

```
KFtilstand(11) = u4;           % Pådrag til varmeelement HE001

% Nye tilstandsestimater
h_est = x1;
t_est = x2;

% Legger ut data som skal lagres
K_P_matrise = [P(1,1);P(1,2);P(2,1);P(2,2);K(1);K(2)];
```

C.6 Estimering av ekstern tilsats (kapittel 7.3)

```
% Utvidet Kalmanfilter for estimering av temperatur i tanken ved
hjelp av pådragsdata og måling av nivå i tanken.
function [h_est, t_est, qis_est, K_P_matrise, KFtilstand]
=fcnEKF(y1, ...
y2, u1, u2, u3, u4, Qv, Rw, SystemKonstanter, KFtilstandInn)

% Innganger:
% y1 - Måling av høyde i tank
% y2 - Måling av temperatur i tank
% u1 - Tabelloppslag av ventilkarakteristikk for LV001
% u2 - Tabelloppslag av ventilkarakteristikk for LV003
% u3 - Pådrag fra vannbatteri (temperatur inn i tank)
% u4 - Pådrag fra varmeelement
% SystemKonstanter - En vektor fra workspace som inneholder
konstanter for vann, tyngdekraft og tank
% KFtilstandInn - Er en vektor med 17 element som er tilstanden til
KF fra forrige tidssteg

% Utganger:
% h_est - er estimat av nivå i tanken
% t_est - er estimat av temperatur i tanken
% K_P_matrise - Kovariansmatrise og Kalman forsterkningsmatrise
% KFtilstand - Er en vektor med 17 element som inneholder tilstanden
til KF

% Data om vann, tyngdeakselerasjon og tank
rho = SystemKonstanter(1);      % Massetetthet vann [kg/m^3]
g   = SystemKonstanter(2);      % Tyngdeakselerasjon [m/s^2]
c_p = SystemKonstanter(3);      % Varmekapasitet vann [j/kg*K]
A    = SystemKonstanter(4);      % Areal tank [m^2]
Kv_LV001 = SystemKonstanter(5); % Ventilkonstant LV001
[m^3/h/sqrt(bar)]
h_LV001 = SystemKonstanter(6);  % Høyde til LV001 [m]
h_max   = SystemKonstanter(7);  % Maks nivå tanken [m]
h_min   = SystemKonstanter(8);  % Min nivå tanken [m]
Tmax    = SystemKonstanter(9);  % Maks temperatur i tanken [m]
Tmin    = SystemKonstanter(10); % Min temperatur i tanken [m]

% Forenkling av uttrykk
c = Kv_LV001*sqrt(rho*g/100000)/3600;

% Samplingstid
T = 0.07;
```

```

% Tuningparametre
Q = Qv;      % Prosesstøy kovariansmatrise
R = Rw;      % Målestøy kovariansmatrise

% Data fra forrige tilstand
x1 = KFtilstandInn(2); % Aposteriori tilstandsestimat
x2 = KFtilstandInn(3); % Aposteriori tilstandsestimat
x3 = KFtilstandInn(4); % Aposteriori tilstandsestimat
P = [KFtilstandInn(5), KFtilstandInn(8), KFtilstandInn(11);
     KFtilstandInn(6), KFtilstandInn(9), KFtilstandInn(12);
     KFtilstandInn(7), KFtilstandInn(10), KFtilstandInn(13)];
     % Kovariansmatrise til aposteriori estimeringsavvik
f1 = KFtilstandInn(14); % Tabelloppslag fra forrige tidssteg
f2 = KFtilstandInn(15); % Tabelloppslag fra forrige tidssteg
f3 = KFtilstandInn(16); % Pådrag fra forrige tidssteg
f4 = KFtilstandInn(17); % Pådrag fra forrige tidssteg

KFtilstand = zeros(1,17); % Oppretter en tom vektor
KFtilstand(1) = KFtilstandInn(1)+1; % Antall steg totalt, inkludert
dette

% Kjent start verdier for estimatene
if KFtilstand(1) <= 1
x1 = y1;
x2 = y2;
x3 = 0;      % Ikke ekstra tilsats initielt
end

% EKF algoritmen
% Steg 1: Beregner apriori tilstandsestimat, x_
q1 = c * f1 * sqrt(x1+h_LV001); % Strømning ut av tanken
x1 = x1 + (T/A)*(f2-q1+x3);
x2 = x2 + (T/(A*x1))*(f2*(f3-x2) + (f4/(c_p*rho)) - x3*x2);

% Steg 2: Beregner transisjonsmatrisen (Phi-matrisen)
Phi = [1 - T*q1/(2*A*(x1 + h_LV001)) , 0 , T/A ;
       -(T/(A*x1^2))*(f3*f2 - x2*f2 + (f4/(c_p*rho))-x3*x2),1-...
       ((T/(A*x1))*(f2 + x3)),-(T/(A*x1))*x2; 0 , 0 , 1 ;];

% Steg 3: Beregner kovariansmatrise til apriori estimeringsavvik, P_
P = Phi*P*Phi' + Q;

% Måler tanken ved et gitt tidspunkt
if (mod(KFtilstand(1),1428)==5) % 1428'te steg. (1428*0.07s=99.96s)

% Steg 4: Målematrise
Dm = [1,0,0;0,1,0]; % Her måles det både x1 og x2

```

```

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

% Steg 6: Måler nivå og temperatur
y_strek = [x1;x2];

% Steg 7: Beregner aposteriori tilstandsestimat, x^
x = [x1;x2;x3] + Km*([y1;y2]-y_strek);
x1 = x(1);
x2 = x(2);
x3 = x(3);

% Steg 8: Beregner kovariansmatrise til aposteriori
estimeringsavvik, P^
P = (eye(3) - Km*Dm)*P;

% Henter ut K matrisen
K = Km;
else
% Steg 4: Målematrise
D = [1,0,0;0,0,0]; % måler x1

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

% Steg 6: Måler nivå
y_strek = x1;

% Steg 7: Beregner aposteriori tilstandsestimatene, x^
x1 = x1 + K(1)*(y1-y_strek);
x2 = x2 + K(2)*(y1-y_strek);
x3 = x3 + K(3)*(y1-y_strek);

% Steg 8: Beregner kovariansmatrise til aposteriori
estimeringsavvik, P^
P = (eye(3) - K*D)*P;
end

% Holder estimatene innenfor gitt grense
if (x1 < h_min); x1 = h_min; end;
if (x1 > h_max); x1 = h_max; end;
if (x2 < Tmin); x2 = Tmin; end;
if (x2 > Tmax); x2 = Tmax; end;
if (x3 < 0); x3 = 0; end;

% Gi ut verdier for neste tidssteg

```

```

KFtilstand(2) = x1;           % Aposteriori tilstandsestimat
KFtilstand(3) = x2;           % Aposteriori tilstandsestimat
KFtilstand(4) = x3;           % Aposteriori tilstandsestimat
KFtilstand(5:13) = P(:)';     % Aposteriori kovariansmatrise
KFtilstand(14) = u1;          % Tabelloppslag for ventilpådrag LV001
KFtilstand(15) = u2;          % Tabelloppslag for ventilpådrag LV003
KFtilstand(16) = u3;          % Temperatur fra vannbatteri (pådrag)
KFtilstand(17) = u4;          % Pådrag til varmeelement HE001

% Nye tilstandsestimater
h_est = x1;
t_est = x2;
qis_est = x3;

% Legger ut kun interessante data som skal lagres
K_P_matrise = [K(1,1);K(2,2);K(3,1);K(3,2);P(1,1);P(2,2);P(3,3)];

```


C.7 Totankanlegg data

```
clear all
close all

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

% Data om tanken
A      = 0.0096;      % areal tanken      [m^2]
Kv_LV001 = 11.25;     % ventilkonstant LV001 [m^3/h/sqrt(bar)]
h_LV001 = 0.05;      % høyde til LV001 [m]
h_max  = 1;          % maks høyde tanken [m]
h_min  = 0.13;       % min høyde tanken [m]
h_init = 0.3;        % initial hoyde [m]
Tmax   = 45;         % maks temperatur i tanken [m]
Tmin   = 0;          % min temperatur i tanken [m]
Tinit  = 25;         % initial temperatur i tanken [m]

% Vektor av data
HentKonstanter
=[0, rho, g, c_p, A, Kv_LV001, h_LV001, h_max, h_min, Tmax, Tmin];

%%%%%%%%%%%% Ventilkarakteristikk LV001 %%%%%%%%%%%%%
u_LV001 = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
f_LV001 = [0 0.051 0.146 0.254 0.345 0.456 0.637 0.775 0.873 0.953
1.00];

%%%%%%%%%%%% Ventilkarakteristikk LV003 %%%%%%%%%%%%%
u_LV003 = [0.00 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
q_LV003 = [6.332e-5 6.249e-5 5.916e-5 4.054e-5 2.089e-5...
1.344e-5 0.781e-5 0.216e-5 0.021e-5 0.00 0.00];

%%%%%%%%%%%% Last inn pådrag og målinger %%%%%%%%%%%%%
% load test_Data_kap3
% load test_Data_kap5
% load test_Data_uten_blanding_kap5
load test_Data_is_kap6
% load test_Data_is_uvariert_pådrag_kap6
% load test_Data_sammeTemp_kap6
% load test_Data_kap7
```