



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Robotteknologi og signalbehandling	Vårsemesteret, 2020 Åpen
Forfatter: Kristian Sandmoen	
Fagansvarlig: Tormod Drengstig Veileder(e): Tormod Drengstig	
Tittel på masteroppgaven: Kalmanfilter anvendt på biologiske prosessmodeller Engelsk tittel: Kalman filter applied to biological process models	
Studiepoeng: 30	
Emneord: Utvidet Kalmanfilter Biologiske prosessmodeller Tilstandsestimering Parameterestimering	Sidetall: 50 + vedlegg/annet: 49 Stavanger, 29.06.2020 dato/år

Forord

Denne masteroppgaven er en avsluttende oppgave for min utdanning i robotteknologi og signalbehandling ved Universitetet i Stavanger, våren 2020.

Jeg vil gjerne takke min veileder Tormod Drengstig for gode tilbakemeldinger og verdifull veiledning.

Kristian Sandmoen

Stavanger 29.06.2020

Sammendrag

I denne masteroppgaven er det blitt utført forsøk for å utforske anvendelsen av Kalmanfilter på biologiske prosessmodeller.

Først har det blitt implementert en modell for et biologisk kontrollmotiv. Deretter har et lineært og et utvidet Kalmanfilter blitt implementert og sammenlignet. Resultatene viser her at det utvidede Kalmanfilteret gir en bedre estimering av tilstandene, samtidig som det håndterer sprang i forstyrrelsene på en bedre måte enn det lineære filteret. Videre har Kalmanfilterets robusthet til avvik i initial verdier og ulike typer responser for forstyrrelser blitt testet. Her viser resultatene at det utvidede Kalmanfilteret håndterer ulike typer forstyrrelser svært godt, mens avvik i initialverdier for tilstandene må håndteres ved å øke den initielle usikkerheten i diagonalen til a posteriori kovariansmatrisen. Sist er det blitt sett på hvilke støy kovariansmatriser som gir de beste tilstandsestimatene. Her viser resultatene at på grunn av det biologiske kontrollmotivets kompenseringsegenskaper, vil den tillagte og den effektive prosesstøyen være forskjellig for systemet. Dette fører til at selv om systemet er blitt tillagt like mye prosess- og målestøy, vil målestøyen ha mer innvirkning enn prosesstøyen. Dette fører til at Kalmanfilteret kan brukes til å fjerne det meste av støyen i systemet.

Oppgaven tar videre for seg kombinert tilstand- og parameterestimering. Her brukes ett augmentert Kalmanfilter for å estimere parametere som ekstra tilstander. Ved å utføre målinger av en eller begge tilstandene i det biologiske motivet gir det utvidede Kalmanfilteret muligheten til å estimere alle de konstante parametere i kontrollmotivet. Deretter brukes det augmenterte Kalmanfilteret til å estimere sprang i parametere tillegg til respons i forstyrrelsen. Her viser resultatene at valget av prosesstøy kovarians for parametere gir muligheter for å estimere sprang i parametere, men estimatet vil da bli mer unøyaktig der parametere er konstante.

I siste delen av oppgaven er en utvidet versjon av det biologiske motivet implementert. Dette motivet inneholder 2 aktiveringsfunksjoner som blir først estimert i denne delen av oppgaven. Resultatene her viser at på samme måte som for sprang i parametere vil valget av prosesstøy kovarians spille en viktig rolle for hvor nøyaktig estimatet skal være i konstante områder, og hvor godt det skal holde følge med brå forandringer i forstyrrelsen.

Innhold

Forord	i
Sammendrag	ii
1 Innledning	1
1.1 Oppgavens formål og problemstilling	1
1.2 Oppgavens oppbygning	1
2 Introduksjon til biologiske prosessmodeller	2
2.1 Homeostase	2
2.2 Kontrollmotiver	2
2.3 Matematisk beskrivelse av biologiske prosessmodeller	3
3 Introduksjon til Kalmanfilter	5
3.1 Modell for prosessen	5
3.2 Forutsetninger for bruk av Kalmanfilter	5
3.3 Kalmanfilter ligninger	6
3.4 Utvidet Kalmanfilter ligninger	7
3.5 Utregning av tilstandsrommodellmatriser	7
3.5.1 Transmisjonsmatrisen Φ og pådragsmatrisen Γ	7
3.5.2 Målematrisen D og prosessforstyrrelsesmatrisen Ω	8
3.6 Observerbarhet	9
4 Kalmanfilter anvendt på biologisk kontrollmotiv	10
4.1 Implementering av kontrollmotiv i simulink	10
4.1.1 Differensialligninger for systemet	10
4.1.2 Valg av parameterverdier	10
4.1.3 Implementasjon av motivet i Simulink	11
4.1.4 Inkludering av prosess- og målestøy	12
4.2 Implementering av Kalmanfilter for motivet	13
4.2.1 Valg av initielle verdier for Kalmanfilteret	13
4.2.2 Matematisk modellering	15
4.2.3 implementering av Kalmanfilteret i Simulink	16
4.3 Utvidet Kalman filter	17
4.3.1 Matematisk modellering	17
4.3.2 implementering av det utvidede Kalmanfilteret	18
4.4 Tuning av initial verdier for det utvidede Kalmanfilteret	19
4.4.1 Avvik i initialverdien for tilstandene A og E	20
4.5 Valg av kovariansmatrisene Q og R	21
4.5.1 Q som tuningsparameter	21

4.5.2	Tuning av parametrene	22
4.5.3	Tuning av Q og R for biologiske kontrollmotiver	25
4.6	Test av ulike typer forstyrrelser	26
4.6.1	Simulink simulering av Kalmanfilter med utfordrende forstyrrelse	27
5	Kombinert tilstand og parameter estimering for biologisk kontrollmotiv	29
5.1	Augmentert Kalmanfilter for kombinert tilstand- og parameterestimering	29
5.2	Estimering av 1 parameter med måling av en tilstand	30
5.2.1	Linearisering og observerbarhet	30
5.2.2	Valg av parameterverdier	31
5.2.3	Simulink simulering	31
5.3	Estimering av 1 parameter med måling av begge tilstander	34
5.3.1	Linearisering og observerbarhet	34
5.3.2	Valg av parameterverdier	34
5.3.3	Simulink simulering	34
5.4	Estimering av 2 parameter med måling av begge tilstander	35
5.4.1	Linearisering og observerbarhet	36
5.4.2	Valg av parameterverdier	37
5.4.3	Simulink simulering	37
5.5	Estimering av parametere med sprang i parameteren	38
5.5.1	Simulink simulering med «spesifikke» støy kovarianser	39
6	Funksjons estimering på utvidet biologisk kontrollmotiv	42
6.1	Utvidelse av modellen	42
6.1.1	Differensiallikninger for systemet	42
6.1.2	Valg av parameterverdier	42
6.2	Simulink simulering	43
6.3	Estimering av funksjonene $\frac{E}{K_A^E+E}$ og $\frac{A}{K_A^A+A}$ med augmentert Kalmanfilter	45
6.3.1	Linearisering og observerbarhet	45
6.3.2	Valg av parameterverdier	45
6.3.3	Simulink simulering	45
7	Diskusjon og Konklusjon	47
7.1	Kapittel 4	47
7.2	Kapittel 5	47
7.3	Kapittel 6	48
	Bibliografi	50
A	Supplerende materiale kapittel 4	51
A.1	Simulink <i>scopes</i> for avvik i initialverdier for A	52
A.1.1	Forsøk med fleksible $\hat{P}(0)$ verdier	52
A.1.2	Forsøk med spesifiserte $\hat{P}(0)$ verdier	53
A.2	Simulink <i>scopes</i> for avvik i initialverdier for E	54

A.2.1	Forsøk med fleksible $\hat{P}(0)$ verdier	55
A.2.2	Forsøk med spesifiserte $\hat{P}(0)$ verdier	56
A.3	Støyreduksjonstest	57
A.3.1	Støyreduksjon med lik prosess- og målestøy kovariansmatrise	57
A.3.2	Støyreduksjon med høyere målestøy kovariansmatrise	57
A.3.3	Støyreduksjon med «spesifiserte» prosess- og målestøy kovariansmatriser	58
A.4	Test av forstyrrelser	59
A.4.1	Rampe	60
A.4.2	Sinus	61
A.4.3	<i>Pseudorandom nummer sekvens</i>	62
B	Supplerende materiale kapittel 5	63
B.1	Estimering av en tilleggs parameter med måling av A	64
B.1.1	Estimering av k_S^A	64
B.1.2	Estimering av k_S^E	65
B.1.3	Estimering av V_{max}	66
B.1.4	Estimering av K_M^E	67
B.2	Estimering av en tilleggs parameter med måling av E	67
B.2.1	Estimering av k_S^A	68
B.2.2	Estimering av k_S^E	69
B.2.3	Estimering av V_{max}	70
B.2.4	Estimering av K_M^E	71
B.3	Estimering av 1 parameter med måling av begge tilstander	71
B.3.1	Estimering av k_S^A med måling av begge tilstander	72
B.3.2	Estimering av k_S^E med måling av begge tilstander	73
B.3.3	Estimering av V_{max} med måling av begge tilstander	74
B.3.4	Estimering av K_M^E med måling av begge tilstander	75
B.4	Estimering av 2 parametere med måling av begge tilstander	75
B.4.1	Estimering av k_S^A og k_S^E med måling av begge tilstander	76
B.4.2	Estimering av k_S^A og V_{max} med måling av begge tilstander	77
B.4.3	Estimering av k_S^A og K_M^E med måling av begge tilstander	78
B.4.4	Estimering av k_S^E og V_{max} med måling av begge tilstander	79
B.4.5	Estimering av k_S^E og K_M^E med måling av begge tilstander	80
B.4.6	Estimering av V_{max} og K_M^E med måling av begge tilstander	81
C	Supplerende materiale kapittel 6	82
C.1	Estimering av funksjonen $\frac{A}{K_a^A + A}$ med augmentert Kalmanfilter	82
C.1.1	Simulink scope av A, E og $\frac{A}{K_a^A + A}$ med måling av A	82
C.1.2	Simulink scope av A, E og $\frac{A}{K_a^A + A}$ med måling av E	83
C.2	Estimering av funksjonen $\frac{E}{K_a^E + E}$ med augmentert Kalmanfilter	84
C.2.1	Simulink scope av A, E og $\frac{E}{K_a^E + E}$ med måling av A	85
C.2.2	Simulink scope av A, E og $\frac{E}{K_a^E + E}$ med måling av E	86

D Simulink og matlabkode eksempler	87
D.1 Simulink	87
D.2 Matlab kode	87
D.2.1 Initialisering av verdier	88
D.2.2 Initialisering av verdier utvidet modell	88
D.2.3 Matlab modell	88
D.2.4 Linearisering	88
D.2.5 Observerbarhet	89
D.2.6 Kalmanfilter	90
D.2.7 Utvidet Kalmanfilter	92
D.2.8 Augmentert Kalmanfilter	94
D.2.9 Augmentert Kalmanfilter med 2 målinger	96

Kapittel 1

Innledning

1.1 Oppgavens formål og problemstilling

Formålet med denne oppgaven er å utforske mulighetene for anvendelse av Kalmanfilter på biologiske prosessmodeller. Et lineært og et utvidet kalmanfilter vil først bli implementert. Det vil så bli utført tester for observerbarhet, robusthet for forstyrrelser, avvik i initialverdier og valg av støy kovariansmatriser. Det vil også bli utført kombinert tilstand- og parameterestimering på prosessmodellene ved å bruke et augmentert Kalmanfilter. Tilslutt vil det bli brukt ett augmentert Kalmanfilter for å estimere en aktiveringsfunksjon.

1.2 Oppgavens oppbygning

Oppgaven er inndelt som følger:

- Kapittel 1 er innledningen og beskriver oppgavens formål og oppbygning.
- Kapittel 2 gir en introduksjon til biologiske prosessmodeller.
- Kapittel 3 gir en introduksjon til Kalmanfilter.
- Kapittel 4 beskriver implementeringen av kalmanfilteret, samt forsøk som er utført for å teste robustheten til Kalmanfilteret.
- Kapittel 5 beskriver forsøk som er utført for å teste mulighetene rundt kombinert tilstand- og parameterestimering for modellen.
- Kapittel 6 beskriver forsøk som er utført for å teste mulighetene for å estimere en aktiveringsfunksjon med kombinert tilstand- og parameterestimering.
- Kapittel 7 gir en konklusjon av resultatene som er oppnådd i oppgaven.

Kapittel 2

Introduksjon til biologiske prosessmodeller

Mesteparten av stoffet i dette kapitlet er hentet fra [[6], [5], [7]]

2.1 Homeostase

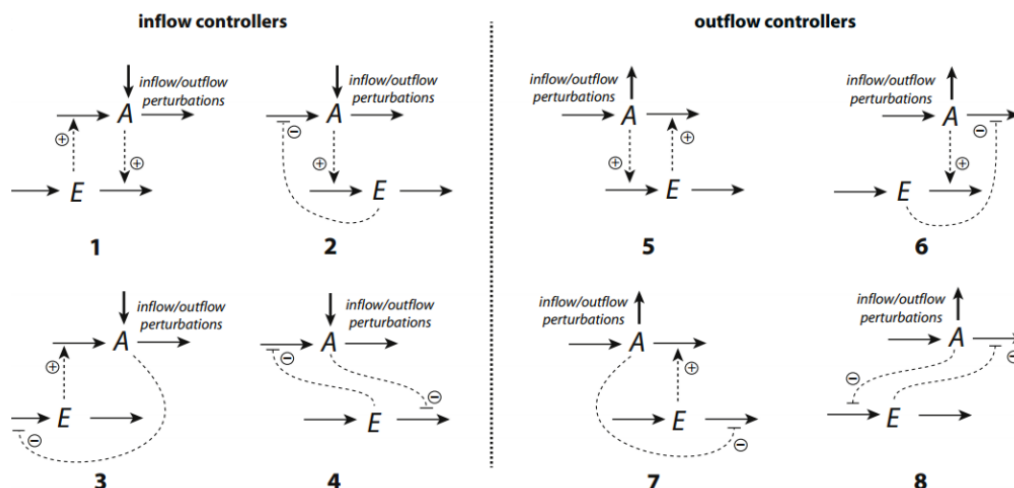
Homeostase er et begrep i fysiologien som beskriver en egenskap (for eksempel kroppstemperatur) som har for mål å holde en verdi tilnærmet konstant i en organismes indre miljø. Kroppstemperaturen til ett menneske ligger på cirka 37.5 °C, selv om vi oppholder oss på en varm strand, eller på ett kaldt fjell. At kroppen klarer å holde kroppstemperaturen tilnærmet konstant oppstår på grunn av homeostase i kroppens indre miljø.[6]. For å oppnå denne typen homeostase i en organisme, er organismens indre miljø bygd opp av kjemiske og biologiske strukturer som styrer de biologiske prosessene.

2.2 Kontrollmotiver

Et biokjemisk nettverk med regulerende egenskaper vil på sin enkleste form inkludere minst to komponenter. Disse to komponentene vil være en *kontrollert* variabel, kalt A , og en *manipulert* variabel, kalt E . Disse to variablene vil skape en negativ tilbakekobling med mål om å holde den *kontrollerte* variabelen i homeostase. I et slikt homeostatisk system kan A være et kjemisk stoff som må reguleres, mens E kan typisk være et hormon, enzym eller transportprotein med formål å regulere A . [5] Et eksempel på et slikt nettverk kan være transport proteinet IRT1 (E) som regulerer opptaket av jern (A) i planter. [7]

I figur 2.1 er det vist 8 biokjemiske strukturer med negativ tilbakekobling, fra nå av kalt kontrollmotiver. Disse kontrollmotivene er inndelt i 2 hovedgrupper, *inflow* og *outflow* kontrollere. *Inflow* kontrolleren regulerer fluksen inn, kalt *syntese*, mens *outflow* kontrolleren regulerer fluksen ut, kalt *degradasjon*. Hovedgruppene deles deretter inn i 2 undergrupper basert på om den *kontrollerte* variabelen blir aktivert (1,3,5,7) eller inhibert (2,4,6,8) av den *manipulerte* variabelen. [5]

De striplede linjene i 2.1 representerer samhandlingen mellom A og E , hvor et plusstegn ved siden av sier at variabelen blir aktivert av den andre, mens ett minustegn sier at variabelen blir inhibert av den andre. De heltrukne linjene inn til A og E , symboliserer syntese av stoffet, og bidrar til at mengden av stoffet øker. De heltrukne linjene ut fra A og E , symboliserer degradasjon av stoffet, og bidrar til at mengden av stoffet synker.



FIGUR 2.1: Skisse av 8 kontrollmotiver. I denne figuren blir syntese av A aktivert av E i 1 og 3, syntesen av A blir inhibert av E i 2 og 4, degradasjon av A blir aktivert av E i 5 og 7, og degradasjon av A blir inhibert av E i 6 og 8. Hentet fra [5]

2.3 Matematisk beskrivelse av biologiske prosessmodeller

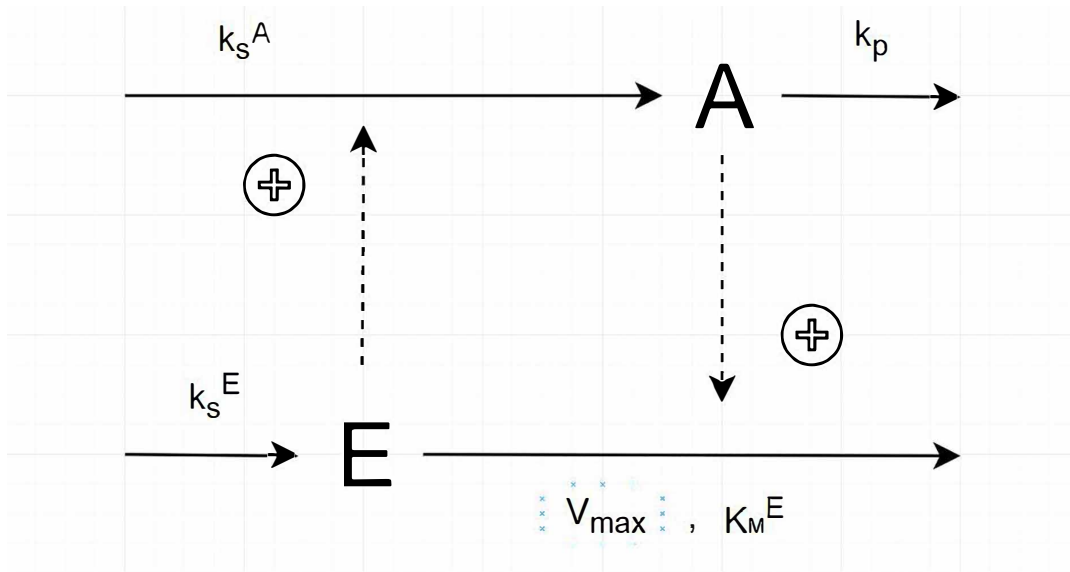
Figur 2.2 viser en skisse av en forenklet *inflow* kontroller 1. I denne skissen vil stoffet A aktivere degradasjonen av E , mens E vil aktivere syntesen av A . Ved bruk av enzym kinematikk er differensialligningene (2.1) og (2.2) funnet[5]:

$$\dot{A} = k_s^A \cdot E - k_p \cdot A \quad (2.1)$$

$$\dot{E} = k_s^E - \frac{V_{max} \cdot E}{K_m^E + E} \cdot A \quad (2.2)$$

Ved å anta at $E \gg K_m^E$ og $\dot{E} = 0$ kan vi finne det teoretiske arbeidspunktet for A ved likning (2.3) [5].

$$\dot{E} = k_s^E - \frac{V_{max} \cdot E}{K_m^E + E} \cdot A = 0 \rightarrow A_{set} = \frac{k_s^E}{V_{max}} \quad (2.3)$$



FIGUR 2.2: Skisse av kontroll motiv 1, inspirasjon til skisse hentet fra [5]

Kapittel 3

Introduksjon til Kalmanfilter

I dette kapitlet gis det en introduksjon til Kalmanfiltrering. Mesteparten av stoffet er hentet fra [4] [10]

3.1 Modell for prosessen

Det diskrete Kalmanfilteret beregner et optimalt tilstandsestimat for en prosess beskrevet med prosesslikningene for en diskret tilstandsrommodell.[4] [10]

$$x(k+1) = \Phi(k)x(k) + \Gamma(k)u(k) + \Omega(k)v(k), \quad (3.1)$$

$$y(k) = D(k)x(k) + E(k)u(k) + w(k). \quad (3.2)$$

Hvor:

- x er tilstandene
- u er pådragene
- y er målingene
- Φ er transmisjonsmatrisen
- Γ er pådragsmatrisen
- Ω er prosessforstyrrelsesmatrisen
- D er målematrisen
- E er direktekoblingsmatrisen
- v er prosesstøyen
- w er målestøyen.

3.2 Forutsetninger for bruk av Kalmanfilter

For at Kalmanfilteret skal være optimalt, bygger Kalmanfilteret på en del forutsetninger/antagelser. Det kreves at systemet er påvirket av både stokastisk

prosesstøy $v(k)$ og målestøy $w(k)$. Det antas også at både prosesstøyen og målestøyen har forventningsverdi lik:

$$E[v(k)] = m_v(k) = 0, \quad (3.3)$$

og

$$E[w(k)] = m_w(k) = 0. \quad (3.4)$$

og kovarianser lik:

$$R_v(\tau, k) = E[v(k + \tau)v^T(k)] = Q(k)\delta(k), \quad (3.5)$$

og

$$R_w(\tau, k) = E[w(k + \tau)w^T(k)] = R(k)\delta(k). \quad (3.6)$$

Q og R er nå kovariansmatriser for henholdsvis prosesstøyen $v(k)$ og målestøyen $w(k)$. Videre krever vi at sekvensene $v(k)$ og $w(k)$ er ukorrelerte:

$$E[v(k + \tau)w^T(k)] = 0. \quad (3.7)$$

til slutt er det et krav om at initialtilstanden $x(0)$ er en hvit stokastisk variabel med forventningsverdi:

$$E[x(0)] = m_0. \quad (3.8)$$

3.3 Kalmanfilter ligninger

Når vi har et system representert med prosesslikningene for diskret tilstandsrommodell (3.1) og (3.2), kan vi bruke den rekursive algoritmen for det diskrete Kalmanfilteret (3.9 - 3.13).

$$\bar{x}(k) = \Phi\hat{x}(k-1) + \Gamma u(k-1) \quad (3.9)$$

$$\bar{P}(k) = \Phi\hat{P}(k-1)\Phi^T + \Omega Q \Omega^T \quad (3.10)$$

$$K(k) = \bar{P}(k)D^T(D\bar{P}(k)D^T + R)^{-1} \quad (3.11)$$

$$\hat{x} = \bar{x}(k) + K(k)(y(k) - D\bar{x}(k)) \quad (3.12)$$

$$\hat{P}(k) = (I - K(k)D)\bar{P}(k) \quad (3.13)$$

3.4 Utvidet Kalmanfilter ligninger

I mange prosesser vil vi ikke ha et lineært system. Det utvidede Kalmanfilteret er en algoritme for å gjøre en lineær tilstandsestimering på et ulineært system. I det utvidede Kalmanfilteret beregnes nå Kalmanforsterkningen K og kovariansmatrisene \hat{P} og \bar{P} utifra den lineære modellen, mens \bar{x} og \bar{y} beregnes utifra den ulineære modellen. Det ulineære systemet er nå gitt av følgende modell:

$$x(k+1) = f[x(k), u(k), \dots] + v(k), \quad (3.14)$$

$$y(k) = g[x(k), \dots] + w(k). \quad (3.15)$$

Den rekursive algoritmen for det utvidede Kalmanfilteret er representert ved påfølgende likninger(3.16 - 3.22):

$$\bar{x}(k) = f[\hat{x}(k-1), u(k-1), \dots] \quad (3.16)$$

$$\Phi(k) = \left. \frac{\partial f(\cdot)}{\partial x} \right|_{\hat{x}(k-1), u(k-1)} \quad (3.17)$$

$$\bar{P}(k) = \Phi(k)\hat{P}(k-1)\Phi^T(k) + \Omega(k)Q\Omega^T(k) \quad (3.18)$$

$$K(k) = \bar{P}(k)D^T(D\bar{P}(k)D^T + R)^{-1} \quad (3.19)$$

$$\bar{y}(k) = g[\bar{x}(k), \dots] \quad (3.20)$$

$$\hat{x} = \bar{x}(k) + K(k)(y(k) - D\bar{x}(k)) \quad (3.21)$$

$$\hat{P}(k) = (I - K(k)D)\bar{P}(k) \quad (3.22)$$

3.5 Utregning av tilstandsrommodellmatriser

3.5.1 Transmisjonsmatrisen Φ og pådragsmatrisen Γ

For å finne transmisjonsmatrisen Φ og pådragsmatrisen Γ er det vanlig å først finne systemmatrisen A_{mat} og den kontinuerlige pådragsmatrisen B_{mat} .

Det utføres deretter diskretisering av A_{mat} og B_{mat} ved hjelp av nullteordens holdeelement og Eulers forovermetode. A_{mat} og B_{mat} finnes ved å linearisere systemets differensialligninger:

$$A_{mat} = \begin{bmatrix} \frac{\partial f_1(\cdot)}{\partial x_1} & \frac{\partial f_1(\cdot)}{\partial x_2} \\ \frac{\partial f_2(\cdot)}{\partial x_1} & \frac{\partial f_2(\cdot)}{\partial x_2} \end{bmatrix}$$

$$B_{mat} = \begin{bmatrix} \frac{\partial f_1(\cdot)}{\partial u} \\ \frac{\partial f_2(\cdot)}{\partial u} \end{bmatrix}$$

Videre finnes Φ ved likning:

$$\Phi = I + A_{mat}T \quad (3.23)$$

$$\Phi = \begin{bmatrix} 1 + A_{mat11}T & A_{mat12}T \\ A_{mat21}T & 1 + A_{mat22}T \end{bmatrix}$$

Og vi finner Γ ved likning:

$$\Gamma = B_{mat}T \quad (3.24)$$

$$\Gamma = \begin{bmatrix} B_{mat1}T \\ B_{mat2}T \end{bmatrix}$$

Der T representerer samplingstiden for Φ og Γ .

3.5.2 Målematrisen D og prosessforstyrrelsesmatrisen Ω

I denne oppgaven er det blitt brukt 3 ulike målematriser:

$$D_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Beskriver målematrisen der A er målt og E estimert

$$D_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Beskriver målematrisen der E er målt og A estimert

$$D_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Beskriver målematrisen der både A og E er målt. Når augmentert Kalmanfilter blir brukt er D utvidet iforhold til antall tilstandsvariable i systemet. Prosessforstyrrelsesmatrisen er i denne oppgaven valgt til:

$$\Omega = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Valget av denne prosessforstyrrelsesmatrisen tilsier at kovariansmatrisen Q forblir uforandret i likningene (3.10) og (3.18). Dette tilsier at kovariansen for prosesstøyen blir styrt direkte av verdiene for Q i denne rapportens eksperimenter. Når augmentert Kalmanfilter er brukt blir Ω utvidet iforhold til antall tilstandsvariable i systemet.

3.6 Observerbarhet

Observerbarhet er en egenskap ved et system, som beskriver sammenhengen mellom de tilstandene vi ønsker å estimere og variablene vi måler. Definisjonen av observerbarhet sier at: Kjennskap til pådragene $u(0), u(1), \dots, u(k-1)$ og målingene $y(0), y(1), \dots, y(k-1)$ er tilstrekkelig for å bestemme initialtilstanden $x(0)$, og kan skrives som:

$$\begin{bmatrix} D \\ D\Phi \\ \vdots \\ D\Phi^{k-1} \end{bmatrix} x(0) = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(k-1) \end{bmatrix} \quad (3.25)$$

$$\Updownarrow \quad (3.26)$$

$$Q_0 x(0) = Y \quad (3.27)$$

$x(0)$ er initial tilstanden til de n tilstandene i systemet. Det likning 3.27 forteller oss er at hvis *observerbarhetsmatrisen* Q_0 har rang n , og rangen til Q_0 er lik systemets *orden*, vil vi ha nok kunnskap gjennom de tilgjengelige pådragene og målingene til å finne initial tilstanden $x(0)$ og systemet er da observerbart.

Kapittel 4

Kalmanfilter anvendt på biologisk kontrollmotiv

I dette kapitlet er en modell for *inflow* kontroller 1 implementert i Simulink. Det har blitt implementert både et lineært Kalmanfilter og et utvidet Kalmanfilter for å estimere tilstandene A og E . Videre har det blitt utført ulike tester for å utforske hvordan algoritmen reagerer på valg av initialverdier, kovariansmatriser og forstyrrelser.

4.1 Implementering av kontrollmotiv i simulink

I dette delkapitlet vil det bli implementert en modell av *inflow kontroller 1* i Simulink.

4.1.1 Differensialligninger for systemet

Ligningene (4.1) og (4.2) beskriver kontrollmotivet: *inflow kontroller 1*. Dette er det samme motivet som er beskrevet i delkapittel 2.3, og vil i dette delkapitlet bli brukt for å implementere en modell i Simulink.

$$\dot{A} = k_s^A \cdot E - k_p \cdot A \quad (4.1)$$

$$\dot{E} = k_s^E - \frac{V_{max} \cdot E}{K_m^E + E} \cdot A \quad (4.2)$$

4.1.2 Valg av parameterverdier

Parameterverdiene i tabell 4.1 er valgt for å få resultater av en størrelsesorden hvor dynamikken til systemet skal komme godt frem i simuleringene. Disse verdiene gir ett arbeidspunkt i $[A, E, k_p] = [5, 5, 1]$, mens begrensningen av k_p vil holde tilstandene A og E godt over 0. Valget av samplingstiden T er gjort for unngå numeriske beregningsfeil i matlab.

Variabel	Verdi
k_s^A	1 mol/L/s
k_s^E	5 mol/L/s
V_{max}	1 mol/L/s
K_M^E	0.003 mol/L/s
k_p	[1 - 5] mol/L/s
T	0.01 s

TABELL 4.1: Verdier for variabler og konstanter brukt i kapittel 3 og 4

4.1.3 Implementasjon av motivet i Simulink

For å studere kompenseringsegenskapene til motivet er det blitt implementert en modell i Simulink. Figur 4.1 viser en simulering av implementasjonen. Det kan fra denne figuren ses hvordan A får ett fall når *inflow-forstyrrelsen* k_p øker, men vil bli regulert tilbake til settpunktet over en kort tidsperiode. Denne reguleringen kommer av at E kompenserer for innvirkningen av forstyrrelsen ved å øke i verdi, og skaper ett nytt arbeidspunkt hvor A har beholdt sin likevektsverdi, mens E og k_p nå har fått nye likevektsverdier.

Ved å sette inn verdier i ligningene (4.1) og (4.2) får vi :

$$\dot{A} = 0 = 1 \cdot 5 - 1 \cdot 5 \quad (4.3)$$

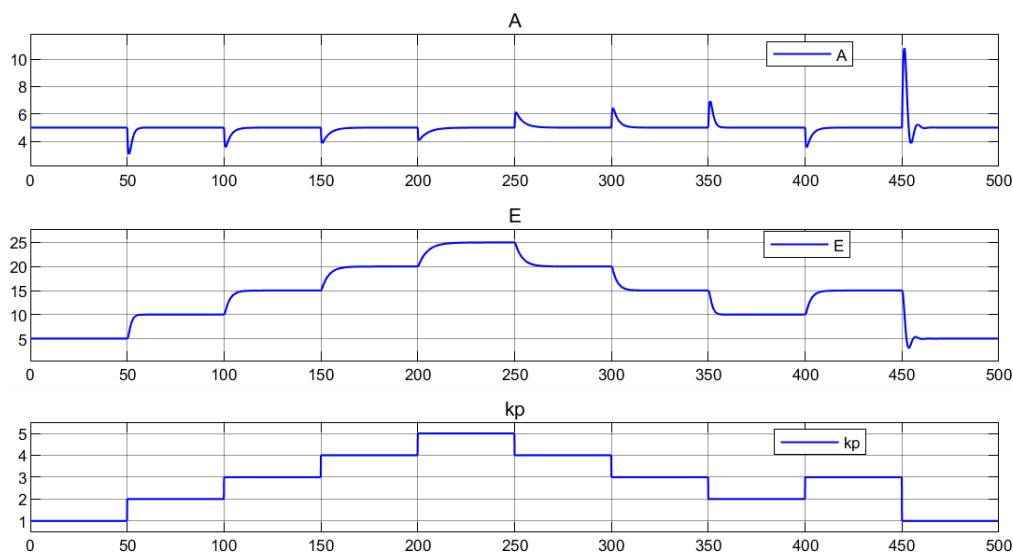
$$\dot{E} = 0 = 5 - \frac{1 \cdot 5}{(\approx 0) + 5} \cdot 5 \quad (4.4)$$

for arbeidspunktet $[A, E, k_p] = [5, 5, 1]$, der K_M^E er tilnærmet lik 0. Denne beregningen kan ses i figur 4.1 fra tidspunkt 0 til 50. I tidspunkt 50 er det lagt inn ett sprang i forstyrrelsen fra 1 til 2. A vil få et kraftig fall, og E vil kompensere for denne økningen i forstyrrelsen, ved å selv øke i verdi. Det vil da bli beregnet ett nytt arbeidspunkt : $[A, E, k_p] = [5, 10, 2]$ som vil gi:

$$\dot{A} = 0 = 1 \cdot 10 - 2 \cdot 5 \quad (4.5)$$

$$\dot{E} = 0 = 5 - \frac{1 \cdot 10}{(\approx 0) + 10} \cdot 5 \quad (4.6)$$

der K_M^E er tilnærmet lik 0. Denne beregningen vil gi \dot{A} og $\dot{E} = 0$ for alle valgene av k_p vist i denne figuren. Valget av $K_M^E \approx 0$ gjøres i dette eksempelet for å unngå desimalverdier for beregningene.



FIGUR 4.1: Simulink scope av kontrollmotivet: inflow kontroller 1

4.1.4 Inkludering av prosess- og målestøy

I mange systemer vil ikke prosessen foregå nøyaktig slik som den matematiske modellen. Denne uforutsigbarheten representeres ofte ved hjelp av en prosesstøy. Når det skal utføres en måling, er vi nødt til å bruke et måleinstrument, usikkerheten i dette instrumentet er det vi kaller målestøy. Inkluderingen av disse støyene er naturlig for å få ett system som på best måte representerer virkeligheten. Inkluderingen av disse støyene er også en av forutsetningene satt i kapittel 3.2 for å bruke ett Kalmanfilteret for tilstandsesimering. I Simulink er *Band-Limited White Noise* blokken brukt for å legge til støy i systemet. Denne blokken gir en støy med middelværdi lik 0 for begge støyene. I dette kapitlet blir prosesstøysens varians valgt til $1 \cdot 10^{-3}$. Når støy legges til i modellen utvides differensialligningene til:

$$\dot{A} = k_s^A \cdot E - k_p \cdot A + v_1 \quad (4.7)$$

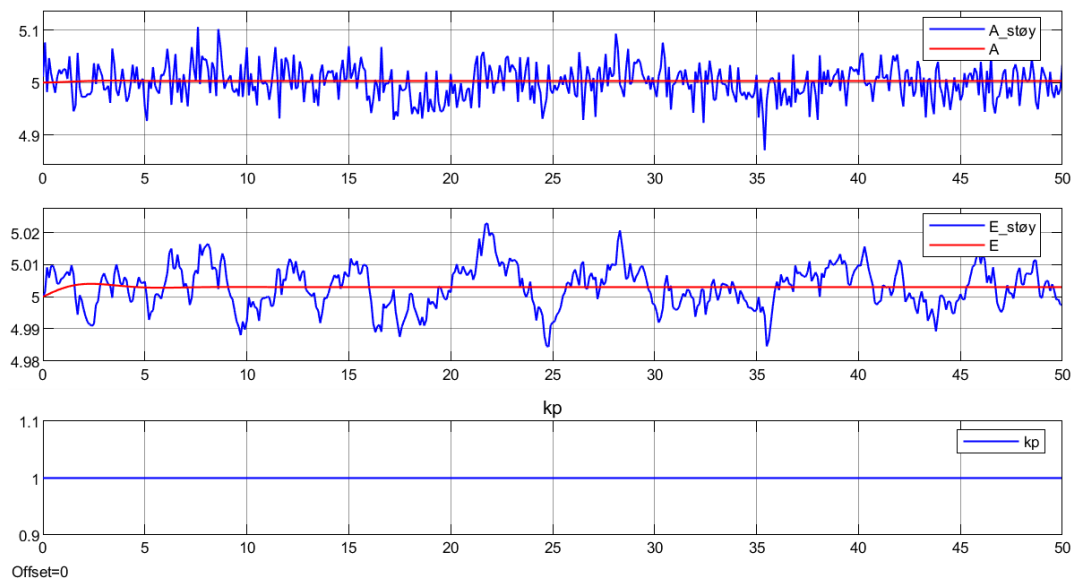
$$\dot{E} = k_s^E - \frac{V_{max} \cdot E}{K_m^E + E} \cdot A + v_2 \quad (4.8)$$

der v_1 og v_2 representerer prosesstøyen for de 2 tilstandene i systemet.

Figur 4.2 viser en sammenligning av de 50 første sekundene av motivet inkludert støy, representert ved blå linje, og motivet uten støy, representert ved rød linje. Det kan ses fra figuren at på grunn av inkluderingen av støy vil responsen til A og E ikke lengre ligge i likevekt i arbeidspunktet: $[A, E, k_p] = [5, 5, 1]$, men vil variere rundt arbeidspunktet.

Prosesstøy	Varians	Middelverdi
v_1	$1 \cdot 10^{-3}$	0
v_2	$1 \cdot 10^{-3}$	0
Målestøy	Varians	Middelverdi
w_1	$1 \cdot 10^{-3}$	0

TABELL 4.2: Verdier for støyen lagt til i systemet



FIGUR 4.2: Simulink scope av kontrollmotivet: inflow kontroller 1 inkludert støy. Blå linje: motiv med støy. Rød linje: motiv uten støy.

4.2 Implementering av Kalmanfilter for motivet

I dette delkapittelet vil de bli implementert ett lineært Kalmanfilter for å estimere tilstandene A og E i Simulink.

4.2.1 Valg av initielle verdier for Kalmanfilteret

For å kunne bruke Kalmanfilteralgoritmen fra delkapittel 3.2 er det nødt til å defineres initielle verdier for parametere i algoritmen. Parameterne som må settes for det lineære kalman filteret er:

- Initielle verdier for tilstandene, $x(0)$.
- Initielle verdier for pådragene, $u(0)$.
- Initielle verdier for a posteriori kovariansmatrisen $\hat{P}(0)$.
- Verdier for kovariansmatrisene til støyen, Q og R .

De første verdiene som må initialiseres er tilstandene A og E , tillegg til pådraget k_p , i denne rapporten kalt forstyrrelsen. Initial verdien for A finnes først ved ligning (2.3), som gir:

$$A_{set} = \frac{k_s^E}{V_{max}} = \frac{5}{1} = 5. \quad (4.9)$$

For å finne E trenger vi verdien for forstyrrelsen k_p , i testene gjort i dette kapittelet starter forstyrrelsen k_p alltid i 1, og initialverdien for k_p velges da til 1. Ved å sette inn verdiene for k_p og A i ligning (2.1) får vi:

$$0 = 5 - \frac{1 \cdot E}{(\approx 0) + E} \cdot 5 \rightarrow E = 5. \quad (4.10)$$

som gir oss arbeidspunktet $[A, E, k_p] = [5, 5, 1]$.

I denne oppgaven er støyen lagt til i systemet med på forhand valgte verdier bestemt i delkapittel 4.1.4. Kovariansmatrisene til støyen velges da lik variansen til støyen fra tabell 4.2, og vi får:

$$Q = \begin{bmatrix} 1 \cdot 10^{-3} & 0 \\ 0 & 1 \cdot 10^{-3} \end{bmatrix}$$

og:

$$R = 1 \cdot 10^{-3}$$

Den siste initielle verdien som må bestemmes er a posteriori kovariansmatrisen $\hat{P}(0)$. Denne verdien er ikke like lett å velge, og må ofte bruke prøve og feile metoden for å finne en optimal verdi. Det denne matrisen sier noe om, er hvor usikker man er i valget av initielle verdier for tilstandene. Hvis det er stor usikkerhet i dette valget, velges ofte $\hat{P}(0)$ til [4]:

$$\hat{P}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Dette valget av $\hat{P}(0)$ vil føre til at Kalmanforsterkningen for den målte tilstanden vil starte nære 1, mens Kalmanforsterkningen for tilstanden som ikke er målt vil starte nær 0. Det at Kalmanforsterkningen starter nær 1 for den målte tilstanden vil føre til at den første verdien som blir estimert for den målte tilstanden, ligger svært nære den faktiske målingen. Det at Kalmanforsterkningen for den ikke målte tilstanden starter nær 0, betyr at det første estimatet vil ligge svært nær estimatet funnet gjennom modellen. Etter litt prøving og feiling velges $\hat{P}(0)$ til:

$$\hat{P}(0) = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.1 \end{bmatrix}$$

Dette valget av $\hat{P}(0)$ vil føre til at Kalmanforsterkningen for både A og E vil starte på verdier nærmere likevektsverdiene for Kalmanfilterforsterkningen og føre til raskere konvergens mot likevektsverdien. Dette vil føre til at både A og E sin første estimerte verdi vil starte nærmere det optimale tilstandsesstimatet, og konvergere raskere mot dette estimatet.

4.2.2 Matematisk modellering

For å kunne implementere Kalmanfilteret i Simulink må først transmisjonsmatrisen Φ og pådragsmatrisen Γ beregnes. Først beregnes systemmatrisen A_{mat} og pådragsmatrisen B_{mat} ved linearisering av differensialligningene (4.7) og (4.10) rundt arbeidspunktet $[A, E, k_p] = [5, 5, 1]$. Hvordan dette utføres er vist i delkapittel 3.5.1. For å utføre linearisering brukes *diff* funksjonen i matlab og A_{mat} og B_{mat} finnes som:

$$A_{mat} = \begin{bmatrix} -1 & 1 \\ -0.9994 & -0.0005 \end{bmatrix}$$

$$B_{mat} = \begin{bmatrix} -5 \\ 0 \end{bmatrix}$$

Deretter brukes ligningene (3.23) og (3.24) for å finne Φ og Γ . Disse beregnes til å bli:

$$\Phi = \begin{bmatrix} 0.99000 & 0.01000 \\ -0.00999 & 0.99999 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} -0.05 \\ 0 \end{bmatrix}$$

Etter at Φ og Γ er funnet må det utføres en observerbarhetstest. Denne testen er utført ved å bruke *obsv*(Φ, D) i matlab som gir:

$$Q_0 = \begin{bmatrix} 1 & 0 \\ 0.99 & 0.01 \end{bmatrix}$$

Ved å bruke Matlab funksjonen *rank*(Q_0) får vi en rang lik 2. Siden vi har 2 tilstandsvariable, A og E vil antall tilstandsvariable som skal estimeres være lik rangen til Q_0 og systemet vil være observerbart.

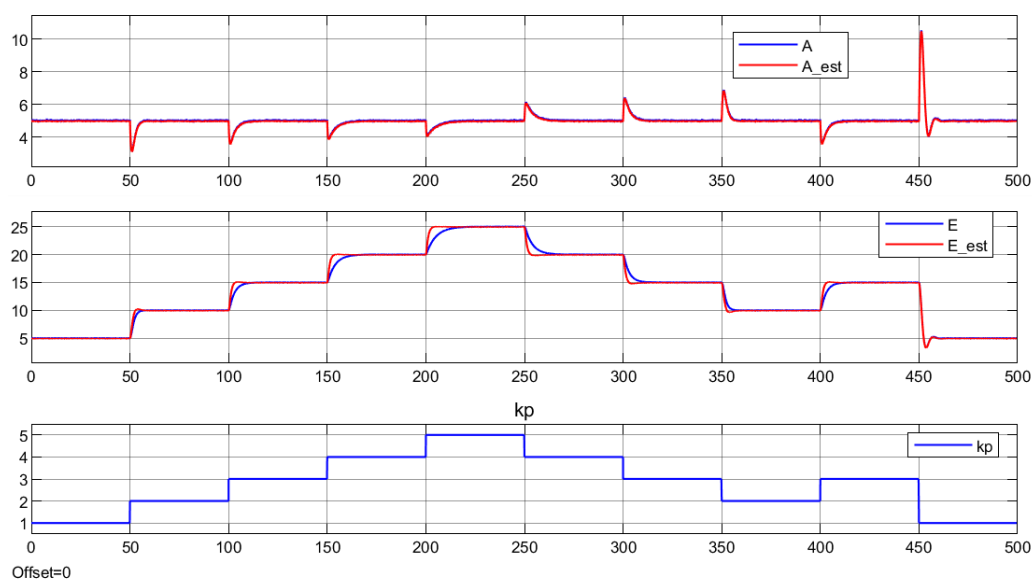
Verdiene i disse beregningene er avrundet, men blir mer nøyaktig beregnet i matlab og Simulink. Eksempler på implementasjonen av lineariseringen og observerbarhetstesten kan finnes i vedlegg D.

4.2.3 implementering av Kalmanfilteret i Simulink

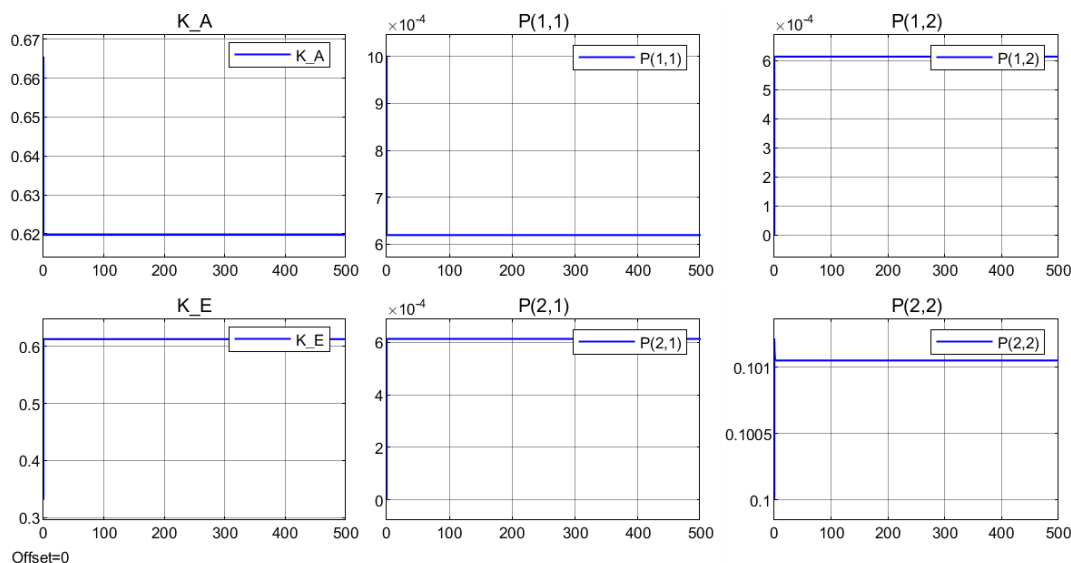
Etter en simulering av Kalmanfilteret med initialverdiene valgt i forrige seksjon, produseres grafene 4.3 og 4.4. I figur 4.3 er estimatene av variablene A og E vist sammen med verdiene fra modellen av kontrollmotivet, mens i figur 4.4 er utviklingen av K og \hat{P} vist.

Ved å se på utviklingen av \hat{P} og K i figur 4.4 kan vi se hvordan den initiale gjetningen av Kalmanforsterkning for A , kalt K_A , ligger mellom 0.66 og 0.67, og vil på få sekunder konvergere til likevektsverdien på cirka 0.62. For Kalmanforsterkningen for E , kalt K_E kan det ses at den initiale gjetningen ligger på mellom 0.3 og 0.4 for denne også raskt konvergerer mot en likevektsverdi på cirka 0.62. At Kalmanforsterkningen for A og E konvergerer mot den samme likevektsverdien er naturlig da prosesstøyen for både A og E er den samme.

Ser vi på utviklingen av A og E i figur 4.3 kan vi se hvordan Kalmanfilteret gir en god estimering av A over hele simuleringen. Ser vi på utviklingen av E kan det ses at kalmanfilteret gir et estimeringsfeil ved tidene der forstyrrelsen k_p blir utsatt for sprang. Det kan også ses at estimeringsfeilen blir større desto lengre vekk fra den originale arbeidspunktet estimatet kommer.



FIGUR 4.3: Simulink scope av A , E og k_p for et lineært Kalmanfilter



FIGUR 4.4: Simulink scope av K og \hat{P} for et lineært Kalmanfilter

Grunnen til denne estimeringsfeilen ligger i ligningene (3.9 - 3.13). Φ og Γ beregnes i disse likningene utifra ett på forhånd satt arbeidspunkt. Dette arbeidspunktet ble satt til $[A, E, k_p] = [5, 5, 1]$ i delkapittel 4.2.1, og brukt til lineariseringen i delkapittel 4.2.2. Arbeidspunktet for A varierer ikke med forstyrrelsen k_p , så denne forblir alltid 5, mens arbeidspunktet for E avhenger av forstyrrelsen. På grunn av ulineariteten i systemet vil det oppstå en estimeringsfeil rundt sprangene som vil bli større desto lengre vekk fra det originale arbeidspunktet estimatene kommer. For å løse dette problemet vil det i neste delkapittel bli implementere et utvidet Kalmanfilter som tar høyde for denne ulineariteten i systemet.

4.3 Utvidet Kalman filter

I denne seksjonen vil det bli utviklet ett utvidet Kalmanfilter for å prøve å forbedre estimatene rundt sprang i forstyrrelsen.

4.3.1 Matematisk modellering

For å implementere det utvidede Kalmanfilteret i Simulink, må vi på samme måte som i forrige delkapittel finne transmisjonsmatrisen Φ , men det er ikke lengre bruk for pådragsmatrisen Γ da a priori estimeringen ikke lengre er avhengig av denne matrisen. For det utvidede Kalmanfilteret lineariseres det ikke lengre rundt ett satt arbeidspunkt, men variablene forblir ukjente og beregnes for hver iterasjon av den rekursive algoritmen. Får ved å bruke *diff* i matlab matrisene:

$$A_{mat} = \begin{bmatrix} -k_p & 1 \\ -\frac{E}{E+0.003} & \frac{AE}{(E+0.003)^2} - \frac{A}{E+0.003} \end{bmatrix}$$

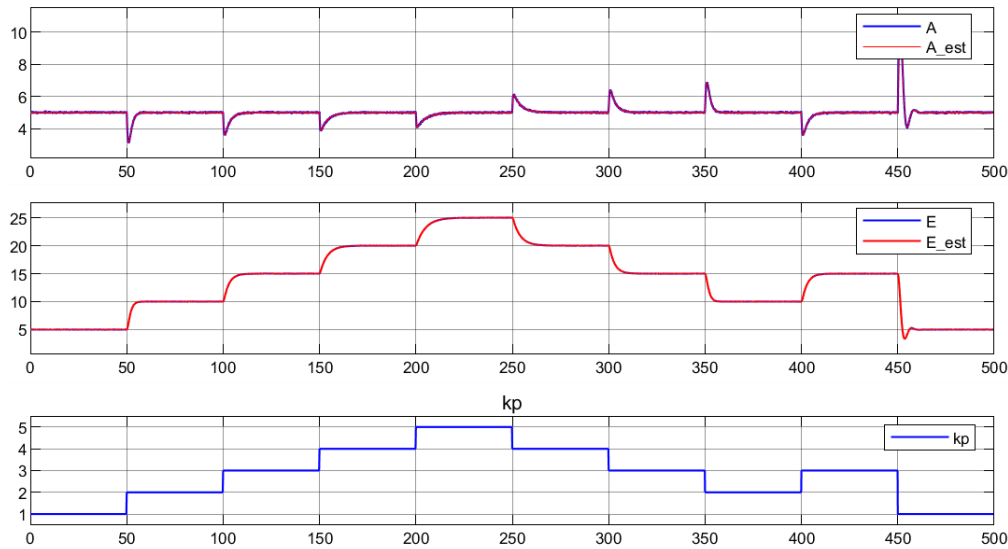
og fra A_{mat} finnes:

$$\Phi = \begin{bmatrix} 1 + (0.01 \cdot A_{11}) & 0.01 \cdot A_{12} \\ 0.01 \cdot A_{21} & 1 + (0.01 \cdot A_{22}) \end{bmatrix}$$

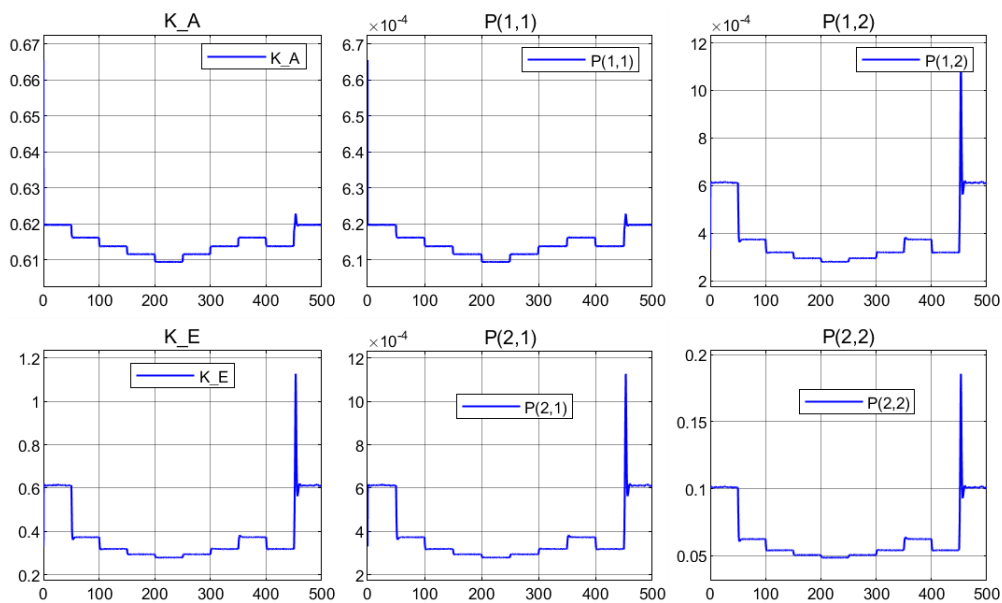
Etter at Φ er funnet utføres det på samme måte som for det lineære Kalmanfilteret en observerbarhetstest. Denne observerbarhetstesten utføres på samme arbeidspunkt som i delkapittel 4.2.2 og sier at systemet er observerbart. Denne observerbarhetstesten er her utført i ett arbeidspunkt, men må i teorien utføres i alle. I denne oppgaven er observerbarhetstestene for utvidede Kalmanfilter først gjort for ett arbeidspunkt, for deretter å sjekke andre arbeidspunkt der det føles nødvendig.

4.3.2 implementering av det utvidede Kalmanfiltert

Ved implementeringen av det utvidede Kalmanfilteret er de samme initialverdiene valgt som for det lineære Kalmanfilteret. Ved å se på det nye estimatet for E i figur 4.5 kan det nå ses at estimeringsfeilen i sprangene til forstyrrelsen er forsvunnet. Dette kan forklares ved å se på likningene (3.14 - 3.22). I motsetning til det lineære Kalmanfilteret er nå Φ varierende med tiden, og vil føre til at Kalmanforsterkningen også vil forandre seg med tiden. Fra figur 4.6 kan det ses hvordan \hat{P} og K ikke lengre konvergerer mot en spesifikk verdi, men forandrer seg når Φ forandrer seg. Ved å sammenligne 4.5 og 4.3 kan det konkluderes med at det utvidede Kalmanfilteret fungerer best for å estimere tilstandene til det biologiske kontrollmotivet.



FIGUR 4.5: Simulink scope av A , E og k_p for et utvidet Kalmanfilter



FIGUR 4.6: Simulink scope av K og \hat{P} for et utvidet Kalmanfilter

4.4 Tuning av initial verdier for det utvidede Kalmanfilteret

I seksjon 4.2.1 ble det valgt initialverdier for Kalmanfilteret som var nær de teoretisk optimale start verdiene for kalman filteret. For de fleste prosesser er

svært nøyaktige initialverdier vanskelig å oppnå. I dette kapittelet vil det bli gjort forsøk for å se hvordan Kalmanfilteret responderer til avvik i de ulike initielle verdiene, og hvilke innvirkninger avvik vil få for estimeringen.

4.4.1 Avvik i initialverdien for tilstandene A og E

Simuleringer fra disse forsøkene kan bli sett i vedlegg [A.1](#) og [A.2](#).

I mange prosesser vil det ikke være mulig å velge nøyaktige initielle verdier for tilstandene. Valget av a posteriori kovariansmatrisen $\hat{P}(0)$ henger også kraftig sammen med valget av de initielle verdiene for A og E . Diagonalen til $\hat{P}(0)$ sier noe om hvor usikkert den initielle gjetningen av tilstandene er. Et valg av $\hat{P}(0)$ til :

$$\hat{P}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

vil føre til at Kalmanforsterkningen til den målte verdien vil starte nær 1, og algoritmen vil derfor velge et første estimat nær den målte verdien. Kalmanforsterkningen til den ikke målte verdien vil starte nær 0, og algoritmen vil derfor velge et første estimat kun basert på modellen. Dette vil gi en «fleksibel» gjetning av initielle verdier, og vil føre til at uansett valg av initialverdi for tilstandene A eller E vil det første estimatet av den målte tilstanden A alltid være ett godt estimat.

For den ikke målte tilstanden vil en fleksibel $\hat{P}(0)$ føre til at det første estimatet av den ikke målte tilstanden E vil følge kun modellen. Dette vil føre til uansett valg av initialverdier for tilstandene vil den ikke målte tilstanden E få et mer ustabil første estimat.

Et annet valg av $\hat{P}(0)$ vil være å forvente gode initielle verdier for tilstandene, da velges $\hat{P}(0)$ som:

$$\hat{P}(0) = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.1 \end{bmatrix}$$

Fordelen med å bruke denne matrisen ligger i at så lange valget av initialverdier for tilstandene A og E er nøyaktige, vil både den målte tilstanden A og den ikke målte tilstanden E ha gode estimater fra første estimat. Ulempen med denne matrisen ligger i at avviket i initialverdi for A og E vil bli tydeligere desto mer avvik vi har. Et valg av en mer «spesifikk» $\hat{P}(0)$ vil også føre til en tregere konvergens mot det riktige estimatet enn det ett «fleksibelt» valg vil.

Det disse forsøkene viser er hvordan valget av initialverdier for tilstandene og initialverdi for a posteriori kovariansmatrisen henger sammen. Det kan ses hvordan jo mer usikker man er på initialverdiene for tilstandene, desto mer «fleksibelt» valg av a posteriori kovariansmatrise burde velges. Dette på grunnlag av både nøyaktighet og hurtig konvergens. Det kan også ses at hvis

initialverdiene for tilstandene har svært nøyaktige verdier, vil en «spesifikk» a posteriori kovariansmatrise alltid være best.

4.5 Valg av kovariansmatrisene Q og R

Ved å se på ligningene for både Kalmanfilteret og det utvidede Kalmanfilteret er det tydelig at både Q og R spiller en viktig rolle i beregningen av Kalmanforsterkningen K . I seksjon 4.2.1 ble Q valgt til:

$$Q = \begin{bmatrix} 1 \cdot 10^{-3} & 0 \\ 0 & 1 \cdot 10^{-3} \end{bmatrix}$$

og R valgt til

$$R = 1 \cdot 10^{-3}$$

Disse verdiene ble valgt på bakgrunn av variansene til v_1 , v_2 og w_1 . I mange prosesser vil variansene til disse støyene være vanskelig å fastslå nøyaktige verdier for. I dette delkapittelet vil det bli gjort forsøk for å teste hvordan valgene av kovariansmatrisene for støyen påvirker estimatene for tilstandene.

4.5.1 Q som tuningsparameter

Kalmanforsterkningen K avhenger både av prosesstøyvariansen Q og målestøyvariansen R . I de fleste tilfeller vil målestøyvariansen være enklest å sette mest nøyaktig, da denne ofte avhenger av et måleinstrument hvor støyen/usikkerheten kan testes. Det er betydelig vanskeligere å anslå prosesstøyvariansen, og man tyr derfor ofte til prøving og feiling for å finne denne verdien. I teorien kan både Q og R tenkes som tuningsparametere [4], så i dette delkapittelet vil det bli sett på hvilke innvirkning forandringer av både Q og R vil få for estimatet.

Fra ligning 3.18 og 3.19 kan det ses hvordan Q gjennom \bar{P} vil påvirke Kalmanforsterkningen. Ved å velge en «stor» Q iforhold til R , som tilsier at vi har mer varians i prosesstøyen enn målestøyen i systemet, vil vi få en «stor» Kalmanforsterkning. Hvis det velges en «liten» Q iforhold til R , som tilsier at vi har mindre varians i prosesstøyen enn målestøyen i systemet, vil vi få en «liten» Kalmanforsterkning. Innvirkningen Kalmanforsterkningen har på systemet er at den beskriver hvor stor andel av avviket mellom den målte verdien og a priori estimatet av den målte verdien ($y(k) - D\bar{x}(k)$ (målingen - a priori estimatet), som skal påvirke de ulike tilstandene som skal estimeres. R vil på grunnlag av det forklart over fungere på motsatt måte av Q .

I dette tilfellet er variansene til prosess- og målestøyen lik. Utifra det sagt over kan det nå forventes en Kalmanforsterkning vil få en «middels» verdi. Det kan også forventes at begge tilstandene vil få en svært lik Kalmanforsterkning. Dette kan vi også se at algoritmen gir i figur 4.6 hvor vi får en

Kalmanforsterkning for både A og E på cirka 0.62 i det første arbeidspunktet.

4.5.2 Tuning av parametrene

I de tidligere delkapitlene er valget definert som: $[Q, R] = [\text{«liten»}, \text{«liten»}]$ valgt. I dette forsøket vil de andre 4 kombinasjonene av «liten» og «stor» verdi for Q og R bli sett på, og sammenlignet. I dette forsøket er verdien for «liten» satt til $1 \cdot 10^{-3}$, mens verdien for «stor» er satt til 1.

I figurene (4.7 - 4.9) er henholdsvis verdiene for A , E og K vist. I grafen øverst til venstre for A og E er $[Q, R] = [\text{«liten»}, \text{«liten»}]$ vist, mens K er representert ved blå strek i 4.9 (blå og svart strek ligger her over hverandre). Dette vil si at det forventes like lite prosess- og målestøy i systemet. I denne simuleringen er da :

$$Q = \begin{bmatrix} 1 \cdot 10^{-3} & 0 \\ 0 & 1 \cdot 10^{-3} \end{bmatrix}, R = 1 \cdot 10^{-3}$$

Ved å se på grafen for A kan vi se hvordan estimatet følger modellen på en god måte, mens den itillegg gir en liten støyreduksjon. Ved å se på verdien for K kan vi se at denne ligger rundt 0.62 som vil si at cirka 62% av feilen mellom a priori estimatet og målingen vil bli lagt til a priori estimatet for å danne a posteriori estimatet (ligning (3.21)). Ved å se på grafen for E kan det her ses at det blir lagt på litt ekstra støy i estimatet i forhold til modellen.

I grafen øverst til høyre for A og E er $[Q, R] = [\text{«stor»}, \text{«liten»}]$ vist, mens K er representert ved grønn strek i 4.9. Dette vil si at det forventes mye prosesstøy og lite målestøy i systemet. I denne grafen er:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 1 \cdot 10^{-3}$$

Ved å se på grafen for A vil estimatet utvikle seg helt likt med modellen. Som vil si at algoritmen velger å følge målingene cirka 100%. Dette kan også ses igjen fra den grønne linjen i figur 4.9 hvor verdien ligger veldig nære 1. Ved å se på grafen for E kan vi se at det her er blitt mer overestimering enn for $[Q, R] = [\text{«liten»}, \text{«liten»}]$.

I grafen nederst til venstre for A og E er $[Q, R] = [\text{«liten»}, \text{«stor»}]$ vist, mens K er representert ved rød strek i 4.9. Dette vil si at vi forventer svært lite prosesstøy, men mye målestøy i systemet. I denne grafen er:

$$Q = \begin{bmatrix} 1 \cdot 10^{-3} & 0 \\ 0 & 1 \cdot 10^{-3} \end{bmatrix}, R = 1$$

Det kan her ses fra grafen for A at det nå har blitt fjernet nesten alt av støy. Verdien for K i 4.9 ligger svært nær 0 som vil si at algoritmen legger svært

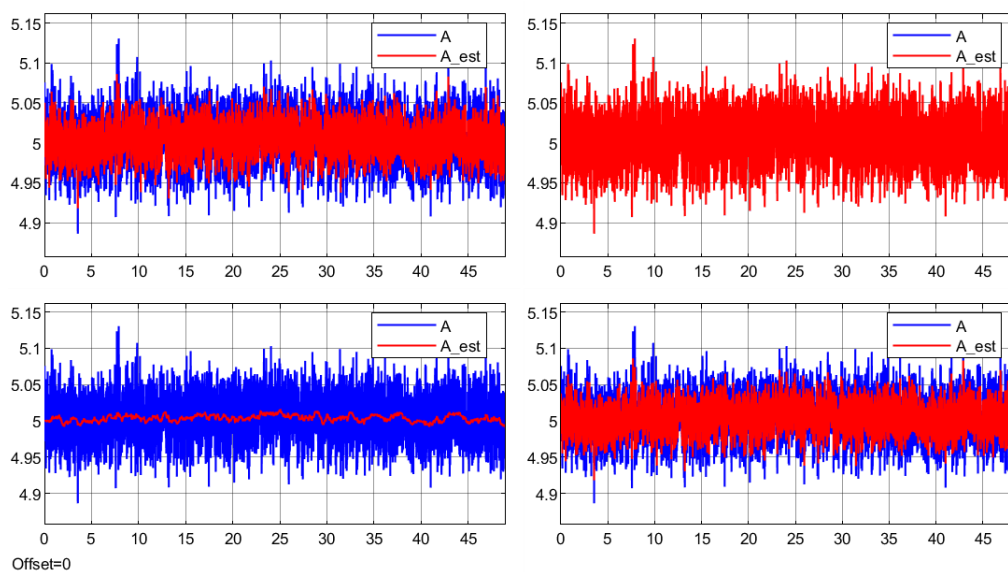
liten vekt på målingen, og mye vekt på modellen. I grafen for E kan det nå ses hvordan vi har fått en bedre estimering av E enn for $[Q, R] = [\text{«liten»}, \text{«liten»}]$.

Den siste grafen nederst til høyre for A og E er $[Q, R] = [\text{stor}^{\prime\prime}, \text{stor}^{\prime\prime}]$ vist, mens K er representert ved svart strek i 4.9. Dette vil si at vi forventer like mye prosess- og målestøy, men av en større mengde enn i $[Q, R] = [\text{liten}^{\prime\prime}, \text{liten}^{\prime\prime}]$. I denne grafen er :

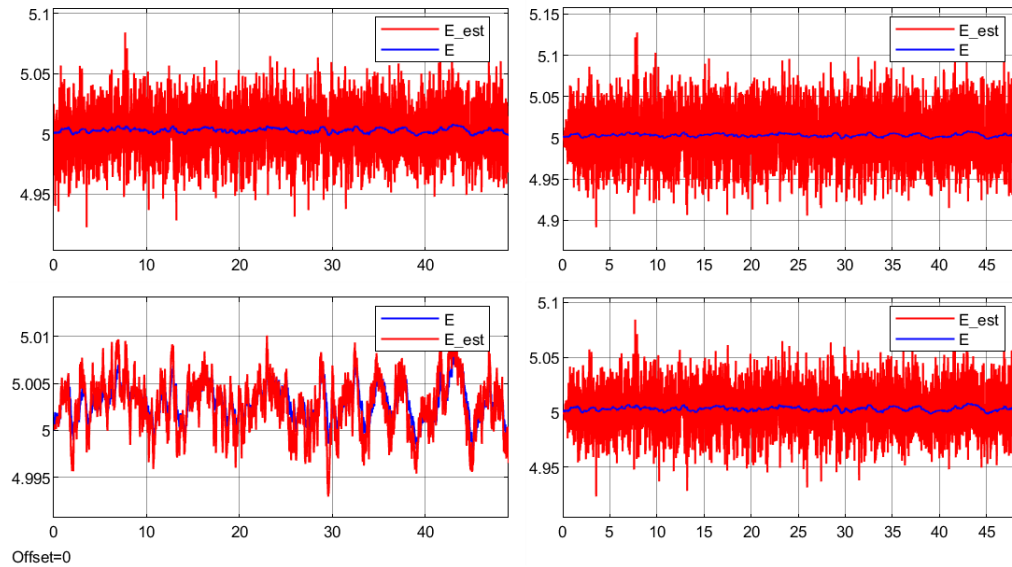
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 1.$$

Ved å se på grafene kan vi her se at algoritmen gir nesten identiske resultater mellom $[\text{stor}^{\prime\prime}, \text{stor}^{\prime\prime}]$ og $[\text{liten}^{\prime\prime}, \text{liten}^{\prime\prime}]$. Dette er fordi forholdet mellom verdiene i Q og R er her helt like.

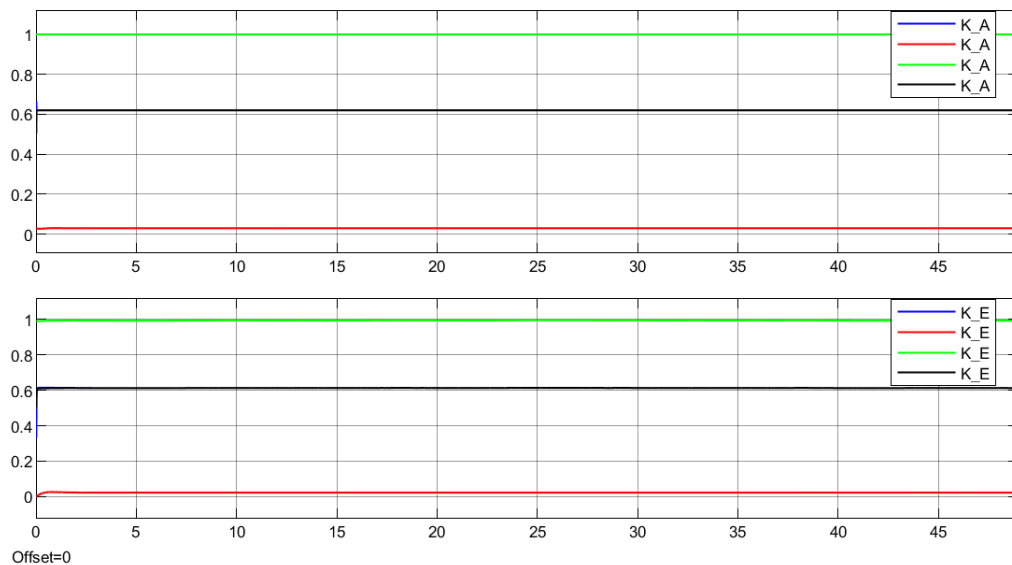
Etter utføringen av disse forsøkene kan man se at å tune Q og R etter variansen til støyen (vist i grafene som $[\text{«liten»}, \text{«liten»}]$) vil gi en støy redusert A , men en støy økning i E . Mens ett valg av «liten» Q og «stor» R gir en bedre estimering av E , men dårligere estimering av A . Dette vil bli undersøkt nærmere i neste delkapittel.



FIGUR 4.7: Simulink scope av A for ulike verdier av Q og R . Rød strek: estimatet av A , blå strek: A fra modellen. Oppe venstre: $[Q, R] = [\text{liten}^{\prime\prime}, \text{liten}^{\prime\prime}]$, oppe høyre $[Q, R] = [\text{stor}^{\prime\prime}, \text{liten}^{\prime\prime}]$, nede venstre: $[Q, R] = [\text{liten}^{\prime\prime}, \text{stor}^{\prime\prime}]$, nede høyre: $[Q, R] = [\text{stor}^{\prime\prime}, \text{stor}^{\prime\prime}]$



FIGUR 4.8: Simulink *scope* av E for ulike verdier av Q og R . Rød strek: estimatet av E , blå strek: E fra modellen. Oppe venstre: $[Q,R] = [\text{liten}, \text{liten}]$, oppe høyre $[Q,R] = [\text{stor}, \text{liten}]$, nede venstre: $[Q,R] = [\text{liten}, \text{stor}]$, nede høyre: $[Q,R] = [\text{stor}, \text{stor}]$



FIGUR 4.9: Simulink *scope* av K for ulike verdier av Q og R . Grønn strek: $[Q,R] = [\text{stor}, \text{liten}]$, rød strek: $[Q,R] = [\text{liten}, \text{ss-tor}]$, blå strek: $[Q,R] = [\text{liten}, \text{liten}]$, svart strek: $[Q,R] = [\text{ss-tor}, \text{stor}]$

4.5.3 Tuning av Q og R for biologiske kontrollmotiver

I delkapittel 4.1.3 ble det vist hvordan kompenseringsegenskapene til kontrollmotivet holder A tilnærmet konstant, mens E øker og synker for å kompensere for forstyrrelser som påvirker A . Støy med den samme variansen blir lagt til for både prosess- og målestøy, som vist i figur 4.10, men i figur 4.11 kan vi se hvordan kun målestøyen blir fullt lagt til for systemet, mens den «effektive» prosesstøyen blir langt lavere for både A og E . Etter en nærmere test viser det seg at den «effektive» variansen for prosesstøyen i systemet er:

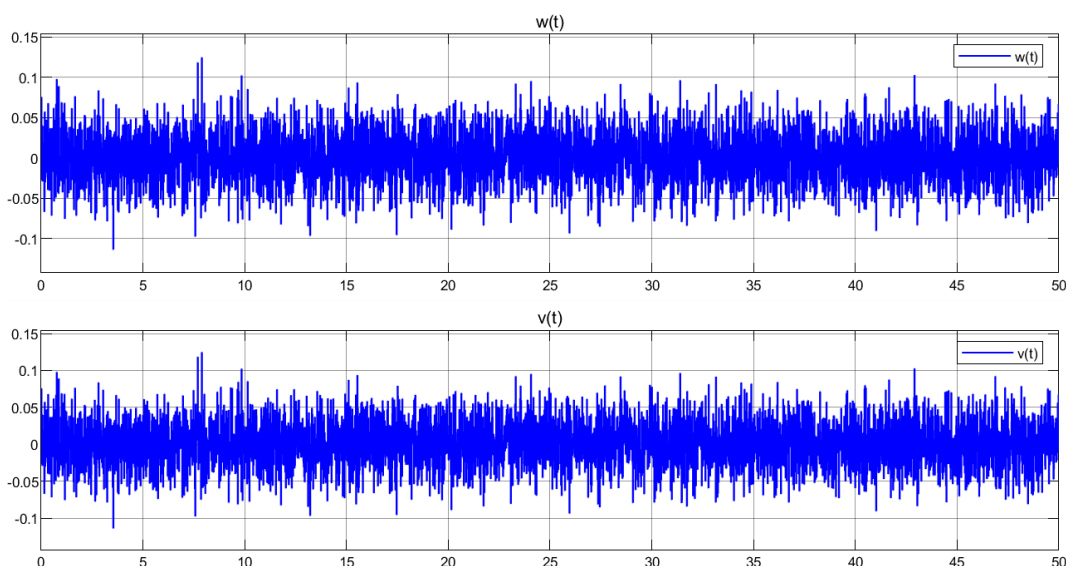
$$v_1 = 1 \cdot 10^{-5} \quad (4.11)$$

$$v_2 = 5 \cdot 10^{-6} \quad (4.12)$$

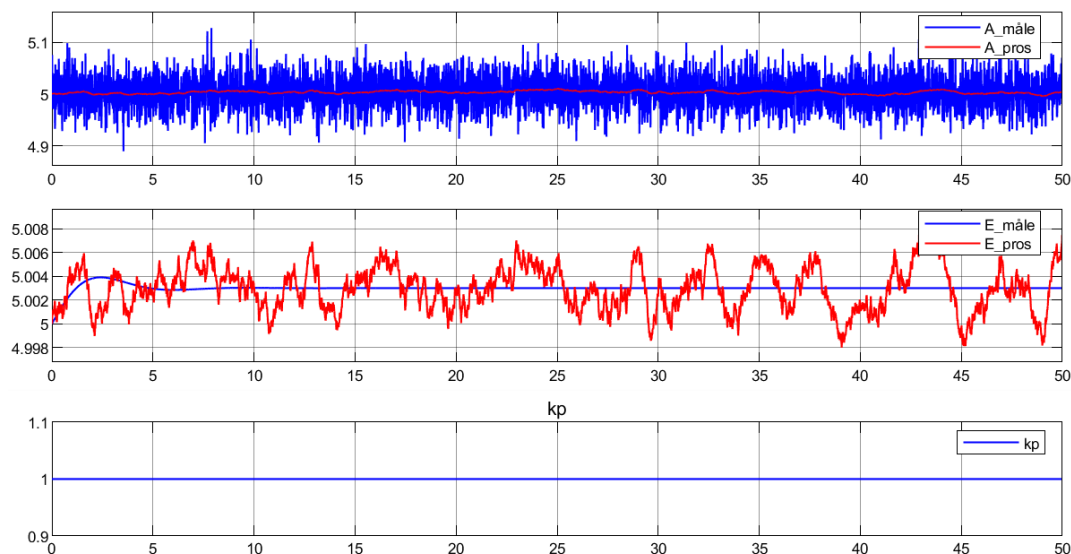
mens den effektive variansen for målestøyen forblir:

$$w_1 = 1 \cdot 10^{-3} \quad (4.13)$$

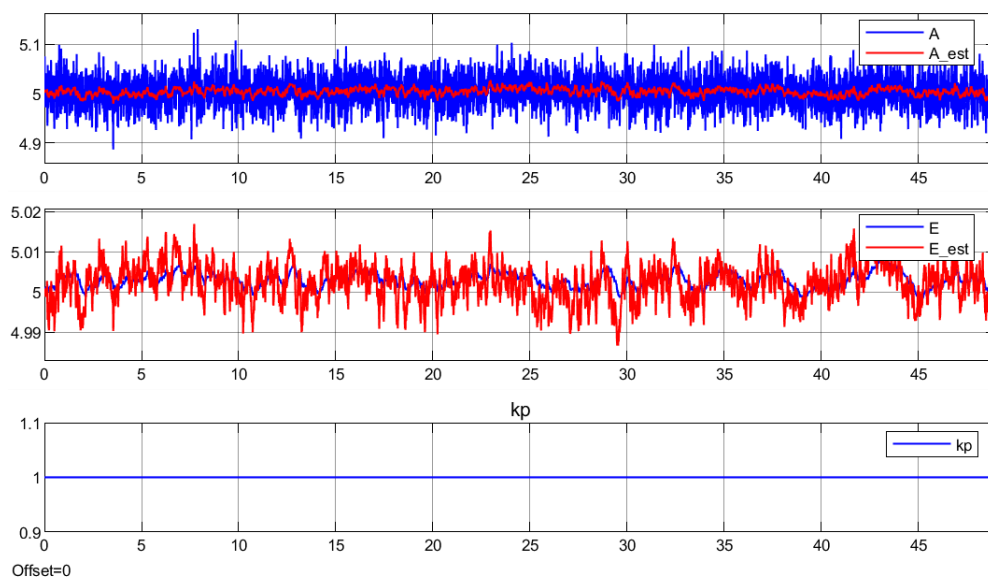
ved å bruke disse Q og R verdiene får vi figuren 4.12 som vil gi en støyreduksjon som i figur A.11. En av hovedformålene med å bruke Kalmanfilter for tilstands estimering er å kunne kvitte seg med mest mulig målestøy, mens prosesstøyen beholdes i systemet. Hvis figurene A.9 og A.11 kan man her se at bruken av den «effektive» støyen funnet her gir ett estimat som ligger nærmere det originale signalet med prosesstøy enn ved å bruke den tillagte prosesstøyen i systemet.



FIGUR 4.10: Simulink scope av prosess- og målestøyen som legges til i systemet



FIGUR 4.11: Simulink *scope* av A og E med enten prosess- eller målestøy lagt til. Blå strek: målestøy lagt til, rød strek prosessstøy lagt til



FIGUR 4.12: Simulink *scope* av A og E med forbedrede Q verdier.

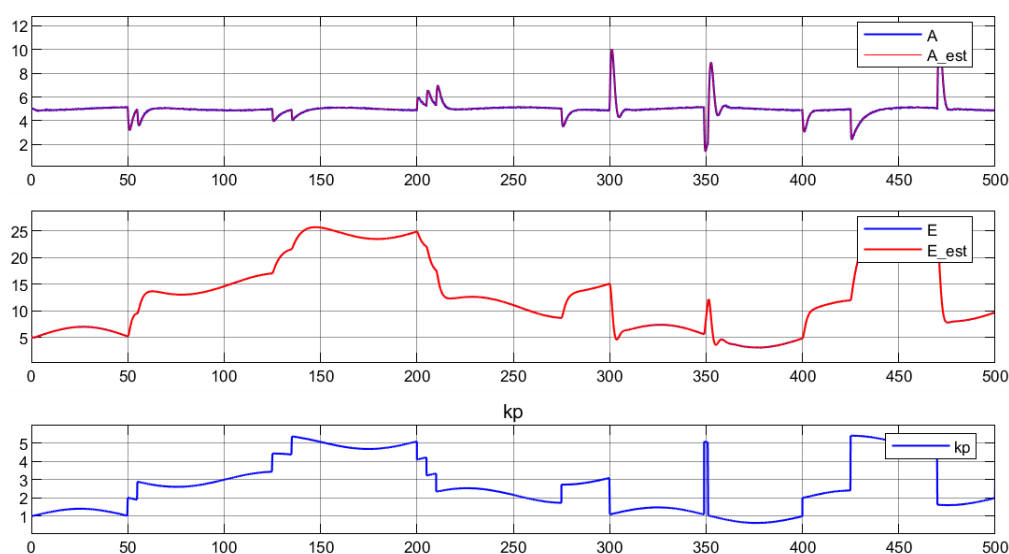
4.6 Test av ulike typer forstyrrelser

For mange prosesser vil ikke alltid forstyrrelsen forandre seg i sprang, slik som brukt i de tidligere forsøkene. I dette delkapittelet vil det bli sett på hvordan det utvidede Kalmanfilteret håndterer ulike typer forstyrrelser.

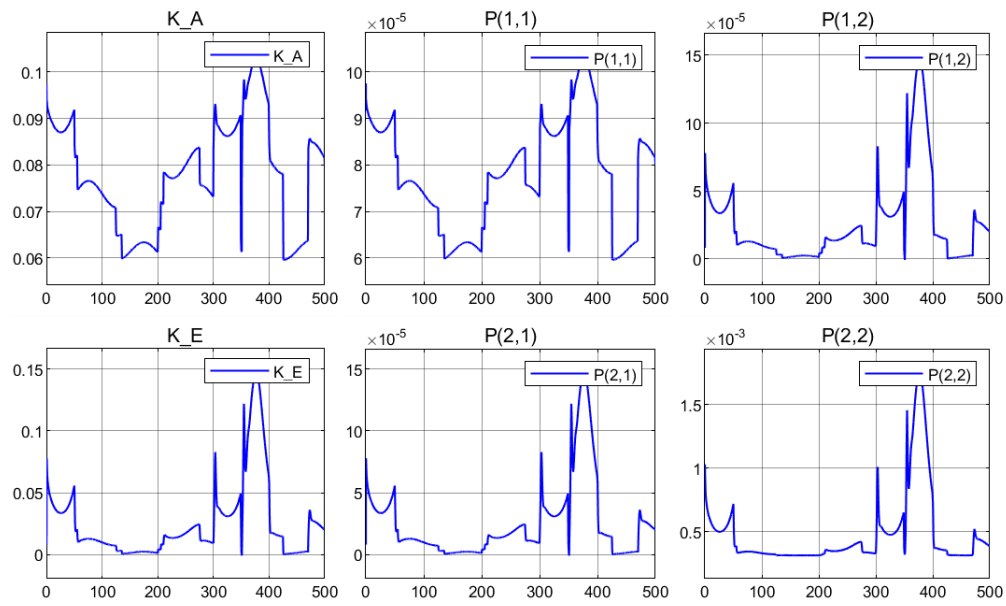
I dette delkapittelet er forstyrrelsesresponsene: rampe, sinus, *pseudorandom nummer sekvens* og tilslutt en blanding av alle utført. Forsøkene gjort med hver enkelt forstyrrelse ligger i vedlegg A.4, mens testen av alle forstyrrelserne samlet er vist under.

4.6.1 Simulink simulering av Kalmanfilter med utfordrende forstyrrelse

I figur 4.13 er utviklingen av tilstandene A og E estimert med en forstyrrelse satt sammen av en rampe, sinus og en *pseudorandom nummer sekvens*. Vi kan se fra denne grafen at det utvidede Kalmanfilteret fint klarer å håndtere forstyrrelser av ulike typer på en veldig god måte.



FIGUR 4.13: Simulink scope av A , E og k_s^A med utfordrende forstyrrelse



FIGUR 4.14: Simulink scope av K og P med utfordrende forstyrrelse

Kapittel 5

Kombinert tilstand og parameter estimering for biologisk kontrollmotiv

I dette kapitlet vil det bli brukt et augmentert Kalmanfilter for å utforske mulighetene for kombinert tilstand- og parameterestimering for det biologiske kontrollmotivet.

5.1 Augmentert Kalmanfilter for kombinert tilstand- og parameterestimering

Utgangspunktet for det augmenterte Kalmanfilteret er som tidligere ligningene (3.1) og (3.2). Når det ønskes å gjøre en parameter estimering vil modellen utvides til å inneholde en ekstra tilstand:

$$x = k \quad (5.1)$$

hvor k er en konstant. Ved å tilføre denne ekstra tilstanden kan modellen utvides til:

$$\dot{A} = k_s^A \cdot E - k_p \cdot A + v_1 \quad (5.2)$$

$$\dot{E} = k_s^E - \frac{V_{max} \cdot E}{K_m^E + E} \cdot A + v_2 \quad (5.3)$$

$$\dot{x} = 0 \quad (5.4)$$

hvor vi bytter ut en av parametrene i differensiallikningene med x , for eksempel k_s^A . Det er også vanlig å anta at parametermodellen inneholder en ukorrelert hvit støy og har en parametermodell på formen:

$$x(k+1) = x(k) + v(k) \quad (5.5)$$

som gjør at prosesstøyens kovariansmatrise Q vil bli utvidet til en 3 ganger 3 matrise. Det er også rimelig å sette variansen til x et par faktorer lavere enn variansen til de andre prosesstøyene i systemet [4]. Dette gjøres fordi det er rimelig å anta at parametertilstandene varierer langsommere enn de originale tilstandsvariablene.

5.2 Estimering av 1 parameter med måling av en tilstand

5.2.1 Linearisering og observerbarhet

I (4.7) og (4.10) er det 4 konstante parametere : k_s^A , k_s^E , V_{max} og K_M^E . Ved å ta k_s^A som et eksempel, blir den augmenterte modellen:

$$\dot{A} = x \cdot E - k_p \cdot A \quad (5.6)$$

$$\dot{E} = k_s^E - \frac{V_{max} \cdot E}{K_M^E + E} \cdot A \quad (5.7)$$

$$\dot{x} = 0 \quad (5.8)$$

hvor k_s^A nå er en ukjent parameter som kan estimeres sammen med A og E .

Ved å linearisere (5.6 - 5.8) får vi systemmatrisen:

$$A_{mat} = \begin{bmatrix} -k_p & x & E \\ -\frac{E \cdot V_{max}}{E + K_M^E} & \frac{A \cdot E \cdot V_{max}}{(E + K_M^E)^2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Etter lineariseringen finnes transmisjonsmatrisen Φ ved diskretering:

$$\Phi = \begin{bmatrix} 1 - (T \cdot k_p) & T \cdot x & T \cdot E \\ -T \cdot \frac{E \cdot V_{max}}{E + K_M^E} & 1 + (T \cdot \frac{A \cdot E \cdot V_{max}}{(E + K_M^E)^2} - \frac{A \cdot V_{max}}{E + K_M^E}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

der T er samplingstiden for systemet.

Linearisering og diskreteringsprosessen blir så utført på de resterende parametrene, og brukes så til å teste observerbarheten til systemet. Resultatene for alle 8 kombinasjonene er vist i tabell 5.2.1. I kolonne tre av tabellen er rangen til observerbarhetsmatrisen vist. I alle 8 testene blir rangen 3, som tilsier at vi kan estimere 3 tilstander for alle kombinasjonene.

Ukjent parameter	Målt variabel	rang
k_s^A	A	3
k_s^A	E	3
k_s^E	A	3
k_s^E	E	3
V_{max}	A	3
V_{max}	E	3
K_M^E	A	3
K_M^E	E	3

5.2.2 Valg av parameterverdier

For å teste parameter estimeringen i Simulink må det fastsettes noen verdier. Initialverdiene for de ulike parametrene som skal estimeres settes til en verdi «litt» høyere enn de ekte verdiene. Initialverdiene for a posteriori kovariansmatrisen finnes ved prøving og feiling under kjøring. Støy kovariansmatrisene settes som:

$$Q = \begin{bmatrix} 1 \cdot 10^{-5} & 0 & 0 \\ 0 & 5 \cdot 10^{-6} & 0 \\ 0 & 0 & 1 \cdot 10^{-9} \end{bmatrix}, R = 1 \cdot 10^{-3}$$

der $Q(1,1)$ og $Q(2,2)$ er funnet i delkapittel 4.5.3, mens $Q(3,3)$ er valgt ett par faktorer lavere.

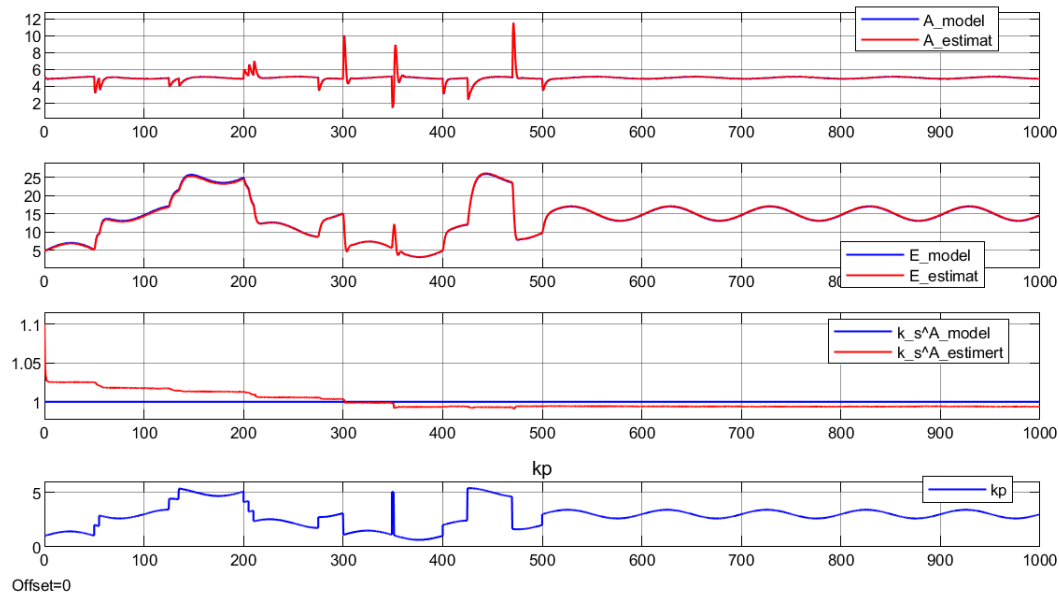
5.2.3 Simulink simulering

I vedlegg B.1 og B.2 er grafer av de 8 kombinasjonene av forsøk som er utført lagt ved. Disse testene viser svært gode resultater for 7 av de 8 kombinasjonene av parametre og målt tilstand. Figurene (5.1 - 5.4) viser tilstands-estimatene, Kalmanforsterkningene og a posteriori kovariansmatrisene for parameterestimering av k_s^A med måling på henholdsvis A og E.

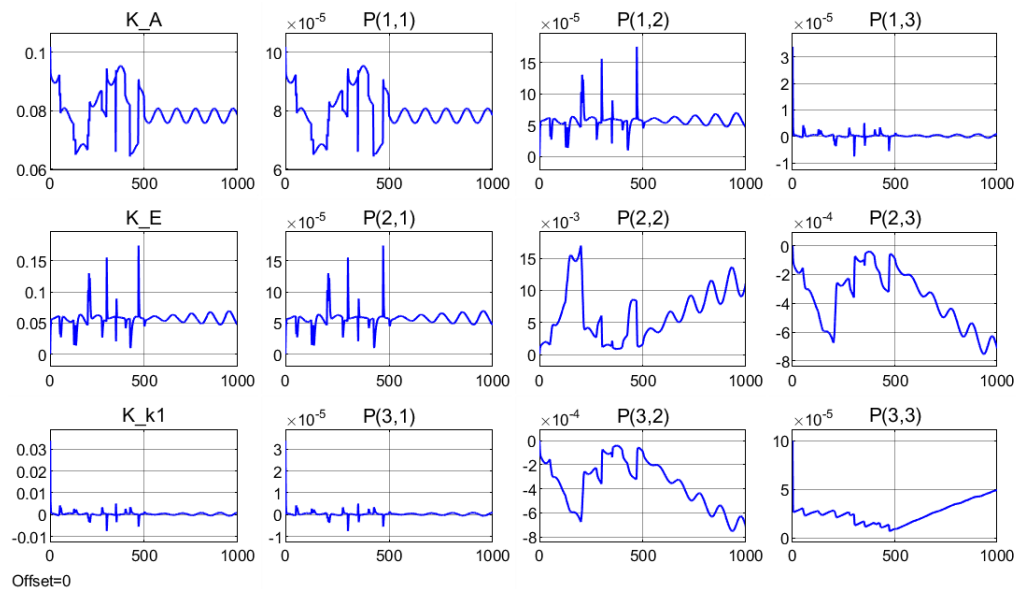
Den største forskjellen mellom disse 2 estimeringene ligger i diagonalen til a posteriori kovariansmatrisene. I figur 5.4 vil usikkerheten for både A ($P(1,1)$) og k_s^A ($P(3,3)$) stabilisere seg, mens den jobber for å motvirke påvirkningen fra forstyrrelsen. Denne typen respons er den samme som vi fikk for testene gjort i kapittel 2, og gir en god og entydig estimering av tilstandene.

I figur 5.2 oppstår en respons der usikkerheten til E ($P(2,2)$) og k_s^A ($P(3,3)$) ikke legger seg rundt et stabilt punkt, slik som i 5.4. Denne responsen vil så få innvirkning på kalmanforsterkingen til k_s^A (K_{k1}) og E (K_E), og gjøre denne svært ustabil. Parameterestimeringen av k_s^A med måling av A er her den parameterestimeringen som gir ett dårlig estimat. Hvis simuleringen kjøres over en lengre periode vil $P(2,2)$ og $P(3,3)$ tilslutt stabilisere seg, men estimatet av E og k_s^A vil ikke stabilisere seg rundt et entydig estimat.

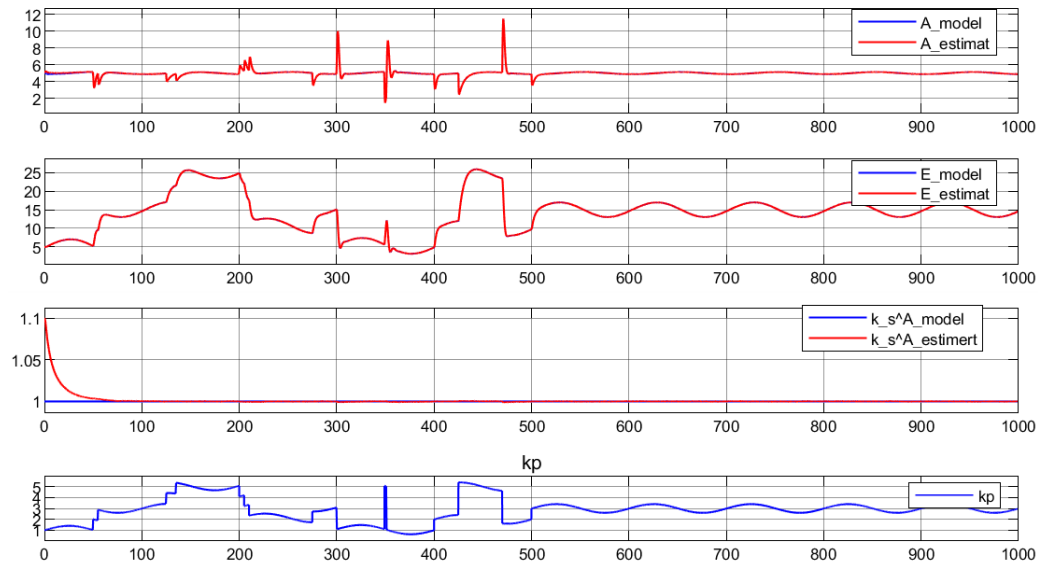
Simulering av de 6 andre kombinasjonene er gjort på samme måte som over, og gir svært gode estimater av både den ikke målte tilstanden og den ukjente parameteren, og er lagt ved i vedlegg B.1 og B.2.



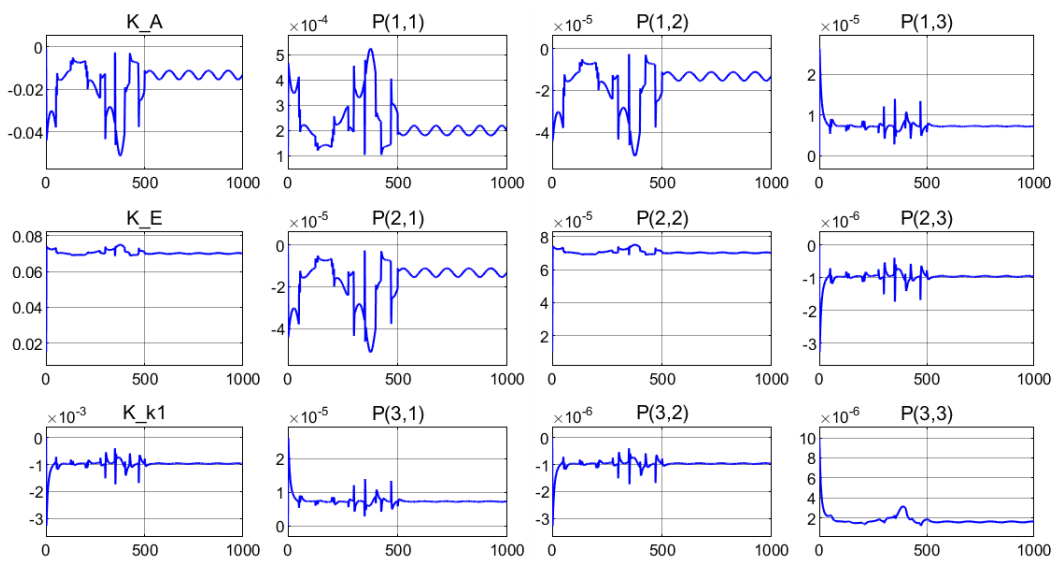
FIGUR 5.1: A , E og k_s^A estimert med måling av A



FIGUR 5.2: Simulink scope av K og P for estimat av k_s^A med måling av A



FIGUR 5.3: A , E og k_s^A estimert med måling av E



FIGUR 5.4: Simulink scope av K og P for estimat av k_s^A med måling av E

5.3 Estimering av 1 parameter med måling av begge tilstander

5.3.1 Linearisering og observerbarhet

Først blir alle systemmatrisene og transmisjonsmatrisene funnet ved linearisering og diskretering. Bruker i denne testen også k_s^A som eksempel og får da samme systemmatrise og transmisjonsmatrise som i 5.2.1. Disse transmisjonsmatrisene blir så brukt for teste observerbarheten til transmisjonsmatrisene.

Ukjent parameter	Målt variabel	rang
k_s^A	A og E	3
k_s^E	A og E	3
V_{max}	A og E	3
K_M^E	A og E	3

I tabell 5.3.1 er de 4 kombinasjonene av parametre med måling av begge variablene vist. Alle kombinasjonene får rang lik 3 i observerbarhetstesten, og systemet er observerbart for alle kombinasjonene.

5.3.2 Valg av parameterverdier

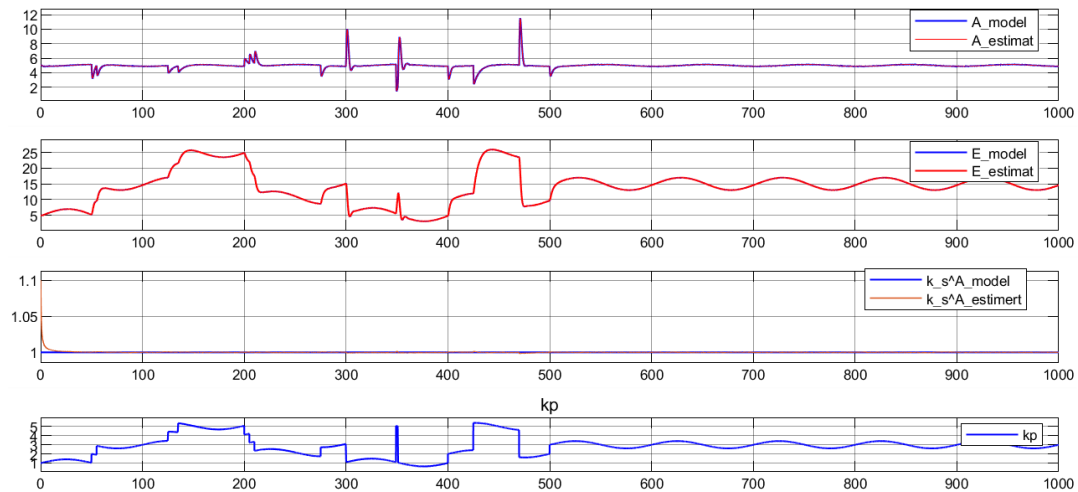
I denne testen brukes samme verdier som valgt i seksjon 5.2.2 untatt R, som settes til:

$$R = \begin{bmatrix} 1 \cdot 10^{-3} & 0 \\ 0 & 1 \cdot 10^{-3} \end{bmatrix}$$

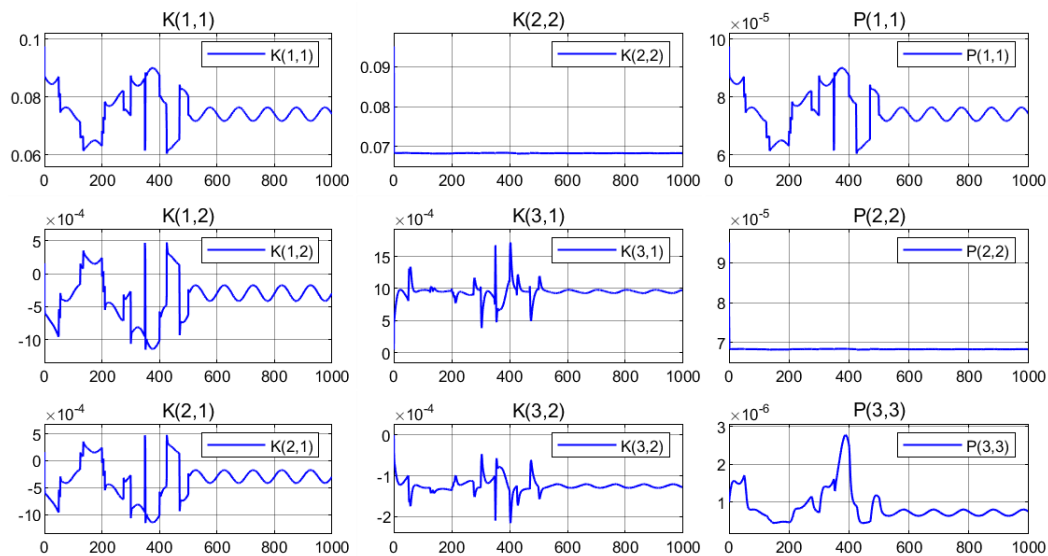
da det nå utføres 2 målinger, må det også introduseres målesstøy for E, og diagonalen til R settes da som variansene til målestøyene.

5.3.3 Simulink simulering

I figur 5.5 vises en suksessfull estimering av k_s^A med måling av begge tilstandene A og E. Ved å sammenligne figur 5.5 og 5.3 kan det også ses hvordan Kalmanfilteret klarer å konvergere til den «ekte» verdien forttere enn med kun 1 måling. De 3 resterende kombinasjonene vil på samme måte som k_s^A gi svært gode resultater. Figurene er lagt ved i vedlegg B.3.



FIGUR 5.5: Simulink scope av A , E og k_s^A for parameter estimering med måling av A og E



FIGUR 5.6: Simulink scope av K og P for estimat av k_s^A med måling av A og E

5.4 Estimering av 2 parameter med måling av begge tilstander

I denne seksjonen vil vi utforske mulighetene for å estimere mer enn 1 parameter sammen med variablene A og E . Det vil da bli introdusert enda en tilstandsvariabel i modellen:

$$x_1 = k_1 \quad (5.9)$$

$$x_2 = k_2 \quad (5.10)$$

5.4.1 Linearisering og observerbarhet

Utfører så linearisering på de 6 kombinasjonene av 2 og 2 parametre. Ved å ta k_s^E og k_s^E som et eksempel for vi den augmenterte modellen:

$$\dot{A} = x_1 \cdot E - k_p \cdot A + v_1 \quad (5.11)$$

$$\dot{E} = x_2 - \frac{V_{max} \cdot E}{K_M^E + E} \cdot A + v_2 \quad (5.12)$$

$$\dot{x}_1 = 0 \quad (5.13)$$

$$\dot{x}_2 = 0 \quad (5.14)$$

hvor k_s^A og k_s^E nå er ukjente parametre som kan estimeres sammen med A og E .

Ved å linearisere (5.11 - 5.14) får vi systemmatrisen:

$$A_{mat} = \begin{bmatrix} -k_p & x & E & 0 \\ -\frac{E \cdot V_{max}}{E + K_M^E} & \frac{A \cdot E \cdot V_{max}}{(E + K_M^E)^2} & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Etter lineariseringen finnes transmisjonsmatrisen Φ ved diskretering:

$$\Phi = \begin{bmatrix} 1 - (T \cdot k_p) & T \cdot x & T \cdot E & 0 \\ -T \cdot \frac{E \cdot V_{max}}{E + K_M^E} & 1 + (T \cdot \frac{A \cdot E \cdot V_{max}}{(E + K_M^E)^2} - \frac{A \cdot V_{max}}{E + K_M^E}) & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

der T er samplingstiden for systemet.

Linearisering og diskreteringsprosessen blir så utført på de resterende parametrene, og brukes så til å teste observerbarheten til systemet. I tabell 5.4.1 er alle 6 kombinasjonene av parametre og målte tilstander vist. I kolonne tre av tabellen er rangen til observerbarhetsmatrisen. Fra denne tabellen kan det her ses at observerbarhetstesten gir en rang av 4 når de 2 ukjente parametrene ligger i forskjellige differensiallikninger, mens den gir en rang av 3 når de 2 ukjente parametrene ligger i samme differensiallikning.

ukjente parametere	Målt variabel	rang
k_s^A og k_s^E	A og E	4
k_s^A og V_{max}	A og E	4
k_s^A og K_M^E	A og E	4
k_s^E og V_{max}	A og E	3
k_s^E og K_M^E	A og E	3
V_{max} og K_M^E	A og E	3

5.4.2 Valg av parameterverdier

Velger samme verdier som i seksjon 5.3.2 untatt at Q som utvides til:

$$Q = \begin{bmatrix} 1 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 5 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 1 \cdot 10^{-9} & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-9} \end{bmatrix}$$

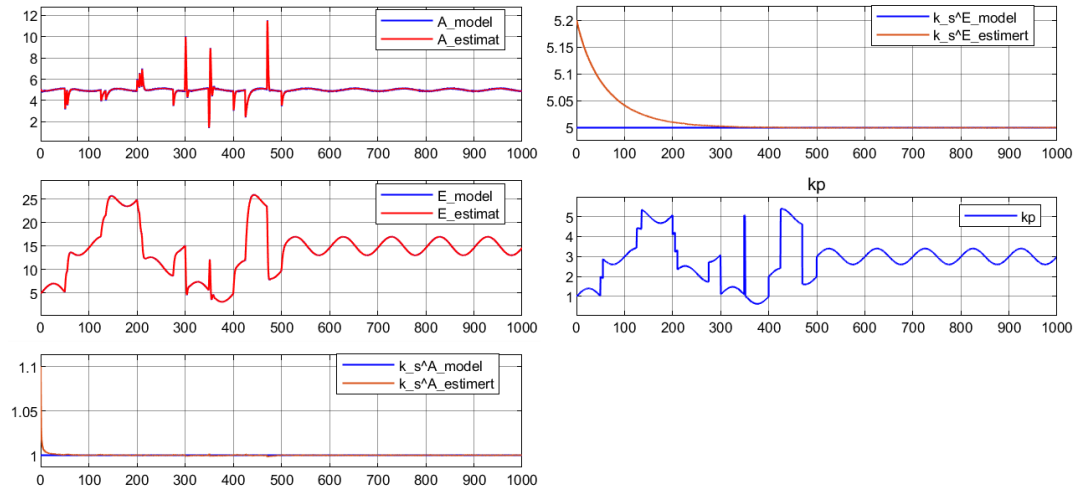
5.4.3 Simulink simulering

Etter testing av de 6 kombinasjonene av parametere vises det at de 3 kombinasjonene som har rang 4 for observerbarhetsmatrisene, gir gode estimater av parametrene. I figur 5.7 vises parameterestimeringen av k_s^A og k_s^E , som er en av de velfungerende parameterestimeringene. Etter testing av de 3 andre kombinasjonene kan det ses mer ujevne resultater.

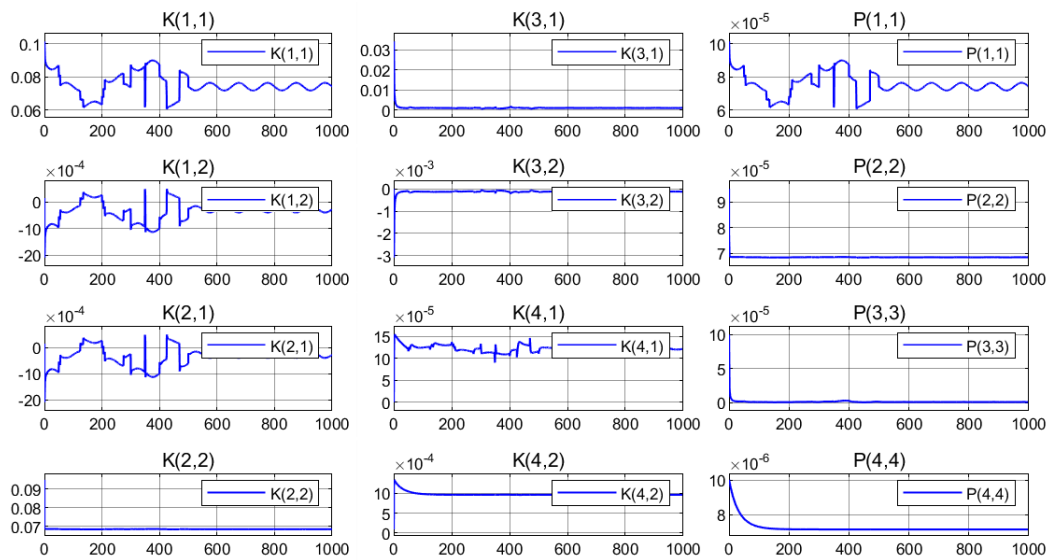
For estimeringen av k_s^E og V_{max} i vedlegg B.4.4 vil ingen av usikkerhetene $P(3,3)$ eller $P(4,4)$ klare å stabilisere seg, og ingen av parametrene vil få et entydig estimat.

For estimeringen av k_s^E og K_M^E i vedlegg B.4.5 vil en av usikkerhetene nærme seg stabil($P(3,3)$), mens den andre ikke vil stabilisere seg($P(4,4)$). og det vil heller ikke bli ett entydig resultat.

For estimeringen av V_{max} og K_M^E i vedlegg B.4.6 vil den ene usikkerheten stabilisere seg ($P(3,3)$) helt, mens den andre ikke vil stabilisere seg ($P(4,4)$). Hvor vi vil få 2 ganske gode estimater.



FIGUR 5.7: Simulink scope av A , E , k_s^A og k_s^E for parameter estimering med måling av A og E



FIGUR 5.8: Simulink scope av K og P for parameterestimering av k_s^A og k_s^E med måling av A og E

5.5 Estimering av parametere med sprang i parameteren

Dette delkapittelet vil bygge på delkapittelet over, der 2 parametere ble estimert med måling av begge tilstandene. I dette delkapittelet vil det legges inn ett sprang i parameteren og se hvordan algoritmen reagerer på dette. I dette

kapittelet blir estimeringen av k_s^A og k_s^E som er vist i delkapittelet over, dette fører til samme linearisering og observerbarhet som brukt tidligere.

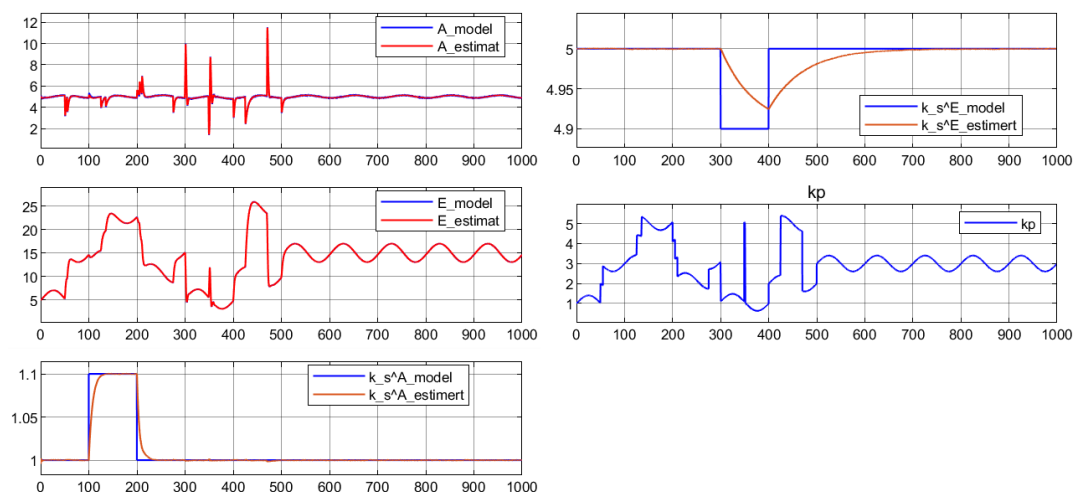
5.5.1 Simulink simulering med «spesifikke» støy kovarianser

For det første forsøket er støy kovariansmatrisene Q og R valgt som i delkapittelet over der parametrene er forventet konstant.

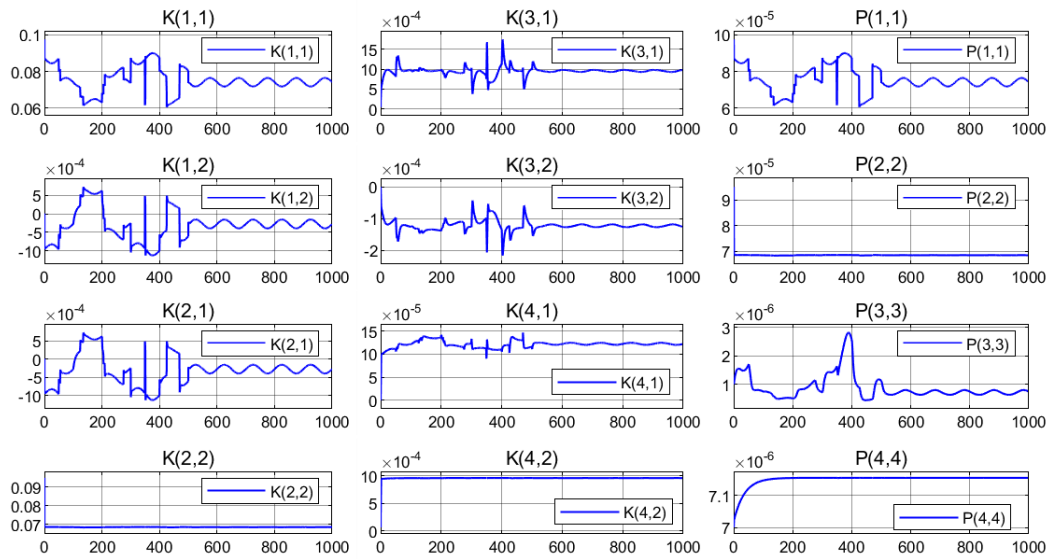
$$Q = \begin{bmatrix} 1 \cdot 10^{-5} & 0 & 0 & 0 \\ 0 & 5 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 1 \cdot 10^{-9} & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-9} \end{bmatrix}, R = 1 \cdot 10^{-3}$$

bruken av denne støy kovariansmatrisen vil føre til at oppdateringene som kommer gjennom Kalmanforsterkningen vil være veldig små. Dette er en fordel da vi forventer at parameteren er konstant. Når det legges inn ett sprang i parameteren kan vi se i figur 5.9 at det vil være gode estimater når parameteren er konstant, men når spranget inntreffer vil algoritmen slite med å komme seg til den nye parameteren.

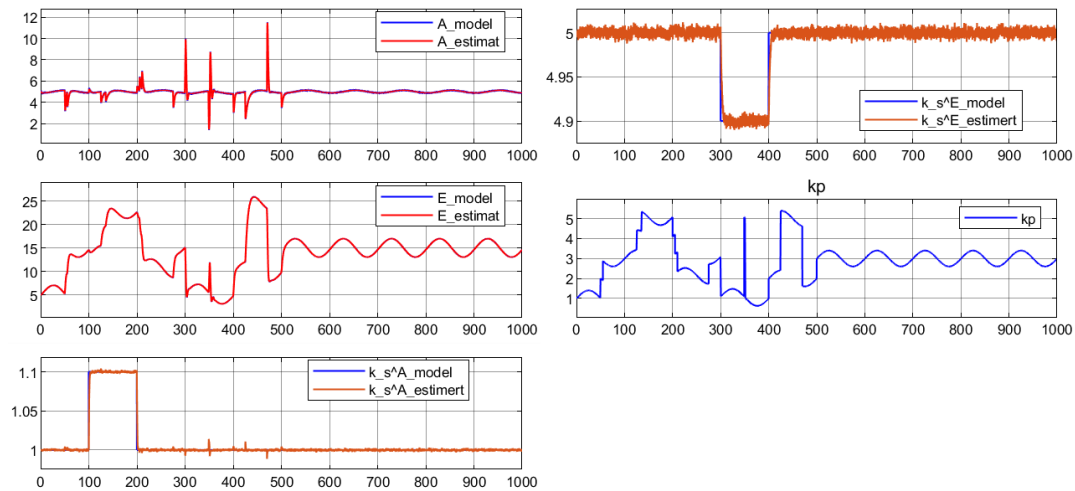
For å kunne ta høyde for slike sprang i parameterne er det nødvendig å senke valgene for $P(3,3)$ og $P(4,4)$. Når $P(3,3)$ og $P(4,4)$ senkes vil Kalmanforsterkningen bli mer sensitiv til forandringer i de målte verdiene, og gi muligheter for større oppdateringer. Det kan nå ses fra figur 5.11 at algoritmen klarer å følge sprangene i parameterne på en bedre måte, men vi har tapt mye nøyaktighet i estimatene når parameterne er konstante.



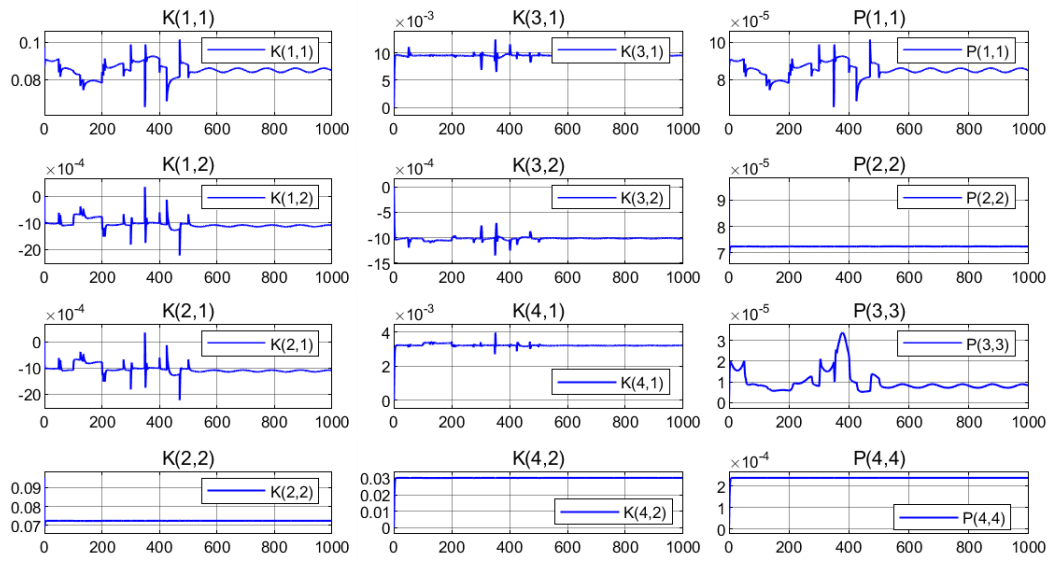
FIGUR 5.9: Simulink scope av A , E , k_s^A og k_s^E for parameter estimering med måling av A og E , med et steg i parameterne



FIGUR 5.10: Simulink scope av K og P for parameterestimering av k_s^A og k_s^E med måling av A og E , med et steg i parameterne



FIGUR 5.11: Simulink scope av A , E , k_s^A og k_s^E for parameter estimering med måling av A og E , med et steg i parameterne



FIGUR 5.12: Simulink scope av K og P for parameterestimering av k_s^A og k_s^E med måling av A og E , med et steg i parameterne

Kapittel 6

Funksjons estimering på utvidet biologisk kontrollmotiv

6.1 Utvidelse av modellen

6.1.1 Differensiallikninger for systemet

I dette kapitlet vil den matematiske modellen for *inflow* kontroller 1 bli utvidet med 2 aktiveringsfunksjoner, kalt F_A og F_E . Disse aktiveringsfunksjonene gjør med differensiallikningene våre er at det legges til ett ledd med en verdi mellom 0 og 1 som begrenser aktiveringene av syntese for tilstandene. Denne utvidelsen av kontroll motivet vil føre til en tregere dynamikk for systemet og det blir også lagt til 2 nyeparametere. De nye differensial likningene for kontroll motivet blir:

$$\dot{A} = k_s^A * \frac{E}{K_a^E + E} - K_p \cdot A + v_1 \quad (6.1)$$

$$\dot{E} = k_s^E - \frac{V_{max} * E}{k_m^E + E} * \frac{A}{K_a^A + A} + v_2 \quad (6.2)$$

6.1.2 Valg av parameterverdier

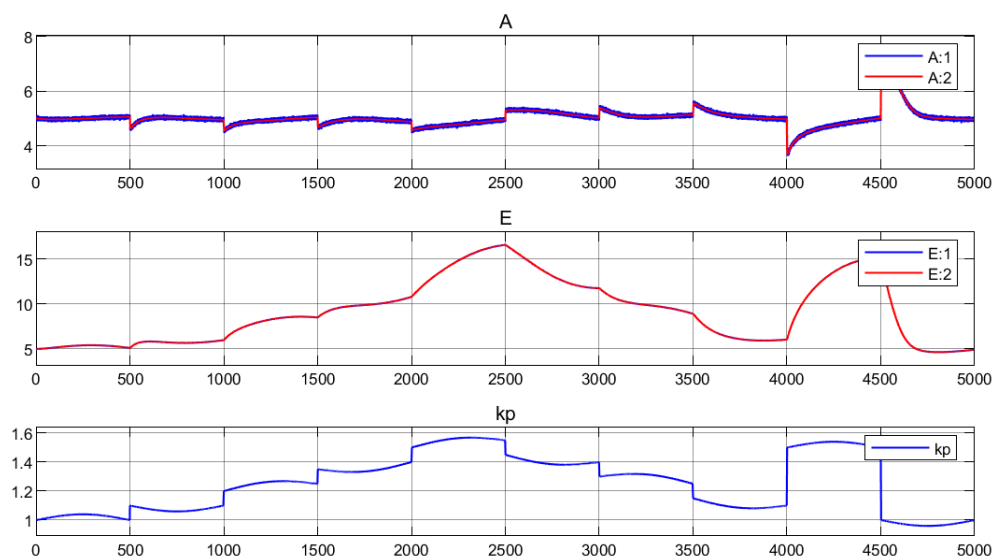
For den utvidede modellen er det nødvendig å forandre på verdiene for parametrene i kontrollmotivet, og det legges til 2 nye. Disse nye verdiene er valgt for å opprettholde arbeidspunkter $[A, E, k_p] = [5, 5, 1]$. forstyrrelsen k_p er også her blitt begrenset for at verdiene for A og E skal holde seg over null.

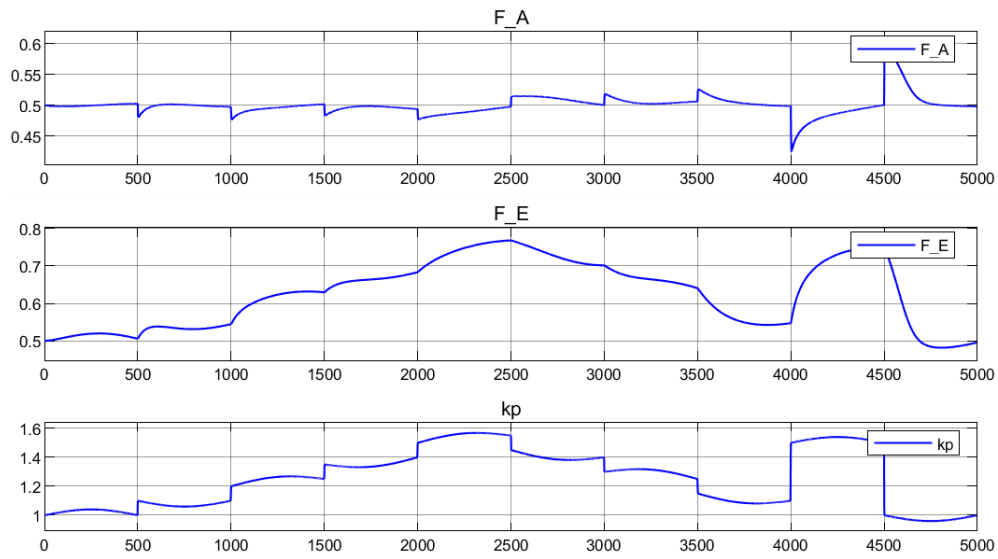
Variabel	Verdi
k_s^A	10 mol/L/s
k_s^E	0.5 mol/L/s
V_{max}	1 mol/L/s
K_M^E	0.003 mol/L/s
K_a^A	5 mol/L/s
K_a^E	5 mol/L/s
k_p	[1 - 1.5] mol/L/s
T	0.01 s

TABELL 6.1: Variabel verdier for kapittel 6

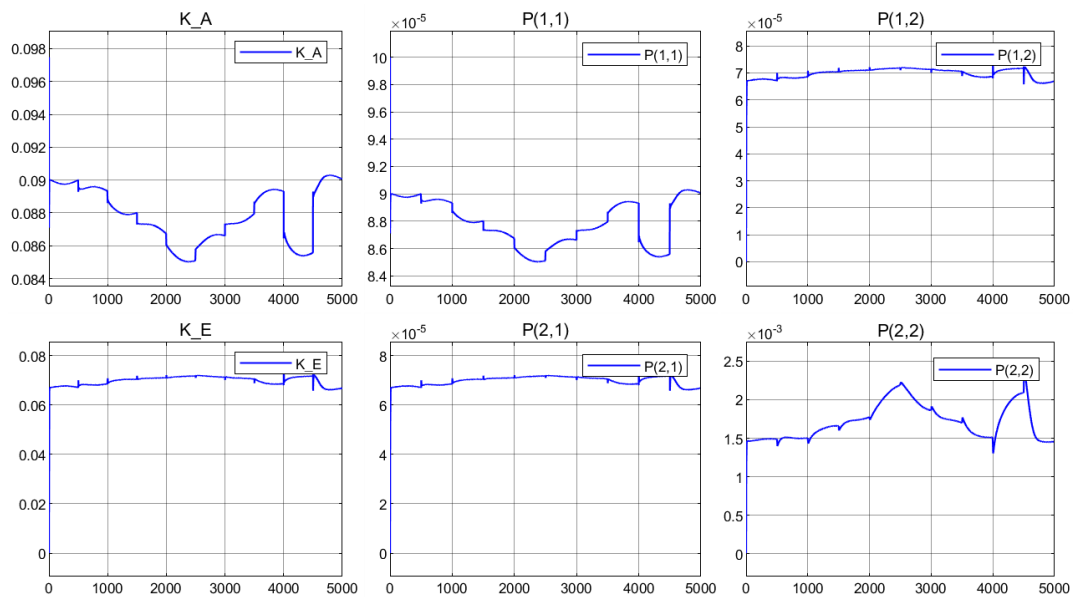
6.2 Simulink simulering

I figur 6.1 kan det ses hvordan det utvidede kalman filter estimerer både A og E på en fortsatt god måte selv for denne modellen. I figur ?? kan det ses hvordan aktiveringsfunksjonene F_A og F_E utvikler seg gjennom simuleringen, og det kan ses hvordan funksjonene speiler tilstandene med en begrenset verdi mellom 0 og 1.

FIGUR 6.1: Simulink scope av A og E for den utvidede modellen



FIGUR 6.2: Simulink scope av $\frac{E}{K_A^E + E}$ (F_E) og $\frac{A}{K_A^A + A}$ (F_A) for den utvidede modellen



FIGUR 6.3: Simulink scope av K og P for den utvidede modellen

6.3 Estimering av funksjonene $\frac{E}{K_A^E + E}$ og $\frac{A}{K_A^A + A}$ med augmentert Kalmanfilter

6.3.1 Linearisering og observerbarhet

Som tidligere utføres linearisering og observerbarhets tester for modellene, og vi får at F_E er observerbart når E måles, men ikke når A måles. Mens F_A er observerbart når A måles, men ikke når E måles.

Ukjent parameter	Målt variabel	rang
F_A	A	3
F_A	E	2
F_E	A	2
F_E	E	3

6.3.2 Valg av parameterverdier

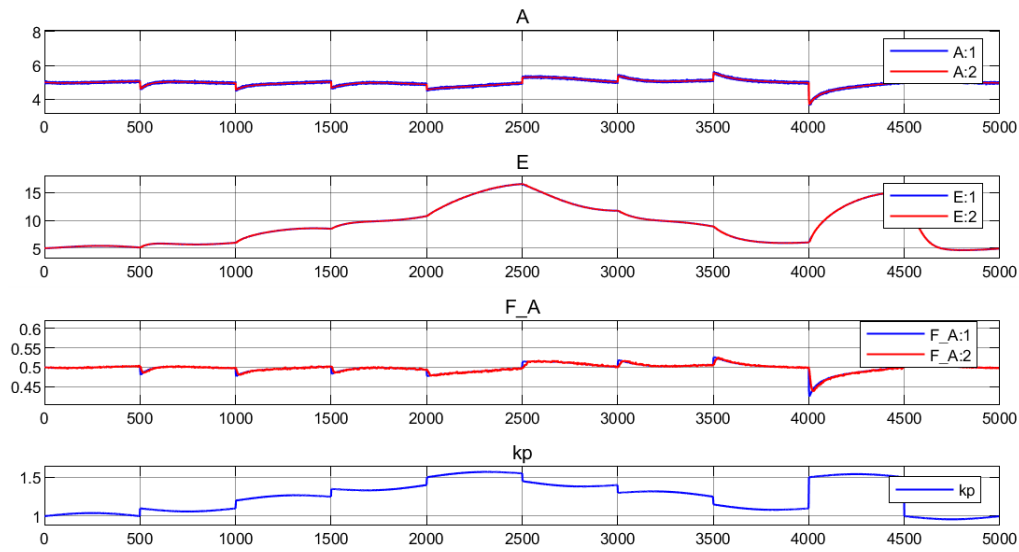
Bruker i dette forsøket samme verdier som tabell 6.1 I tillegg velges støy kovariansmatrisene til:

$$Q = \begin{bmatrix} 1 \cdot 10^{-5} & 0 & 0 \\ 0 & 5 \cdot 10^{-6} & 0 \\ 0 & 0 & 1 \cdot 10^{-7} \end{bmatrix}, R = 1 \cdot 10^{-3}$$

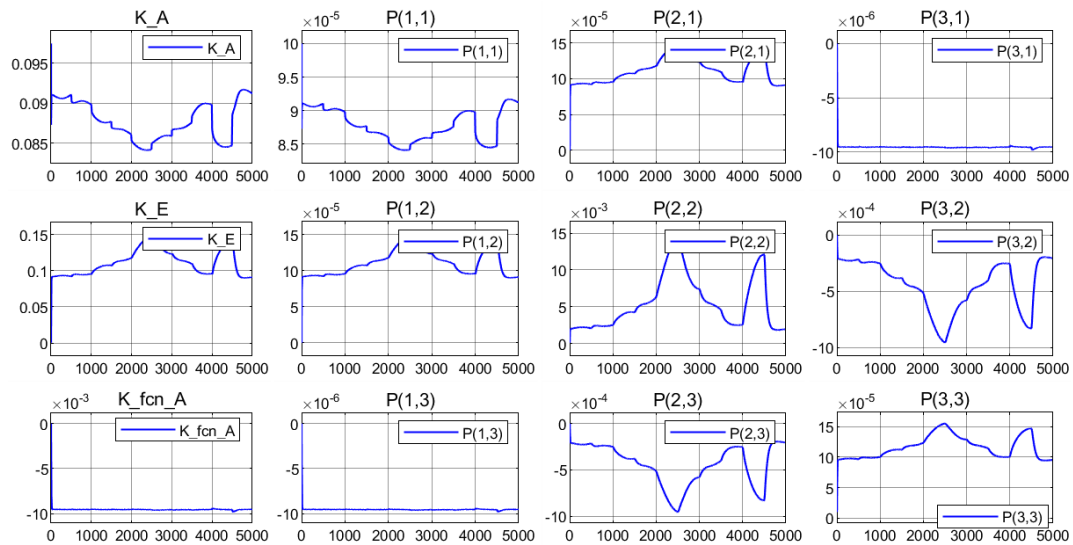
der $Q(1,1)$ og $Q(2,2)$ er funnet i delkapittel 4.5.3, mens $Q(3,3)$ er valgt litt lavere enn de 2 andre tilstandene. Ett lavere valg av $Q(3,3)$ vil gjøre at estimatet av F_A og F_E vil slite mer med å holde følge ved tider med store sprang. Hvis $Q(3,3)$ velges større vil estimatene av F_A og F_E følge sprangene bedre, men funksjonene vil ta til seg mer støy.

6.3.3 Simulink simulering

I figur 6.4 er estimeringen av F_A med måling av A vist. Fra figuren kan det ses at Kalmanfilteret gir en god estimering av F_A frem til det oppstår sprang i forstyrrelsen, hvor vi vil få små estimeringsfeil grunnet valget av $Q(3,3)$. Her ville det vært mulig å velge en større $Q(3,3)$ verdi for å holde følge i sprangene, men som konsekvens av dette vil estimatet for de andre delene av signalet få mer påvirkning av støy.



FIGUR 6.4: Simulink scope av A , E , F_A og k_p med måling av A



FIGUR 6.5: Simulink scope av K og P for estimering av F_E med måling av A

Kapittel 7

Diskusjon og Konklusjon

7.1 Kapittel 4

I kapittel 4 har det blitt implementert og testet både et lineært Kalmanfilter og et utvidet Kalmanfilter. Etter implementeringen av det lineære Kalmanfilteret var det tydelig at kontrollmotivene sin ulinearitet gav tydelige estimeringsfeil når arbeidspunktet for linearisering ble forlatt. Etter implementeringen av det utvidede Kalmanfilteret ble det produsert gode estimater for både den kontrollerte og den manipulerte variabelen. Ulempen med å bruke det utvidede Kalmanfilteret er at systemmatrisen og transmisjonsmatrisen må beregnes på nytt hver iterasjon. Siden det lineære filteret gir gode estimater i områder nær arbeidspunktet det er linearisert for, vil det for eksempel være mulig å implementere en kombinasjon av algoritmene som kan benytte det lineære Kalmanfilteret i situasjoner der forstyrrelsen er tilnærmet konstant, mens det utvidede benyttes når forstyrrelsen er mindre stabil.

Etter implementasjonen ble algoritmen introdusert for ulike typer avvik og forstyrrelser. Etter disse testene er det klart at det utvidede Kalmanfilteret er robust for ulike typer forstyrrelser. Valg av initielle verdier for a posteriori kovariansmatrise og initielle verdier for tilstandene må velges i samsvar med usikkerheten rundt de ekte initielle verdiene for tilstandene. Valget av a posteriori kovariansen og de initielle verdiene for tilstandene blir mindre og mindre viktig desto lengre tid algoritmen kjører. Hvis det er høye krav om hvor fort algoritmen skal fungere optimalt, er det nødvendig med svært gode initielle verdier, mens hvis algoritmen kan «trene» en stund i systemet, vil valget av initielle verdier bli mindre viktig.

7.2 Kapittel 5

I kapittel 5 har det blitt gjort forsøk for å teste kombinert tilstand- og parameter estimering for de biologiske kontrollmotivene. I første del er det brukt et augmentert Kalmanfilter for å estimere 1 av motivets parametre sammen med de originale estimatene av variablene. Etter implementasjon av det augmenterte Kalmanfilteret ble 3 av 4 tilfeller med måling av den kontrollerte variabelen estimert på en god måte, mens 4 av 4 tilfeller ble estimert på en

god måte ved måling av den manipulerede variabelen. I tilfellet i dette forsøket hvor det ikke ble et godt estimat gav observerbarhetstesten en rang som tilsa at parameteren skulle være observerbar. Under testene viste det seg at a posteriori kovariansen til tilstanden som skulle estimeres reagerer på samme måte som a posteriori kovariansene til de ikke observerbare kombinasjonene i delkapittel 5.4. Her kan det være mulig at observerbarhetstesten ikke gir riktig rang, det er også mulig at numeriske feil står for denne feilen.

I andre del av forsøket er det estimert 1 av motivets parametre ved måling av både den kontrollerte og den manipulerede variabelen. I dette forsøket ble 4 av 4 parametre estimert med gode resultater. I disse forsøkene viser det seg også at når begge tilstandene måles, vil algoritmen rette opp for avvik i de initielle verdiene raskere enn da 1 tilstand er målt. Hvis det er mulig å estimere begge tilstandene vil dette alltid være lønnsomt når det skal utføres parameterestimering, vis det er mulig å «trene» algoritmen med måling av begge tilstander, for så å gå over til å kun måle 1 tilstand vil dette også være profitabelt med mindre initialverdiene er veldig gode.

I tredje del av forsøket er det estimert 2 av motivets parametre samtidig med måling av både den kontrollerte og den manipulerede variabelen. I denne testen viste det seg at kun 3 av 6 kombinasjoner var observerbare. De 3 observerbare kombinasjonene var kombinasjoner der de 2 parametrene var en del av hver sin differensiallikning. Her viser resultatene at de 3 observerbare kombinasjonene gav god estimering, mens de 3 andre gav varierende resultater hvor ingen gav entydige estimater for begge parameterne som skulle bli estimert.

I siste del av forsøket er det estimert 2 parametre samtidig med måling av både den kontrollerte og den manipulerede variabelen, men det er her lagt inn sprang i parameterne. I dette forsøket kommer det frem at algoritmen håndterer sprang i parameterne, men hvor godt sprangene håndteres kommer an på prosesstøy kovariansen til parameterne. Hvis en lav prosesstøy kovarians er valgt vil parameterne bli estimert mer nøyaktig i konstante områder, men slite med store sprang. Hvis prosesstøy kovariansen til parameterne øker, vil estimatene bli mer unøyaktig for konstante områder, men håndtere sprangene på en bedre måte. Når man bruker et utvidet Kalmanfilter vil det også være mulig å introdusere varierende støy kovariansmatriser [4]. Implementasjon av dette kan gi oss muligheten til å kombinere lav støy kovarians for situasjoner der det er tilnærmet konstant verdi for parameteren, og øke støy kovariansen i situasjoner der det er større sprang.

7.3 Kapittel 6

I kapittel 6 er det blitt brukt en utvidet versjon av kontrollmotivet. Denne versjonen introduserer en funksjon som begrenset aktiveringen av syntesen mellom den kontrollerte og den manipulerede variabelen. Her viser det seg

at kun aktiveringsfunksjonen som aktiverer syntesen til den målte variabelen er observerbar. Etter forsøket viser det seg at algoritmen finner entydige estimater for funksjonene for alle 4 kombinasjonene, selv om bare 2 av de er observerbare gjennom observerbarhetstesten. I dette forsøket viser det seg på samme måte som med sprang i parameteren at valget av prosesstøy kovarians for funksjonen vil ha innvirkning på hvor godt den estimerer tilnærmet konstante områder, og større sprang. Her vil det også være mulig å implementere en variasjon i støy kovariansen for situasjoner med tilnærmet konstant forstyrrelse og situasjoner med større sprang i forstyrrelsen.

Bibliografi

- [1] Per Holck og Anton Hauge. *homeostase i Store medisinske leksikon på snl.no*. Hentet fra <https://sml.snl.no/homeostase> (2020/06/25).
- [2] Kristian Thorsen og Geir B. Riscoll og Daniel M. Tveit og Peter Ruoff og Tormod Drengstig. *Tuning of Physiological Controller Motif*. Hentet fra: DOI: 10.3384/ecp1714231.
- [3] Atle Gjengedal. «Sammenligning av ekf- og ukf- kalmanfilter anvendt på totankprosess.» Masteroppgave, Universitetet i Stavanger, 2009.
- [4] Finn Haugen. *Regulering av dynamiske systemer*. 2. utg. Tapir forlag, 1996.
- [5] T Drengstig og I W Ni og K Throsen og X M Xu og P Ruoff. «A Basic Set of Homeostatic Controller Motifs». I: *Biophysical Journal* 103 (2012), s. 2000–2010.
- [6] T Drengstig og K Throsen og I W Jolma og P Ruoff. «Fysiologiske reguleringsmekanismer». I: *NATUREN* 4 (2013), s. 128–137.
- [7] Tormod Drengstig og Kristian Thorsen og Peter Ruoff. «Control theoretic properties of physiological controller motifs». I: *ICSSE 2013. IEEE International Conference on System Science and Engineering* (jul. 2013), s. 165–170.
- [8] Duy Viet Nguyen. «Kalmanfilter anvendt på totankprosess for temperatur estimering». Masteroppgave, Universitetet i Stavanger, 2017.
- [9] Karl Skretting. *Notat 1 - Systemidentifikasjon*. Hentet fra <http://www.ux.uis.no/karlsk/E-LE620/notat1.pdf>. 2019.
- [10] Karl Skretting. *Notat 3 - Systemidentifikasjon*. Hentet fra <http://www.ux.uis.no/karlsk/E-LE620/notat3.pdf>. 2019.

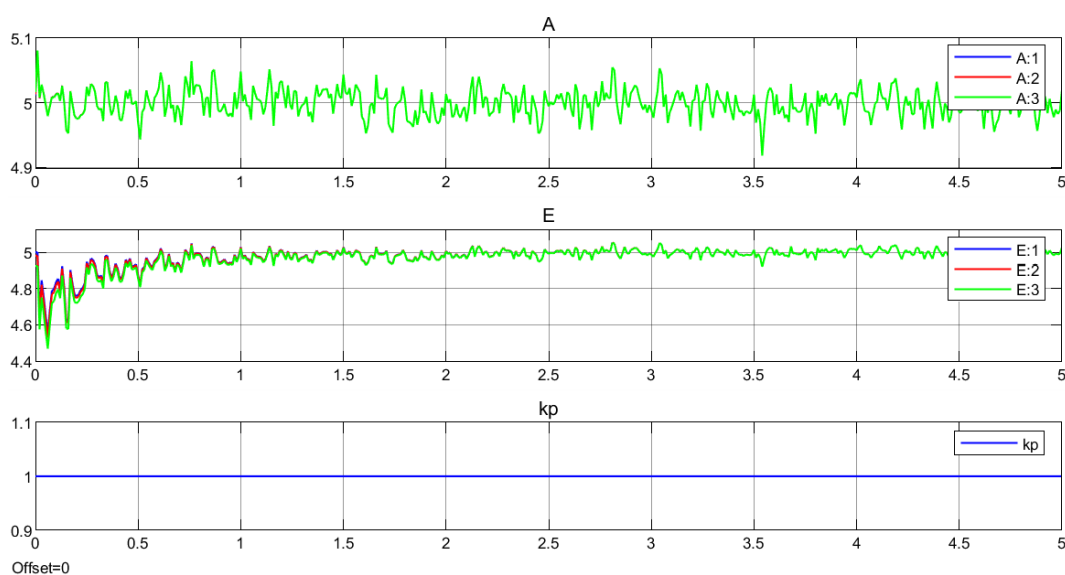
Tillegg A

Supplerende materiale kapittel 4

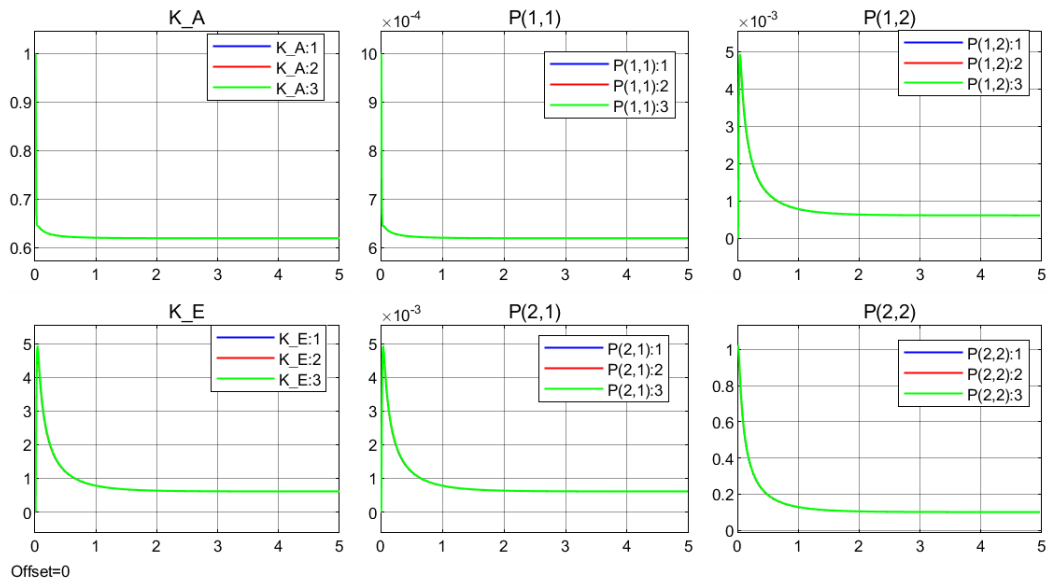
A.1 Simulink scopes for avvik i initialverdier for A

Disse simuleringene viser hvordan forskjellige initielle verdier for posteriori kovariansematrisen påvirker avvik i initielle verdier for tilstanden A . Disse simuleringene hører til delkapittel 4.4

A.1.1 Forsøk med fleksible $\hat{P}(0)$ verdier

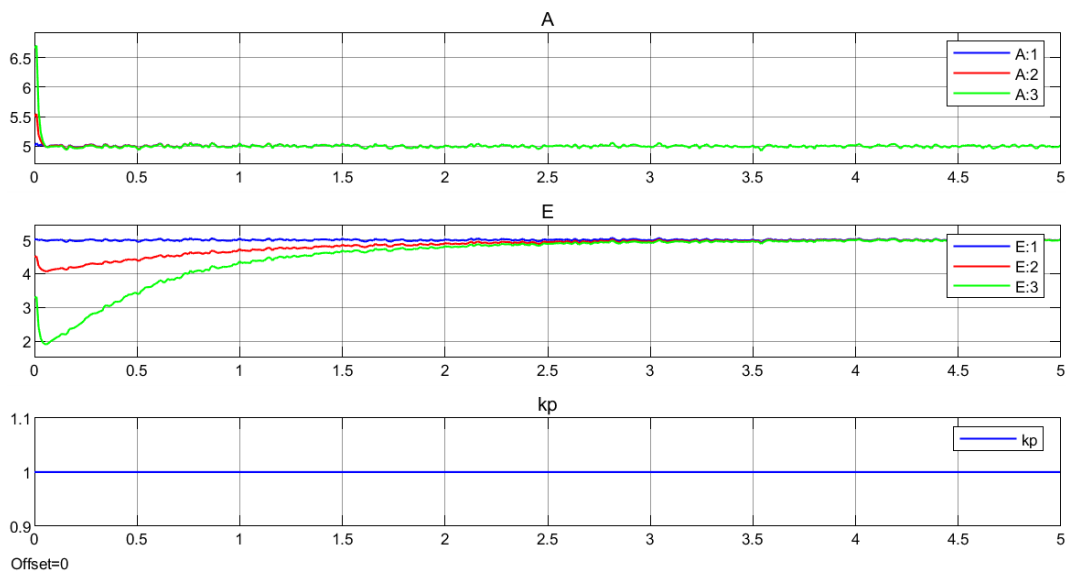


FIGUR A.1: Simulink scope av A , E og k_p med avvik i A . Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [6.5, 5, 1]$, grønn strek: $[A, E, k_p] = [10, 5, 1]$

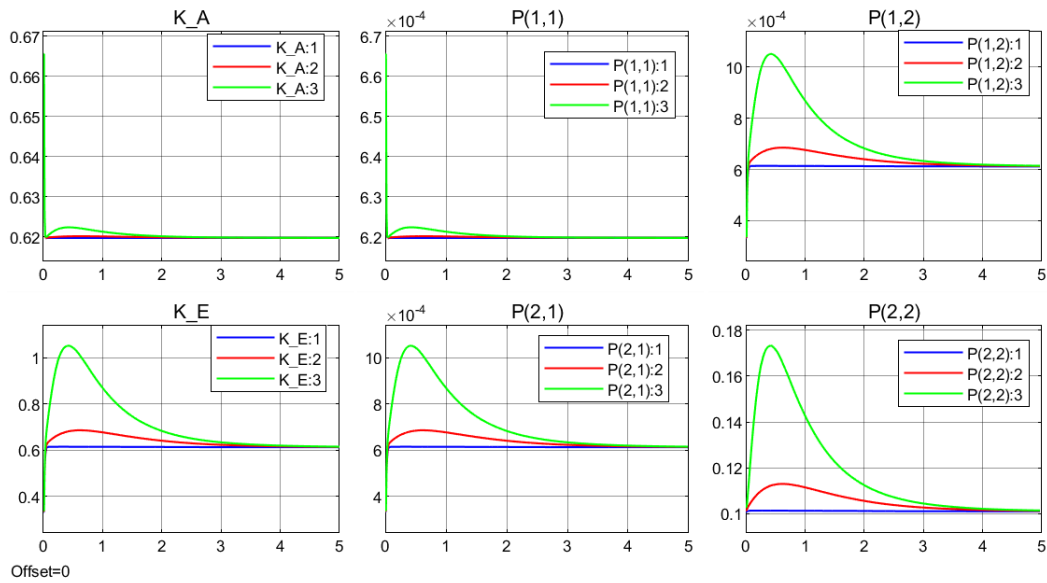


FIGUR A.2: Simulink *scope* av K og P med avvik i A. Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [6.5, 5, 1]$, grønn strek: $[A, E, k_p] = [10, 5, 1]$

A.1.2 Forsøk med spesifiserte $\hat{P}(0)$ verdier



FIGUR A.3: Simulink *scope* av A, E og k_p med avvik i A. Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [6.5, 5, 1]$, grønn strek: $[A, E, k_p] = [10, 5, 1]$

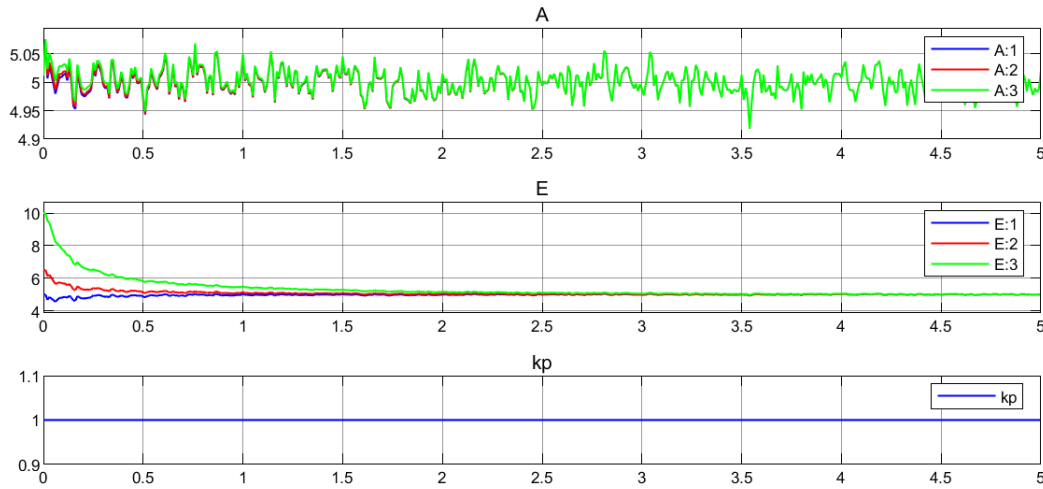


FIGUR A.4: Simulink *scope* av K og P med avvik i A. Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [6.5, 5, 1]$, grønn strek: $[A, E, k_p] = [10, 5, 1]$

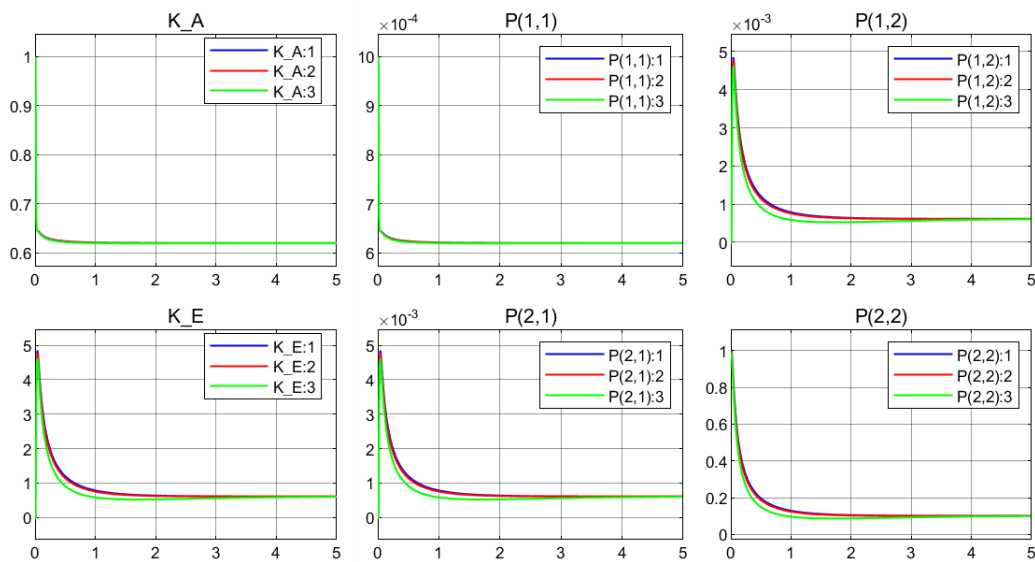
A.2 Simulink *scopes* for avvik i initialverdier for E

Disse simuleringene viser hvordan forskjellige initielle verdier for a posteriori kovariansematrisen påvirker avvik i initielle verdier for tilstanden E . Disse simuleringene hører til delkapittel 4.4

A.2.1 Forsøk med fleksible $\hat{P}(0)$ verdier

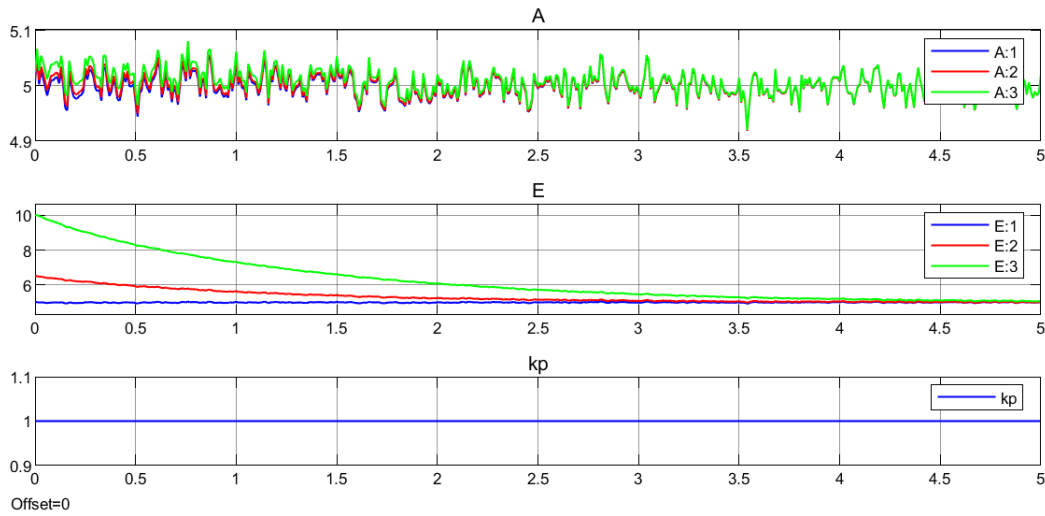


FIGUR A.5: Simulink scope av A , E og k_p med avvik i E . Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [5, 6.5, 1]$, grønn strek: $[A, E, k_p] = [5, 10, 1]$

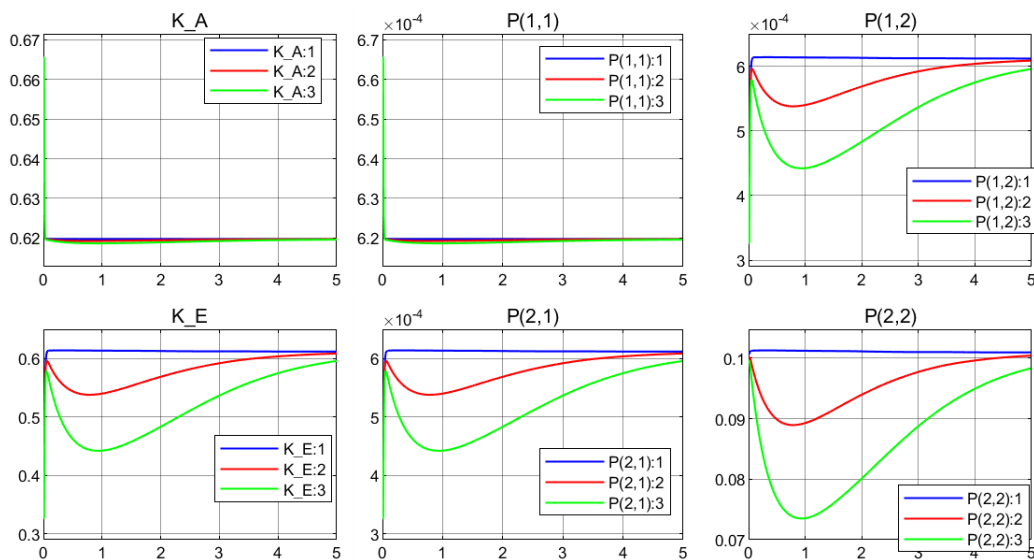


FIGUR A.6: Simulink scope av K og P med avvik i E . Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [5, 6.5, 1]$, grønn strek: $[A, E, k_p] = [5, 10, 1]$

A.2.2 Forsøk med spesifiserte $\hat{P}(0)$ verdier



FIGUR A.7: Simulink scope av A , E og k_p med avvik i E . Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [5, 6.5, 1]$, grønn strek: $[A, E, k_p] = [5, 10, 1]$



FIGUR A.8: Simulink scope av K og P med avvik i E . Blå strek: $[A, E, k_p] = [5, 5, 1]$, rød strek: $[A, E, k_p] = [5, 6.5, 1]$, grønn strek: $[A, E, k_p] = [5, 10, 1]$

A.3 Støyreduksjonstest

I dette vedlegget kan man se snutter av simuleringer som viser hvordan ulike valg av prosessstøy kovariansmatrise gir ulik reduksjon av målestøy i systemet. Disse simuleringene hører til kapittel 4.5.

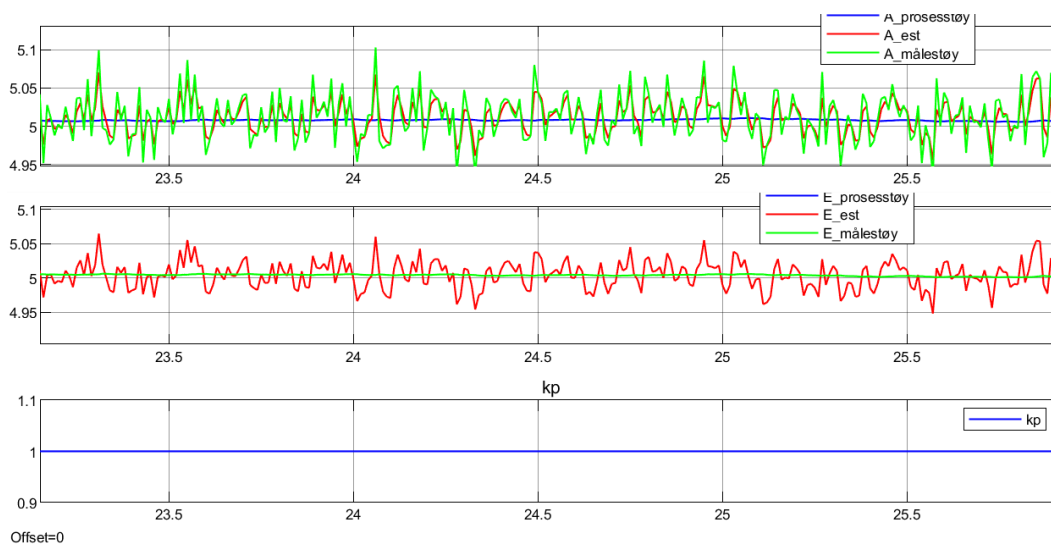
A.3.1 Støyreduksjon med lik prosess- og målestøy kovariansmatrise

I dette forsøket er:

$$Q = \begin{bmatrix} 1 \cdot 10^{-3} & 0 \\ 0 & 1 \cdot 10^{-3} \end{bmatrix}$$

og:

$$R = 1 \cdot 10^{-3}$$



FIGUR A.9: Simulink *scope* av A , E og k_p for test av estimering oppmot modell med kun prosessstøy oppmot modell med prosess- og målestøy. Blå strek: kun prosessstøy, rød strek: estimat, grønn strek: prosess- og målestøy

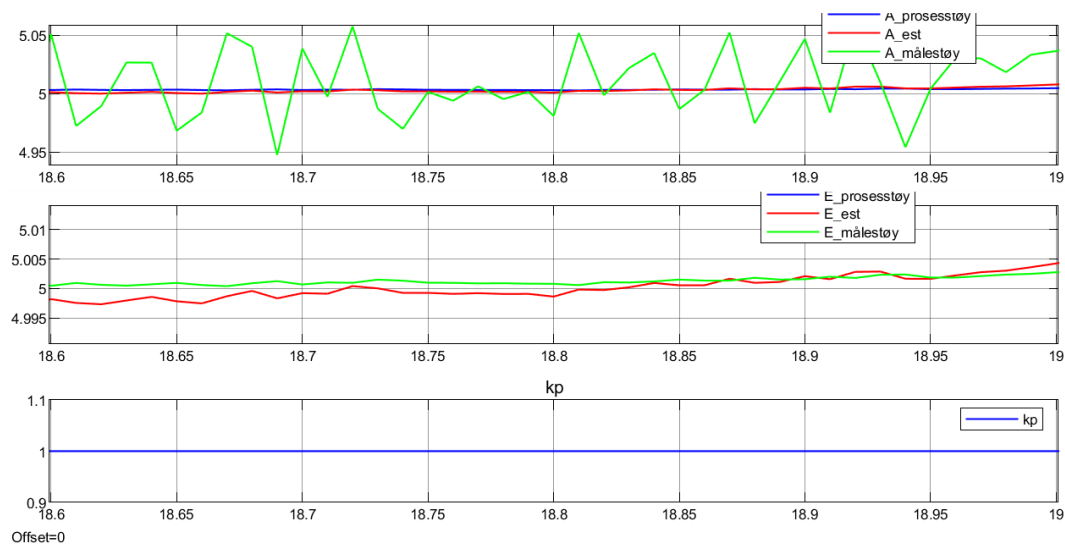
A.3.2 Støyreduksjon med høyere målestøy kovariansmatrise

I dette forsøket er:

$$Q = \begin{bmatrix} 1 \cdot 10^{-3} & 0 \\ 0 & 1 \cdot 10^{-3} \end{bmatrix}$$

og:

$$R = 1$$



FIGUR A.10: Simulink scope av A , E og k_p for test av estimering oppmot modell med kun prosesstøy oppmot modell med prosess- og målestøy. Blå strek: kun prosesstøy, rød strek: estimat, grønn strek: prosess- og målestøy

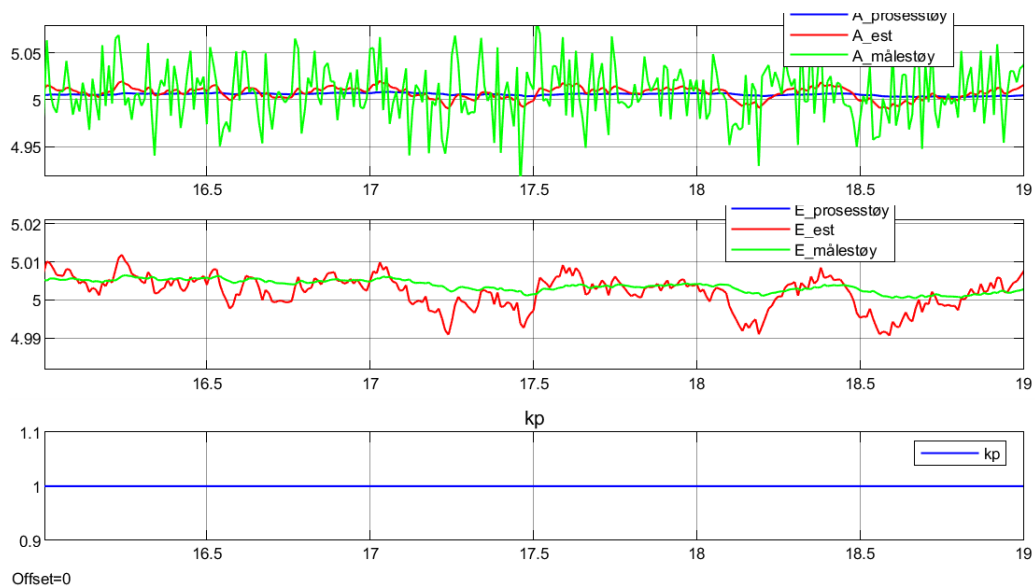
A.3.3 Støyreduksjon med «spesifiserte» prosess- og målestøy kovariansmatriser

I dette forsøket er:

$$Q = \begin{bmatrix} 1 \cdot 10^{-5} & 0 \\ 0 & 5 \cdot 10^{-6} \end{bmatrix}$$

og:

$$R = 1 \cdot 10^{-3}$$

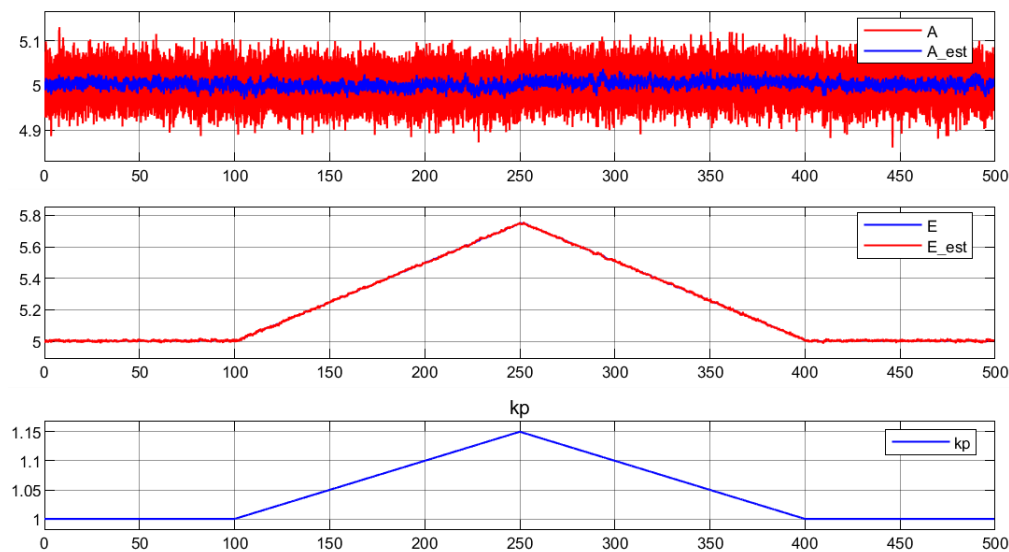


FIGUR A.11: Simulink *scope* av A , E og k_p for test av estimering oppmot modell med kun prosesstøy oppmot modell med prosesstøy og målestøy. Blå strek: kun prosesstøy, rød strek: estimat, grønn strek: prosesstøy og målestøy

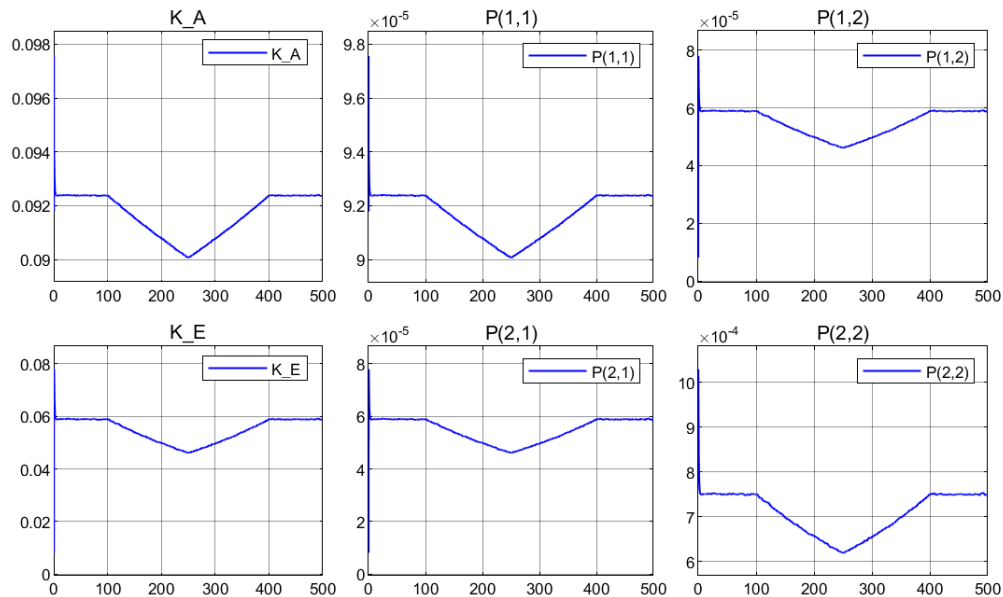
A.4 Test av forstyrrelser

I dette vedlegget er simuleringer som viser ulike typer responser lagt ved. Disse simuleringene hører til delkapittel [4.6](#)

A.4.1 Rampe

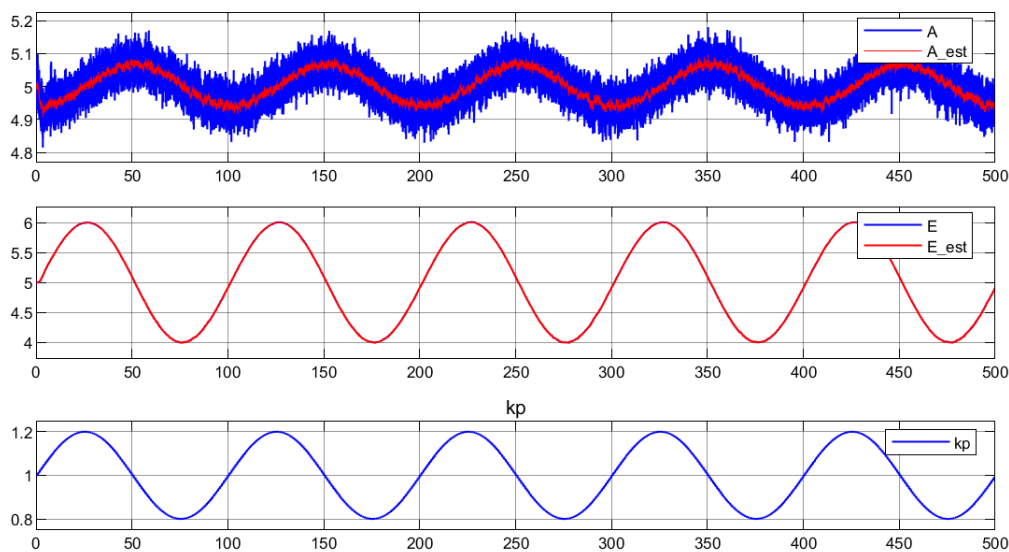


FIGUR A.12: Simulink scope av A , E og k_p med en rampe som forstyrrelse

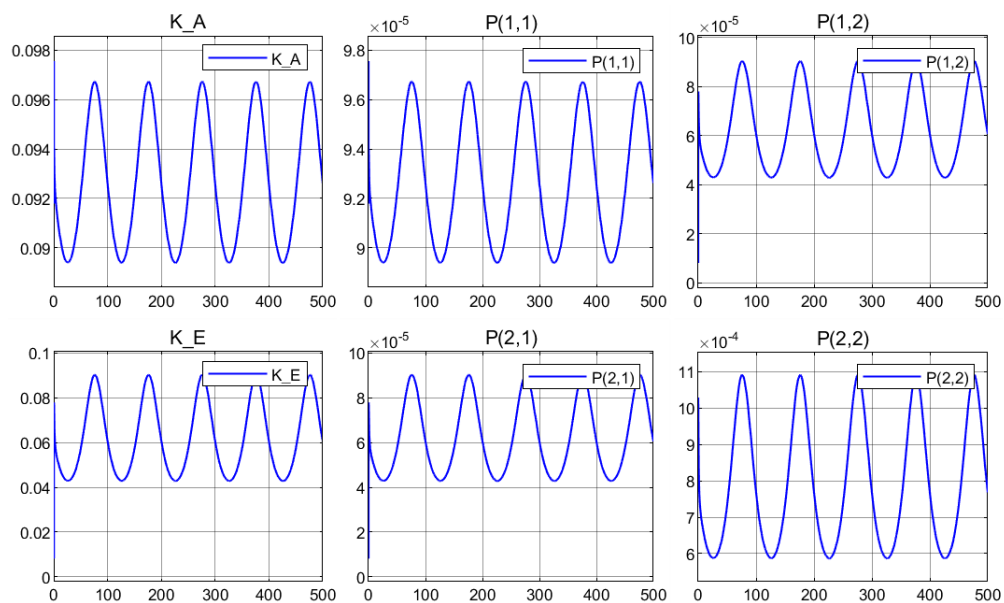


FIGUR A.13: Simulink scope av K og P med en rampe som forstyrrelse

A.4.2 Sinus

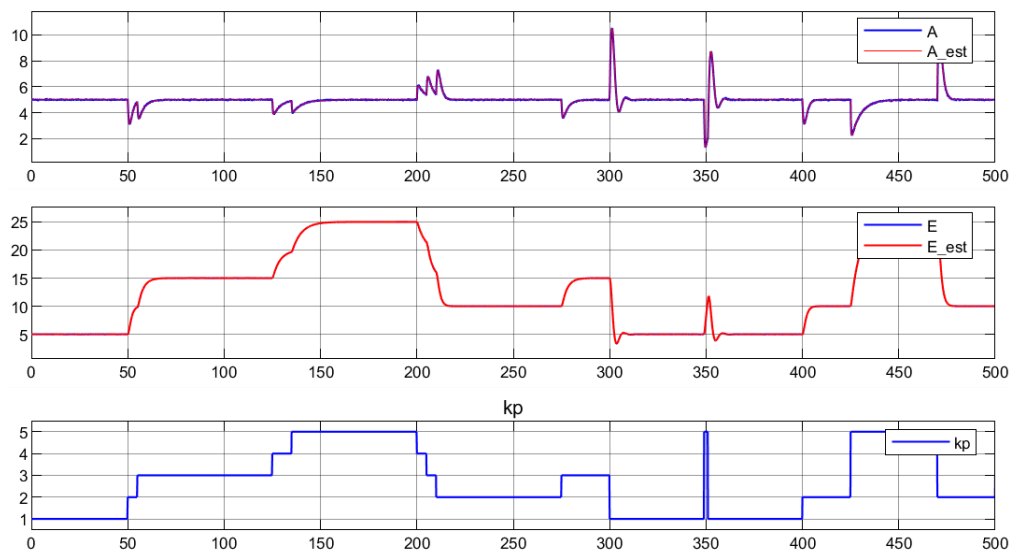


FIGUR A.14: Simulink scope av A , E og k_p med en sinus som forstyrrelse

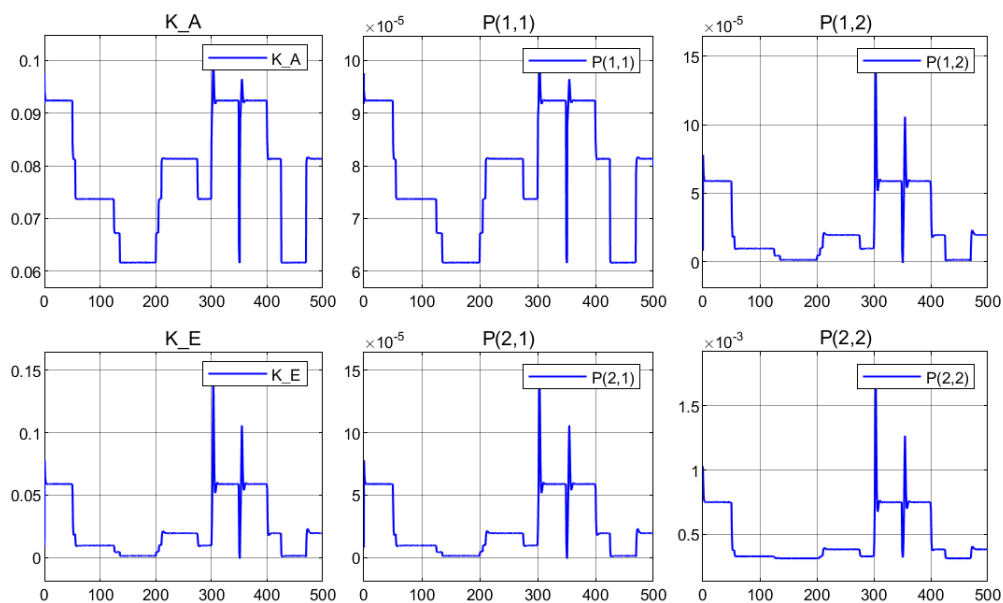


FIGUR A.15: Simulink scope av K og P med en sinus som forstyrrelse

A.4.3 Pseudorandom nummer sekvens



FIGUR A.16: Simulink scope av A , E og k_p med en Pseudorandom nummer sekvens som forstyrrelse



FIGUR A.17: Simulink scope av K og P med en Pseudorandom nummer sekvens som forstyrrelse

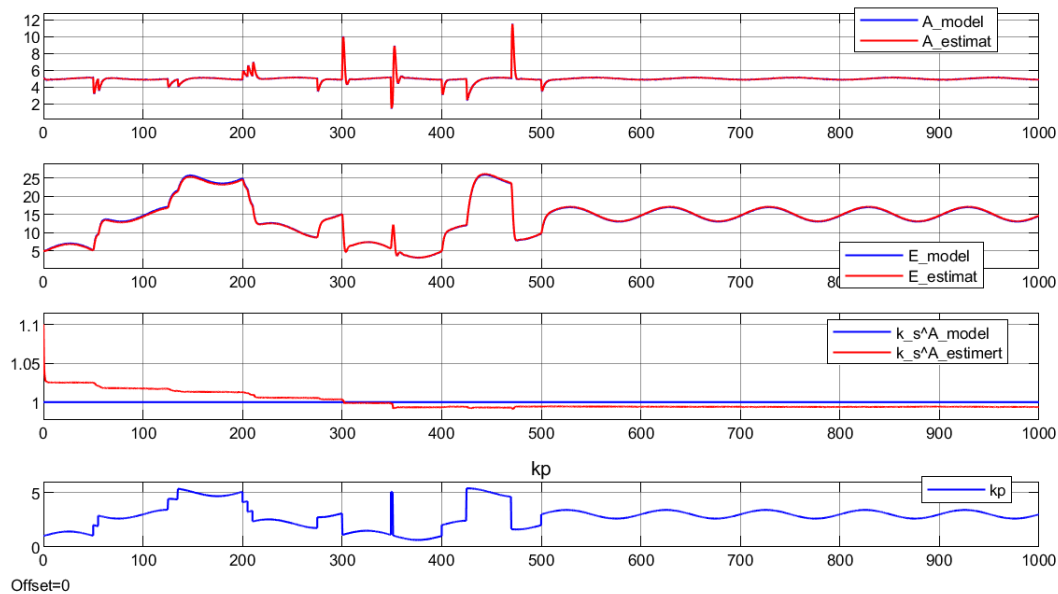
Tillegg B

Supplerende materiale kapittel 5

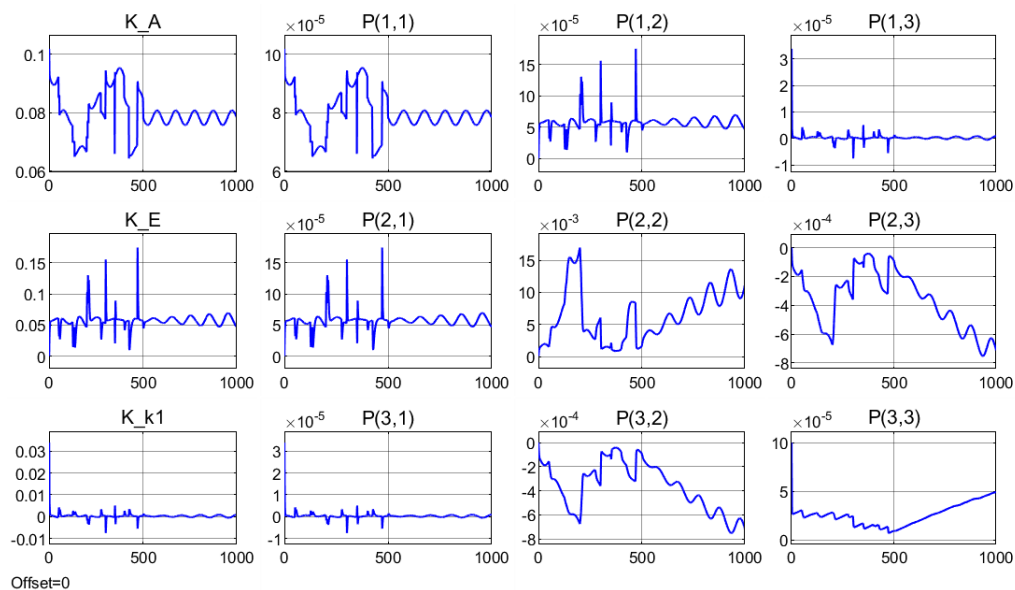
B.1 Estimering av en tilleggs parameter med måling av A

I dette vedlegget er simuleringer av alle 4 kombinasjoner av parametere som er testet i delkapittel 5.2 med en måling av tilstanden A

B.1.1 Estimering av k_S^A

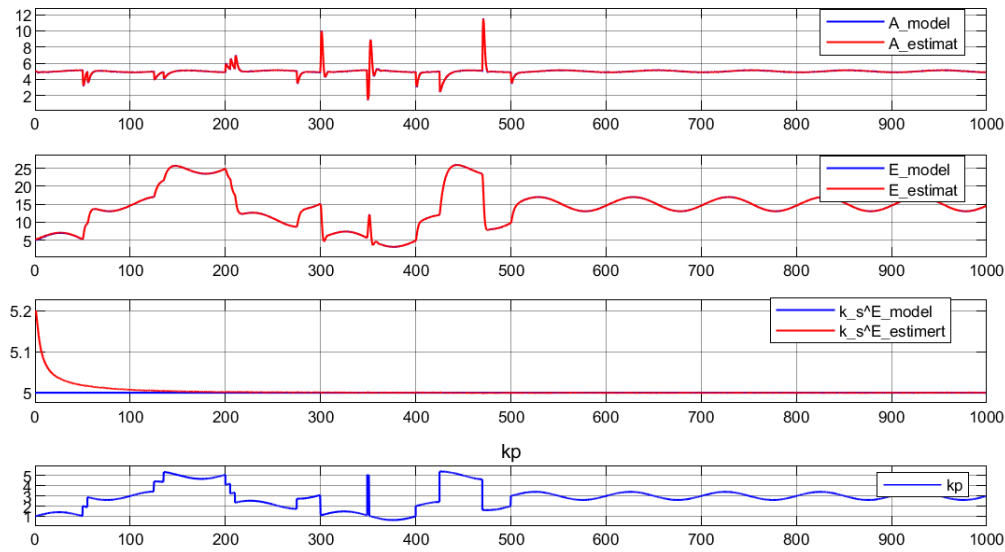


FIGUR B.1: Simulink scope av A, E og k_S^A med måling av A

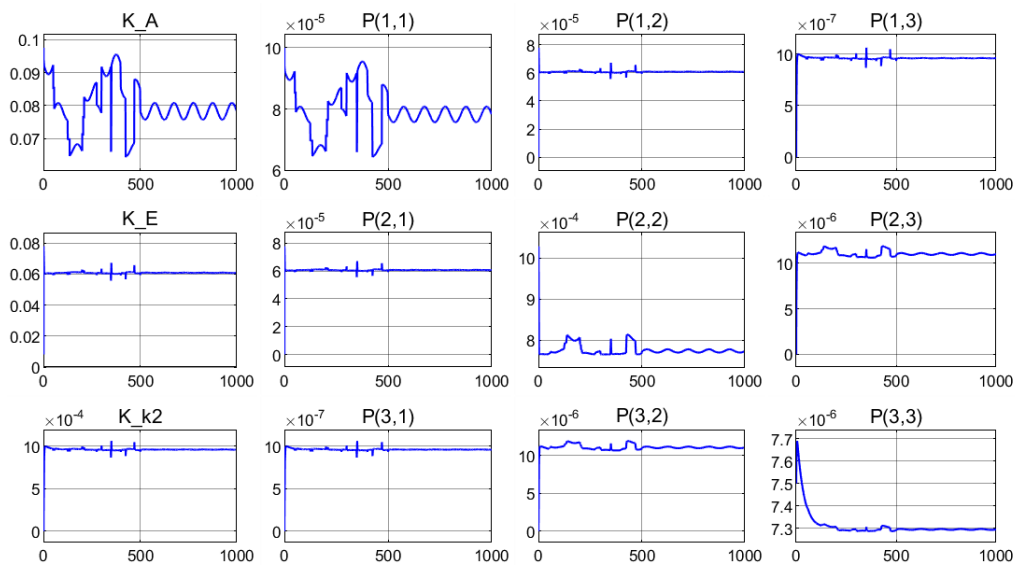


FIGUR B.2: Simulink scope av K og P for estimering av k_S^A

B.1.2 Estimering av k_S^E

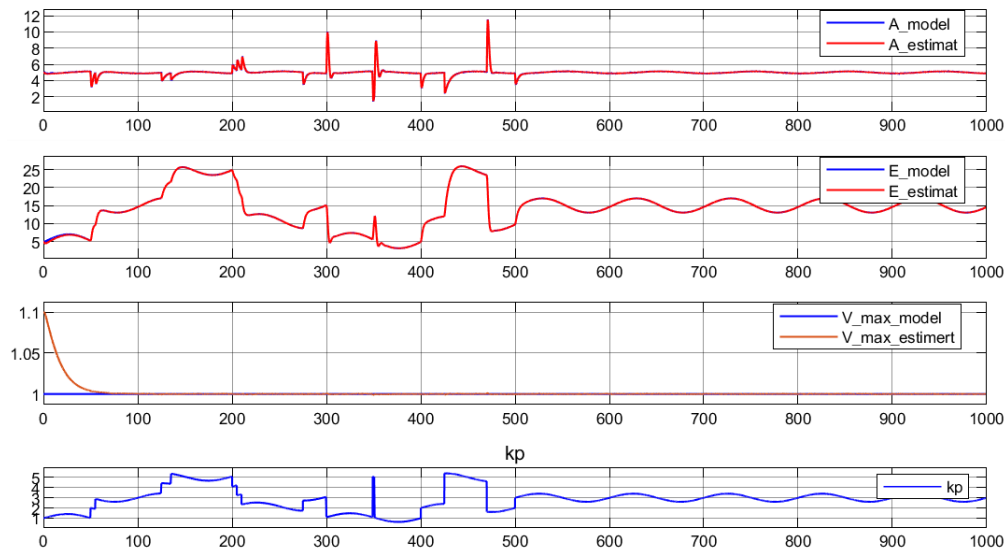


FIGUR B.3: Simulink scope av A, E og k_S^E med måling av A

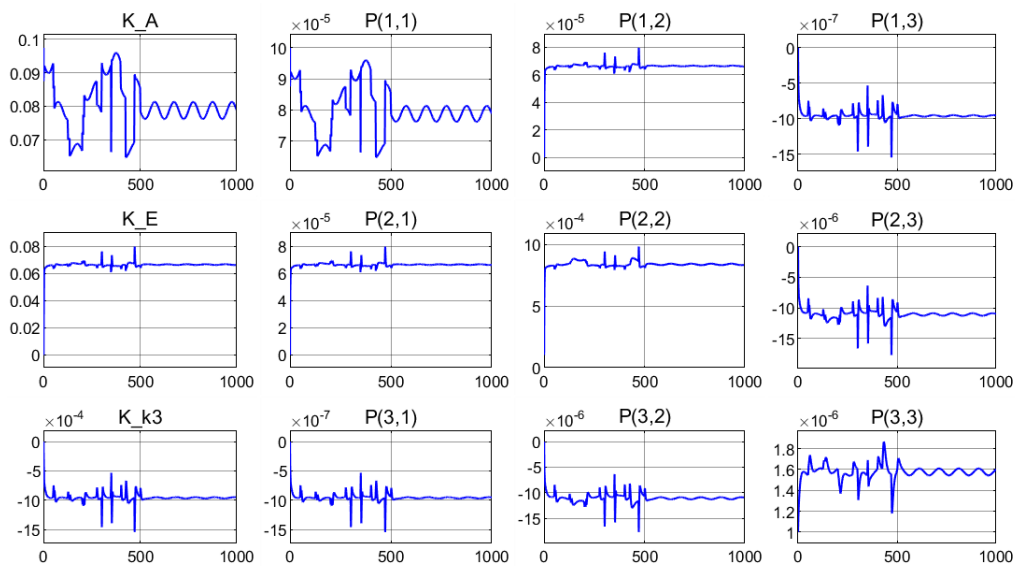


FIGUR B.4: Simulink scope av K og P for estimering av k_S^E

B.1.3 Estimering av V_{max}

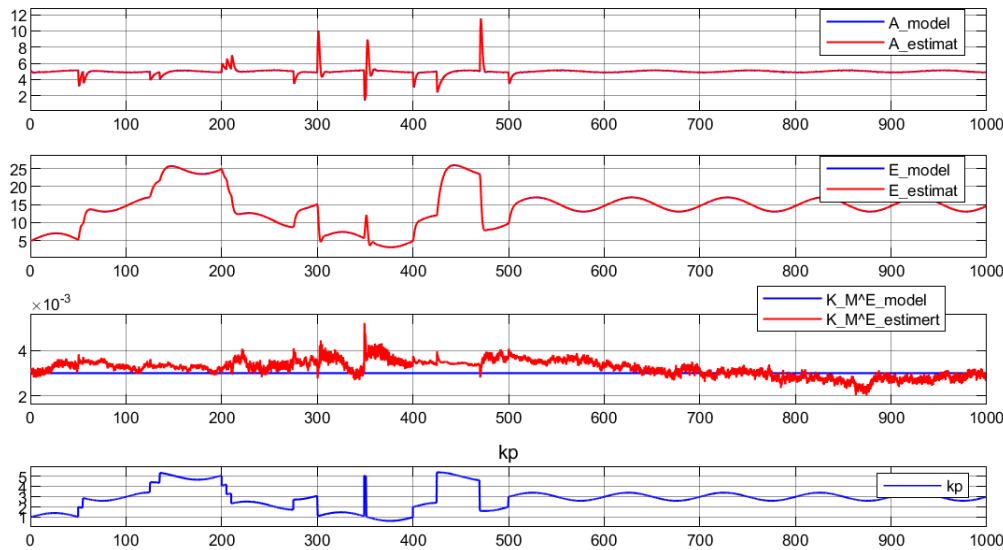


FIGUR B.5: Simulink scope av A, E og V_{max} med måling av A

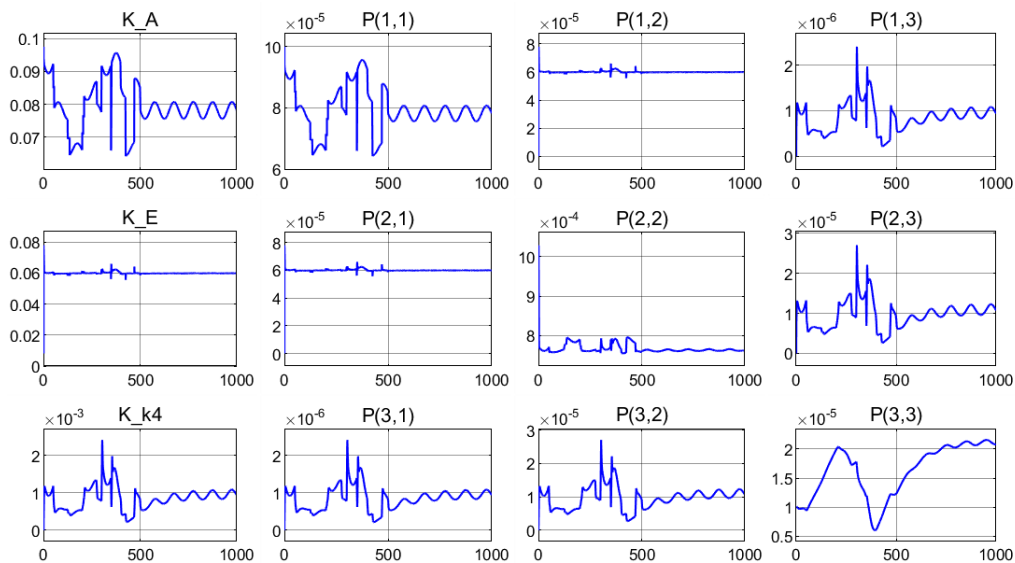


FIGUR B.6: Simulink scope av K og P for estimering av V_{max}

B.1.4 Estimering av K_M^E



FIGUR B.7: Simulink scope av A, E og K_M^E med måling av A

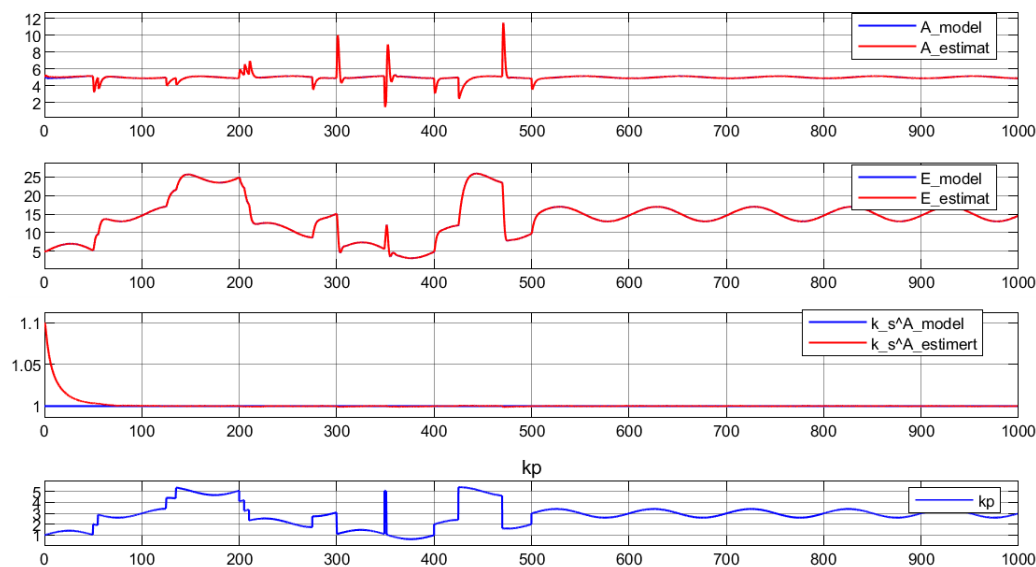


FIGUR B.8: Simulink scope av K og P for estimering av K_M^E

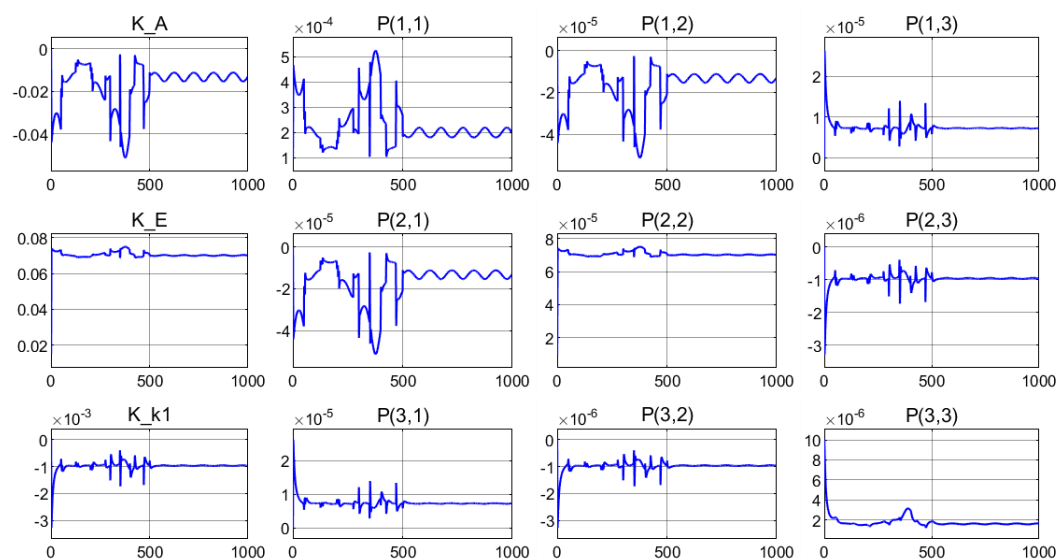
B.2 Estimering av en tilleggs parameter med måling av E

I dette vedlegget er simuleringer av alle 4 kombinasjoner av parametere som er testet i delkapittel 5.2 med en måling av tilstanden E

B.2.1 Estimering av k_S^A

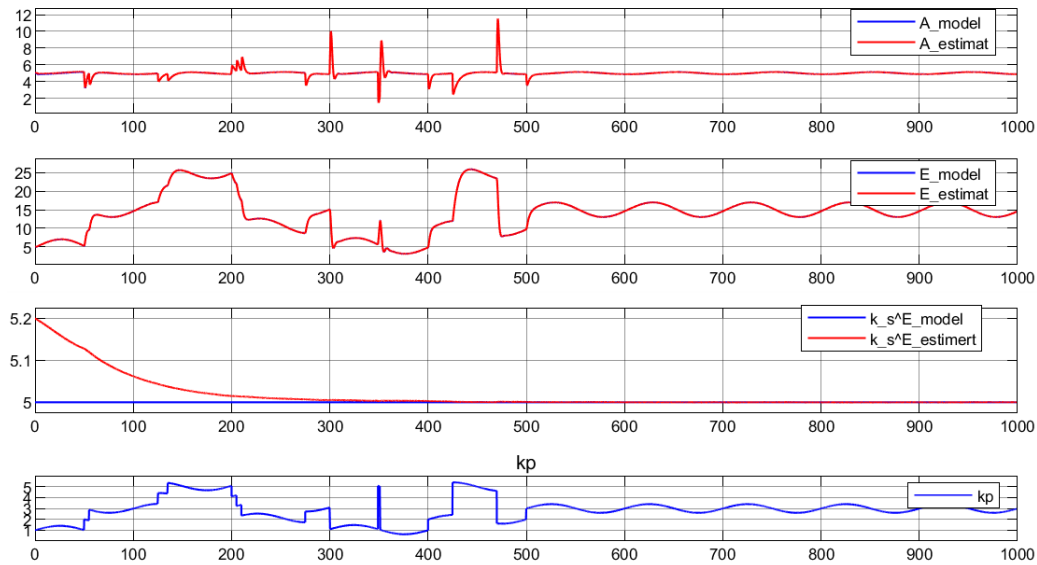


FIGUR B.9: Simulink scope av A, E og k_S^A med måling av E

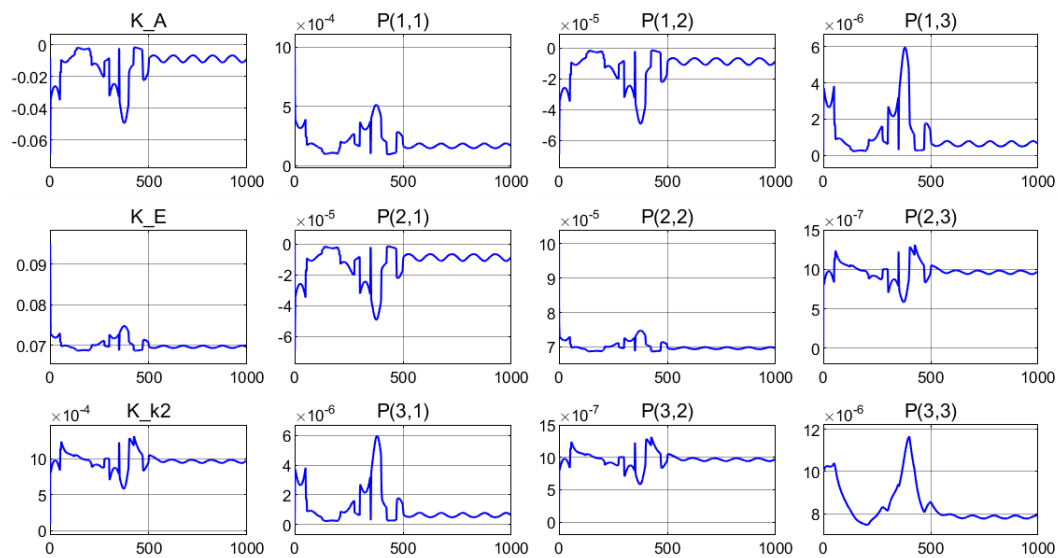


FIGUR B.10: Simulink scope av K og P for estimering av k_S^A

B.2.2 Estimering av k_S^E

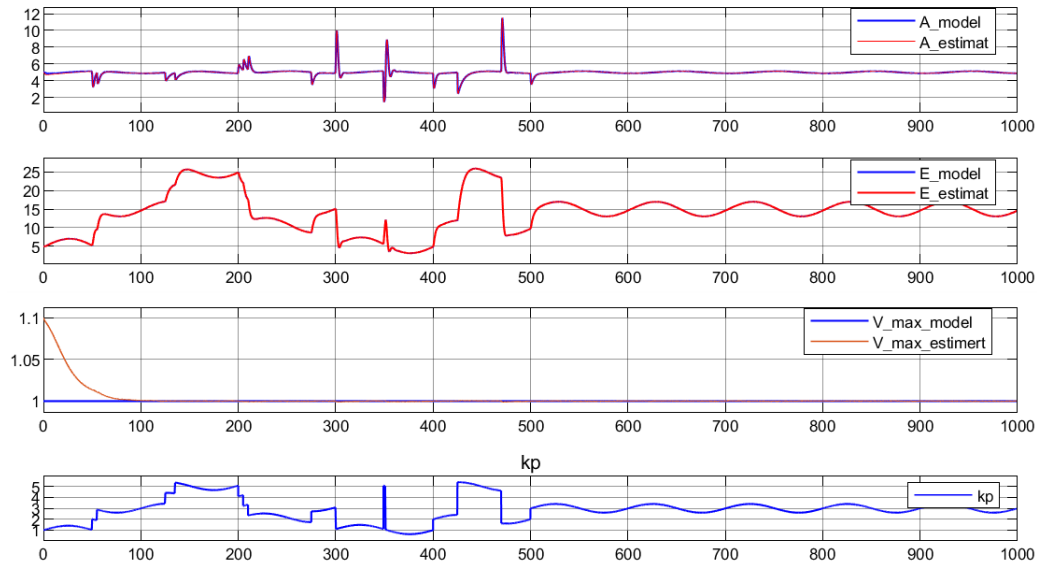


FIGUR B.11: Simulink scope av A , E og k_S^E med måling av E

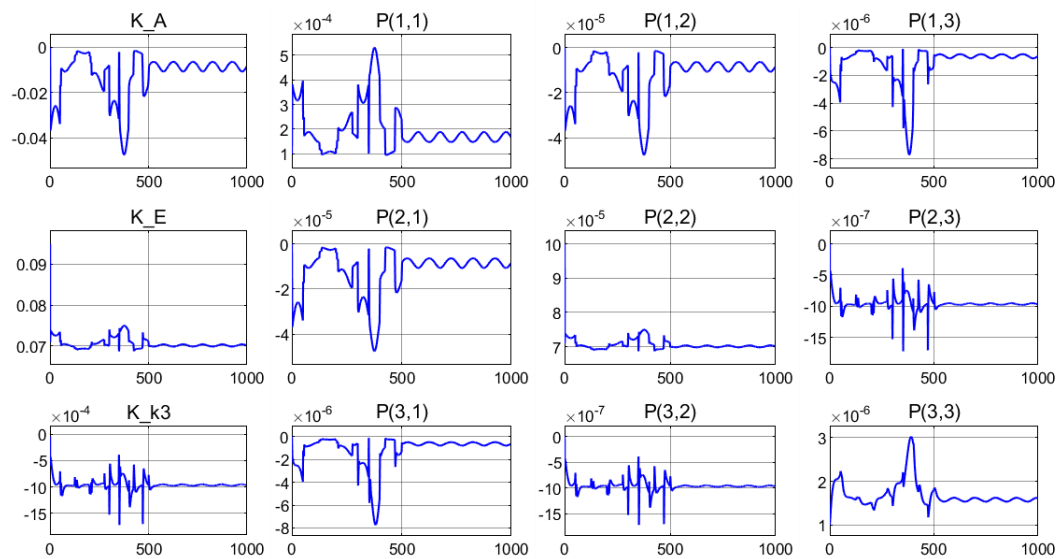


FIGUR B.12: Simulink scope av K og P for estimering av k_S^E

B.2.3 Estimering av V_{max}

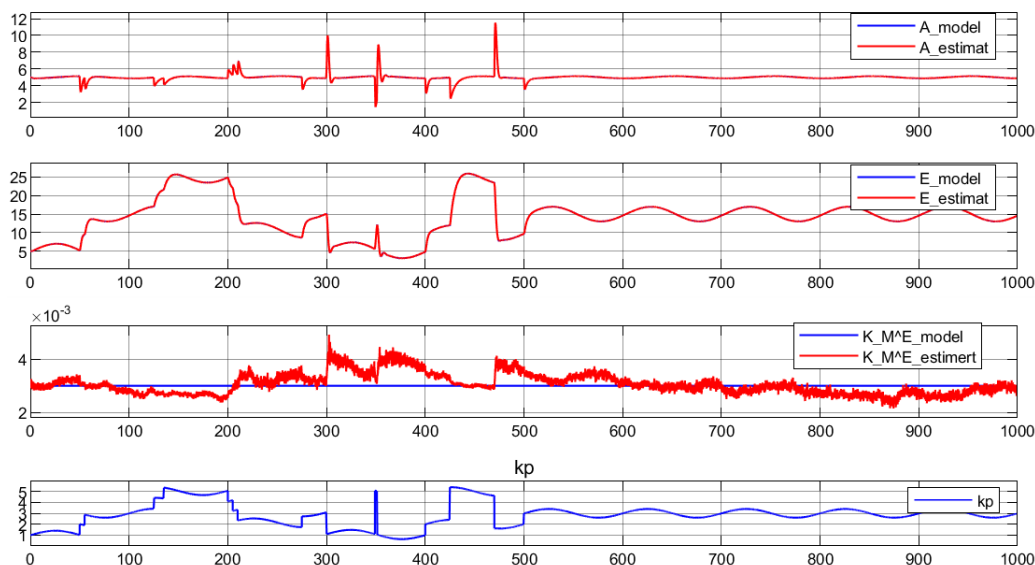


FIGUR B.13: Simulink scope av A, E og V_{max} med måling av E

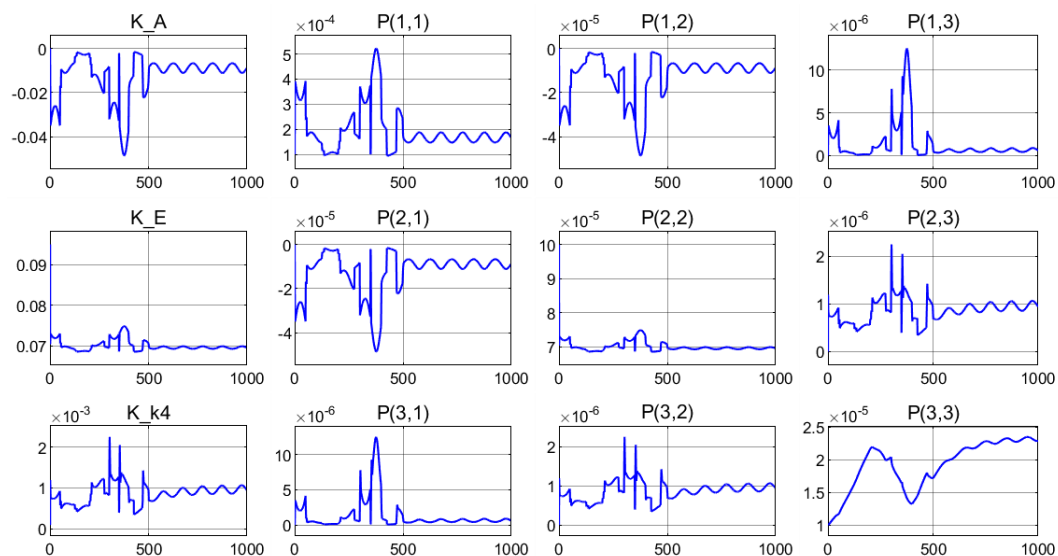


FIGUR B.14: Simulink scope av K og P for estimering av V_{max}

B.2.4 Estimering av K_M^E



FIGUR B.15: Simulink scope av A, E og K_M^E med måling av A

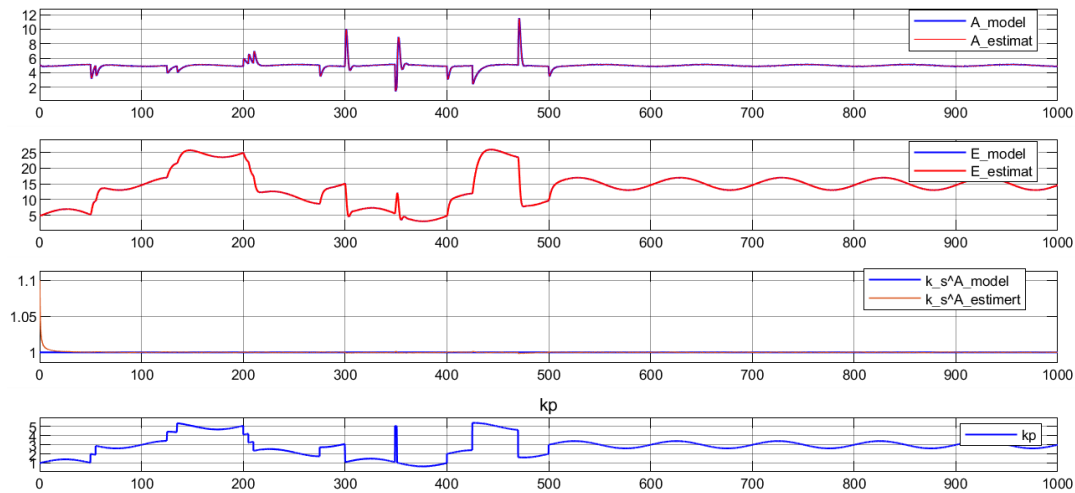


FIGUR B.16: Simulink scope av K og P for estimering av K_M^E

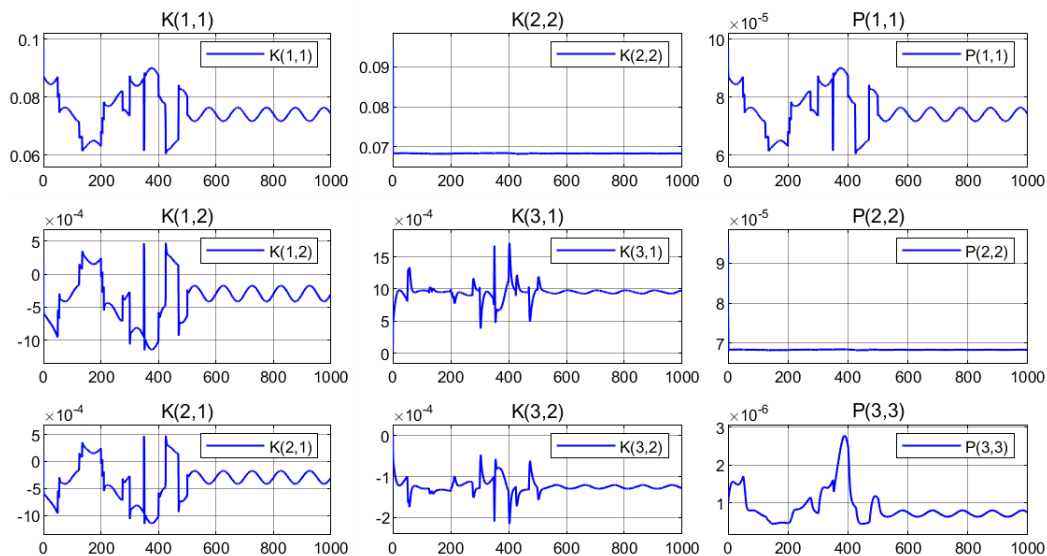
B.3 Estimering av 1 parameter med måling av begge tilstander

I dette vedlegget er simuleringer av alle 4 kombinasjoner av parametere som er testet i delkapittel 5.3 med en måling av tilstanden A

B.3.1 Estimering av k_s^A med måling av begge tilstander

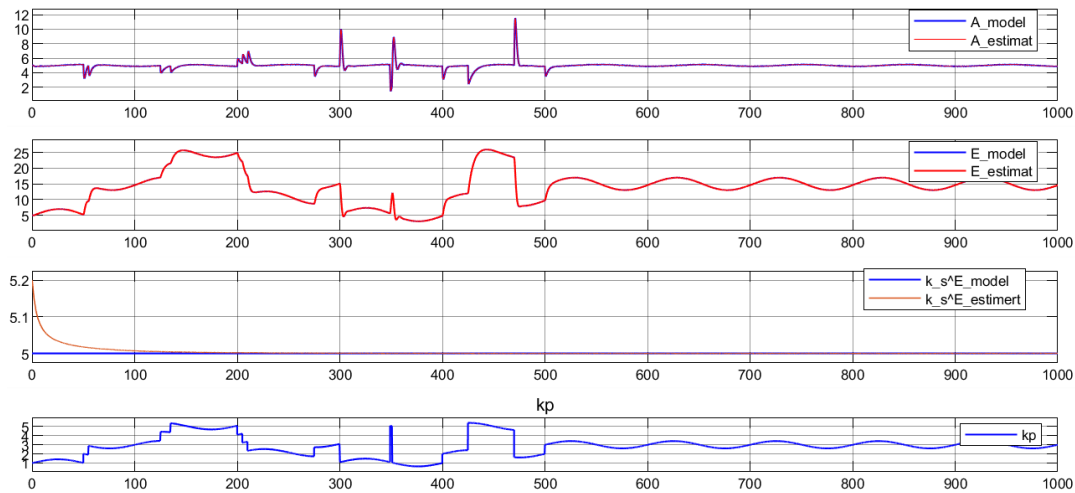


FIGUR B.17: Simulink scope av A , E og k_s^A for parameterestimering med måling av 2 tilstander

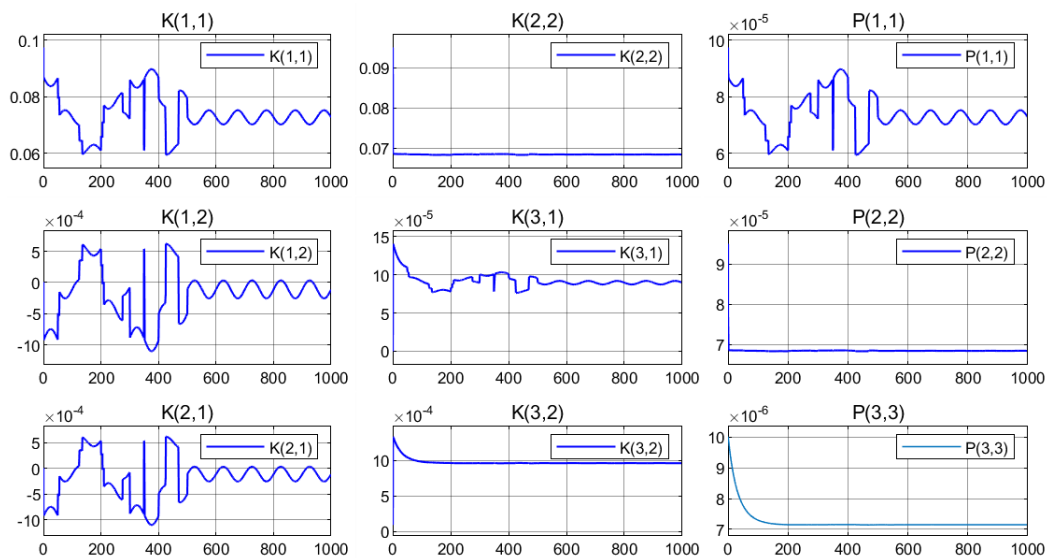


FIGUR B.18: Simulink scope av K og P for parameterestimering av k_s^A med måling av 2 tilstander

B.3.2 Estimering av k_s^E med måling av begge tilstander

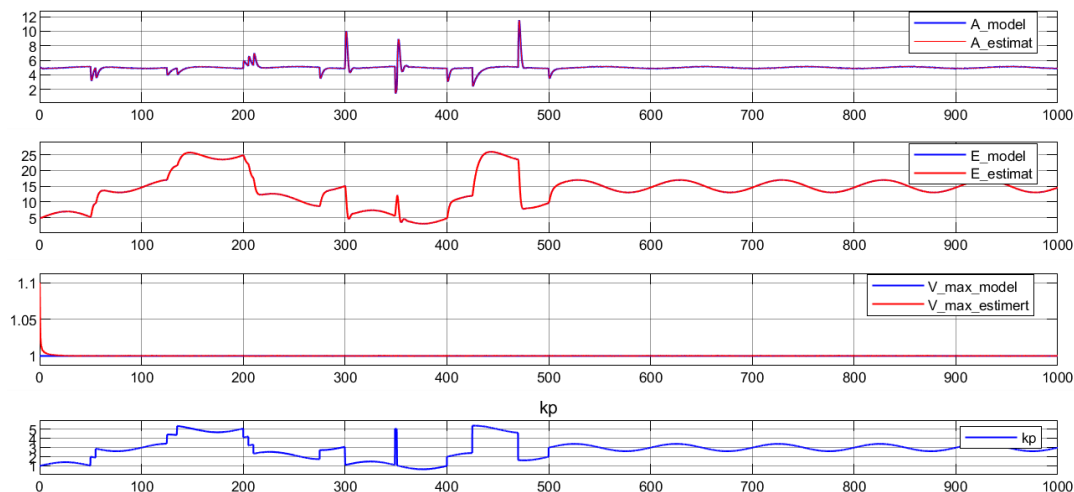


FIGUR B.19: Simulink scope av A , E og k_s^E for parameterestimering med måling av 2 tilstander

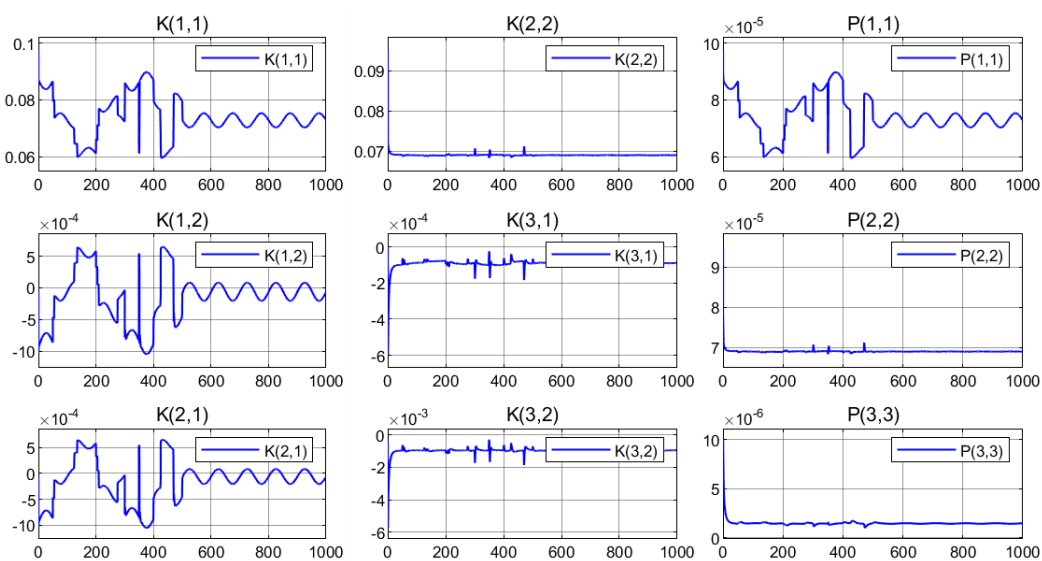


FIGUR B.20: Simulink scope av K og P for parameterestimering av k_s^E med måling av 2 tilstander

B.3.3 Estimering av V_{max} med måling av begge tilstander

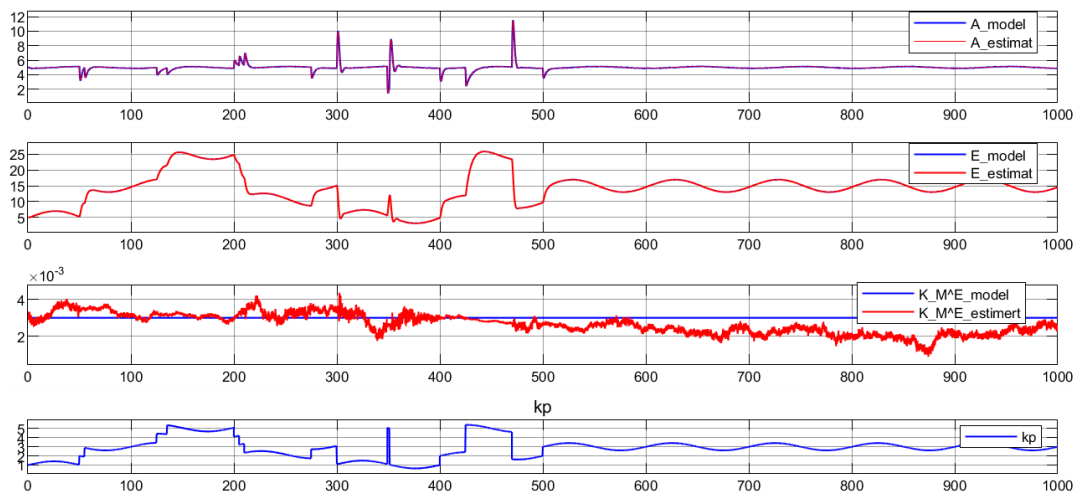


FIGUR B.21: Simulink scope av A , E og V_{max} for parameterestimering med måling av 2 tilstander

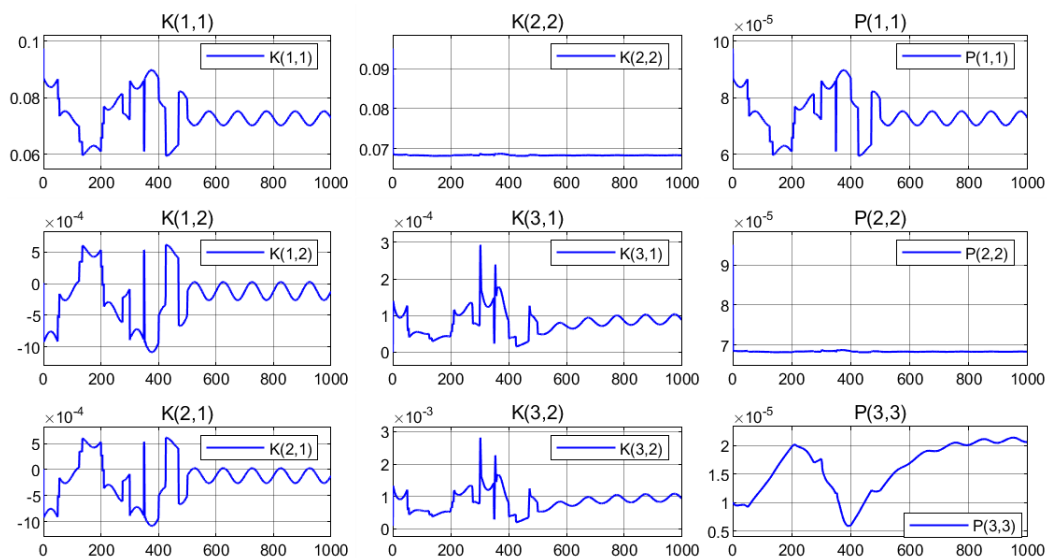


FIGUR B.22: Simulink scope av K og P for parameterestimering av k_S^E med måling av 2 tilstander

B.3.4 Estimering av K_M^E med måling av begge tilstander



FIGUR B.23: Simulink scope av A , E og K_M^E for parameterestimering med måling av 2 tilstander

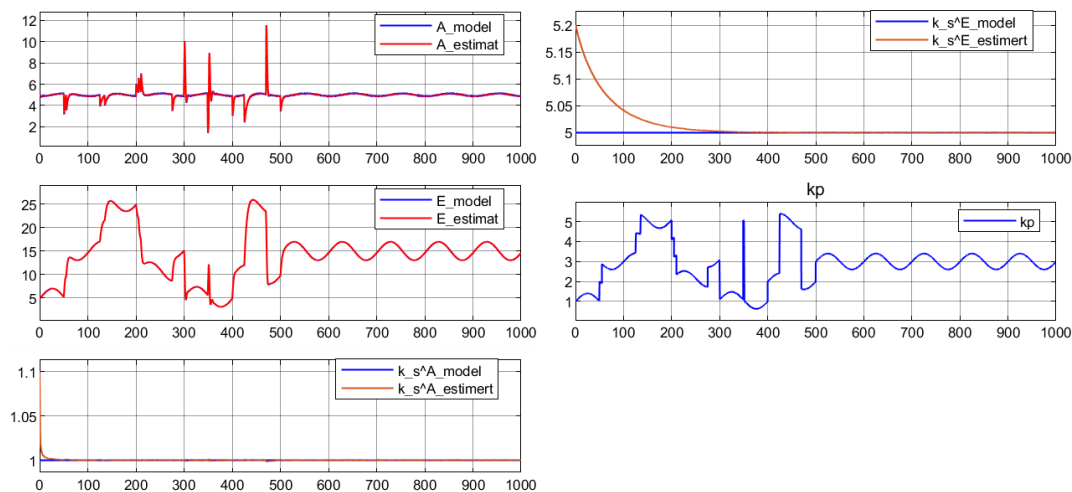


FIGUR B.24: Simulink scope av K og P for parameterestimering av k_S^E med måling av 2 tilstander

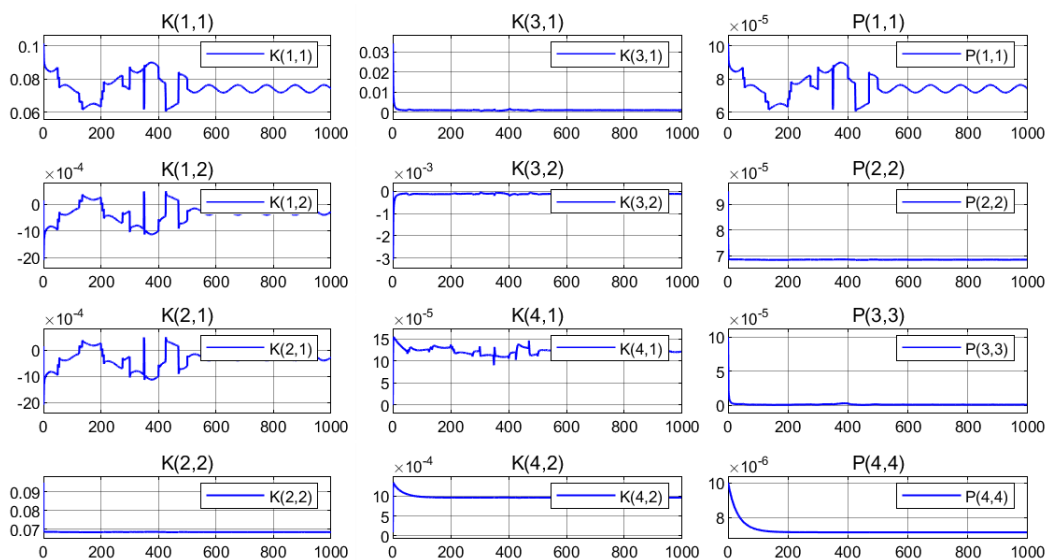
B.4 Estimering av 2 parametere med måling av begge tilstander

I dette vedlegget er simuleringer av alle 6 kombinasjoner av parametere som er testet i delkapittel 5.4 med en måling av begge tilstander.

B.4.1 Estimering av k_s^A og k_s^E med måling av begge tilstander

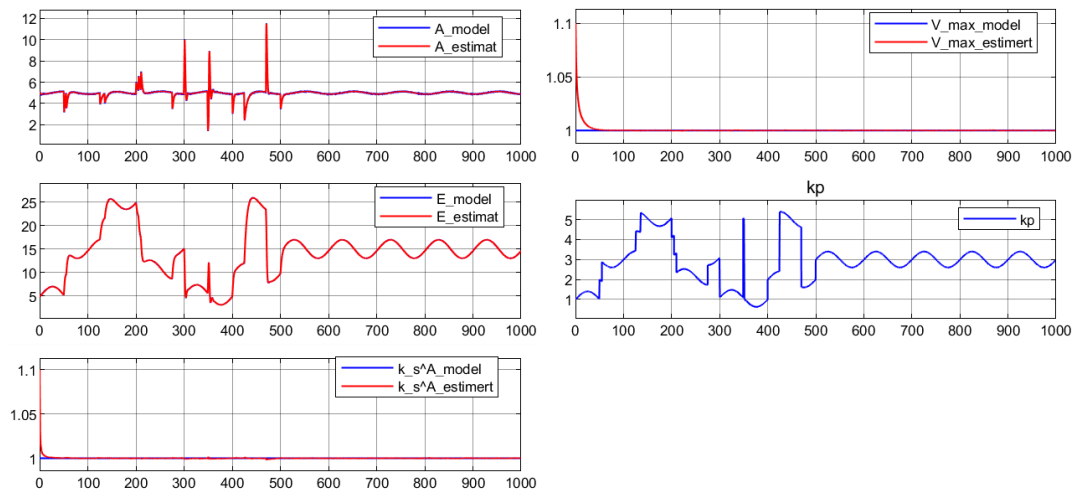


FIGUR B.25: Simulink scope av A , E , k_s^A og k_s^E estimert med måling av begge tilstandene

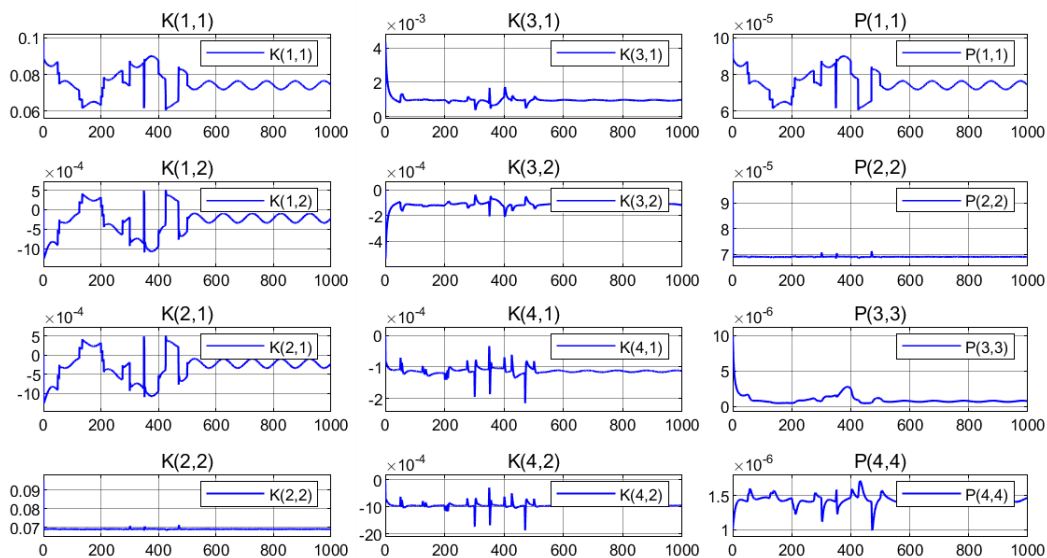


FIGUR B.26: Simulink scope av K og P for parameterestimering av k_s^A og k_s^E med måling av begge tilstandene

B.4.2 Estimering av k_s^A og V_{max} med måling av begge tilstander

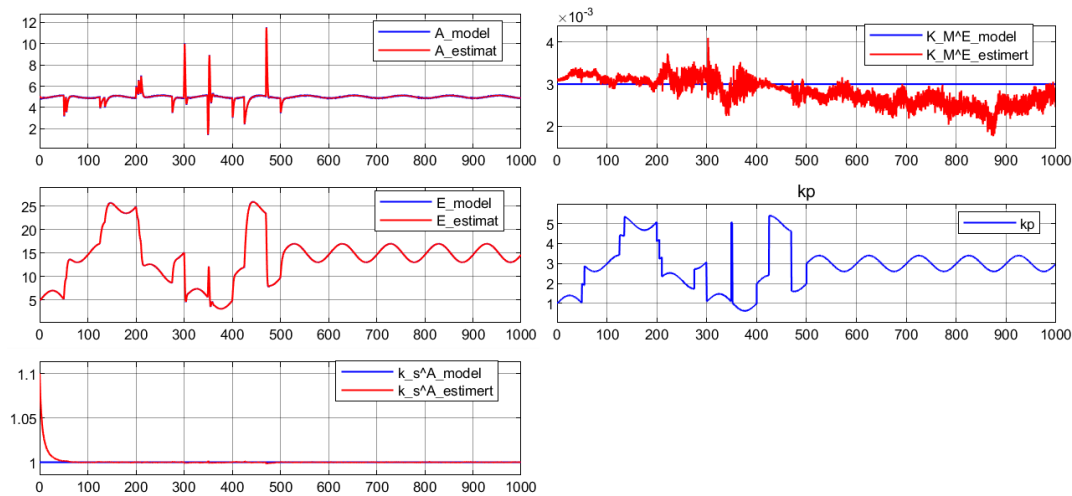


FIGUR B.27: Simulink scope av A , E , k_s^A og V_{max} estimert med måling av begge tilstandene

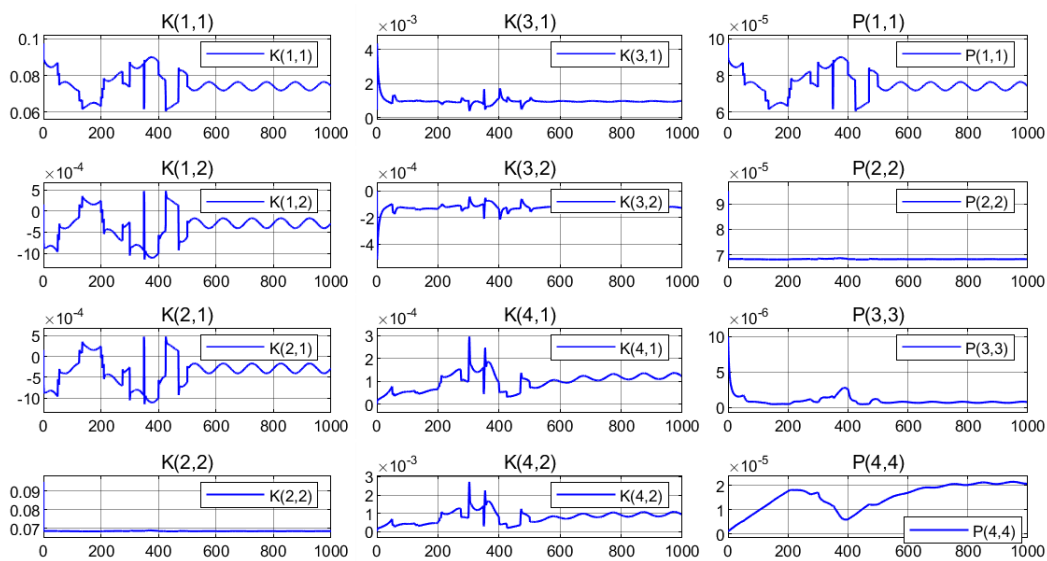


FIGUR B.28: Simulink scope av K og P for parameterestimering av k_s^A og V_{max} med måling av begge tilstandene

B.4.3 Estimering av k_s^A og K_M^E med måling av begge tilstander

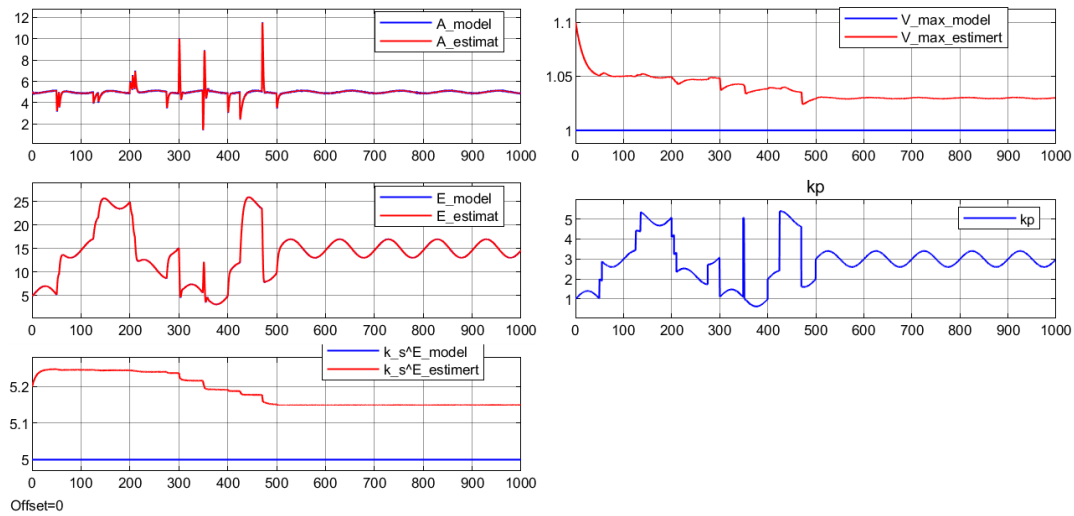


FIGUR B.29: Simulink *scope* av A , E , k_s^A og K_M^E estimert med måling av begge tilstandene

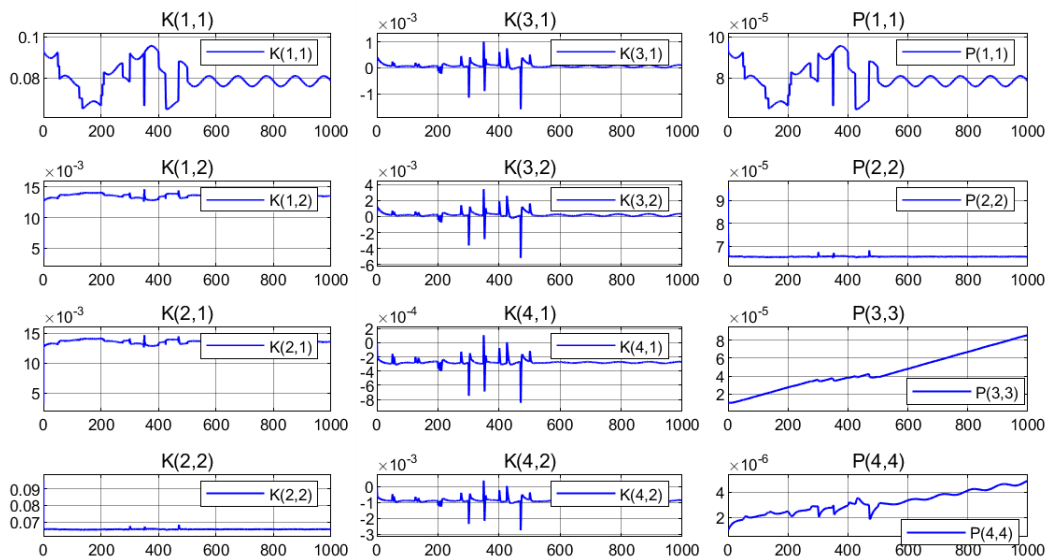


FIGUR B.30: Simulink *scope* av K og P for parameterestimering av k_s^A og K_M^E med måling av begge tilstandene

B.4.4 Estimering av k_s^E og V_{max} med måling av begge tilstander

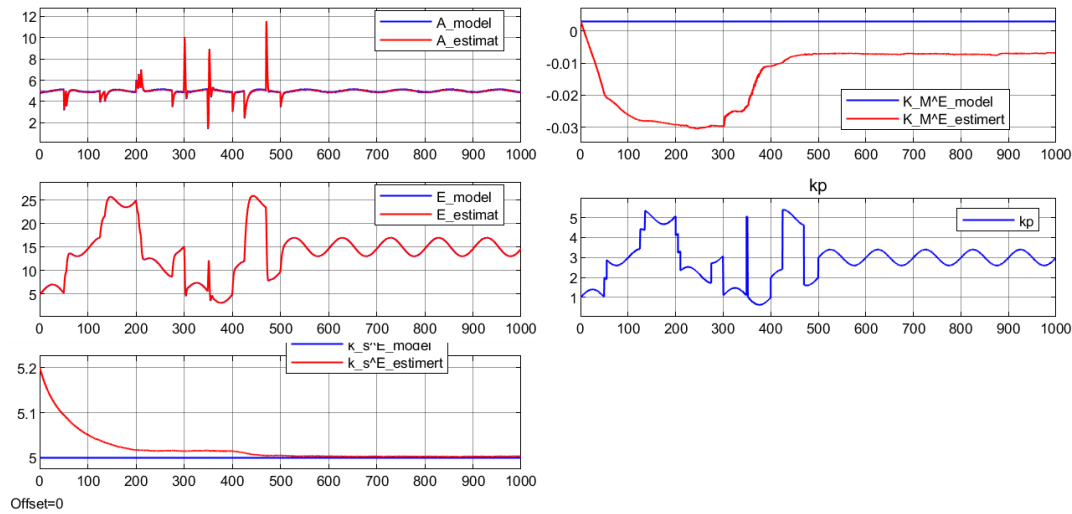


FIGUR B.31: Simulink scope av A , E , k_s^E og V_{max} estimert med måling av begge tilstandene

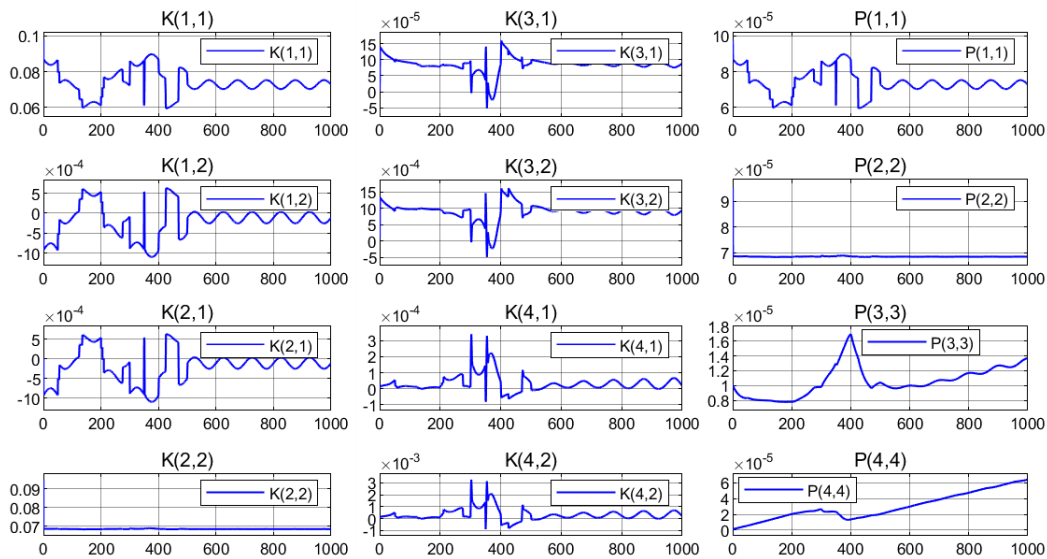


FIGUR B.32: Simulink scope av K og P for parameterestimering av k_s^E og V_{max} med måling av begge tilstandene

B.4.5 Estimering av k_s^E og K_M^E med måling av begge tilstander

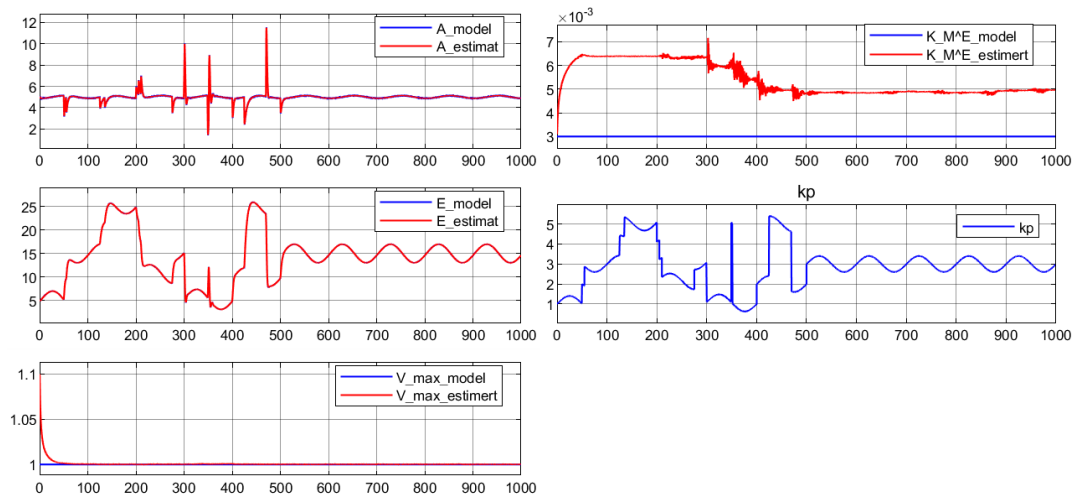


FIGUR B.33: Simulink scope av A , E , k_s^E og K_M^E estimert med måling av begge tilstandene

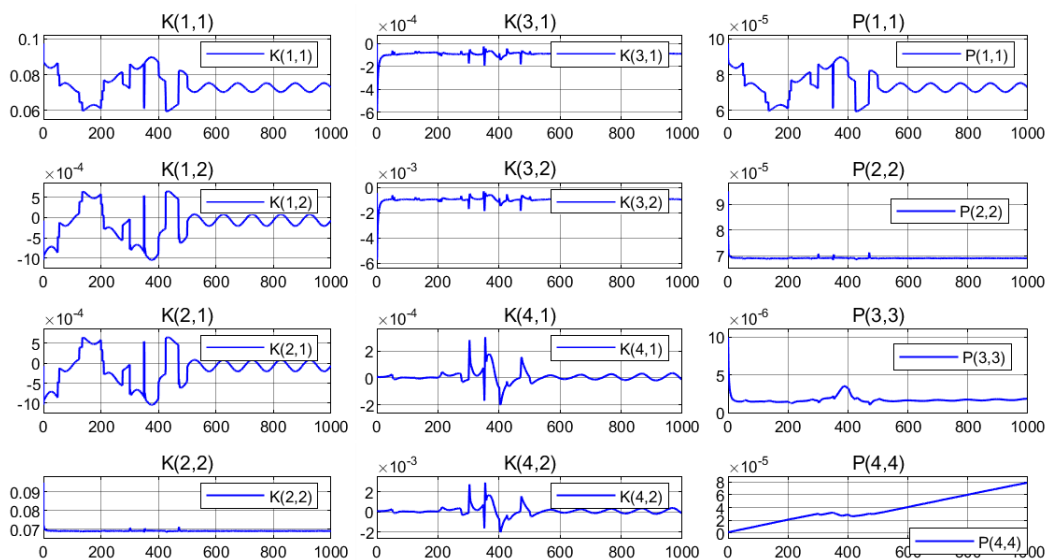


FIGUR B.34: Simulink scope av K og P for parameterestimering av k_s^E og K_M^E med måling av begge tilstandene

B.4.6 Estimering av V_{max} og K_M^E med måling av begge tilstander



FIGUR B.35: Simulink scope av A , E , V_{max} og K_M^E estimert med måling av begge tilstandene



FIGUR B.36: Simulink scope av K og P for parameterestimering av V_{max} og K_M^E med måling av begge tilstandene

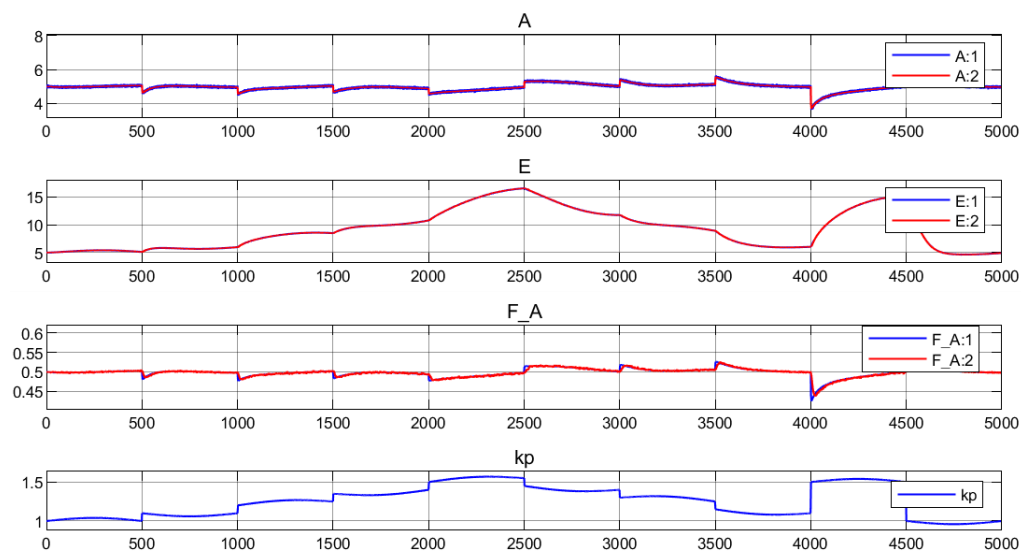
Tillegg C

Supplerende materiale kapittel 6

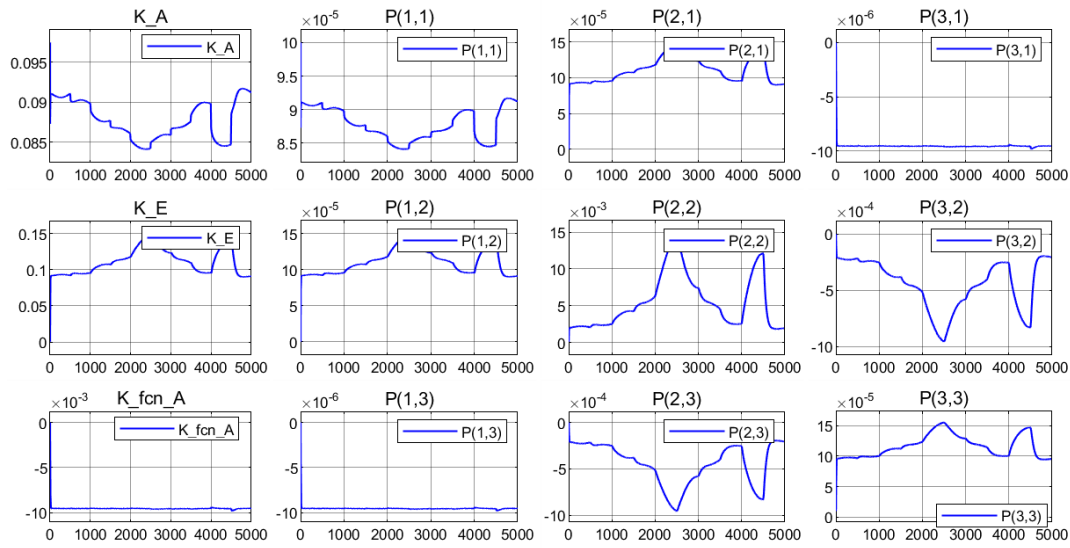
C.1 Estimering av funksjonen $\frac{A}{K_a^A + A}$ med augmentert Kalmanfilter

I dette vedlegget er simuleringer av estimeringen av funksjonen F_A lagt ved. Disse simuleringene hører til delkapittel 6.3

C.1.1 Simulink scope av A , E og $\frac{A}{K_a^A + A}$ med måling av A

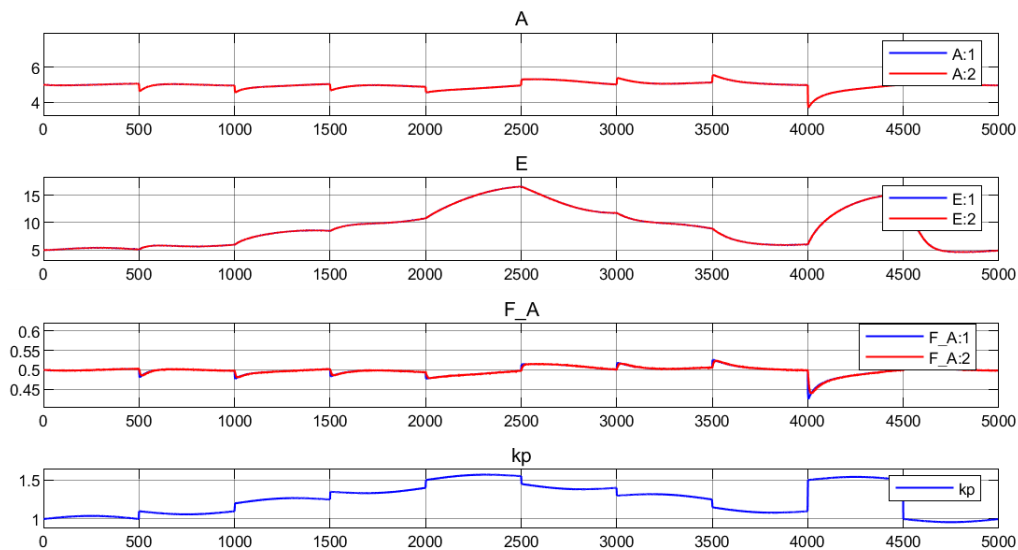


FIGUR C.1: Simulink scope av A , E , F_A og k_p med måling av A

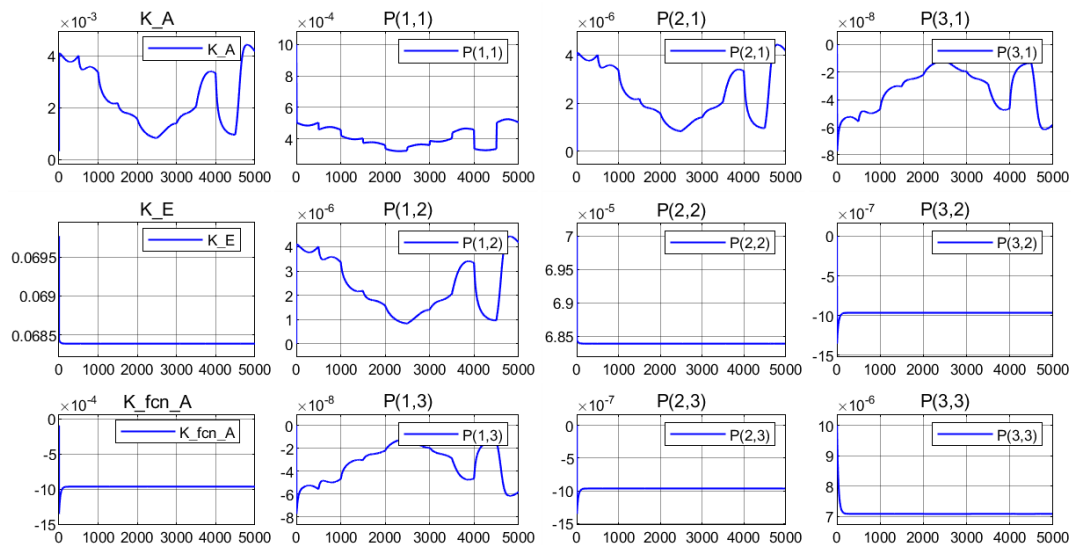


FIGUR C.2: Simulink scope av K og P for estimering av F_E med måling av A

C.1.2 Simulink scope av A , E og $\frac{A}{K_a^A + A}$ med måling av E



FIGUR C.3: Simulink scope av A , E , F_A og k_p med måling av E

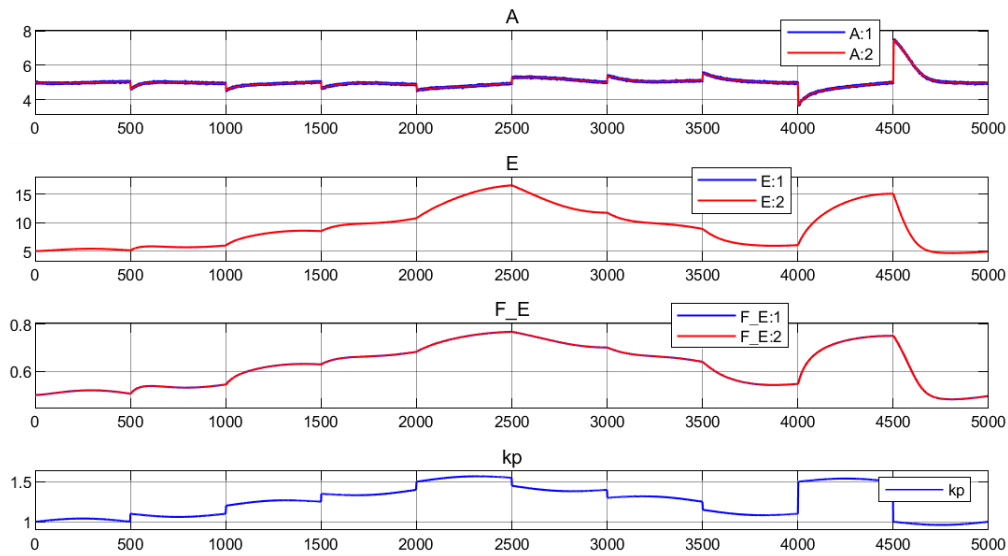


FIGUR C.4: Simulink *scope* av K og P for estimering av F_E med måling av E

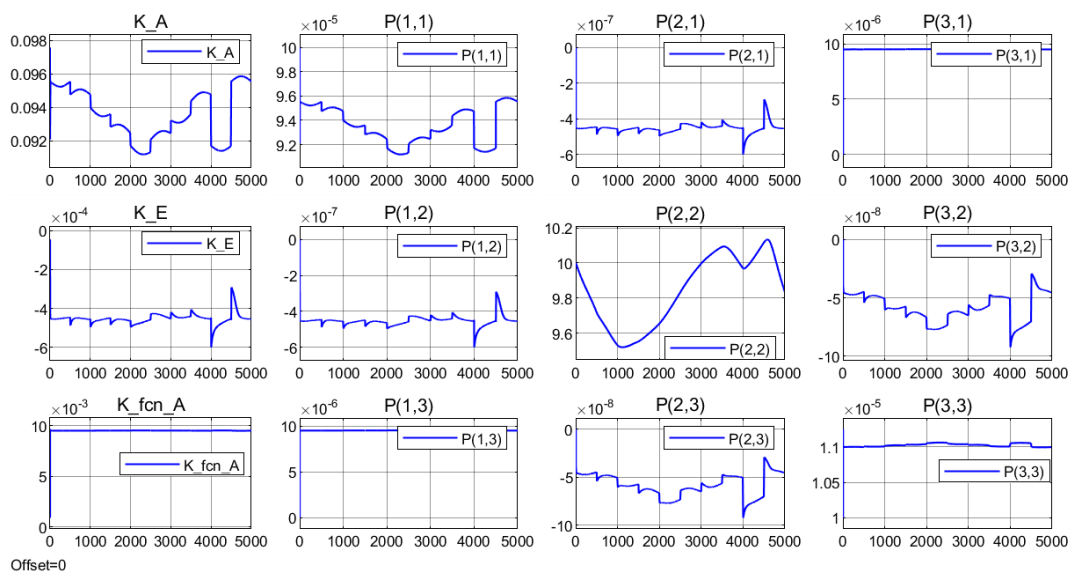
C.2 Estimering av funksjonen $\frac{E}{K_a^E + E}$ med augmentert Kalmanfilter

I dette vedlegget er simuleringer av estimeringen av funksjonen F_E lagt ved. Disse simuleringene hører til delkapittel 6.3

C.2.1 Simulink scope av A , E og $\frac{E}{K_a^E + E}$ med måling av A

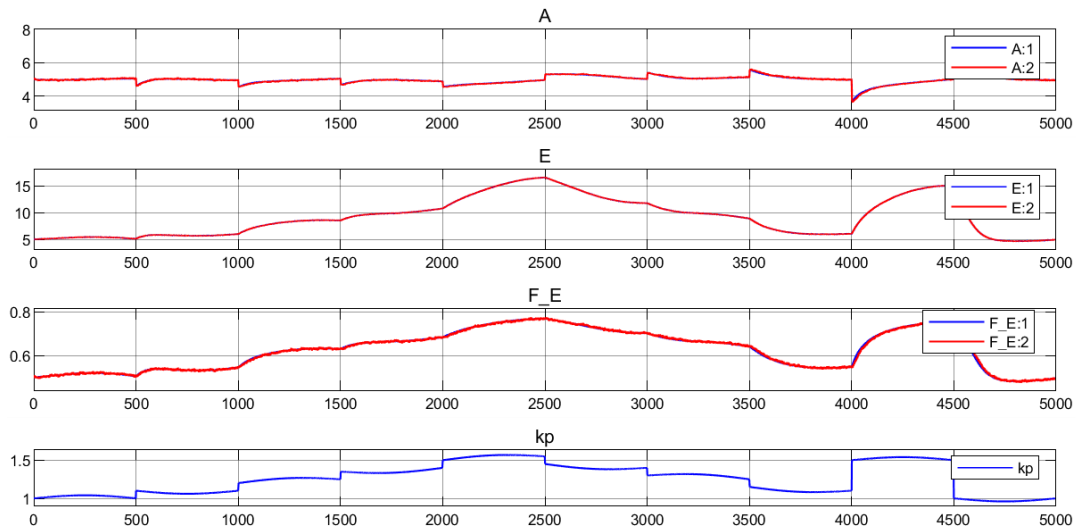


FIGUR C.5: Simulink scope av A , E , F_E og k_p med måling av A

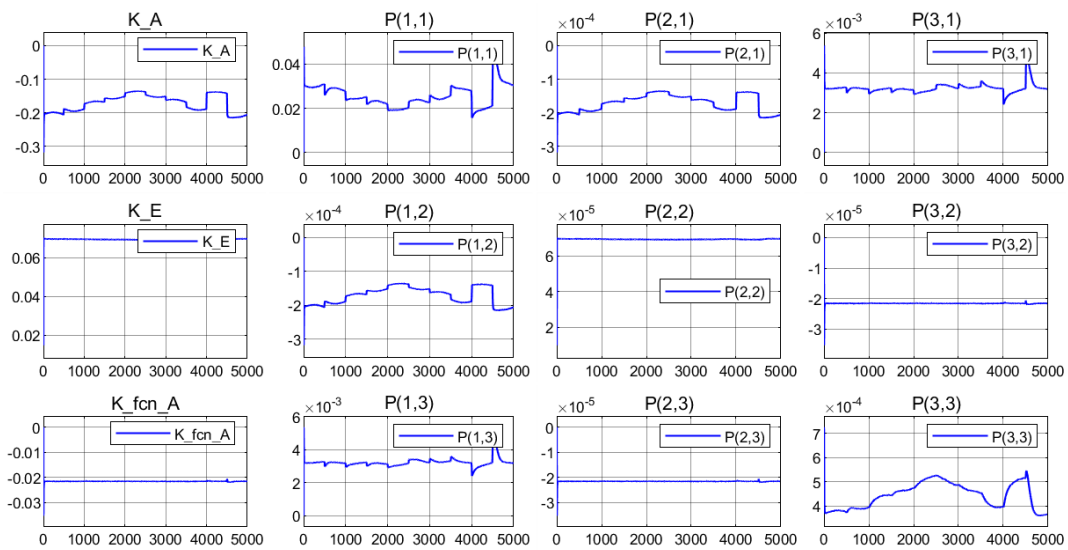


FIGUR C.6: Simulink scope av K og P for estimering av F_A med måling av A

C.2.2 Simulink scope av A , E og $\frac{E}{K_a^E + E}$ med måling av E



FIGUR C.7: Simulink scope av A , E , F_E og k_p med måling av E



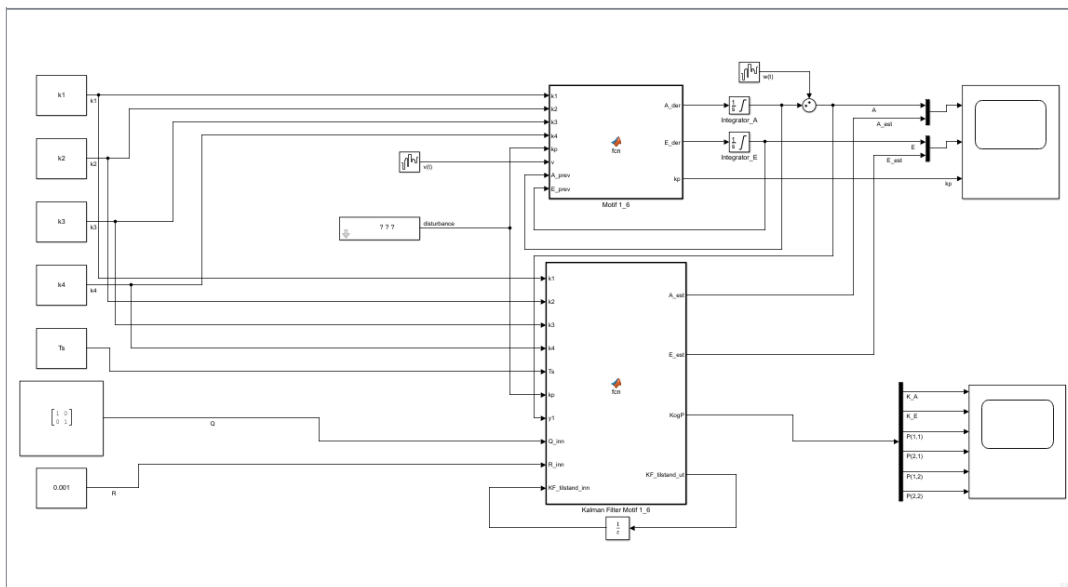
FIGUR C.8: Simulink scope av K og P for estimering av F_A med måling av E

Tillegg D

Simulink og matlabkode eksempler

D.1 Simulink

Her vises et eksempel bilde av Simulink implementasjonen av Kalmanfilteret. Denne implementasjonen er grunnlaget for implementasjonene brukt i de ulike forsøkene, og vil inneholde små forandringer iforhold til de ulike forsøkene som utføres.



FIGUR D.1: Bilde av implementasjon av modell og Kalmanfilter i Simulink

D.2 Matlab kode

I denne seksjonen vises eksempel koder av funksjoner og beregninger brukt gjennom utførelsen av denne oppgaven. Disse kodene vil inneholde små forandringer iforhold til de ulike forsøkene som utføres, men er bygget på den

samme strukturen.

D.2.1 Initialisering av verdier

I denne Matlabfilen initialiseres parametrene brukt i modellen.

```
1 clear all;
2
3 k1 = 1; %k_s^A
4 k2 = 5; %k_s^E
5 k3 = 1; %V_max
6 k4 = 0.003; %K_M^E
7
8 Ts = 0.1;
```

D.2.2 Initialisering av verdier utvidet modell

I denne Matlabfilen initialiseres parametrene brukt i den utvidede modellen.

```
1 clear all;
2
3 k1 = 10; %k_s^A
4 k2 = 0.5; %k_s^E
5 k3 = 1; %V_max
6 k4 = 0.003; %K_m^E
7 k5 = 5; %k_a^E
8 k6 = 5; %k_a^A
9
10 Ts = 0.01;
```

D.2.3 Matlab modell

Denne Matlabfilen viser hvordan modellen for den biologiske kontrolleren er lagt inn som en funksjonsblokk i Simulink.

```
1 function [A_der, E_der, kp] = fcn(k1, k2, k3, k4, kp, v, A_prev,
   E_prev)
2
3 A_der = k1 * E_prev - kp * A_prev + v;
4 E_der = k2 - (((k3 * E_prev) / (k4 + E_prev)) * A_prev) + v;
5
6 end
```

D.2.4 Linearisering

Denne Matlabfilen viser hvordan lineariseringen er gjort i Matlab. Denne filen blir utvidet når flere tilstander blir lagt til.

```

1 syms A_der E_der x1 k2 k3 k4 kp v v2 E A T
2
3 A_der = (x1*E) - (kp*A) + v;
4 E_der = k2 - ((k3*E*A)/(k4+E));
5 x1_der = 0;
6
7 a11 = diff(A_der,A);
8 a12 = diff(A_der,E);
9 a13 = diff(A_der,x1);
10 a21 = diff(E_der,A);
11 a22 = diff(E_der,E);
12 a23 = diff(E_der,x1);
13 a31 = diff(x1_der,A);
14 a32 = diff(x1_der,E);
15 a33 = diff(x1_der,x1);
16
17 A_mat = [a11,a12,a13;a21,a22,a23;a31,a32,a33]
18
19 b1 = diff(A_der,kp);
20 b2 = diff(E_der,kp);
21 b3 = diff(x1_der,kp);
22
23 B_mat = [b1;b2;b3]
24
25 Phi = eye(3) + (A_mat*T)
26
27 Gamma = B_mat*T

```

D.2.5 Observerbarhet

Denne Matlabfilen viser hvordan observerbarhetestesten er utført i Matlab. Denne filen blir utvidet når flere tilstander blir lagt til.

```

1 clear all;
2
3 k1 = 1; %k_s^A
4 k2 = 5; %k_s^E
5 k3 = 1; %V_max
6 k4 = 0.003; %K_m^E
7
8 T = 0.01;
9
10 kp = 1;
11 A = 5;
12 E = 5;
13
14 A_mat = [-kp, k1, E; -(E*k3)/(E + k4), ((A*E*k3)/(E+k4)
    ^2) - (A*k3)/(E + k4), 0; 0, 0, 0];

```



```
15
16
17 D = [1,0,0];
18
19 Phi = eye(3) + (A_mat*T);
20
21 Ob = obsv(Phi,D)
22
23 rang = rank(Ob)
```

D.2.6 Kalmanfilter

Denne Matlabfilen viser implementasjonen av det lineære Kalmanfilteret i en funksjonsblokk i Simulink.

```
1 function [A_est, E_est, KogP, KF_tilstand_ut] = fcn(k1, k2,
2           k3, k4, Ts, kp, y1, Q_inn, R_inn, KF_tilstand_inn)
3
4           %Initielle verdier.
5           A_start = 5;
6           E_start = 5;
7           kp_start = 1;
8
9           T = Ts;
10
11          if KF_tilstand_inn(1) <= 1 %Initial verdier settes
12              her
13                  x1 = A_start;
14                  x2 = E_start;
15                  u = kp_start;
16                  P = [0.001,0;0,100];
17          else
18                  x1 = KF_tilstand_inn(2); % Her g r algoritmen
19                      rekkursivt, tar inn verdier fra forrige
20                      iterasjon
21                  x2 = KF_tilstand_inn(3);
22                  u = kp;
23                  P = [KF_tilstand_inn(4), KF_tilstand_inn(5); ...
24                      KF_tilstand_inn(6), KF_tilstand_inn(7)];
25          end
26
27          Q = Q_inn;
28          R = R_inn;
29
30          all = -kp_start; %A_mat funnet fra lineariseringen,
31              med konstante verdier
```

```
29     a12 = k1;
30     a21 = -((k3*E_start)/(k4+E_start));
31     a22 = ((k3*A_start*E_start)/(k4 + E_start)^2) - ((k3
        *A_start)/(k4+E_start));
32
33     A_mat = [a11 , a12 ; a21 , a22 ];
34
35     b1 = -A_start; % B_mat funnet fra lineariseringen
36     b2 = 0;
37
38     B_mat = [b1 ; b2 ];
39
40     d1 = 1;
41     d2 = 0;
42
43     D = [d1 , d2 ];
44
45     E = 0;
46
47     phi = eye(2) + (A_mat*T); %utregning av Phi og Gamma
48
49     gamma = B_mat*T;
50
51     omega = [1 , 0 ; 0 , 1];
52
53     x = [x1 ; x2 ];
54     y = y1;
55
56
57     x_strek = (phi*x) + (gamma*u); %Kalmanfilter
        ligningene funnet i delkapittel (3.3)
58     P_strek = (phi*P*phi') + (omega*Q*omega');
59
60     K = (P_strek*D') / ((D*P_strek*D') + R);
61
62     x_hatt = x_strek + (K*(y-(D*x_strek)));
63
64     P_hatt = (eye(2)-(K*D))*P_strek;
65
66
67     KF_tilstand_ut = zeros(7,1); %sender verdiene som
        skal brukes i neste iterasjon ut.
68
69     KF_tilstand_ut(1) = KF_tilstand_inn(1)+1;
70     KF_tilstand_ut(2) = x_hatt(1);
71     KF_tilstand_ut(3) = x_hatt(2);
72     KF_tilstand_ut(4:7) = P_hatt(:)';
```

```
73
74     A_est = x_hatt(1); %verdier som skal til scope.
75     E_est = x_hatt(2);
76
77     KogP = zeros(6,1);
78
79     KogP(1) = K(1);
80     KogP(2) = K(2);
81     KogP(3:6) = P_hatt(:);
82
83
84
85 end
```

D.2.7 Utvidet Kalmanfilter

Denne Matlabfilen viser implementasjonen av det utvidede Kalmanfilteret i en funksjonsblokk i Simulink.

```
1 function [A_est, E_est, KogP, KF_tilstand_ut] = fcn(k1, k2,
2           k3, k4, Ts, kp, y1, Q_inn, R_inn, KF_tilstand_inn)
3
4     %Initielle verdier.
5     A_start = 5;
6     E_start = 5;
7     kp_start = 1;
8
9     T = Ts;
10
11     if KF_tilstand_inn(1) <= 1 %Initial verdier settes
12         her
13             x1 = A_start;
14             x2 = E_start;
15             u = kp_start;
16             P = [0.001, 0; 0, 10];
17     else
18         x1 = KF_tilstand_inn(2); % Her g r algoritmen
19             rekkursivt, tar inn verdier fra forrige
20             iterasjon
21         x2 = KF_tilstand_inn(3);
22         u = kp;
23         P = [KF_tilstand_inn(4), KF_tilstand_inn(5); ...
24             KF_tilstand_inn(6), KF_tilstand_inn(7)];
25     end
26
27     Q = Q_inn;
```

```
26 R = R_inn;
27
28 a11 = -u; %A_mat funnet fra lineariseringen , med
    variable verdier
29 a12 = k1;
30 a21 = -((k3*x2)/(k4+x1));
31 a22 = ((k3*x1*x2)/(k4 + x2)^2) - ((k3*x1)/(k4+x1));
32
33 A_mat = [a11 , a12 ; a21 , a22 ];
34
35
36 d1 = 1;
37 d2 = 0;
38
39 D = [d1 , d2 ];
40
41 E = 0;
42
43 phi = eye(2) + (A_mat*T); %utregning av Phi
44
45 omega = [1 , 0 ; 0 , 1];
46
47 y = y1;
48
49
50 x1_strek = x1 + (T*((k1*x2)-(u*x1))); %utvidet
    Kalmanfilter ligningene funnet i delkapittel
    (3.4)
51 x2_strek = x2 + (T*(k2 - (((k3*x2)/(k4+x2))*x1)));
52
53
54 P_strek = (phi*P*phi') + (omega*Q*omega');
55
56 K = (P_strek*D') / ((D*P_strek*D') + R);
57
58 x1_hatt = x1_strek + (K(1)*(y-(x1_strek)));
59 x2_hatt = x2_strek + (K(2)*(y-(x1_strek)));
60
61 P_hatt = (eye(2)-(K*D))*P_strek;
62
63
64 KF_tilstand_ut = zeros(7,1); %sender verdiene som
    skal brukes i neste iterasjon ut.
65
66 KF_tilstand_ut(1) = KF_tilstand_inn(1)+1;
67 KF_tilstand_ut(2) = x1_hatt;
68 KF_tilstand_ut(3) = x2_hatt;
```

```

69     KF_tilstand_ut(4:7) = P_hatt(:)';
70
71     A_est = x1_hatt; %verdier som skal til scope.
72     E_est = x2_hatt;
73
74     KogP = zeros(6,1);
75
76     KogP(1) = K(1);
77     KogP(2) = K(2);
78     KogP(3:6) = P_hatt(:);
79
80
81
82 end

```

D.2.8 Augmentert Kalmanfilter

Denne Matlabfilen viser implementasjonen av det augmenterte Kalmanfilteret med 1 måling og 1 ekstra tilstand i en funksjonsblokk i Simulink

```

1 function [A_est, E_est, k1_est, KogP, KF_tilstand_ut] = fcn(
    k2, k3, k4, Ts, kp, y1, Q_inn, R_inn, KF_tilstand_inn)
2
3
4     %Initielle verdier.
5     A_start = 5;
6     E_start = 5;
7     x3_start = 1.1;
8     kp_start = 1;
9
10    T = Ts;
11
12    if KF_tilstand_inn(1) <= 1 %Initial verdier settes
        her
13        x1 = A_start;
14        x2 = E_start;
15        x3 = x3_start;
16        u = kp_start;
17        P = [0.0001, 0, 0; 0, 0.00001, 0; 0, 0, 0.000027]; %mer
            passende initial verdi.
18    else
19        x1 = KF_tilstand_inn(2); % Her g r algoritmen
            rekkursivt, tar inn verdier fra forrige
            iterasjon
20        x2 = KF_tilstand_inn(3);
21        x3 = KF_tilstand_inn(4);
22        u = kp;

```

```

23     P = [KF_tilstand_inn(5),KF_tilstand_inn(6),
24           KF_tilstand_inn(7);...
25           KF_tilstand_inn(8),KF_tilstand_inn(9),
26           KF_tilstand_inn(10);...
27           KF_tilstand_inn(11),KF_tilstand_inn(12),
28           KF_tilstand_inn(13)];
29
30     end
31
32     Q = Q_inn;
33     R = R_inn;
34
35     a11 = -u; %A_mat funnet fra lineariseringen, med
36     variable verdier
37     a12 = x3;
38     a13 = x2;
39     a21 = -((k3*x2)/(k4+x2));
40     a22 = ((k3*x1*x2)/((k4 + x2)^2) - ((k3*x1)/(k4+x2)))
41     ;
42     a23 = 0;
43     a31 = 0;
44     a32 = 0;
45     a33 = 0;
46
47     A_mat = [a11 ,a12 ,a13 ;a21 ,a22 ,a23 ;a31 ,a32 ,a33 ];
48
49     d1 = 1;
50     d2 = 0;
51     d3 = 0;
52
53     D = [d1 ,d2 ,d3 ];
54
55     E = 0;
56
57     phi = eye(3) + (A_mat*T); %utregning av Phi
58
59     omega = [1 ,0 ,0;0 ,1 ,0;0 ,0 ,1];
60
61     y = y1;
62
63     x1_strek = x1 + (T*((x3*x2)-(u*x1))); %utvidet
64     Kalmanfilter ligningene funnet i delkapittel
65     (3.4), augmentert med 1 tilstand
66     x2_strek = x2 + (T*(k2 - (((k3*x2)/(k4+x2))*x1)));
67     x3_strek = x3;

```

```

63     P_strek = (phi*P*phi') + (omega*Q*omega');
64
65     K = (P_strek*D') / ((D*P_strek*D') + R);
66
67     x1_hatt = x1_strek + (K(1)*(y-x1_strek));
68     x2_hatt = x2_strek + (K(2)*(y-x1_strek));
69     x3_hatt = x3_strek + (K(3)*(y-x1_strek));
70
71     P_hatt = (eye(3)-(K*D))*P_strek;
72
73
74     KF_tilstand_ut = zeros(13,1); %sender verdiene som
       skal brukes i neste iterasjon ut.
75
76     KF_tilstand_ut(1) = KF_tilstand_inn(1)+1;
77     KF_tilstand_ut(2) = x1_hatt;
78     KF_tilstand_ut(3) = x2_hatt;
79     KF_tilstand_ut(4) = x3_hatt;
80     KF_tilstand_ut(5:13) = P_hatt(:)';
81
82     A_est = x1_hatt; %verdier som skal til scope.
83     E_est = x2_hatt;
84     k1_est = x3_hatt;
85
86     KogP = zeros(12,1);
87
88     KogP(1) = K(1);
89     KogP(2) = K(2);
90     KogP(3) = K(3);
91     KogP(4:12) = P(:);
92
93
94
95 end

```

D.2.9 Augmentert Kalmanfilter med 2 målinger

Denne Matlabfilen viser implementasjonen av det augmenterte Kalmanfilteret med 2 måling og 2 ekstra tilstander i en funksjonsblokk i Simulink

```

1 function [A_est, E_est, k1_est, k2_est, KogP, KF_tilstand_ut]
   = fcn(k3, k4, Ts, kp, y1, y2, Q_inn, R_inn, KF_tilstand_inn)
2
3
4     %Initielle verdier.
5     A_start = 5;
6     E_start = 5;
7     x3_start = 1.1;

```

```

8     x4_start = 5.2;
9     kp_start = 1;
10
11    T = Ts;
12
13    if KF_tilstand_inn(1) <= 1 %Initial verdier settes
14        her
15            x1 = A_start;
16            x2 = E_start;
17            x3 = x3_start;
18            x4 = x4_start;
19            u = kp_start;
20            P =
21                [0.0001,0,0,0;0,0.0001,0,0;0,0,0.0001,0;0,0,0,0.00001];
22                %mer passende initial verdi.
23    else
24        x1 = KF_tilstand_inn(2); % Her g r algoritmen
25        rekkursivt, tar inn verdier fra forrige
26        iterasjon
27        x2 = KF_tilstand_inn(3);
28        x3 = KF_tilstand_inn(4);
29        x4 = KF_tilstand_inn(5);
30        u = kp;
31        P = [KF_tilstand_inn(6),KF_tilstand_inn(7),
32            KF_tilstand_inn(8),KF_tilstand_inn(9);...
33            KF_tilstand_inn(10),KF_tilstand_inn(11),
34            KF_tilstand_inn(12),KF_tilstand_inn(13);...
35            KF_tilstand_inn(14),KF_tilstand_inn(15),
36            KF_tilstand_inn(16),KF_tilstand_inn(17);...
37            KF_tilstand_inn(18),KF_tilstand_inn(19),
38            KF_tilstand_inn(20),KF_tilstand_inn(21)];
39    end
40
41    Q = Q_inn;
42    R = R_inn;
43
44    a11 = -u; %A_mat funnet fra lineariseringen, med
45    variable verdier
46    a12 = x3;
47    a13 = x2;
48    a14 = 0;
49    a21 = -((k3*x2)/(k4+x2));
50    a22 = ((k3*x1*x2)/(k4 + x2)^2) - ((k3*x1)/(k4+x2));
51    a23 = 0;
52    a24 = 1;
53    a31 = 0;

```



```
45     a32 = 0;
46     a33 = 0;
47     a34 = 0;
48     a41 = 0;
49     a42 = 0;
50     a43 = 0;
51     a44 = 0;
52
53     A_mat = [ a11 , a12 , a13 , a14 ; a21 , a22 , a23 , a24 ; a31 , a32 , a33
54             , a34 ; a41 , a42 , a43 , a44 ];
55
56     d11 = 1;
57     d12 = 0;
58     d13 = 0;
59     d14 = 0;
60     d21 = 0;
61     d22 = 1;
62     d23 = 0;
63     d24 = 0;
64
65     D = [ d11 , d12 , d13 , d14 ; ...
66         d21 , d22 , d23 , d24 ];
67
68     E = 0;
69
70     phi = eye(4) + (A_mat*T); %utregning av Phi
71
72
73     omega = [ 1 , 0 , 0 , 0 ; 0 , 1 , 0 , 0 ; 0 , 0 , 1 , 0 ; 0 , 0 , 0 , 1 ];
74
75
76     %kalman filter
77
78     x1_strek = x1 + (T*((x3*x2)-(u*x1))); %utvidet
79     Kalmanfilter ligningene funnet i delkapittel
80     (3.4) , augmentert med 2 tilstander
81     x2_strek = x2 + (T*(x4 - (((k3*x2)/(k4+x2))*x1)));
82     x3_strek = x3;
83     x4_strek = x4;
84
85     P_strek = (phi*P*phi') + (omega*Q*omega');
86
87     K = (P_strek*D') / ((D*P_strek*D') + R);
88
89     x_strek = [ x1_strek ; x2_strek ; x3_strek ; x4_strek ];
```

```
89     y = [y1;y2];
90
91     x_hatt = x_strek + (K*(y-D*x_strek));
92
93
94     P_hatt = (eye(4)-(K*D))*P_strek;
95
96
97     KF_tilstand_ut = zeros(21,1); %sender verdiene som
    skal brukes i neste iterasjon ut.
98
99     KF_tilstand_ut(1) = KF_tilstand_inn(1)+1;
100    KF_tilstand_ut(2) = x_hatt(1);
101    KF_tilstand_ut(3) = x_hatt(2);
102    KF_tilstand_ut(4) = x_hatt(3);
103    KF_tilstand_ut(5) = x_hatt(4);
104    KF_tilstand_ut(6:21) = P_hatt(:)';
105
106    A_est = x_hatt(1); %verdier som skal til scope.
107    E_est = x_hatt(2);
108    k1_est = x_hatt(3);
109    k2_est = x_hatt(4);
110
111    KogP = zeros(12,1);
112
113    KogP(1) = K(1,1);
114    KogP(2) = K(1,2);
115    KogP(3) = K(2,1);
116    KogP(4) = K(2,2);
117    KogP(5) = K(3,1);
118    KogP(6) = K(3,2);
119    KogP(7) = K(4,1);
120    KogP(8) = K(4,2);
121    KogP(9) = P_hatt(1,1);
122    KogP(10) = P_hatt(2,2);
123    KogP(11) = P_hatt(3,3);
124    KogP(12) = P_hatt(4,4);
125
126
127
128
129 end
```