

Stabilisering av slug-strømning ved bruk av topside-målinger og ulineær MPC

Sten Even Smoge Hatlen

Master i teknisk kybernetikk
Oppgaven levert: Juni 2008
Hovedveileder: Morten Hovd, ITK

Oppgavetekst

Oppgaven går ut på å se på bruk av ulineær MPC, evt. kombinert med ulineær tilstandsestimering, for å stabilisere slug-strømning over et større driftsområde enn det som er oppnåelig ved bruk av lineære regulatorer

Oppgaven gitt: 07. januar 2008
Hovedveileder: Morten Hovd, ITK

Sammendrag

Riser-indusert slugging er et strømningsmønster som kan oppstå i flerfase strømming i stigerøret opp til oljeplattformer og karakteriseres av vesentlige strømnings- og trykkvariasjoner. Den støtvide strømmingen forårsaket av riser-indusert slugging kan medføre store produksjonsproblemer for prosessanlegget nedstrøms, og det trengs derfor en metode for å håndtere problemet. En velkjent metode for å unngå slugging er å redusere åpningen til strupeventil ved toppen av stigerøret. I denne rapporten er det foreslått å bruke ulineær MPC kombinert med ulineær tilstandsestimering for å styre pådrag til strupeventil optimalt.

Denne rapporten beskriver en algoritme for ulineær modell prediktiv regulering som bruker en "single shooting" optimaliseringsmetode med prestabilisering. Metoden er en iterativ metode som løser en serie med QP-problemer, i likhet med sekvensiell kvadratisk programmering, for å finne optimalt pådrag. Det er brukt tilstandsestimering, både EKF og UKF, for å estimere systemets tilstander for prediksjon. Algoritmene for NMPC og tilstandsestimering er realisert i Matlab for testing og simulering.

Det er brukt en forenklet dynamiske modell av riser-indusert slugging. Slug-strømning har blitt stabilisert for en bestemt ventilåpning ved å bruke NMPC og tilstandsestimering på den forenklede modellen.

Forord

Denne masteroppgaven markerer en avslutning på mitt masterstudium i Teknisk Kybernetikk ved NTNU. Jeg vil takke min veileder Morten Hovd ved institutt for Teknisk Kybernetikk, NTNU, for veiledning og mange gode innspill på oppgaven. Jeg vil også takke Åsmund Våge Fannemel for nyttige innspill og diskusjoner.

Sten Even Hatlen
Trondheim, 19. juni 2008

Innhold

1	Innledning	1
2	Modell Prediktiv Regulering	3
2.1	En kort presentasjon av historien til MPC	3
2.2	Egenskaper til MPC	5
2.2.1	Modellen	5
2.2.2	Forstyrrelser	5
2.2.3	Begrensninger	6
2.2.4	Stabilitet	6
2.3	MPC formulering	7
2.3.1	Objektiv funksjon	9
2.3.2	Begrensninger	12
3	Ulineær Modell Prediktiv Regulering	15
3.1	NMPC	15
3.1.1	NMPC metoder	15
3.2	SQP-algoritme	16
3.2.1	Initialverdier	17
3.2.2	Nominell tilstandstrajektorie	17
3.2.3	Integrasjonsmetode	18
3.2.4	Linearisering og diskretisering	18
3.2.5	Diskretiseringsmetoder	19
3.2.6	QP-Problem	20
3.2.7	Begrensninger	24
3.2.8	Linjesøk	25
3.3	Konvergenskriterie	26
3.4	CSTR - ekspempel	29
4	Tilstandsestimering	33
4.1	Kalmanfilter	33
4.2	Ulineær estimering	33
4.2.1	Extended Kalmanfilter	33
4.3	Unscented Kalmanfilter	34

5	Riser-indusert slugging	39
5.1	Slug-strømming	39
5.2	Storkaas modell for modellering av slugging	40
6	Stabilisering av slug-strømming	43
6.1	Oppsett	43
6.2	Tilstandsestimering	43
6.3	NMPC på Storkaas modell	45
6.3.1	Simulering	46
7	Konklusjon	51
8	Videre arbeid	53
8.1	Ulineær MPC	53
8.2	Tilstandsestimering	54
8.3	Modell	54
	Bibliografi	55
A	Modell av slugging i stigerør	57
A.1	Antagelser i modellen	57
A.2	Modell ligninger	58
A.3	Notasjon	60

Kapittel 1

Innledning

Denne rapporten vil ta for seg regulering av slug-strømning i rørledninger. Slug-strømning er et ustabilisert strømmønster hvor olje/vann og gassen strømmer støtvis og separat i oljeledninger. Denne uregelmessige strømmingen forårsaker utfordringer og problemer for oljeselskapene.

I denne rapporten vil det fokuseres på riser-slugging, dette er støtvis strømning i stigerøret fra havbunnen og opp til separator på oljeplattformen. Slugger i stigerøret kan bli ganske store og kan i verste tilfeller fylle hele stigerøret. Inngangsseparatoren er ikke stor nok til å motta så store slugger. Selv små slugger vil medføre dårlig olje/vann separasjon og slitasje på utstyr.

En vanlig metode for å unngå slugging er å redusere åpningen til strupeventil på oljeledningen før inngangsseparator. Blir ventilåpningen liten nok vil fenomenet slugging fjernes helt [25]. Imidlertid vil en redusert ventilåpning medføre lavere oljeproduksjon. Det er derfor hensiktsmessig å bruke reguleringsløsninger for å kunne stabilisere den ustabile oljestrømmen også for større ventilåpninger.

Det er forskjellige alternative målinger som kan brukes til regulering av ventilen. Disse målingene kan deles inn i topside-målinger og målinger tilgjengelig oppstrøms stigerøret. Det vil her fokuseres på topside målinger da disse er lettere tilgjengelig. Bakdelen med å bruke topside målinger er at systemet blir vanskeligere å stabilisere.

For å stabilisere slug-strømning er det her sett på bruk av modell prediktiv regulering (MPC) for å optimalisere pådragsbruk til ventilen. På grunn av den ulineære dynamikken til slug-strømning vil det brukes ulineær modell prediktiv regulering (NMPC). En NMPC algoritme er utviklet og implementert i Matlab for simulering av systemet.

Regulering ved bruk av NMPC forutsetter kunnskap om systemets tilstander for prediksjon. Det er derfor brukt tilstandsestimering for å estimere tilstander fra målinger tilgjengelig topside. Her er det fokusert på to ulineære estimeringsmetoder: Extended kalman-

filter (EKF) og Unscented kalmanfilter (UKF).

For modellering av systemet er det brukt en forenklet modell av riser-indusert slugging. Modellen er utviklet av E.Storkås [25] og er brukt til prosess-simulering, prediksjon i NMPC og modell for tilstandsestimering.

Her følger en kort beskrivelse av de forskjellige kapitlene:

- *Kapittel 2* gir en innføring i MPC. Kapitlet starter med en kort presentasjon av historien til MPC. Det utledes en lineær MPC formulering basert på prestabilisering. Denne prestabiliseringen danner grunnlag for formulering av NMPC i kapittel 3.
- *Kapittel 3* tar for seg formulering av NMPC-algoritmen som vil brukes i denne rapporten. Kapitlet starter med en innføring i forskjellige SQP metoder før stegene i selve algoritmen beskrives. Det avsluttes med en anvendelse av NMPC-algoritmen på en CSTR-modell.
- *Kapittel 4* gir en kort introduksjon av ulineær tilstandsestimering EKF og UKF. Kapitlet avsluttes med en algoritme for skalert UKF.
- *Kapittel 5* gir en kort innføring i de aktuelle typene av slug-strømning. Videre presenteres modellen av riser-indusert slugging som vil brukes i denne rapporten.
- *Kapittel 6* presenterer resultat fått ved simulering av NMPC med tilstandsestimering på den dynamiske prosessen. Det er vist estimeringsfeil med både EKF og UKF for det nominelle tilfellet. Det er også vist simuleringer av stabilisering av slug-strømning ved bruk av NMPC og tilstandsestimator.
- *Kapittel 7* inneholder konklusjon og summerer resultat i denne rapporten.
- *Kapittel 8* foreslår retninger for videre arbeid.
- *Tillegg A* inneholder detaljert beskrivelse av modellen brukt til prosess simulering, prediksjon i NMPC og modell for tilstandsestimering.

Kapittel 2

Modell Prediktiv Regulering

Modell prediktiv regulering (MPC) er en type algoritmer som bruker en eksplisitt prosessmodell for å forutsi fremtidig respons til et system. For hvert reguleringsintervall optimaliseres den fremtidige responsen til systemet ved å beregne fremtidig optimalt pådrag. Det første elementet i den optimale pådragsvektoren brukes deretter som pådrag for systemet. Syklusen gjentas deretter for neste reguleringsintervall.

Den eksplisitte prosessmodellen gjør det mulig å bruke hele dynamikkområdet til prosessen under utregning av pådraget til prosessen. Siden MPC algoritmen betrakter prosessmodellen for fremtidig respons gir dette muligheter til å forutsi fremtidig påvirkning av støy. Dette gjør det mulig å motvirke effekten forstyrrelser har på systemet ved bruk av foroverkobling og tilbakekobling. En av de store fordelene med MPC sammenlignet med andre reguleringsmetoder er inkluderingen av begrensninger.

2.1 En kort presentasjon av historien til MPC

Denne delen tar for seg kort om historien og utviklingen til MPC

LQG

Dagens reguleringsmetoder kan spores tilbake til arbeidet til Kalman [11] på 1960 tallet som prøvde å finne optimal løsning på lineære reguleringsystem. Kalman studerte "Linear Quadratic Regulator" (LQR) som baserer seg på optimalisering av en kvadratisk objektiv funksjon. Kombinert LQR regulator og kalmanfilter blir kalt "Lineær Quadratic Gaussian" (LQG). Begrensninger på pådrag, tilstander og utganger var ikke tatt hensyn til under utvikling av LQG [20].

IDCOM

Den første beskrivelesen av en MPC-applikasjon som har blitt presentert er gitt i [22]. Deres IDCOM (Identifikasjon og kontroll) ble beskrevet av forfatterene som "modell prediktiv heuristic control"(MPHC). IDCOM baserte seg på en impuls respons modell

av systemet som er gyldig kun for prosesser som er stabile i åpen sløyfe. Den iterative reguleringsalgoritmen gjorde det mulig med begrensninger på innganger og utganger. I [22] blir innganger til prosessen delt inn i manipulerede variabler (MV) og forstyrrelses variabler (DV), som ikke kan reguleres. Utganger fra prosessen blir referert til som regulerede variabler (CV).

Inkludering av begrensninger i reguleringen medførte at MPC ble mer attraktiv for prosessindustrien. De argumenterte også med et økonomisk fortrinn ved å bruke MPC; at bedre dynamisk regulering tillater utgangen å operere nærmere begrensningene.

DMC

Dynamisk matrise regulering (DMC) [2] ligner på MPHC, men bruker en step-respons modell istedenfor en impuls-respons modell. Ved å bruke step-respons modell kan predikerte fremtidige forandringer på utgangen representeres som en lineær kombinasjon av fremtidige forandringer på pådraget. Matrisen som forbinder de to kalles en Dynamisk Matrise [20]. Optimale innganger ble dermed utregnet som en løsning på et minste kvadraters problem. DMC var en ren derterministisk formulering og inkluderte ikke noen støymodell.

Nyere MPC formuleringer baserer seg på prinsippene til DMC. DMC hadde enorm innvirkning på industrien og har blitt brukt i utstrakt grad av oljeindustrien [16]. Sammen representerer IDCOM og DMC 1. generasjons MPC teknologi.

QDMC

I Kvadratisk DMC (QDMC) blir begrensninger behandlet på en mer konsistent måte. I [5] vises det hvordan objektiv funksjonen i DMC kan omskrives som et standard QP problem. Dette er en fordel fordi det tillater å bruke standard optimaliserings algoritmer. Det vises også hvordan begrensningene kan implementeres eksplisitt. Dette gir en systematisk måte å implementere begrensninger på pådrag og utganger. QDMC kan betraktes som 2. generasjons MPC teknologi [20].

Senere utvikling

Begrensninger kan deles opp i harde og myke. Et problem med QDMC er at hvis de harde begrensningene overskrides vil optimerings problemet bli infeasible. I senere MPC versjoner har det også blitt implementert myke begrensninger som tillater begrensningene å overskrides. I senere tid har det også blitt tatt hensyn til modell usikkerhet (robust design) og forbedret identifikasjons teknologi.

2.2 Egenskaper til MPC

2.2.1 Modellen

I MPC optimiseres manipulerede variabler ut fra kunnskap om prosessens fremtidige respons. Den fremtidige prediksjonen utføres ved å bruke en modell av prosessen og derfor er modellen et viktig element i en MPC regulator. God regulering av prosessen er avhengig av at modellen klarer å prediktere den fremtidige responsen tilstrekkelig nøyaktig.

Lineære modeller

Flere MPC algoritmer bruker step- og impulsresponsmodeller som utledes fra konvolusjonsintegral. Et problem med disse modellene er at de er begrenset til asymptotisk stabile prosesser [6]. Modifisering av algoritmene kan gjøre det mulig å behandle integrerende prosesser, men virkelig ustabile prosesser er generelt ikke mulig. En fordel med disse modellene er at de er lett å utlede fra test data fra prosessen.

Ulineære modeller

Av og til er en ulineær modell tilgjengelig hvor ligningene fåes fra kunnskap om underliggende masse- og energibalanse til systemet. Dette kalles en *first-principles modell*. For komplekse industrielle systemer er slike modeller ofte kostbar å utvikle. Den ulineære modellen kan lineariseres om et lineariseringspunkt. En utfordring her ligger i at dersom modellen er veldig ulineær vil den lineariserte modellen bare være gyldig nært lineariseringspunktet.

2.2.2 Forstyrrelser

Forstyrrelser kan motvirkes på to måter: ved bruk av foroverkobling og tilbakekobling. Fordelen med foroverkobling er at den ikke må vente til påvirkningen av forstyrrelsen har tredd i kraft, men kan gjøre justeringer i forkant av forstyrrelsen. Forstyrrelser som kan forutsies i forkant inkluderes i modellen.

Da det ikke er mulig å lage en perfekt modell av forstyrrelsene må foroverkobling kombineres med tilbakekobling. Tilbakekobling kan da ta hensyn til umålte forstyrrelser. Forstyrrelser som virker inn på tilbakekoblingen kan motvirkes ved å gå ut fra at forstyrrelsen som har påvirket utgangen vil forbli konstant for all fremtid [13]. Dette gjøres ved å regne ut bias som er avviket mellom faktisk utgang og prediktert utgang:

$$d_k = y_k^m - y_k \quad (2.1)$$

denne biasen kan legges til modellen for underliggende prediksjoner

$$y_{k+j} = g(x_{k+j}) + d_k \quad (2.2)$$

Derimot for integrerende og ustabile prosesser vil modellen med konstant forstyrrelse på utgangen ikke virke fordi estimatoren inneholder de ustabile polene til prosessen [17]. Dette kan løses ved bruk modeller av forstyrrelsen på inngang eller tilstand. Dersom målestøyen er stor vil det være fordelsmessig å bruke et Kalmanfilter, som vist i [17].

2.2.3 Begrensninger

Begrensninger kan generelt deles opp i myke og harde. Myke begrensninger kan overskrides, men amplituden til overskridelsen straffes generelt med en kvadratisk straff i objektiv funksjonen. Harde begrensninger er av typen som aldri bør overskrides. Bruk av harde begrensninger på utgangen bør brukes med forsiktighet i MPC da forstyrrelser lett kan medføre at regulatoren blir infeasible.

2.2.4 Stabilitet

To begreper som ofte brukes i forbindelse med stabilitet er nominell stabilitet (*NS*) og robust stabilitet (*RS*). I [24] er nominell stabilitet definert som stabilitet for det nominelle systemet; som er systemet upåvirket av modellfeil og støy. Videre er robust stabilitet definert som stabilitet for en gitt usikkerhets-mengde; som er det nominelle systemet påvirket av modellfeil og støy. Heretter vil stabilitet være ensbetydende med nominell stabilitet.

Uendelig prediksjons horisont

For å garantere stabilitet i MPC må man forsikre seg om at begrensningene overholdes over den uendelige horisonten og ikke bare for reguleringshorisonten. Siden man ofte må velge en kort reguleringshorisont for å begrense kompleksiteten til utregningene vil stabilitet i mange tilfeller være et praktisk problem.

Problem med stabilitet oppstår ofte når prediksjonshorisonten ikke er lang nok, og det viser seg at stabilitet ofte kan oppnås ved å gjøre prediksjonshorisonten lengre, eller til og med uendelig. En metode for å oppnå stabilitet er å evaluere responsen til fremtidige CV over et endelig antall tidsintervall, såkalt *endelig prediksjonshorisont* [20]. Lengden til denne horisonten (H_p) er en tunings parameter og settes generelt lang nok til at den fanger inn steady-state effekten til alle de fremtidige MV. Dette er en tilnærming til uendelig prediksjonshorisont-løsningen og for å sikre den stabilitet som oppnås ved uendelig prediksjons horisont er det vanlig å legge til en begrensning på den siste tilstanden, denne begrensningen er ment for å tvinge tilstandene til en gitt verdi ved slutten av prediksjonshorisonten.

En annen metode for å oppnå stabilitet er å bruke uendelig prediksjonshorisont horisont. Dette kan oppnås ved å formulere reguleringsproblemet over uendelig horisont ved å bruke endelig antall parametre. Her brukes det en egen vekt på siste tilstand i reguleringshorisonten. Det er denne metoden det vil fokuseres på i denne rapporten.

Prestabilisering

Numeriske problemer kan oppstå når systemets oppførsel skal predikteres, spesielt aktuelt er dette dersom systemet er ustabil. Utregning av fremtidig oppførsel til systemet involverer å finne systemmatrisen A^i . Dette kan medføre numeriske problemer, spesielt dersom systemet er ustabil. Når systemet er ustabil har A-matrisen egenverdier utenfor enhetssirkelen og enkelte element i A^i kan bli veldig stor i forhold til andre ettersom i øker.

Prestabilisering er en effektiv metode for å unngå problemer relatert til termineringsbegrensninger og dårlig kondisjonering av matriser [13]. Vi introduserer tilbakekobling K for å stabilisere systemet. I tillegg settes pådraget fra MPC-regulatoren til c_k . Dette gir pådraget

$$u_k = Kx_k + c_k \quad (2.3)$$

2.3 MPC formulering

I dette avsnittet vil det utledes en formulering for lineær MPC. Den lineære MPC formuleringen er konstruert med tanke på bruk av såkalt prestabilisering. Denne prestabiliseringen vil i neste avsnitt brukes til å utlede en ulineær MPC formulering.

Lineær MPC baserer seg på en linearisert prosessmodell og forutsetter at den lineariserte modellen er gyldig for hele operasjonsområdet. Den lineariserte modellen er gitt av

$$x_{k+1} = Ax_k + Bu_k + Ed_k \quad (2.4)$$

$$y_k = Cx_k \quad (2.5)$$

MPC formuleringen som vil brukes her baserer seg på prestabilisering. Tilbakekobling med prestabilisering:

$$u_k - u_k^d = -K(x_k - x_k^d) + c_k \quad k = 0, \dots, H_u - 1 \quad (2.6)$$

$$u_k - u_k^d = -K(x_k - x_k^d) \quad k \geq H_u \quad (2.7)$$

Objektiv funksjonen er gitt som

$$V(k) = \sum_{i=1}^{H_p} \left\{ \left\| \hat{y}_{k+i|k} - y_{k+i}^d \right\|_Q^2 + \left\| \hat{u}_{k+i-1|k} - u_{k+i-1}^d \right\|_R^2 \right\} \quad (2.8)$$

Her brukes betegnelsen H_u for pådragshorisont, H_p er prediksjonshorisont, Q er vektning på utgang og R er vektning på pådrag. Ved å dele opp objektiv funksjonen i regulerings- og prediksjonshorisont får vi

$$V(k) = \sum_{i=1}^{H_u-1} \left\| \hat{y}_{k+i|k} - y_{k+i}^d \right\|_Q^2 + \sum_{i=0}^{H_u-1} \left\| \hat{u}_{k+i|k} - u_{k+i}^d \right\|_R^2 + \sum_{i=H_u}^{H_p} \left\| \hat{y}_{k+i|k} - y_{k+i}^d \right\|_Q^2 \quad (2.9)$$

Begrensninger er gitt ved

$$x_{k+1} = Ax_k + Bu_k \quad (2.10)$$

$$y_k = Cx_k + Du_k \quad (2.11)$$

$$u_{min} \leq u_k \leq u_{maks} \quad (2.12)$$

$$y_{min} \leq y_k \leq y_{maks} \quad (2.13)$$

hvorav likhetsbegrensningene vil være implisitt gitt av prosessmodellen. Dersom pådraget i (2.6) settes inn i tilstandsligningen i (2.4) vi uttrykket

$$\begin{aligned} x_{k+1} &= Ax_k + B(-K(x_k - x_k^d) + c_k + u_k^d) + Ed_k \\ &= (A - BK)x_k + B_k x_k^d + Bc_k + Bu_k^d + BEd_k \end{aligned} \quad (2.14)$$

Dersom vi definerer

$$\bar{A} = A - BK \quad \bar{u} = \begin{bmatrix} c_k \\ u_k^d \\ d_k \\ x_k^d \end{bmatrix} \quad \bar{B} = \begin{bmatrix} B \\ B \\ BE \\ BK \end{bmatrix}^T \quad (2.15)$$

systemet i (2.14) kan dermed forenkles til følgende

$$x_{k+1} = \bar{A}x_k + \bar{B}\bar{u}_k \quad (2.16)$$

Systemet i (2.16) vil være stabilt dersom K velges slik at \bar{A} har egenverdier innenfor enhets sirkelen. Vi vil gå ut fra at K er ubegrenset optimal for uendelig prediksjonshorisont, altså en LQR regulator. Det er kjent at uendelig horisont gir stabilitet for det nominelle systemet (dvs. $x_k \rightarrow x_k^d$ når $k \rightarrow \infty$). Det er ønskelig å få med uendelig horisont i objektiv funksjonen (2.8), men samtidig ønsker vi objektiv funksjonen formulert over endelig horisont for å begrense utregningskostnaden. Siste leddet i (2.9) kan omformuleres slik det er gjort i [13]. Med uendelig prediksjonshorisont får vi

$$\sum_{i=Hu}^{\infty} \left\| \hat{y}_{k+i|k} - y_{k+i}^d \right\|_Q^2 = (x_{k+Hu} - x_{k+Hu}^d)^T \left[\sum_{i=0}^{\infty} (\bar{A}^T)^i C^T Q C \bar{A}^i \right] (x_{k+Hu} - x_{k+Hu}^d) \quad (2.17)$$

Lar vi

$$\bar{Q} = \sum_{i=0}^{\infty} (\bar{A}^T)^i C^T Q C \bar{A}^i \quad (2.18)$$

Denne summen konvergerer dersom \bar{A} er stabil. Dette gir matrise Lyapunov ligning som kan løses for \bar{Q} dersom \bar{A}, C og Q er gitt.

$$\bar{A}^T \bar{Q} \bar{A} = \sum_{i=1}^{\infty} (\bar{A}^T)^i C^T Q C \bar{A}^i = \bar{Q} - C^T Q C \quad (2.19)$$

2.3.1 Objektiv funksjon

Objektiv funksjonen kan nå formuleres over endelig horisont H_u og er gitt på matriseform som

$$V(k) = (Y_{k+1} - Y_{k+1}^d)^T \tilde{Q} (Y_{k+1} - Y_{k+1}^d) + (U_k - U_k^d)^T \tilde{R} (U_k - U_k^d) \quad (2.20)$$

hvor

$$\tilde{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Q & 0 \\ 0 & 0 & \cdots & 0 & \bar{Q} \end{bmatrix} \quad \tilde{R} = \begin{bmatrix} R & 0 & \cdots & 0 & 0 \\ 0 & R & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & R \end{bmatrix} \quad (2.21)$$

$$Y_{k+1} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+H_u-1} \\ x_{k+H_u} \end{bmatrix} \quad U_k = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+H_u-2} \\ u_{k+H_u-1} \end{bmatrix} \quad (2.22)$$

Legg merke til at siste element i \tilde{Q} og Y_{k+1} er forskjellig fra de andre på grunn av formulering over endelig horisont.

Det er ønskelig å uttrykke objektiv funksjonen i (2.20) ved hjelp av matriser og optimaliseringsvariabel. Det må derfor finnes uttrykk for $(Y_{k+1} - Y_{k+1}^d)$ og $(U_k - U_k^d)$. Vi har at

$$Y_{k+1} = \bar{C} X_{k+1} \quad (2.23)$$

hvor

$$\bar{C} = \begin{bmatrix} C & 0 & \cdots & 0 & 0 \\ 0 & C & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & C & 0 \\ 0 & 0 & \cdots & 0 & I \end{bmatrix} \quad (2.24)$$

$$X_{k+1} = P_{x1} x_k + P_{u^d1} U_k^d + P_{x^d1} X_k^d + P_{d1} D_k + P_{c1} \hat{c}_k \quad (2.25)$$

Legg merke til at siste element i \bar{C} er ulik de andre. Vektorene er gitt av

$$U_k^d = \begin{bmatrix} u_k^d \\ u_{k+1}^d \\ \vdots \\ u_{k+Hu-2}^d \\ u_{k+Hu-1}^d \end{bmatrix} \quad X_k^d = \begin{bmatrix} x_k^d \\ x_{k+1}^d \\ \vdots \\ x_{k+Hu-2}^d \\ x_{k+Hu-1}^d \end{bmatrix} \quad D_k^d = \begin{bmatrix} d_k^d \\ d_{k+1}^d \\ \vdots \\ d_{k+Hu-2}^d \\ d_{k+Hu-1}^d \end{bmatrix} \quad \hat{c}_k = \begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ c_{k+Hu-2} \\ c_{k+Hu-1} \end{bmatrix} \quad (2.26)$$

og matrisene er gitt av

$$P_{x1} = \begin{bmatrix} \bar{A} \\ \bar{A}^2 \\ \vdots \\ \bar{A}^{Hu-1} \\ \bar{A}^{Hu} \end{bmatrix} \quad P_{u^d1} = \begin{bmatrix} B & 0 & \dots & 0 & 0 \\ \bar{A}B & B & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{A}^{Hu-2}B & \bar{A}^{Hu-3}B & \dots & B & 0 \\ \bar{A}^{Hu-1}B & \bar{A}^{Hu-2}B & \dots & \bar{A}B & B \end{bmatrix} \quad (2.27)$$

$$P_{x^d1} = \begin{bmatrix} BK_k & 0 & \dots & 0 & 0 \\ \bar{A}BK_k & BK_{k+1} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{A}^{Hu-2}BK_k & \bar{A}^{Hu-3}BK_{k+1} & \dots & BK_{k+Hu-2} & 0 \\ \bar{A}^{Hu-1}BK_k & \bar{A}^{Hu-2}BK_{k+1} & \dots & \bar{A}BK_{k+Hu-2} & BK_{k+Hu-1} \end{bmatrix} \quad (2.28)$$

$$P_{d1} = \begin{bmatrix} E & 0 & \dots & 0 & 0 \\ \bar{A}E & E & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{A}^{Hu-2}E & \bar{A}^{Hu-3}E & \dots & E & 0 \\ \bar{A}^{Hu-1}E & \bar{A}^{Hu-2}E & \dots & \bar{A}E & E \end{bmatrix} \quad (2.29)$$

$$P_{c1} = \begin{bmatrix} B & 0 & \dots & 0 & 0 \\ \bar{A}B & B & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{A}^{Hu-2}B & \bar{A}^{Hu-3}B & \dots & B & 0 \\ \bar{A}^{Hu-1}B & \bar{A}^{Hu-2}B & \dots & \bar{A}B & B \end{bmatrix} \quad (2.30)$$

Ved å kombinere (2.23), (2.25) og (2.20) får vi

$$Y_{k+1} - Y_{k+1}^d = \bar{C}(P_{x1}x_k + P_{u^d1}U_k^d + P_{x^d1}X_k^d + P_{d1}D_k + P_{c1}\hat{c}_k) - Y_{k+1}^d \quad (2.31)$$

som gir

$$Y_{k+1} - Y_{k+1}^d = \bar{C}P_{c1}\hat{c}_k - A_{k+1} \quad (2.32)$$

hvor A_{k+1} er avviket mellom referansen Y_{k+1}^d og den uavhengige prosessen dersom $\hat{c} = 0$ (fri respons).

$$A_{k+1} = Y_{k+1}^d - \bar{C}(P_{x1}x_k + P_{u^d1}U_k^d + P_{x^d1}X_k^d + P_{d1}D_k) \quad (2.33)$$

Av (2.20) og (2.32) får vi uttrykket

$$Y_{k+1} - Y_{k+1}^d = \hat{c}^T(\bar{C}P_{c1})^T \tilde{Q} \bar{C} P_{c1} \hat{c} - 2A_{k+1}^T \tilde{Q} \bar{C} P_{c1} \hat{c} \quad (2.34)$$

her har element som er uavhengig av optimaliseringsvariabel \hat{c} blitt fjernet.

Nå gjenstår det å finne et uttrykk for $(U_k - U_k^d)$ i (2.20). Ved å kombinere (2.4) og (2.6) får vi

$$U_k = P_{x2}x_k + P_{u^d2}U_k^d + P_{x^d2}X_k^d + P_{d2}D_k + P_{c2}\hat{c}_k \quad (2.35)$$

Hvor matrisene er gitt av

$$P_{x2} = \begin{bmatrix} -K_k \\ -K_{k+1}\bar{A} \\ -K_{k+2}\bar{A}^2 \\ \vdots \\ -K_{k+Hu-2}\bar{A}^{Hu-2} \\ -K_{k+Hu-1}\bar{A}^{Hu-1} \end{bmatrix} \quad (2.36)$$

$$P_{u^d2} = \begin{bmatrix} I & 0 & \cdots & 0 & 0 \\ -K_{k+1}B & I & \cdots & 0 & 0 \\ -K_{k+2}\bar{A}B & -K_{k+2}B & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -K_{k+Hu-1}\bar{A}^{Hu-2}B & -K_{k+Hu-1}\bar{A}^{Hu-3}B & \cdots & -K_{k+Hu-1}B & I \end{bmatrix} \quad (2.37)$$

$$P_{x^d2} = \begin{bmatrix} K_k & 0 & \cdots & 0 \\ -K_{k+1}BK_k & K_{k+1} & \cdots & 0 \\ -K_{k+2}\bar{A}BK_k & -K_{k+2}BK_{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -K_{k+Hu-1}\bar{A}^{Hu-2}BK_k & -K_{k+Hu-1}\bar{A}^{Hu-3}BK_{k+1} & \cdots & K_{k+Hu-1} \end{bmatrix} \quad (2.38)$$

$$P_{d2} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ -K_{k+1}E & 0 & \cdots & 0 & 0 \\ -K_{k+2}\bar{A}E & -K_{k+2}E & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -K_{k+Hu-1}\bar{A}^{Hu-2}E & -K_{k+Hu-1}\bar{A}^{Hu-3}E & \cdots & -K_{k+Hu-1}E & 0 \end{bmatrix} \quad (2.39)$$

$$P_{c2} = \begin{bmatrix} I & 0 & \cdots & 0 & 0 \\ -K_{k+1}B & I & \cdots & 0 & 0 \\ -K_{k+2}\bar{A}B & -K_{k+2}B & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -K_{k+Hu-1}\bar{A}^{Hu-2}B & -K_{k+Hu-1}\bar{A}^{Hu-3}B & \cdots & -K_{k+Hu-1}B & I \end{bmatrix} \quad (2.40)$$

Ved å kombinere (2.20) og (2.35) får vi

$$U_k - U_k^d = \bar{T}_k + P_{c2}\hat{c}_k \quad (2.41)$$

hvor

$$\bar{T}_k = P_{x2}x_k + P_{u^d2}U_k^d + P_{x^d2}X_k^d + P_{d2}D_k - U_k^d \quad (2.42)$$

Videre gir dette

$$(U_k - U_k^d)^T \tilde{R}(U_k - U_k^d) = \hat{c}^T P_{c2}^T \tilde{R} P_{c2} \hat{c} + 2\bar{T}_k^T \tilde{R} P_{c2} \hat{c} \quad (2.43)$$

Av (2.34) og (2.43) kan objektfunksjonen skrives som

$$V(k) = \hat{c}^T W_c \hat{c} + \Sigma \hat{c} \quad (2.44)$$

$$W_c = (\bar{C}P_{c1})^T \tilde{Q} \bar{C} P_{c1} + P_{c2}^T \tilde{R} P_{c2} \quad (2.45)$$

$$\Sigma = 2[\bar{T}_k^T \tilde{R} P_{c2} - A_{k+1}^T \tilde{Q} \bar{C} P_{c1}] \quad (2.46)$$

2.3.2 Begrensninger

Begrensninger på utgangen kan uttrykkes som

$$y_{min} \leq y_{k+1} \leq y_{max} \quad (2.47)$$

Over hele prediksjonshorizonten kan begrensningene uttrykkes med matriser

$$D_y Y_{k+1} \leq d_y \quad (2.48)$$

Dette gir

$$\underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & C \\ 0 & 0 & \cdots & 0 & -C \end{bmatrix}}_{D_y} \underbrace{\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ y_{k+4} \\ \vdots \\ y_{k+Hu-1} \\ x_{k+Hu} \end{bmatrix}}_{Y_{k+1}} \leq \underbrace{\begin{bmatrix} y_{max} \\ -y_{min} \\ y_{max} \\ -y_{min} \\ \vdots \\ y_{max} \\ -y_{min} \end{bmatrix}}_{d_y} \quad (2.49)$$

For å finne et uttrykk for Y_{k+1} , gitt av optimaliseringsvariabel \hat{c} kan vi bruke (2.23) og (2.25)

$$D_{yc}\hat{c} \leq d_{yc} \quad (2.50)$$

hvor

$$D_{yc} = D_y\bar{C}P_{c1} \quad (2.51)$$

$$d_{yc} = d_y - D_y\bar{C}(P_{x1}x_k + P_{u^d1}U_k^d + P_{x^d1}X_k^d + P_{d1}D_k) \quad (2.52)$$

Begrensninger på pådrag er gitt ved

$$u_{min} \leq u_k \leq u_{max} \quad (2.53)$$

uttrykker dette over hele reguleringshorisonten med matriser

$$D_u U_k \leq d_u \quad (2.54)$$

dette gir

$$\underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & -1 \end{bmatrix}}_{D_u} \underbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ u_{k+3} \\ \vdots \\ u_{k+Hu-2} \\ u_{k+Hu-1} \end{bmatrix}}_{U_k} \leq \underbrace{\begin{bmatrix} u_{max} \\ -u_{min} \\ u_{max} \\ -u_{min} \\ \vdots \\ u_{max} \\ -u_{min} \end{bmatrix}}_{d_u} \quad (2.55)$$

For å finne et uttrykk for U_k , gitt av optimaliseringsvariabel \hat{c} kan vi kombinere (2.35) og (2.54), dermed får vi

$$D_{uc}\hat{c} \leq d_{uc} \quad (2.56)$$

hvor

$$D_{uc} = D_u P_{c2} \quad (2.57)$$

$$d_{uc} = d_u - D_u(P_{x2}x_k + P_{u^d2}U_k^d + P_{x^d2}X_k^d + P_{d2}D_k) \quad (2.58)$$

Alle begrensningene kan nå ved matriser og optimiseringsvariabel \hat{c} ved å kombinere (2.50) og (2.56)

$$N\hat{c} \leq b - (Mx_k + M_{u^d}U_k^d + M_{x^d}X_k^d + M_d D_k) \quad (2.59)$$

hvor

$$N = \begin{bmatrix} D_y \bar{C} P_{c1} \\ D_u P_{c2} \end{bmatrix} \quad b = \begin{bmatrix} d_y \\ d_u \end{bmatrix} \quad M = \begin{bmatrix} D_y \bar{C} P_{x1} \\ D_u P_{x2} \end{bmatrix} \quad (2.60)$$

$$M_{u^d} = \begin{bmatrix} D_y \bar{C} P_{u^d1} \\ D_u P_{u^d2} \end{bmatrix} \quad M_{x^d} = \begin{bmatrix} D_y \bar{C} P_{x^d1} \\ D_u P_{x^d2} \end{bmatrix} \quad M_d = \begin{bmatrix} D_y \bar{C} P_{d1} \\ D_u P_{d2} \end{bmatrix} \quad (2.61)$$

Kapittel 3

Ulineær Modell Prediktiv Regulering

3.1 NMPC

I noen tilfeller er prosessmodellen veldig ulineær og prosessen er utsatt for store og hyppige forstyrrelser. I tillegg kan refreansen til systemet forandres og ofte spenne seg over et stort område av prosessdynamikken. Dette er faktorer som rettferdiggjør bruk av NMPC [21]. Hensikten til NMPC regulatoren er å stabilisere systemet i et stabilitetsområde og med hensyn til enkelte grenser som den lineære regulatoren vi bryte. Målet er at det skal lede til forbedret ytelse sammenlignet med lineær MPC.

Ulineær MPC baserer seg på en ulineær modell

$$\dot{x} = f(x, u) \tag{3.1}$$

Den ulineære modellen kan gi en ikke-kvadratisk kostnadsfunksjon og i tillegg kan begrensningene være ulineære, dette gir opphav til et ulineært program. At optimeringsproblemet er ikke-konvekst for ulineær MPC presenterer teoretiske og beregningsmessige utfordringer som ikke er tilstede i lineær MPC.

Lineær MPC har ofte dårlig prediksjonskapasitet noe som medfører redusert ytelse. I tillegg vil kvadratisk kostnadsfunksjon og lineære begrensninger ofte ikke reflektere behovene til industrien. Ulineær MPC derimot takler disse punktene på en mer konsistent måte.

3.1.1 NMPC metoder

I dag er de fleste metoder for å løse NMPC basert på Sekvensiell Kvadratisk Programmering (SQP) [7]. SQP baserer seg på å iterativt løse en kvadratisk tilnærming til objektifunksjonen og en lineær tilnærming av begrensningene. Et QP problem løses for å finne passende søkeretning. Deretter utføres typisk et linjesøk langs søkeretningen for

å finne optimal steglengde i den aktuelle retningen. Dersom den optimale løsningen til QP problemet ikke er et minimum til det opprinnelige problemet, kan vi lage et nytt QP problem ved å tilnærme det opprinnelige problemet i et område rundt den aktuelle løsningen. Dette gir en ny løsning. Prosessen kan gjentas inntil et minimum til det opprinnelige problemet finnes.

Det finnes forskjellige metoder i litteraturen for hvordan QP spesifiseres for å finne søkeretningen. Metodene kan deles inn i indirekte metoder og direkte metoder. Nå for tiden er direkte metoder mest brukt på grunn av deres anvendbarhet og robusthet [3]. Tre direkte metoder er: single shooting, samtidig tilnærming (samtidig simulering og optimisering) og multiple shooting.

I single shooting vil QP-delen av SQP algoritmen bestå av to deler: sekvensiell simulering og optimisering. Først simuleres modellen ved å kun bruke pådraget $(u_0, u_1, \dots, u_{L-1})$ som optimiseringsvariabel, dermed fjernes tilstandene $(x_0, x_1, \dots, x_{L-1})$ som optimiseringsvariabel. Videre løses et QP problem for å finne søkeretning. En fordel er at modelligningene er tilfredsstillt langs hele prediksjonshorisonen. En bakdel med denne tilnærmingen er at ustabile systemer er vanskelig å bearbeide [3].

I samtidig tilnærming er modellen implementert som en eksplisitt likhets begrensning. Dette gjør at både $(u_0, u_1, \dots, u_{L-1})$ og $(x_0, x_1, \dots, x_{L-1})$ er optimiseringsvariabler. En klar fordel er at divergens til tilstandsvariabler vil detekteres siden fremtidige tilstander er optimiseringsvariabler. En bakdel med denne tilnærmingen er at det blir stort antall optimiseringsvariabler og dermed regnekrevende.

Multiple shooting kan sees på som en kombinasjon av de to andre tilnærmingene. Her deles reguleringshorisonen inn i flere sub-horisonter. Modellen simuleres for hver sub-horison og det legges til som likhets begrensning at tilstanden ved slutten av hver sub-horison skal være lik den første tilstanden ved neste sub-horison. Fordel med denne tilnærmingen er at den detekterer divergens til tilstandsvariabler og er dermed bedre egnet for ustabile systemer. Inndelingen i sub-horisonter gir færre optimiseringsvariabler enn for samtidig tilnærming.

3.2 SQP-algoritme

I dette avsnittet vil det utledes en NMPC algoritme basert på SQP. Det vil brukes prestabilisering for bedre håndtering av ustabile systemer. Selve algoritmen for NMPC er beskrevet i Algoritme 2.

Trinnene i SQP algoritmen kan kort oppsummeres:

1. Bestem initialverdier
2. Finn nominell trajektorie

3. Lineariser rundt nominell trajektorie
4. Finn søkeretning (neste pådrag) ved å løse lokalt QP problem
5. Linjesøk langs løsningsvektor
6. Sjekk konvergenzkriterie

3.2.1 Initialverdier

Her er det fornuftig å velge nominell pådragsvektor fra forrige intervall [19], bare forskjøvet ett tidssteg.

$$\bar{U}_k = \begin{bmatrix} \bar{u}_k \\ \bar{u}_{k+1} \\ \vdots \\ \bar{u}_{k+Hu-2} \\ \bar{u}_{k+Hu-1} \end{bmatrix} \quad (3.2)$$

3.2.2 Nominell tilstandstrajektorie

Den nominelle tilstandstrajektorien finnes ved å integrere tilstandsligningen $\dot{x} = f(x, u, d)$ over reguleringshorisonten H_u , gitt at initialtilstanden x_k og nominell pådragsvektor \bar{U}_k er kjent. Her settes den nominelle pådragsvektoren like lang som reguleringshorisonten. Integrering gir den nominelle tilstandstrajektorien

$$\bar{x}_k = \int f(x, \bar{u}, d) dt \quad (3.3)$$

$$\bar{X}_{k+1} = \begin{bmatrix} \bar{x}_{k+1} \\ \bar{x}_{k+2} \\ \vdots \\ \bar{x}_{k+Hu-2} \\ \bar{x}_{k+Hu-1} \\ \bar{x}_{k+Hu} \end{bmatrix} \quad (3.4)$$

Dermed får vi

$$\bar{Y}_{k+1} = \bar{C} \bar{X}_{k+1} \quad (3.5)$$

hvor

$$\bar{C} = \begin{bmatrix} C & 0 & \cdots & 0 & 0 \\ 0 & C & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & C & 0 \\ 0 & 0 & \cdots & 0 & I \end{bmatrix} \quad (3.6)$$

3.2.3 Integrasjonsmetode

Vi ønsker en integrasjonsmetode som tilsvarende systemets respons i åpen og lukket sløyfe. Er systemet ustabil i åpen sløyfe skal også integrasjonsmetoden representere dette. Forbedret Eulers metode er gitt ved

$$\mathbf{k}_1 = \mathbf{f}(x_n, u_n) \quad (3.7)$$

$$\mathbf{k}_2 = \mathbf{f}(x_n + h\mathbf{k}_1, u_n) \quad (3.8)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}(\mathbf{k}_1 + \mathbf{k}_2) \quad (3.9)$$

Steglengden h må tilpasses slik at egenverdiene λ til systemet er innenfor stabilitetsområdet til integrasjonsmetoden. Stabilitetsområdet på den reelle akse for forbedret Eulers metode blir:

$$-\frac{2}{h} \leq \lambda \leq 0 \quad (3.10)$$

Eventuelt kan 4. ordens Runge Kutta metode (RK4) brukes

$$\mathbf{k}_1 = \mathbf{f}(x_n, u_n) \quad (3.11)$$

$$\mathbf{k}_2 = \mathbf{f}(x_n + \frac{h}{2}\mathbf{k}_1, u_n) \quad (3.12)$$

$$\mathbf{k}_3 = \mathbf{f}(x_n + \frac{h}{2}\mathbf{k}_2, u_n) \quad (3.13)$$

$$\mathbf{k}_4 = \mathbf{f}(x_n + h\mathbf{k}_3, u_n) \quad (3.14)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (3.15)$$

RK4 har derimot et lite område i høyre halvplan som er stabilt, dermed må vi være påpasselig på at systemet ikke har noen egenverdier i dette området. RK4 tillater bruk av større steglengde enn Forbedret Eulers metode, men samtidig er også RK4 mer regnekrevende.

3.2.4 Linearisering og diskretisering

For å kunne bruke standard QP algoritmer må den ulineære modellen lineariseres. Her velges det å lineariseres rundt den nominelle trajektorien. Da det ikke lineariseres om et likevektspunkt vil tilstanden x ikke forbli konstant og dermed får vi et tidsvarierende lineært system. Denne modellen vil bare være gyldig for et bestemt område rundt det lineariserte punktet. Gitt det ulineære systemet

$$\begin{aligned} \dot{x} &= f(x, u, d) \\ y &= h(x, u, d) \end{aligned} \quad (3.16)$$

Diskretisering linearisert om nominellt pådrag \bar{U}_k og nominell tilstandsvektor \bar{X}_{k+1} gir den lineariserte diskrete modellen:

$$x_{k+1} = A_k x_k + B_k u_k \quad (3.17)$$

$$y_k = C_k x_k \quad (3.18)$$

Pådraget som brukes

$$u_k = \alpha(x_k) + c_k \quad (3.19)$$

hvor $\alpha(x_k)$ er et prestabiliserende pådrag og c_k er pådraget fra MPC regulatoren. Her velges $\alpha(x_k)$ til $-K_k(x_k)$ hvor K_k er forsterknings-matrise som sørger for at systemmatrisen i (3.22) får egenverdier innenfor enhetssirkelen. Med referansefølging får vi pådraget

$$u_k = u_k^d - K_k(x_k - x_k^d) + c_k \quad (3.20)$$

Settes pådraget inn i (3.17)

$$\begin{aligned} x_{k+1} &= (A_k - B_k K_k) x_k + B_k u_k^d + B_k x_k^d + B_k c_k \\ &= \bar{A}_k x_k + B_k u_k^d + B_k x_k^d + B_k c_k \end{aligned} \quad (3.21)$$

hvor

$$\bar{A}_k = A_k - B_k K_k \quad (3.22)$$

3.2.5 Diskretiseringsmetoder

Ved valg av diskretiseringsmetode er det forskjellige hensyn å ta. For å velge en passende diskretiseringsmetode må det tas hensyn til hvor spredt egenverdiene til det aktuelle systemet er. Det må også tas hensyn til hvor store egenverdiene er for å kunne velge en steglengde som gjør diskretiseringen numerisk stabil. Den eksakte løsningen på diskretiserin er gitt ved

$$x_{k+1} = e^{Ah} x_k + \left(\int_0^h e^{Av} dv \right) B u_k \quad (3.23)$$

hvor $v = kh + h - \tau$ og pådraget u_k forutsettes konstant under integralet. A og B er lineariserte matriser fått ved perturbering av systemet, $A = \partial f / \partial x$ og $B = \partial f / \partial u$.

Forward Euler

Forward Euler baserer seg på at for små tidssteg er $e^{Ah} \approx I + Ah$; altså de to første leddene av Taylorrekken. Forward Euler på systemet i (3.16)

$$\begin{aligned} \frac{x_{k+1} - x_k}{h} &= \frac{\partial f}{\partial x} \Big|_{(\cdot)} x_k + \frac{\partial f}{\partial u} \Big|_{(\cdot)} \frac{\partial u}{\partial x} \Big|_{(\cdot)} x_k + \frac{\partial f}{\partial u} \Big|_{(\cdot)} \frac{\partial u}{\partial x^d} \Big|_{(\cdot)} x_k^d + \\ &\quad \frac{\partial f}{\partial u} \Big|_{(\cdot)} \frac{\partial u}{\partial c} \Big|_{(\cdot)} c_k + \frac{\partial f}{\partial u} \Big|_{(\cdot)} \frac{\partial u}{\partial u_k^d} \Big|_{(\cdot)} u_k^d \end{aligned} \quad (3.24)$$

Matrisene i (3.21) blir dermed

$$\begin{aligned}
A_k &= I + h \frac{\partial f}{\partial x} \Big|_{(\cdot)} & B_k &= h \frac{\partial f}{\partial u} \Big|_{(\cdot)} \\
-K_k &= \frac{\partial f}{\partial x} \Big|_{(\cdot)} \frac{\partial u}{\partial x^d} \Big|_{(\cdot)} = K_k & C_k &= \frac{\partial y}{\partial x} \Big|_{(\cdot)} \\
\frac{\partial u}{\partial c} \Big|_{(\cdot)} &= I & \frac{\partial u}{\partial u_k^d} \Big|_{(\cdot)} &= I
\end{aligned} \tag{3.25}$$

Forward Euler er numerisk stabil dersom $h\lambda$ er innenfor en sirkel med radius 1 omkring -1 på den reelle akse, her er h steglengden og λ er egenverdiene til Jacobian $J = \partial \mathbf{f}(\mathbf{x}, t) / \partial \mathbf{x}$.

Bilineær transformasjon

Systemer som har en stor spredning i egenverdier til Jacobian betegnes som stive systemer. Stive systemer går under betegnelsen som systemer som er vanskelig å løse med eksplisitte metoder. En mulig tilnærming til e^{Ah} i (3.23)

$$e^{Ah} \approx (I + \frac{1}{2}Ah)(I - \frac{1}{2}Ah)^{-1} \tag{3.26}$$

Denne metoden kalles bilineær transformasjon eller Tustin tilnærming (trapezoidal regel). Dette er en implisitt Runge-Kutta metode og den er stabil for alle λ som har negativ reell del. Flere metoder for diskretisering kan finnes i [4].

3.2.6 QP-Problem

Som for det lineære tilfellet vil QP-problemet bestå i å minimere objektiv funksjonen gitt på matriseform

$$V_k = (\hat{Y}_{k+1} - Y_{k+1}^d)^T \tilde{Q} (\hat{Y}_{k+1} - Y_{k+1}^d) + (\hat{U}_k - U_k^d)^T \tilde{R} (\hat{U}_k - U_k^d) \tag{3.27}$$

hvor

$$\hat{U}_k = \begin{bmatrix} \hat{u}_k \\ \hat{u}_{k+1} \\ \vdots \\ \hat{u}_{k+Hu-2} \\ \hat{u}_{k+Hu-1} \end{bmatrix}, \quad \hat{Y}_{k+1} = \begin{bmatrix} \hat{y}_{k+1} \\ \hat{y}_{k+2} \\ \vdots \\ \hat{y}_{k+Hu-1} \\ \hat{x}_{k+Hu} \end{bmatrix} \tag{3.28}$$

Ved å kombinere (3.18) og (3.21) får vi den lineære modellen linearisert rundt nominell trajektorie

$$y_{k+1} = C(\bar{A}_k x_k + B_k u_k^d + B_k x_k^d + B_k c_k) \quad (3.29)$$

Et uttrykk for \hat{Y}_{k+1} kan finnes ved å bruke en 1. ordens Taylor tilnærming omkring den nominelle system utgangen \bar{Y}_{k+1}

$$\hat{Y}_{k+1} = \bar{Y}_{k+1} + \left. \frac{\partial Y_{k+1}}{\partial \hat{c}} \right|_{(\bar{Y}_{k+1})} \Delta \hat{c}_k \quad (3.30)$$

Her brukes hatten "ˆ" for å indikere at utgangen er en første ordens tilnærming linearisert omkring nominell trajektorie. Her er $\Delta \hat{c}_k$ optimeringsvariabel som finnes ved å løse QP-problemet

$$\Delta \hat{c}_k = \begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ c_{k+Hu-2} \\ c_{k+Hu-1} \end{bmatrix} \quad (3.31)$$

Ved å kombinere (3.29) og den partiellderiverte i (3.30) får vi

$$\frac{\partial y_{k+1}}{\partial c_k} = \frac{\partial y_{k+1}}{\partial x_{k+1}} \cdot \frac{\partial x_{k+1}}{\partial c_k} = C_{k+1} B_k \quad (3.32)$$

$$\frac{\partial y_{k+2}}{\partial c_k} = \frac{\partial y_{k+2}}{\partial x_{k+2}} \cdot \frac{\partial x_{k+2}}{\partial x_{k+1}} \cdot \frac{\partial x_{k+1}}{\partial c_k} = C_{k+2} \bar{A}_{k+1} B_k \quad (3.33)$$

$$\frac{\partial y_{k+2}}{\partial c_{k+1}} = \frac{\partial y_{k+2}}{\partial x_{k+2}} \cdot \frac{\partial x_{k+2}}{\partial c_{k+1}} = C_{k+2} B_{k+1} \quad (3.34)$$

Dermed blir sensitivetsmatrisen

$$S_y = \begin{bmatrix} CB_k & 0 & \cdots & 0 & 0 \\ C\bar{A}_{k+1}B_k & CB_{k+1} & \cdots & 0 & 0 \\ C\bar{A}_{k+2}\bar{A}_{k+1}B_k & C\bar{A}_{k+2}B_{k+1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C \prod_{i=2}^{Hu-1} \bar{A}_{k+Hu-i} B_k & C \prod_{i=2}^{Hu-2} \bar{A}_{k+Hu-i} B_{k+1} & \cdots & CB_{k+Hu-2} & 0 \\ \prod_{i=1}^{Hu-1} \bar{A}_{k+Hu-i} B_k & \prod_{i=1}^{Hu-2} \bar{A}_{k+Hu-i} B_{k+1} & \cdots & \bar{A}_{k+Hu-1} B_{k+Hu-2} & B_{k+Hu-1} \end{bmatrix} \quad (3.35)$$

Her vil den dynamiske matrisen S_y representere prosessens sensitivitet til forandringer i optimeringsvariabel omkring den nominelle trajektorien. Elementet nederst til venstre i sensitivetsmatrisen S_y er det samme som observerbarhetsmatrisen for det lineariserte systemet (dersom C settes lik identitetsmatrisen). Full observabilitet tilfredstiller dermed å garantere full rang til S_y siden alle andre element er forkortelser av elementet nederst til høyre. Full rang til S_y impliserer observerbarhet av alle ustabile tilstander, og systemet kan dermed stabiliseres.

Uttrykket for (3.30) kan skrives som

$$\hat{Y}_{k+1} = \bar{Y}_{k+1} + S_y \Delta \hat{c}_k \quad (3.36)$$

Finner bidraget fra utgangen til objektiv funksjonen ved å sette inn for første ordens tilnærming linearisert omkring nominell trajektorie \hat{Y}_{k+1} i (3.27)

$$V_y = (\bar{Y}_{k+1} + S_y \Delta \hat{c}_k - Y_{k+1}^d)^T \tilde{Q} (\bar{Y}_{k+1} + S_y \Delta \hat{c}_k - Y_{k+1}^d) \quad (3.37)$$

Fjerner ledd uavhengig av optimaliseringsvariabel $\Delta \hat{c}_k$, dermed får vi

$$V_y = (\Delta \hat{c}_k)^T (S_y)^T \tilde{Q} S_y \Delta \hat{c}_k + 2(\bar{Y}_{k+1})^T \tilde{Q} S_y \Delta \hat{c}_k - 2(Y_{k+1}^d)^T \tilde{Q} S_y \Delta \hat{c}_k \quad (3.38)$$

Tilsvarende for Y_{k+1} kan det også finnes et uttrykk for U_{k+1} ved å bruke en 1. ordens Taylor tilnærming. Pådraget er basert på den lineære modellen som er linearisert omkring nominell trajektorie \bar{U}_k og \bar{X}_{k+1} som i (3.20)

$$u_k = u_k^d - K_k(x_k - x_k^d) + c_k \quad (3.39)$$

Et estimat for U_k kan finnes ved å bruke en 1. ordens taylor tilnærming omkring det nominelle pådraget \bar{U}_k

$$\hat{U}_k = \bar{U}_k + \left. \frac{\partial U_k}{\partial \hat{c}} \right|_{(\bar{u}_k)} \Delta \hat{c}_k \quad (3.40)$$

Sensitivitetsmatrisen som representerer de partiellderiverte blir

$$S_u = \begin{bmatrix} I & 0 & \cdots & 0 \\ -K_{k+1} B_k & I & \cdots & 0 \\ -K_{k+2} \bar{A}_{k+1} B_k & -K_{k+2} B_{k+1} & \cdots & 0 \\ -K_{k+3} \bar{A}_{k+2} \bar{A}_{k+1} B_k & -K_{k+3} \bar{A}_{k+2} B_{k+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -K_{k+Hu-2} \prod_{i=3}^{Hu-1} \bar{A}_{k+Hu-i} B_k & -K_{k+Hu-2} \prod_{i=3}^{Hu-2} \bar{A}_{k+Hu-1} B_{k+1} & \cdots & I \\ -K_{k+Hu-1} \prod_{i=2}^{Hu-1} \bar{A}_{k+Hu-i} B_k & -K_{k+Hu-1} \prod_{i=2}^{Hu-2} \bar{A}_{k+Hu-i} B_{k+1} & \cdots & -K_{k+Hu-1} B_{k+Hu-2} \end{bmatrix} \quad (3.41)$$

ved å bruke sensitivitetsmatrisen får vi

$$\hat{U}_k = \bar{U}_k + S_u \Delta \hat{c}_k \quad (3.42)$$

I (3.27) erstattes U_k med den 1. ordens tilnærmingen \hat{U}_k . Pådrags-delen av kostnads-funksjonen kan nå finnes ved å kombinere (3.42) og (3.27).

$$V_u = (\bar{U}_k + S_u \Delta \hat{c}_k - U_k^d)^T \tilde{R} (\bar{U}_k + S_u \Delta \hat{c}_k - U_k^d) \quad (3.43)$$

Som gir

$$V_u = (\Delta \hat{c}_k)^T (S_u)^T \tilde{R} S_u \Delta \hat{c}_k + 2(\bar{U}_k)^T \tilde{R} S_u \Delta \hat{c}_k - 2(U_k^d)^T \tilde{R} S_u \Delta \hat{c}_k \quad (3.44)$$

Av (3.38) og (3.44) får vi kostnadsfunksjonen

$$V_k = (\Delta \hat{c}_k)^T W \Delta \hat{c}_k + \Sigma \Delta \hat{c}_k \quad (3.45)$$

her er

$$W = (S_y)^T \tilde{Q} S_y + (S_u)^T \tilde{R} S_u \quad (3.46)$$

$$\Sigma = 2[(\bar{Y}_{k+1})^T \tilde{Q} S_y - (Y_{k+1}^d)^T \tilde{Q} S_y + (\bar{U}_k)^T \tilde{R} S_u - (U_k^d)^T \tilde{R} S_u] \quad (3.47)$$

Etter at søkeretningen er funnet bør vi ha fått en ganske god trajektorie frem i tid, dersom det velges å lineariseres rundt denne igjen må pådragsvektoren det skal integreres rundt oppdateres. Fra denne iterasjonen har vi estimatet

$$\hat{U}_{k,j} = \bar{U}_{k,j} + S_{u_{k,j}} \lambda \Delta \hat{c}_{k,j} \quad (3.48)$$

Nominellt pådrag for neste iterasjon blir dermed

$$\bar{U}_{k,j+1} = \left[\hat{u}_{k,j}^T \quad \hat{u}_{k+1,j}^T \quad \cdots \quad \hat{u}_{k+Hu-2,j}^T \quad \hat{u}_{k+Hu-1,j}^T \right]^T \quad (3.49)$$

Terminal kostnad

For at optimiseringsproblemet i (3.27) skal kunne formuleres over endelig horisont bruker vi en terminal kostnad. Dette medfører at siste element i \tilde{Q} -matrisen forskjellig fra de andre. Siste element i \tilde{Q} -matrisen finnes ved å løse en matrise Lyapunov ligning med hensyn på \bar{Q} , dersom \bar{A}, C og Q er gitt, som vist i (2.17) - (2.19)

$$\bar{A}^T \bar{Q} \bar{A} = \sum_{i=1}^{\infty} (\bar{A}^T)^i C^T Q C \bar{A}^i = \bar{Q} - C^T Q C \quad (3.50)$$

Legg også merke til at terminal kostnaden medfører siste element i (3.6), (3.28) og (3.35) blir forskjellig fra de andre. Ved å bytte ut Q i (3.50) med $Q = Q + K^T R K$ vil også pådraget vektlegges over den uendelige horisonten.

Terminalkostnaden forutsetter at K er en forsterkningsmatrise som sørger for at $\bar{A}_k \dots \bar{A}_{k+Hu-1}$ får egenverdier innenfor enhetssirkelen. Det kan være vanskelig å finne en slik som stabiliserer systemet over den delen av tilstandsrommet som er interessant, men som et første forsøk kan man prøve LQR-regulatoren. Videre forutsetter den endelige formuleringen at den lineariserte matrisen \bar{A}_{k+Hu} er gyldig over den uendelige horisonten.

Ved bruk av dynamisk K-matrise er et alternativ er å løse den diskrete-tids Ricatti ligningen. Denne ligningen gir en terminalkostnad \bar{Q} som vektlegger både tilstander og pådrag over den uendelige horisonten.

$$A^T \bar{Q} A - \bar{Q} - (A^T \bar{Q} B)(B^T \bar{Q} B + R)^{-1}(B^T \bar{Q} A) + Q = 0 \quad (3.51)$$

K finnes ved å løse (3.51) med hensyn på \bar{Q}

$$K = (B^T \bar{Q} B + R)^{-1}(B^T \bar{Q} A) \quad (3.52)$$

Disse to ligningene kan løses i Matlab med funksjonen *dare* og kan løses for hvert enkelt sample gitt matrisene Q , R og de lineariserte matrisene A_k og B_k . I praksis viste det seg at det gikk noe raskere å løse Lyapunovligningen enn Ricattiligningen, noe som er logisk siden Ricattiligningen som løses ved Matlab-funksjonen *dare* i tillegg returnerer forsterkningsmatrisen K .

3.2.7 Begrensninger

Begrensninger på pådrag og utgang kan skrives som

$$u_{min} \leq u_k \leq u_{max} \quad (3.53)$$

$$y_{min} \leq y_{k+1} \leq y_{max} \quad (3.54)$$

Begrensningene basert på den lineariserte modellen uttrykket med matriser over pådrags og prediksjonshorisonten

$$D_u U_k \leq du \quad (3.55)$$

$$D_y Y_{k+1} \leq dy \quad (3.56)$$

Her er D_u og D_y definert i henholdsvis (2.55) og (2.49). Vi ønsker begrensningene gitt av optimeringsvariabel $\Delta \hat{c}_k$. Vi bruker (3.42) og (3.36) og setter inn for U_k og Y_{k+1} .

$$D_u(\bar{U}_k + S_u \Delta \hat{c}_k) \leq du \quad (3.57)$$

$$D_y(\bar{Y}_{k+1} + S_y \Delta \hat{c}_k) \leq dy \quad (3.58)$$

Ved å kombinere (3.57) og (3.58) får vi de lineariserte begrensningene uttrykt ved optimeringsvariabel $\Delta \hat{c}$

$$N \Delta \hat{c} \leq b \quad (3.59)$$

$$N = \begin{bmatrix} D_u S_u \\ D_y S_y \end{bmatrix} \quad b = \begin{bmatrix} du - D_u \bar{U}_k \\ dy - D_y \bar{Y}_{k+1} \end{bmatrix} \quad (3.60)$$

3.2.8 Linjesøk

Etter at søkeretningen er funnet kan det kjøres linjesøk for å bestemme hvor langt steg vi skal forflytte oss i den aktuelle retningen. I linjesøket vil vi at det skal skje en reduksjon i objektiv funksjonen samtidig som begrensningene overholdes. I dette avsnittet formuleres et linjesøk som tar hensyn til reduksjon i objektiv funksjonen og som straffer brudd på myke begrensninger.

Begrensningene er gitt av (3.55)

$$g_{uk} = D_u \hat{U}_k - du \quad (3.61)$$

$$g_{yk} = D_y \hat{Y}_{k+1} - dy \quad (3.62)$$

hvor \hat{U}_k og \hat{Y}_{k+1} er gitt av (3.42) og (3.36). Brudd på begrensninger kan formuleres som

$$\gamma_k = \sum_{i=1}^{2 \times Hu} \omega_{u,i} [g_{uk,i}]^+ + \sum_{i=1}^{2 \times Hu} \omega_{y,i} [g_{yk,i}]^+ \quad (3.63)$$

her er ω_u og ω_y vektorer for de aktuelle begrensningene hvor det kan legges mindre vekt på brudd nært i tid enn på brudd langt frem i tid. Det kan legges forskjellig vekt på brudd på begrensninger for pådrag og utgang

$$\omega_u = [\omega_{u_1} \quad \omega_{u_2} \cdots \omega_{u_{2Hu}}], \quad \dim = 1 \times 2Hu \quad (3.64)$$

$$\omega_y = [\omega_{y_1} \quad \omega_{y_2} \cdots \omega_{y_{2Hu}}], \quad \dim = 1 \times 2Hu \quad (3.65)$$

Slik begrensningene er formulert er det kun positive element i vektorene g_{u_k} og g_{y_k} som vil være brudd på begrensningene. Funksjonen som er brukt i (3.63) er derfor definert som

$$[z]^+ = \max(0, z) \quad (3.66)$$

En straffefunksjon for brudd på begrensninger kan nå formuleres lignende den i [18].

$$F_k(\lambda \Delta \hat{c}_k) = V_k(\lambda \Delta \hat{c}_k) + b \gamma_k(\lambda \Delta \hat{c}_k) + \alpha \gamma_k(\lambda \Delta \hat{c}_k)^2 \quad (3.67)$$

Her er α og b er positive skalarer. Det er det tatt med både lineær og kvadratisk vektning for brudd på begrensninger. Den kvadratiske vektningen vil i liten grad straffe små brudd på begrensningene. Derfor er det viktig at den lineære vektningen b velges tilstrekkelig stor for å straffe små brudd på begrensningene. V_k er objektiv funksjonen for det aktuelle steget og er gitt i (3.27). Objektiv funksjonen V_k og bruddet på begrensningene γ_k finnes ved å evaluere den ulineære modellen for pådraget med den aktuelle steglengden λ

$$\hat{U}_{k,j} = \bar{U}_{k,j} + S_{u_{k,j}} \lambda \Delta \hat{c}_{k,j} \quad (3.68)$$

For hver gang ny F_k skal beregnes må det oppdateres estimat av U_k og Y_{k+1} for objektiv funksjonen og for begrensningene i (3.61) og (3.62). Hovedmomentene for en backtracking linjesøk algoritme er gitt i Algoritme 1

Harde begrensninger

Begrensningene beskrevet for linjesøket over er myke begrensninger, det vil si begrensninger som kan brytes til en viss grad. Ofte er det ikke ønskelig eller fysisk mulig at en begrensning overskrides. Det er tilfellet ved for eksempel regulering av ventiler hvor ventilåpningen må være mellom null og hundre prosent. For linjesøket kan harde begrensninger inkluderes for inngangen ved å bruke (3.57)

$$D_u(\bar{U}_k + \lambda S_u \Delta \hat{c}_k) \leq du \quad (3.69)$$

Dette gir

$$\lambda = (D_u)^{-1}(du - D_u \bar{U}_k)(S_u \Delta \hat{c}_k)^{-1} \quad (3.70)$$

Etter at søkretningen \hat{c}_k er funnet, er det bare vektoren λ som er ukjent. Ligningen over kan dermed løses med hensyn på λ . Det minste elementet i vektoren λ brukes som største steglengde i søkeretningen i linjesøket. Selvfølgelig kan ikke lambda være mindre enn null. Ny verdi for λ må finnes hver gang linjesøksrutinen kalles.

3.3 Konvergenzkriterie

For å sikre at nominelt pådrag gir sensitivitetmatriser som representerer systemdynamikken tilfredsstillende må QP-problemet løses tilstrekkelig mange ganger. For å sikre QP-problemet løses tilstrekkelig mange ganger, er det implementert et konvergenzkriterie lignende det i [12]. Her settes en grense for hvor stor endring i pådrag som skal tillates

$$\|\bar{U}_{k,j+1} - \bar{U}_{k,j}\| \leq \Delta u_{min} \quad (3.71)$$

Dette implementeres som

$$\|\lambda S_u \Delta \hat{c}_k\| \leq \Delta u_{min} \quad (3.72)$$

En liten verdi på Δu betyr at det nominelle pådraget frem i tid er ganske optimalt for den lineariserte prediksjonen av systemet, og at lineariseringen er gjort tilstrekkelig mange ganger.

Algorithm 1 Linjesøk

```
procedure LINJESOK( $\bar{U}_{k,j}$ ,  $Su_{k,j}$ ,  $\Delta\hat{c}_{k,j}$ )  
   $\tau \in (0, 1)$ ,  $n_{LS} = 1$  ▷ Initialiserer variabler  
   $\lambda_{opt} = 1$   
   $LS \leftarrow true$   
   $\phi_{nom} \leftarrow F_k(\bar{U}_{k,j})$  ▷ Straffefunksjon for nominellt pådrag  
   $\phi_2 \leftarrow F_k(\bar{U}_{k,j} + Su_{k,j}\lambda_{opt}\Delta\hat{c}_{k,j})$   
  while  $LS$  do  
     $\lambda_{ny} \leftarrow \tau\lambda_{opt}$   
     $\phi_1 \leftarrow \phi_2$   
     $\phi_2 \leftarrow F_k(\bar{U}_{k,j} + Su_{k,j}\lambda_{ny}\Delta\hat{c}_{k,j})$   
    if  $\phi_1 \leq \phi_{nom}$  and  $\phi_1 \leq \phi_2$  then ▷ Tester om steglengde er optimal  
       $LS \leftarrow false$   
    else if  $n_{LS} == n_{LSmax}$  then  
       $LS \leftarrow false$   
       $print \leftarrow$  linjesøk avbrutt av maks antall iterasjoner  
      if  $\phi_2 \leq \phi_{nom}$  then  
         $\lambda_{opt} \leftarrow \lambda_{ny}$   
      else  
         $\lambda_{opt} \leftarrow 0$   
      end if  
    else  
       $\lambda_{opt} \leftarrow \lambda_{ny}$   
       $n_{LS} = n_{LS} + 1$   
    end if  
  end while  
  return  $\lambda_{opt}$   
end procedure
```

Algorithm 2 NMPC

```
 $QP \leftarrow true, n_{QP_{max}} \leftarrow verdi$   
 $Initialiser \bar{U}_k$  ▷ Finn initialverdier til pådraget  
for  $k \leftarrow 1, \dots$  do  
   $j \leftarrow 1$   
  while QP do  
     $\bar{X}_{k+1,j} \leftarrow integrasjon(\bar{U}_{k,j}, x_k)$   
     $A_{big}, B_{big} \leftarrow LineariserDiskretiser(\bar{X}_{k+1,j}, \bar{U}_{k,j})$   
     $\hat{Y}_{k+1} = \bar{Y}_{k+1} + S_y \Delta \hat{c}_{k,j}$  ▷ Estimat for QP problem  
     $\hat{U}_k = \bar{U}_k + S_u \Delta \hat{c}_{k,j}$  ▷ Estimat for QP problem  
     $\Delta \hat{c}_{k,j} \leftarrow LokaltQP(\hat{Y}_{k+1,j}, \hat{U}_{k,j})$  ▷ Finner søkeretning  
     $\lambda_k \leftarrow LinjeSok(\bar{U}_{k,j}, S_{u_{k,j}}, \Delta \hat{c}_{k,j})$   
    if konvergenskriterium then ▷ Sjekker konvergenskriterium  
       $QP \leftarrow false$   
    else if  $n_{QP} == n_{QP_{max}}$  then  
       $QP \leftarrow false$   
       $print \leftarrow søkeretning$  QP avbrutt av maks antall iterasjoner  
    else  
       $\bar{U}_{k,j+1} = \bar{U}_{k,j} + S_{u_{k,j}} \lambda_{k,j} \Delta \hat{c}_{k,j}$   
       $n_{QP} \leftarrow n_{QP} + 1$   
       $j \leftarrow j + 1$   
    end if  
  end while  
   $u_k = \bar{u}_k + \lambda_k \Delta c_k$  ▷ Setter ut pådraget (første element i pådragsvektor)  
   $\bar{U}_{k+1,1} = \bar{U}_{k,j}$  ▷ Initialiserer pådrag for neste tidssteg  
end for
```

3.4 CSTR - eksempel

For å vise funksjonaliteten til NMPC algoritmen vil det presenteres et eksempel. Eksempelet er en ideell isotermisk CSTR, også brukt i [19] og [14]. Prosessmodellen er gitt ved

$$\frac{dx_1}{dt} = u_1 + u_2 - k_1\sqrt{x_1} \quad (3.73)$$

$$\frac{dx_2}{dt} = (C_{B1} - x_2)\frac{u_1}{x_1} + (C_{B2} - x_2)\frac{u_2}{x_1} - \frac{k_2x_2}{(1+x_2)^2} \quad (3.74)$$

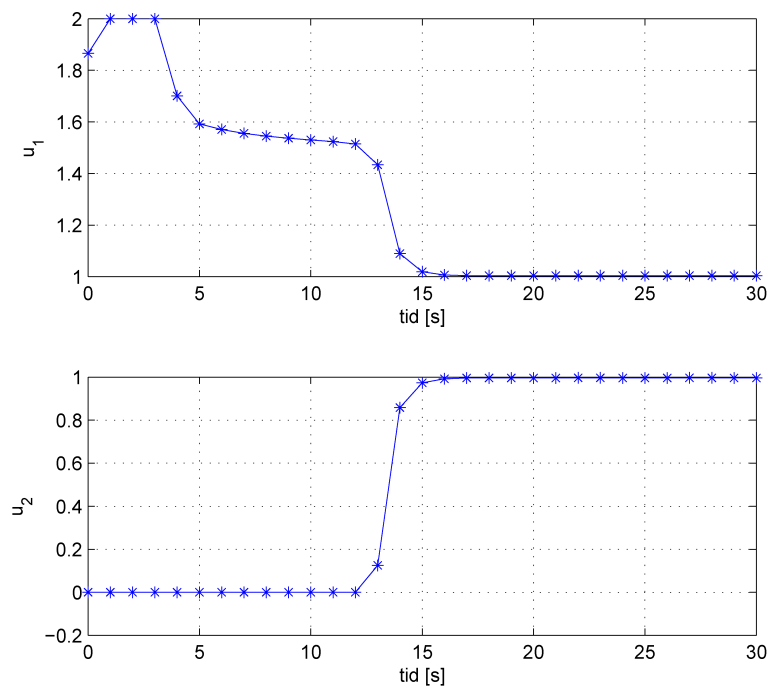
Parametrene er satt til $k_1 = 0.2$, $k_2 = 1$, $C_{B1} = 24.9$ og $C_{B2} = 0.1$. Manipulerte variabler er gitt av (u_1, u_2) . Vi går ut fra at begge tilstandsvariablene kan måles og at prosessutgangen er gitt av (x_1, x_2) . Ved å sette $(u_1, u_2) = (1, 1)$ har modellen tre likevektspunkt, ved $x_1 = 100$ og $x_2 = (0.633, 2.72, 7.07)$. Det midterste likevektspunktet er ustabil, mens de to andre er stabile.

Målet er å holde tilstandene og de manipulerte variabler ved det ustabile punktet $x_{eq} = (100, 2.72)$ og $u_{eq} = (1, 1)$, for å teste NMPC-algoritmen. Systemet er simulert med samplingtid på $T_s = 1s$. Reguleringshorisont er satt til $H_u = 10$. Vektene settes til

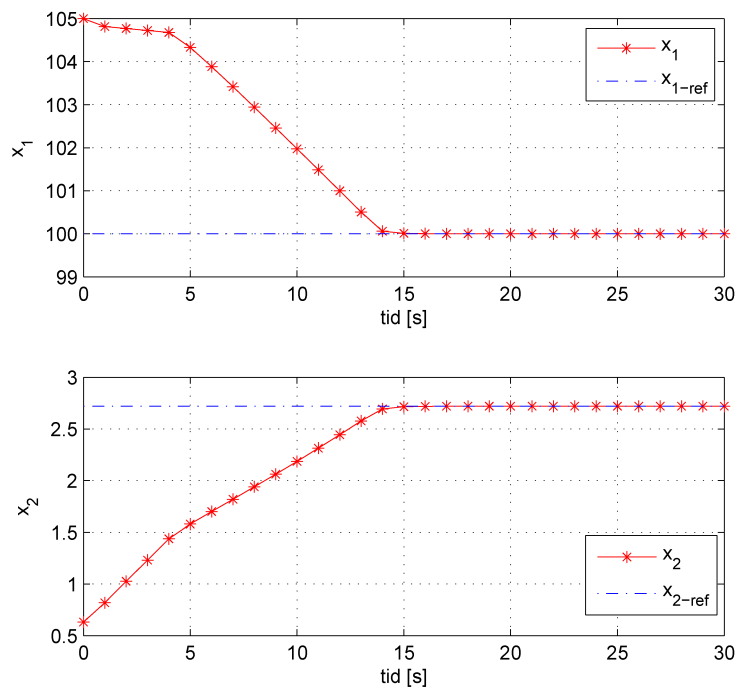
$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 160 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.75)$$

For å teste hvordan algoritmen håndterer bergrensninger, vil det settes bergrensninger på pådraget. Begrensningen som brukes er $0 \leq u_1 \leq 2$, $0 \leq u_2 \leq 2$. Som vist i Figur 3.1 er begrensningene aktivert for begge pådragene.

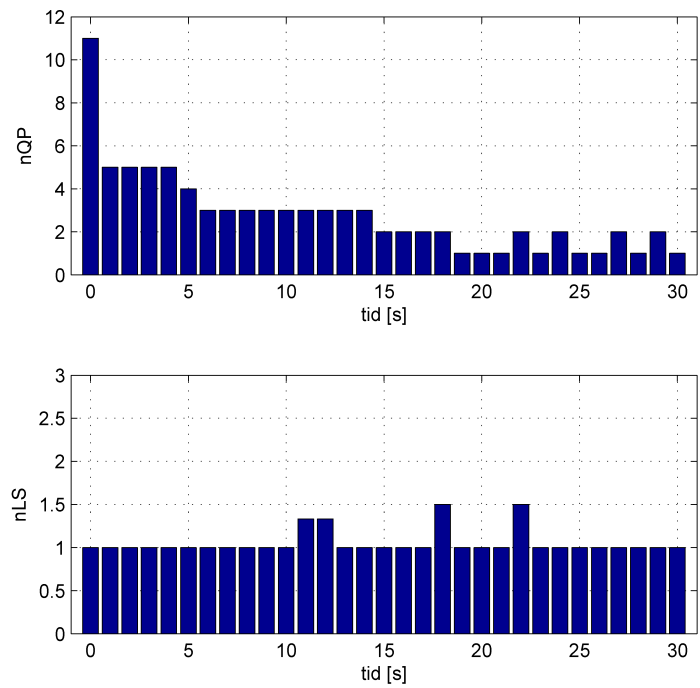
Initialtilstand er satt til $(x_1, x_2) = (105, 0.633)$, dette gir responsen som vist i Figur 3.2. Det er ikke brukt noen begrensninger på utgang. Konvergenzkriteriet for NMPC-algoritmen er satt til $\Delta u_{min} = 0.0005$. I Figur 3.3 kan vi se dette konvergenzkriteriet medfører at QP-problemet løses maks 11 ganger. Parameteren i linjesøket som bestemmer verdien til ny λ er satt til $\tau = 0.9$. Av Figur 3.3 ser vi at dette gir gjennomsnittlig få iterasjoner i linjesøket.



Figur 3.1: Pådrag



Figur 3.2: Tilstander med referanse



Figur 3.3: Iterasjoner av QP-problemet (øverste figur) og gjennomsnittlig antall iterasjoner i linjesøket (nederste figur)

Kapittel 4

Tilstandsestimering

Dette kapitlet vil ta for seg tilstandsestimering for ulineære systemer. Denne tilstandsestimeringen er ment for å kombineres med NMPC.

Vanligvis er det kun utgangen på systemet som måles, og ikke de faktiske tilstandene. For å kunne stabilisere systemet med NMPC er det ønskelig å ha tilgang til alle ustabile tilstander. Dette kan gjøres ved å bruke metoder for tilstandsestimering.

4.1 Kalmanfilter

Kalmanfilteret er et filter som estimerer tilstandene til et dynamisk system fra en serie målinger påvirket av støy [11]. Kalmanfilter baserer seg på såkalt rekursiv estimering. Dette er en filtreringsmetode hvor bare forrige måling og tilstandsestimat er nødvendig for å regne ut neste estimat. Det opprinnelige kalmanfilteret er begrenset til lineære systemer. De fleste praktiske systemer er derimot ulineære. Ulineariteten kan være assosiert med enten prosessmodellen eller med observasjonsmodellen eller ved begge to.

4.2 Ulineær estimering

Siden kalmanfilteret ble presentert i 1960 årene har standarden for ulineær estimering vært å linearisere ulineariteten i systemet. Lineariseringen gjøres vanligvis på to forskjellige måter. Den kan gjøres omkring en nominell trajektorie i tilstandsrommet som ikke avhenger av målingene, og denne metoden kalles linearisert kalmanfilter [1]. Alternativet er å linearisere omkring kalmanfilterets estimerte trajektorie. Den siste metoden går under betegnelsen Extended Kalmanfilter (EKF).

4.2.1 Extended Kalmanfilter

EKF er sannsynligvis den mest brukte estimatoren for ulineære tilstandsrom systemer. EKF bruker kalmanfilter på ulineære systemer ved å linearisere alle ulineære modeller slik at de tradisjonelle lineære kalmanfilter-ligningene kan brukes. Lineariseringen i

EKF baserer seg på en første ordens tilnærming av en Taylorrekke. Tilstandsfordelingen tilnærmes med Gaussiske tilfeldige variabler som forsinkes analytisk gjennom den første ordens lineariseringen av systemet. Disse tilnærmingene kan medføre store feil i middelverdien og kovariansen til de transformerte Gaussiske tilfeldige variablene. Dette gjelder spesielt når modellen er veldig ulineær. Dette kan videre medføre suboptimal ytelse og i verste fall divergens av filteret.

Ytelsen til EKF kan bli betraktelig dårligere når forutsetningen om lokal linearitet om lineariseringspunktet ikke er gjeldende. Dette er aktuelt når høyere ordens dynamikk til systemet blir signifikant. Det har også vist seg at EKF er vanskelig å implementere og tune [10].

I de senere årene har det blitt presentert en ny type filter som kalles Unscented Kalmanfilter (UKF). Dette filteret har vanligvis vist seg å være en forbedring av EKF for ulineære systemer og vil presenteres i neste avsnitt. Et eksempel på middelverdi og kovarians i EKF og UKF er vist i Figur 4.2.

4.3 Unscented Kalmanfilter

Unscented kalmanfilter (UKF) er en ulineær estimator som gir samme ytelse som kalmanfilteret for lineære system, og i tillegg kan brukes på ulineære system uten å ta i bruk lineariseringsmetoder som brukes i EKF. UKF baserer seg på at det er lettere å tilnærme en sannsynlighetsfordeling enn det er å tilnærme en vilkårlig ulineær funksjon eller transformasjon [8].

En mengde med punkt, sigma punkt, velges deterministisk slik at de får bestemte egenskaper (har en gitt middelverdi og kovarians). Dette medfører at høyere-ordens informasjon om fordelingen kan fanges med et fast lavt antall med punkt [10]. Den ulineære funksjonen brukes på hvert enkelt punkt for å gi en mengde med transformerte punkt. Statistikken til disse transformerte punktene kan deretter regnes ut for å finne et estimat av ulineær transformert middelverdi og kovarians. En illustrasjon av unscented transformasjon er vist i Figur 4.1.

Unscented transformasjon

Først velges en mengde på $2L + 1$ vektete sample-punkt, såkalte sigma-punkt, som velges deterministisk slik at de representerer den faktiske middelverdien \bar{x} og kovariansen P_x til variabelen x . Dette gjøres ved å velge følgende punkt [9]:

$$\mathcal{X}_0 = \bar{x} \tag{4.1}$$

$$\mathcal{X}_i = \bar{x} + ((\sqrt{L + \lambda})P_x)_i \quad i = 1, \dots, L \tag{4.2}$$

$$\mathcal{X}_i = \bar{x} - ((\sqrt{L + \lambda})P_x)_i \quad i = L + 1, \dots, 2L \tag{4.3}$$

her er L størrelsen til tilstandsvektor x . Vektene for hvert enkelt sigma punkt:

$$W_0^{(m)} = \frac{\lambda}{L + \lambda} \quad i = 0 \quad (4.4)$$

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \quad i = 0 \quad (4.5)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L + \lambda)} \quad i = 1, \dots, 2L \quad (4.6)$$

Her settes λ til

$$\lambda = \alpha^2(L + \kappa) - L \quad (4.7)$$

Stegene i den skalerte unscented transformasjonen blir dermed [15]

1. Velg parametere α , β og κ . Velg $0 \leq \alpha \leq 1$ for å skalere spredningen på sigma-punktene. Denne bør velges til et lite tall for å unngå å sample resultat som ikke er lokale, dette gjelder spesielt når ulineariteten er stor. $\beta \geq 0$ er en vekt som kan brukes for å ta med kunnskap om høyere ordens moment av fordelingen. Normalt velges $\beta = 2$ for gaussisk fordeling [9]. Velg $\kappa \geq 0$ for å garantere at kovariansmatrisen blir positiv semidefinit.
2. Regn ut de skalerte sigma-punkt for $i = 0, \dots, 2L$ og de respektive vektene ved å bruke (4.1)-(4.7).
3. Finn de transformerte sigma-punktene ved å overføre skalerte sigma-punkt gjennom den ulineære modellen.

$$\mathcal{Y}_i = f(\mathcal{X}_i) \quad i = 0, \dots, 2L \quad (4.8)$$

4. Statistikken til \mathcal{Y} finnes deretter ved å bruke statistikk på de transformerte punktene. Dette gir middellverdi \bar{y} , kovarians P_y og krysskovarians P_{xy} .

Algoritme skalert UKF

Algoritmen baserer seg på den ulineære modellen

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \quad (4.9)$$

$$y_k = h(x_k, u_k) + v_k \quad (4.10)$$

her går vi ut fra at prosessstøy og målestøy kan beskrives som additiv støy.

Algoritmen for skalert UKF hvor støyen er antatt som "additiv" er som følger [15]:

- Initialisering

$$\hat{x}_0 = E[x_0] \quad (4.11)$$

$$P_{x_0} = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \quad (4.12)$$

$$\bar{w} = E[w] \quad (4.13)$$

$$\bar{v} = E[v] \quad (4.14)$$

$$R_w = E[(w - \bar{w})(w - \bar{w})^T] \quad (4.15)$$

$$R_v = E[(v - \bar{v})(v - \bar{v})^T] \quad (4.16)$$

- For hvert tidssteg $k = 1, 2, \dots$

1. Regn ut sigmapunkt og vektorer som i (4.1)-(4.7)

$$\mathcal{X}_{k-1} = \left[\hat{x}_{k-1} \quad \hat{x}_{k-1} + \gamma\sqrt{P_{k-1}} \quad \hat{x}_{k-1} - \gamma\sqrt{P_{k-1}} \right] \quad (4.17)$$

2. Tidsoppdatering av ligningene

$$\mathcal{X}_{k|k-1}^* = f(\mathcal{X}_{k-1}, u_{k-1}) \quad (4.18)$$

$$\hat{x}_k^- = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{X}_{i,k|k-1}^* \quad (4.19)$$

$$P_{x_k}^- = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{X}_{i,k|k-1}^* - \hat{x}_k^-)(\mathcal{X}_{i,k|k-1}^* - \hat{x}_k^-)^T + R_w \quad (4.20)$$

3. Finner sigmapunkt for måle-oppdatering

$$\mathcal{X}_{k|k-1} = \left[\hat{x}_k^- \quad \hat{x}_k^- + \gamma\sqrt{P_{x_k}^-} \quad \hat{x}_k^- - \gamma\sqrt{P_{x_k}^-} \right] \quad (4.21)$$

4. Oppdaterer estimat med målinger

$$\mathcal{Y}_{k|k-1} = h(\mathcal{X}_{k|k-1}, u_k) \quad (4.22)$$

$$\hat{y}_k^- = \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Y}_{i,k|k-1} \quad (4.23)$$

$$P_{y_k} = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{Y}_{i,k|k-1} - \hat{y}_k^-)(\mathcal{Y}_{i,k|k-1} - \hat{y}_k^-)^T + R_v \quad (4.24)$$

$$P_{x_k y_k} = \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{X}_{i,k|k-1} - \hat{x}_k^-)(\mathcal{Y}_{i,k|k-1} - \hat{y}_k^-)^T \quad (4.25)$$

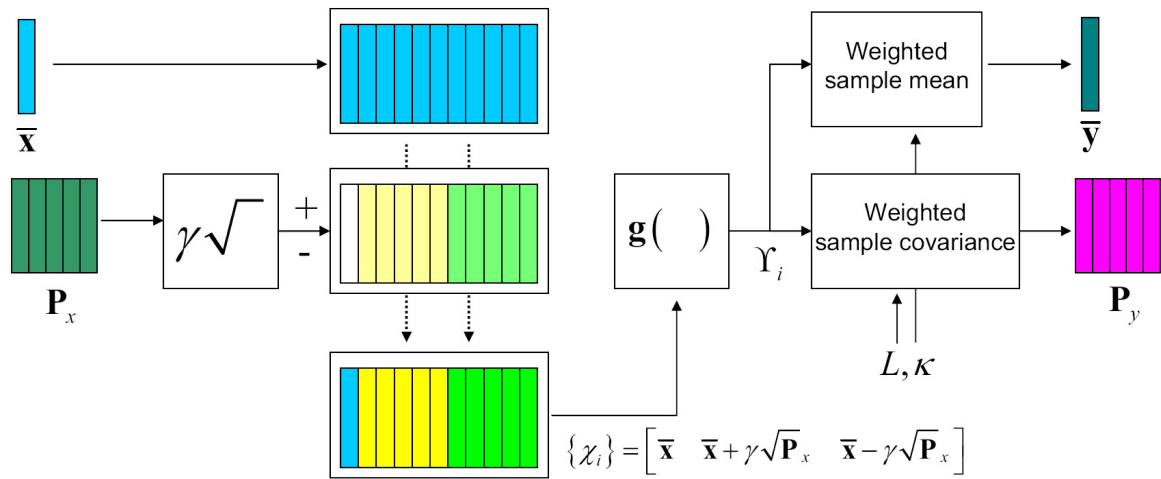
$$K_k = P_{x_k y_k} P_{y_k}^{-1} \quad (4.26)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - \hat{y}_k^-) \quad (4.27)$$

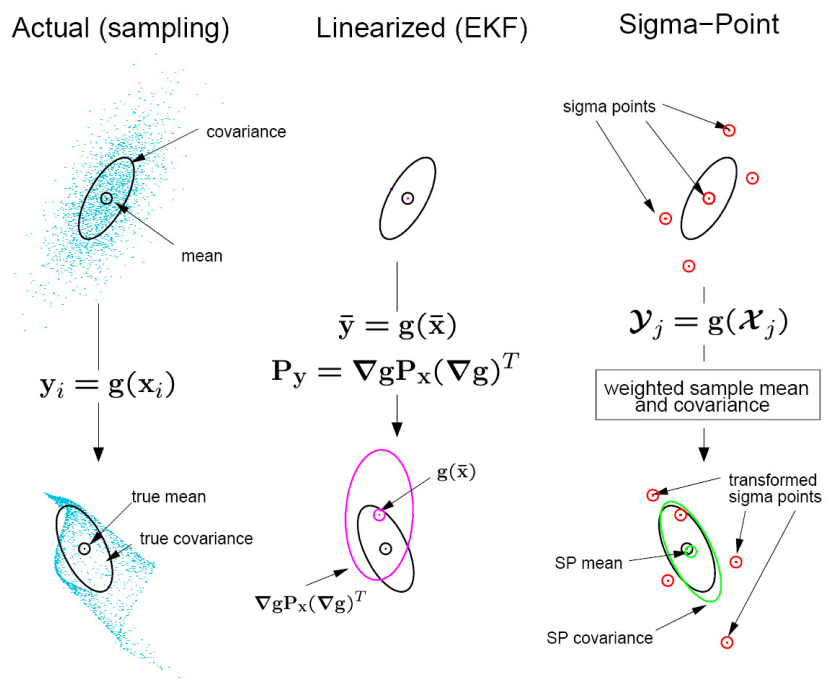
$$P_{x_k} = P_{x_k}^- - K_k P_{y_k} K_k^T \quad (4.28)$$

- Parametre

γ i (4.17) en sammensatt skaleringsfaktor (4.1) og er gitt av $\gamma = \sqrt{L + \lambda}$. R_w er kovarians til prosess-støy og R_v er kovarians til måle-støy. \bar{w} og \bar{v} er lik nullvektor dersom støyen antas å være Gaussisk fordelt.



Figur 4.1: Unscented transformasjon [15].



Figur 4.2: EKF vs UKF. Illustrasjon av nøyaktighet til middelværdi og kovarians forsinket gjennom det ulineære systemet [15].

Kapittel 5

Riser-indusert slugging

5.1 Slug-strømning

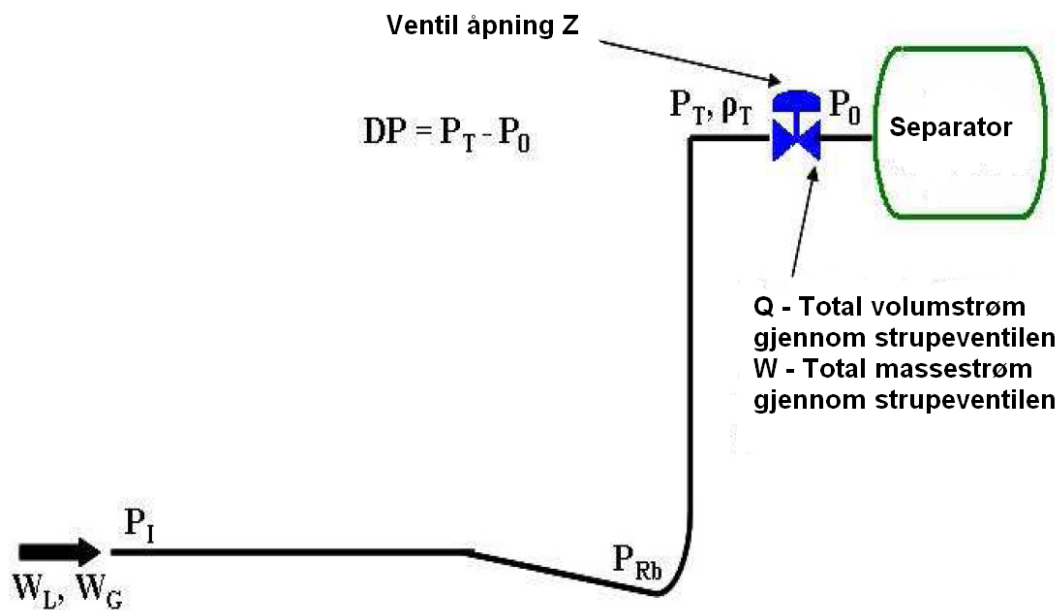
Olje produsert offshore transporteres gjennom rørledninger som en kompleks blanding av olje, vann, gass og sand. Et vanlig strømmønster er kjent som slug-strømning. Slug-strømning er et strømmønster i flerfase rørledninger som karakteriseres ved at væsken strømmer støtvis langs røret i en konsentrert masse.

Slug-strømning kan deles inn i fire forskjellige typer [25]: hydrodynamisk slugging, terrengeindusert slugging, transient slugging og riser-indusert slugging. Hydrodynamisk slugging forårsakes av hastighetsforskjeller mellom fasene og kan oppstå i nesten horisontale rørledninger. Disse sluggene er vanligvis ganske korte og opptrer ofte. Terrengeindusert slugging forårsakes av lave punkt i geometrien til rørledningen. Denne typen slugging kan inneholde mye væske og derfor forårsake store trykkvariasjoner i systemet. Transient slugging forårsakes av forandringer i operasjonsbetingelser.

Fokuset i denne rapporten vil være på riser-indusert slugging. Riser-indusert slugging er slugger som oppstår i stigerøret fra havbunnen opp til seperatoren, som vist i Figur 5.1. Slugger i stigerøret forårsakes av væskeansamling i bunnen av stigerøret. Væskeansamlingen blokkerer rørledningen og vil fortsette å vokse inntil differansetrykket over sluggen blir høyt nok til å blåse sluggen ut av stigerøret. Denne typen, muligens kombinert med terrengeindusert slugging, er den mest alvorlige for olje/vann-baserte system.

Utvikling av slugging i flerfase rørlinjer er et viktig tema for oljeprodusentene. Den uregelmessige strømmingen resulterer i dårlig olje/vann seperasjon, begrenser produksjonskapasiteten og forårsaker faking. Mye penger har derfor blitt brukt for å unngå problemer assosiert med slugg-strømning, for eksempel ved å installere buffere eller å øke volumet på inngangsseparatoren for å gi tilstrekkelig buffer kapasitet. Dette er ofte dyre og ikke tilstrekkelig gode løsninger.

En alternativ løsning er å detektere oppbygningen av slugger og deretter bruke choke-



Figur 5.1: Stigerør med målinger

ventilen (strupeventilen) til rørledningen for å redusere slug-strømning. Reduseres ventilaåpningen under en bestemt grense, og holdes konstant der, vil slugging unngås [25]. Hovedproblemet med denne tilnærmingen er at struper vi ventilen for mye vil dette medføre høyere trykk i rørledningen og redusert oljeproduksjon.

Det kan derfor være en fordel å bruke reguleringsløyfer med tilbakekobling for å hindre slugging ved å regulere ventilaåpningen. Det er da mulig å stabilisere trykket i rørledningen selv for store ventilaåpninger. I [25] er det vist at ved å bruke målinger i Figur 5.1, kan slug-strømning stabiliseres ved aktiv bruk av ventilaåpningen.

Trykkmåling i bunnen av stigerøret (P_{Rb}) og trykkmåling oppstrøms stigerøret (P_I) krever subsea instrumentering, noe som er meget kostbart. Det vil derfor være mer praktisk om målinger topside kan brukes. Målinger som er tilgjengelig topside er vist i Figur 5.1. Det er differansetrykk over ventilaåpning DP , volumstrøm Q , massestrøm W og tetthet ρ_T oppstrøm strupeventil.

5.2 Storakaas modell for modellering av slugging

Simuleringer i denne rapporten er gjort med utgangspunkt i Storakaas dynamiske modell av riser-indusert slugging. Modellen er en forenklet ulineær dynamisk modell og har vist seg å gi en god representasjon av hvordan riser-indusert slugging opptrer i praksis.

Modellen forutsetter at systemet kan beskrives ved å koble sammen tre balanse volum. Massen til væsken i stigerøret og i bunnen av stigerøret modelleres med et enkelt volum. Gassvolumet i tilførselsrøret antas konstant, men med varierende masse til gassen. To dynamiske tilstander er tatt med for henholdsvis massen til gassen i stigerøret og i tilførselsrøret. Dette gir modellen tre dynamiske tilstander:

- Masse til væsken m_L i stigerøret og i bunnen av stigerøret
- Masse til gassen m_{G1} i tilførselsrøret
- Masse til gassen m_{G2} i stigerøret

Differensial ligningene til modellens tilstander:

$$\frac{d}{dt}m_L = w_{L,inn} - w_{L,ut} \quad (5.1)$$

$$\frac{d}{dt}m_{G1} = w_{G,inn} - w_{G1} \quad (5.2)$$

$$\frac{d}{dt}m_{G2} = w_{G1} - w_{G,ut} \quad (5.3)$$

Her er $w_{L,inn}$ og $w_{G,inn}$ henholdsvis masse-rate [kg/s] av væske inn i systemet og masserate av gass inn i systemet. $w_{L,ut}$ og $w_{G,ut}$ er masserate av væske og gass ut av systemet, og w_{G1} er masserate av gass gjennom tilførselsrøret. Nærmere detaljer om modellen kan finnes i Appendiks A.

Kapittel 6

Stabilisering av slug-strømning

I dette kapittelet er det tatt med simuleringer av tilstandsestimering kombinert med NMPC med formålet å stabilisere oljestrømmen inn til separator. Det er sett på stabilisering av slug-strømning for det nominelle systemet omkring en ventilåpning på $z = 0.2$.

6.1 Oppsett

Fødestrømmen inn til systemet er nominelt konstant med $w_{L,inn} = 8.64kg/s$ (olje) og $w_{G,inn} = 0.362kg/s$ (gass). Trykket etter strupeventilen (P_0) er nominelt konstant på 50 bar. Dermed er ventilåpning Z den eneste frihetsgraden i systemet. Tilførselen av væske $W_{L,inn}$ og gass $W_{G,inn}$ ansees som prosessforstyrrelser.

6.2 Tilstandsestimering

I praksis måles kun utgangene til systemet og ikke de faktiske tilstandene. For å få tilgang til tilstandene i systemet (5.1) - (5.3) må det brukes en form for tilstandsestimering. Her er det testet med Unscented kalmanfilter og Extended kalmanfilter. Det ble brukt målepunkter $DP[Bar]$ og $Q[m^3/s]$ for tilstandsestimeringen.

Fødestrømmen i tilførselsrøret vil normalt variere til en viss grad. Av den grunn er fødestrømmen av væske $w_{L,inn}$ og gass $w_{G,inn}$ tatt med i estimeringen. Dermed kan endringer i fødestrømmen detekteres av modellen som brukes i MMPC algoritmen og tilstandsestimeringen. Dette medfører bedre prediksjon i NMPC og bedre tilstandsestimat. Systemet vil dermed bli mer robust for prosessforstyrrelser.

UKF

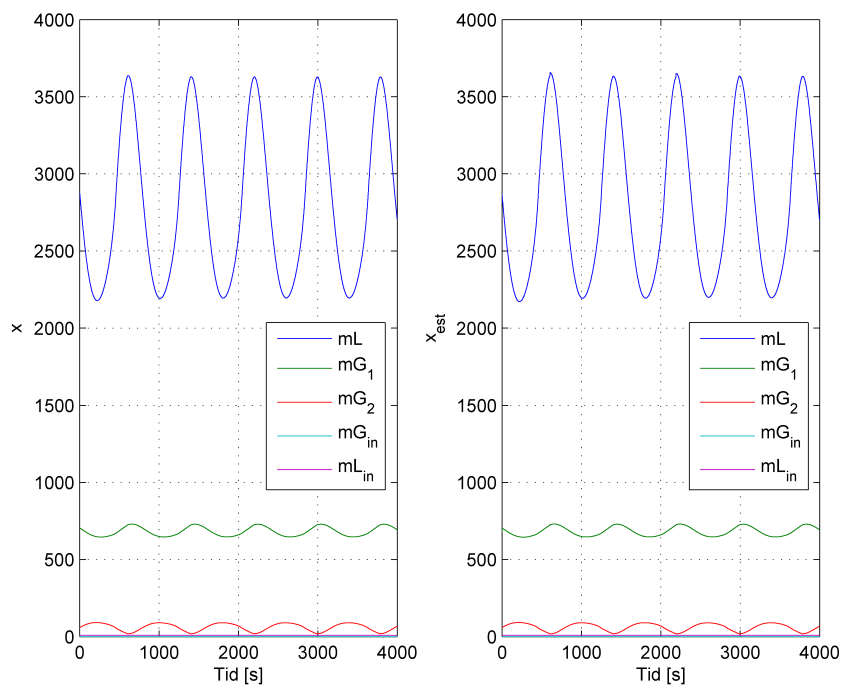
Kovariansmatrisen til tilstander i UKF ble satt til

$$R_w = \begin{bmatrix} 4e - 6 \cdot x_1 & 0 & 0 & 0 & 0 \\ 0 & 4e - 8 \cdot x_2 & 0 & 0 & 0 \\ 0 & 0 & 4e - 8 \cdot x_3 & 0 & 0 \\ 0 & 0 & 0 & 4e - 11 \cdot d_1 & 0 \\ 0 & 0 & 0 & 0 & 4e - 11 \cdot d_2 \end{bmatrix} \quad (6.1)$$

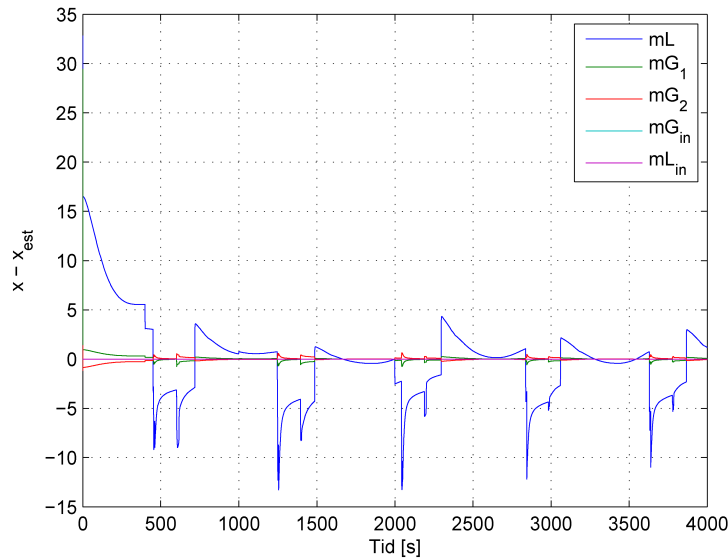
her er x_1, x_2 og x_3 systemets tilstander, henholdsvis m_L , m_{G1} og m_{G2} . Mens d_1 og d_2 er nominell fødestrøm. Kovariansmatrisen til målestøy er gitt av

$$R_v = \begin{bmatrix} 1e - 7 & 0 \\ 0 & 1e - 3 \end{bmatrix} \quad (6.2)$$

her er første element kovarians til målepunkt DP og siste element er kovarians til målepunkt Q . I Figur 6.1 vises systemets faktiske tilstander og tilstander estimert av UKF med målepunkter DP og Q og i Figur 6.2 vises estimeringsfeil med samme målepunkter.



Figur 6.1: Faktiske tilstander (til venstre) og estimerte tilstander (til høyre). Det er brukt UKF med målepunkter DP og Q



Figur 6.2: Estimeringsfeil. Det er brukt UKF med målepunkter DP og Q

EKF

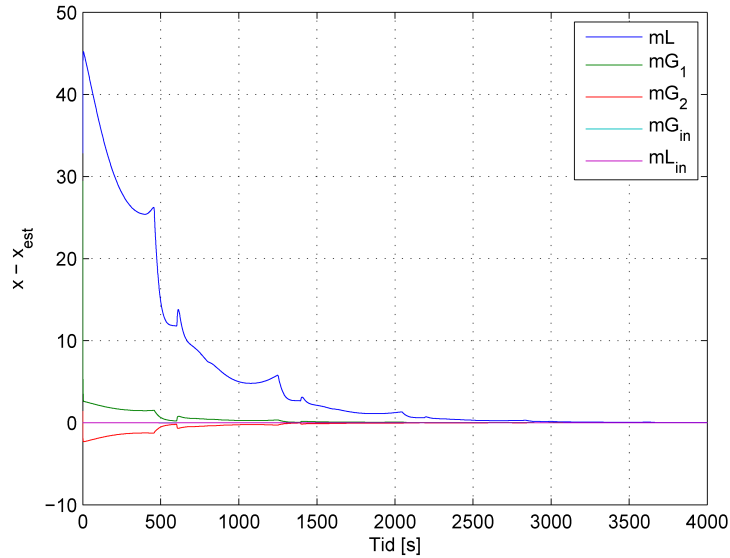
Normalt avhenger filter forsterkningen K_f av tilstandsestimatet \hat{x} , men her er det brukt en konstant forsterkning, utregnet fra en linearisert modell om stasjonær tilstand. Dermed er det ingen behov for kontinuerlig linearisering, som det vanligvis er for EKF. Denne tilnærmingen ble brukt i [25] og [23] med gode resultater. I Figur 6.3 er det vist estimeringsfeil med målepunkt DP for EKF.

Kommentar til tilstandsestimering

For det nominelle systemet ga en modifisert EKF generelt mer stabil estimering av tilstandene enn UKF. I UKF avhenger filter forsterkningen av tilstandsestimatet mens i den modifiserte EKF er det brukt en konstant filter forsterkningen. Under estimeringen av den ulineære modellen skjer det med jevne tidsrom sprang i estimert utgang i UKF. Dette medfører at filter forsterkningen i UKF endrer seg raskt noe som medfører sprang i estimeringsfeil til estimerte tilstander, som vist i Figur 6.2. Den dynamiske tilpasningen til filter forsterkningen i UKF vil derimot medføre at det kan forventes at UKF vil estimere tilstandene bedre når prosessen er påvirket av prosess-støy.

6.3 NMPC på Storkaas modell

Ved linearisering av Storkaas modell omkring forskjellige operasjonspunkt kan det vises at egenverdiene til Jacobian er relativt spredt, noe som gir et stivt problem. For å unngå diskretiseringsfeil og numerisk ustabilitet er det derfor valgt å diskretisere med



Figur 6.3: Estimeringsfeil. Det er brukt EKF med målepunkt DP

den implisitte metoden ”Tustin tilnærming” i (3.26). I NMPC-algoritmen er systemet integrert med en 4. ordens Runge Kutta og denne metoden viste seg å fungere bra i praksis.

Vektene i NMPC-algoritmen for tilstander og pådrag er satt til:

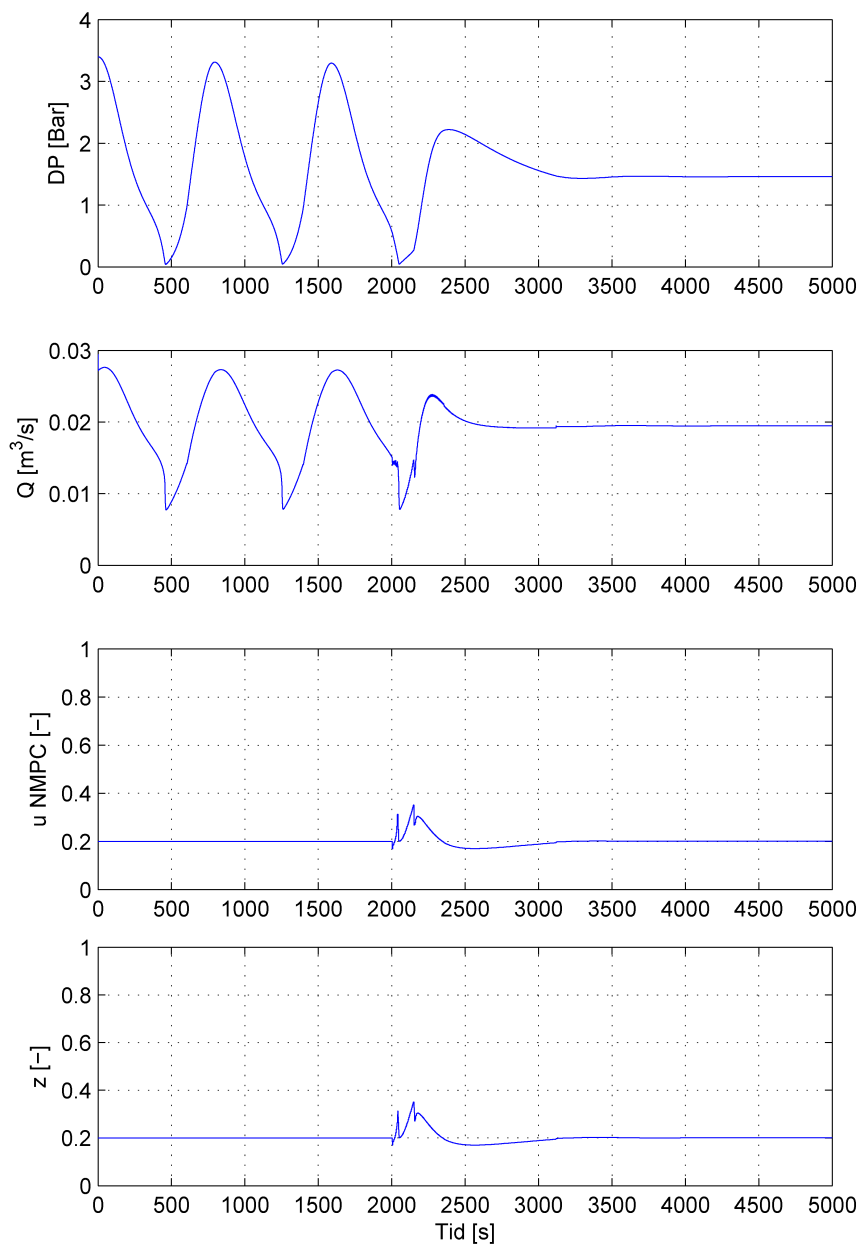
$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = [55e6] \quad (6.3)$$

Det er valgt å sette ut pådrag fra NMPC-algoritmen hvert 4. sekund, dette kan rettfærdiggjøres utfra den trege dynamikken til strupe-ventilen. Sampling tiden i NMPC-algoritmen er satt til $T_s = 0.1s$, dermed vil en reguleringshorisont på $H_u = 100$ prediktere modellens oppførsel 10s frem i tid.

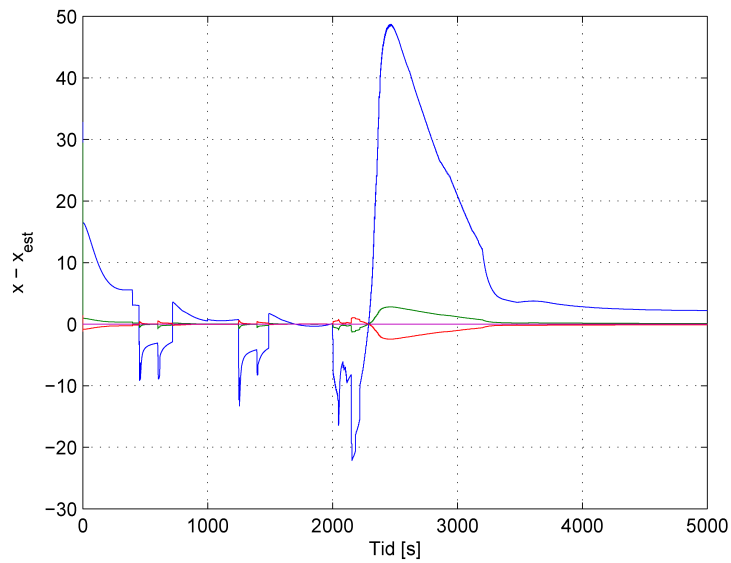
6.3.1 Simulering

Det er simulert for to forskjellige tilfeller. I det første tilfellet er reguleringshorisonten satt til $H_u = 10$, her er det brukt UKF med målepunkter DP og Q. I det andre tilfellet er reguleringshorisonten $H_u = 100$, her er det brukt EKF med målepunkt DP. I begge tilfellene holdes ventilåpningen konstant på $z = 0.2$ de første 2000 sekunder før NMPC aktiveres.

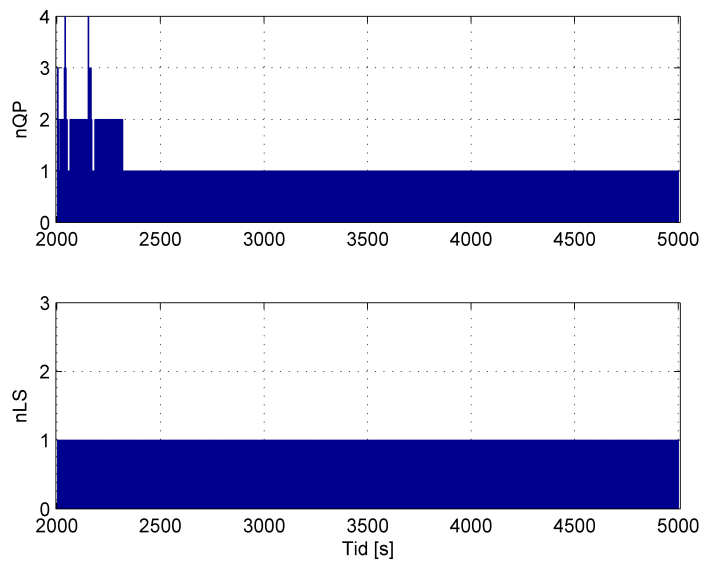
For det første tilfellet, $H_u = 10$, ble konvergenzkriteriet i NMPC-algoritmen satt til $\Delta u_{min} = 0.01$, mens for det andre tilfellet ble det brukt $\Delta u_{min} = 0.1$ for å unngå for mange iterasjoner av QP-problemet.



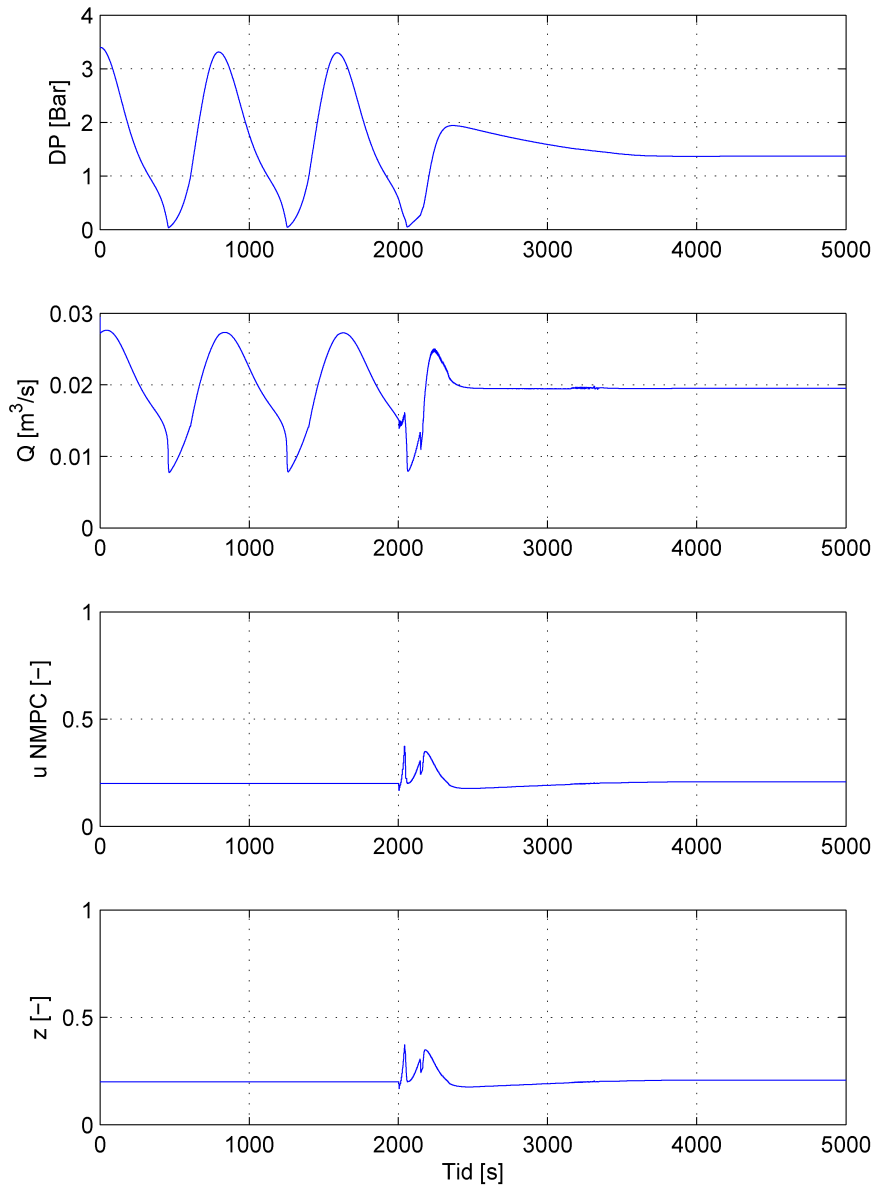
Figur 6.4: Stabilisering om ventilåpning $z = 0.2$. Reguleringshorisont $H_u = 10$. Det er brukt UKF med målepunkter DP og Q. De to nederste figurene er pådrag ut av NMPC ($uNMPC$) og ventilåpning z .



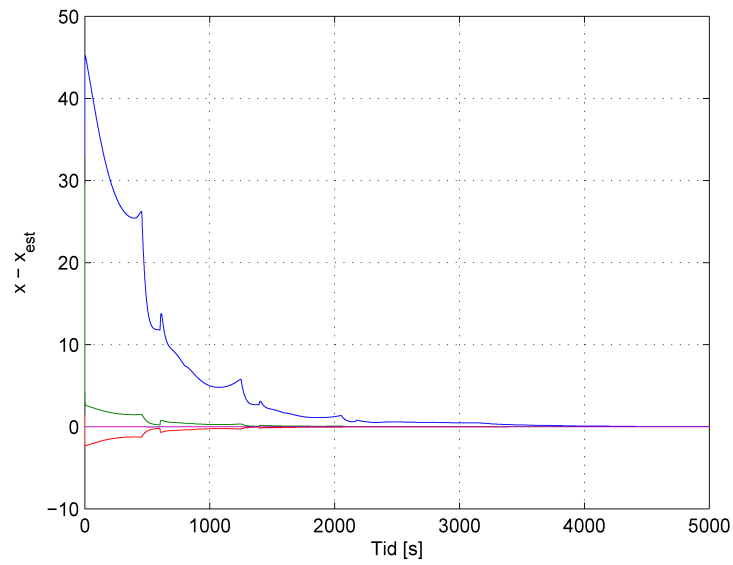
Figur 6.5: Estimeringsfeil med målepunkter DP og Q. Reguleringshorisont $H_u = 10$.



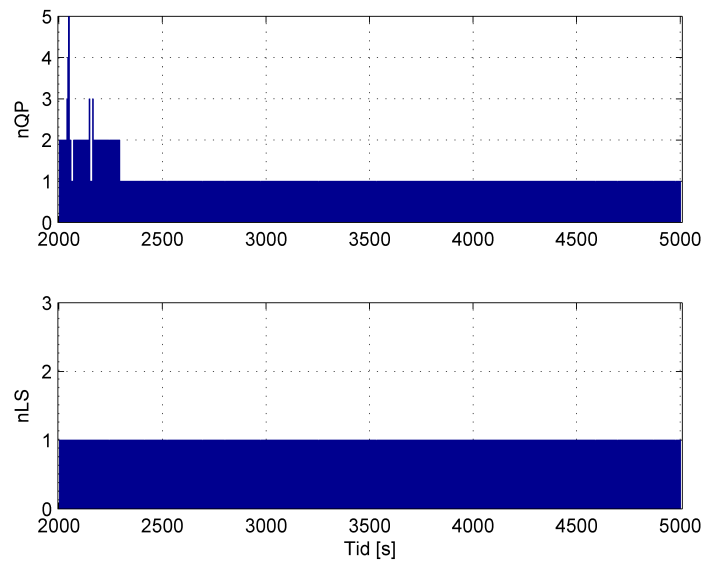
Figur 6.6: Iterasjoner av QP-problemet (øverste figur) og gjennomsnittlig antall iterasjoner for linjesøket (nederste figur) for NMPC algoritmen. Reguleringshorisont $H_u = 10$. Det er brukt UKF med målepunkter DP og Q



Figur 6.7: Stabilisering om ventilåpning $z = 0.2$. Reguleringshorisont $H_u = 100$. Det er brukt EKF med målepunkt DP. De to nederste figurene er pådrag ut av NMPC ($uNMPC$) og ventilåpning z .



Figur 6.8: Estimeringsfeil med målepunkt DP. Reguleringshorisont $H_u = 10$.



Figur 6.9: Iterasjoner av QP-problemet (øverste figur) og gjennomsnittlig antall iterasjoner for linjesøket (nederste figur) for NMPC algoritmen. Reguleringshorisont $H_u = 100$. Det er brukt EKF med målepunkt DP

Kapittel 7

Konklusjon

I denne rapporten har det blitt utviklet og implementert en NMPC-algoritme basert på en "single shooting" optimaliseringsmetode med prestabilisering. Det har også blitt implementert UKF og EKF for tilstandsesimering.

For det nominelle systemet ga en modifisert EKF generelt mer stabil estimering av tilstandene enn UKF. Derimot kan det forventes at UKF vil estimere tilstandene bedre ved påvirkning av prosess-støy.

NMPC kombinert med ulineær tilstandsestimering ble brukt for å styre pådrag til strupeventil optimalt. Det ble brukt en forenklet modell av riser-indusert slugging til prosess simulering, prediksjon i NMPC og modell for tilstandsestimering. Slug-strømning ble stabilisert for en bestemt ventilåpning ved å bruke NMPC og tilstandsestimering på den forenklete modellen.

Kapittel 8

Videre arbeid

8.1 Ulineær MPC

Det er mer komplisert å gjøre settpunktsendringer ved å bruke tilstander som referanse enn det er ved å bruke målinger. Slik den ulineære MPC formuleringen er oppbygd nå er C-matrisen satt som identitetsmatrise. Dette medfører at tilstandene i systemet reguleres utfra referansen til tilstandene. For Storakaas modell, beskrevet i Tillegg A, er målingene tilgjengelig gjennom den ulineære funksjonen $y = h(x, u)$, hvor x er systemets tilstander og u er pådraget. Predikterte målinger kan derfor finnes ved å bruke den ulineære funksjonen på de predikterte tilstandene.

Integralvirkning

I praksis vil målinger være utsatt for målestøy, og faktisk utgang kan skrives som $y = h(x, u) + w$. For å unngå stasjonæravvik forårsaket av målestøy og modellfeil bør det implementeres en form for integralvirkning i NMPC-algoritmen. En måling som ser lovende ut for å brukes i integralvirkning er differansetrykk DP over strupeventil.

Valg av K-matrise for prestabilisering

Dersom det velges av konstant K-matrise er det ikke sikkert at denne klarer å stabilisere systemet for den delen av tilstandsrommet systemet vil operere i. Dersom det velges å bruke en dynamisk K-matrise er et alternativ å bruke en LQ regulator. For å finne denne må en diskrete algebraisk Ricatti ligning (DARE) løses. Dette er imidlertid regnekrevende ettersom Ricatti ligningen må løses for hvert tidssteg over reguleringshorisonten. Dersom reguleringshorisonten $Hu = 50$ medfører dette at Ricattiligningen utregnes 50 ganger.

Valg av NMPC formulering

Et mulig alternativ til å bruke prestabilisering med tilstandstilbakekobling kan være å optimalisere systemet direkte. Da bør det brukes en "simultan" dynamisk optimaliserings-

metode (og ikke en sekvensiell, som kanskje er lettest å implementere). Det bør da brukes en "terminal state weight" i NMPC-formuleringen. Helst basert på "quasi-infinite horizon", eller lignende.

8.2 Tilstandsestimering

I tillegg til systemets tilstander estimerer UKF fødestrømmen i tilføreselsrøret. Store endringer i fødestrømmen kan medføre at UKF blir ustabil og at estimeringsfeilen divergerer. Det bør undersøkes hvordan UKF håndterer forskjellige endringer i fødestrømmen.

NMPC-algoritmen baserer seg på tilstandsestimatene fra kalmanfilteret. Det bør undersøkes hvor stor estimeringsfeilen kan bli før NMPC får problemer med å stabilisere systemet.

8.3 Modell

I denne rapporten er det brukt Storkaas modell av slugging i stigerøret som prosessmodell. Dette er en forenklet modell av det virkelige systemet. Modellen har få antall tilstander og egner seg derfor til å brukes som modell for tilstandsestimeringen og NMPC. For å få mer realistiske svar på hvordan kombinasjonen av tilstandsestimering og NMPC vil fungere i praksis bør det også testes å simulere med en OLGA-modell som prosessmodell.

Bibliografi

- [1] R.G. Brown og P.Y.C. Hwang *Introduction to random signals and applied Kalman filtering*, 3rd ed., John Wiley and Sons, 1997
- [2] C.R. Cutler og B.L. Ramaker *Dynamic matrix control- a computer control algorithm*, Proceedings of the Joint Automatic Control Conference, San Francisco, USA, 1980
- [3] M. Diehl *Efficient numerical solution methods for fast NMPC*, Workshop: Model predictive control for fast nonlinear systems: existing approaches, challenges, and applications. Before 45th CDC, San Diego, USA, 12. Desember, 2006
- [4] O. Egeland og J.T. Gravdahl *Modeling and Simulation for Automatic Control*, Marine Cybernetics, 2002
- [5] C.E. Garcia og A.M. Morshedi *Quadratic programming solution of dynamic matrix control*, Chemical Engineering Communication, 46:73-87, 1986
- [6] M. Hovd *Kompendium i Prosessregulering, Videregående kurs*, NTNU, 2006
- [7] L. Imsland *Introduction to Model Predictive Control*, NTNU, 2007
- [8] S.J. Julier og J.K. Uhlmann *A New Extension of the Kalman Filter to Nonlinear Systems*, In The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II, SPIE, 1997
- [9] S.J. Julier *The Scaled Unscented Transformation*, Proceedings of the American Control Conference, 2002
- [10] S.J. Julier og J.K. Uhlmann *Unscented Filtering and Nonlinear Estimation*, In Proceedings of The IEEE, Vol.92, NO.3, 2004
- [11] R.E. Kalman *A New Approach to Linear Filtering and Prediction Problems*, Transactions of the ASME-Journal of Basic Engineering, 82:34-35, 1960
- [12] W.C. Li og L.T. Biegler *Multistep, Newton-type control strategies for constrained, nonlinear processes*, Chem. Eng. Res. Des., 67:562-577, 1989
- [13] J.M. Maciejowski. *Predictive Control with constraints*, Prentice Hall, 2002

- [14] F. Martinsen, L.T. Biegler, B.A. Foss. *A new optimization algorithm with application to nonlinear MPC*, Journal of Process Control, 14(8):853-865, 2004
- [15] R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*, Phd thesis, OGI School of Science and Engineering at Oregon Health and Science University, 2004
- [16] M. Morari og J.H. Lee *Model predictive control: past, present and future*, Computers and Chemical Engineering, 23:667-682, 2003
- [17] K.R. Muske og J.B. Rawlings *Model predictive control with linear models*, AIChE J., 39(2):262-287, 1993
- [18] J. Nocedal og S.J. Wright *Numerical Optimization*, 2nd ed., Springer, 2006
- [19] N.M.C. Oliveira og L.T. Biegler *An extension of Newton-type algorithms for nonlinear process control*, Automatica J. IFAC 31(2), 281-286, 1995
- [20] S. Joe Qin og T.A. Badgwell *A survey of industrial model predictive control technology*, Control Engineering Practice, 11(7):733-764, 2003
- [21] S.J. Qin og T.A. Badgwell *An overview of Nonlinear Model Predictive Control Applications*, 2000
- [22] J. Richalet, A. Rault, J.L. Testud og J. Papon *Algorithmic control of industrial processes*, In Proceedings of the 4th IFAC symposium on identification and system parameter estimation, 119-1167, 1976
- [23] C.F. Trudvang. *Modellbasert stabiliserende regulering av gravitasjonsindusert slugging i pipeline-riser systemer*, Diploma Thesis, NTNU, 2003
- [24] S. Skogestad og I. Postlethwaite. *Multivariable Feedback Control*, 2nd ed., John Wiley & Sons Ltd, UK, 2005
- [25] E. Storkaas. *Stabilizing Control and Controllability: Control solutions to avoid slug flow in pipeline-riser systems*, PhD thesis, NTNU, 2005

Tillegg A

Modell av slugging i stigerør

Her er det tatt med en en forenklet modell av slugging i stigerør. Modellen er utledet av E.Storkås [25].

A.1 Antagelser i modellen

1. Konstant væsketetthet i føderøret (neglisjerer væskedynamikk i føderøret)
2. Konstant gassvolum V_{G1} i føderøret. Dette kommer av punkt 1 dersom vi også neglisjerer variasjoner i væskevolum p.g.a. varierende væskenhå h_1 i bunnpunkt av stigerøret
3. Kun et kontrollvolum for væsken (kontrollvolumet inkluderer stigerøret og en del av føderøret)
4. To kontrollvolum for gass, separert av bunnpunktet i stigerøret. De henger sammen gjennom trykk - strømning sammenheng
5. Ideell gasslov
6. Stajsonær trykkbalanse mellom stigerør og tilførselspunkt (mellom P_1 og P_2)
7. Forenklet ventilmodell for gass og væske som forlater stigerøret
8. Konstant temperatur

A.2 Modell ligninger

Massebalanser:

$$\frac{d}{dt}m_L = w_{L,inn} - w_{L,ut} \quad (\text{A.1})$$

$$\frac{d}{dt}m_{G1} = w_{G,inn} - w_{G1} \quad (\text{A.2})$$

$$\frac{d}{dt}m_{G2} = w_{G1} - w_{G,ut} \quad (\text{A.3})$$

Tilstandsavhengige interne ligninger

$$P_1 = \frac{m_{G1}RT}{V_{G1}M_G} \quad (\text{A.4})$$

$$\rho_{G1} = \frac{m_{G1}}{V_{G1}} \quad (\text{A.5})$$

$$V_L = \frac{m_L}{\rho_L} \quad (\text{A.6})$$

$$h_1A_1 + V_{LR} = V_L \quad (\text{A.7})$$

$$V_T = A_2(H_2 + L_3) \quad (\text{A.8})$$

$$V_{G2} = V_T - V_{LR} \quad (\text{A.9})$$

$$\rho_{G2} = \frac{m_{G2}}{V_{G2}} \quad (\text{A.10})$$

$$\alpha_L = \frac{V_{LR}}{V_T} \quad (\text{A.11})$$

$$P_2 = \frac{m_{G2}RT}{V_{G2}M_G} \quad (\text{A.12})$$

$$\bar{\rho} = \frac{m_{G2} + V_{LR}\rho_L}{V_T} \quad (\text{A.13})$$

$$\bar{\rho}g(H_2 + H_3) - \rho_Lgh_1 = P_1 - P_2 \quad (\text{A.14})$$

$$\alpha_{LT} = (V_{LR} > H_2A_2)\alpha_{LT}^* + \frac{w^n}{1 + w^n}(\alpha_L - (V_{LR} > H_2A_2)\alpha_{LT}^*) \quad (\text{A.15})$$

$$\alpha_{LT}^* = \frac{V_{LR} - A_2H_2}{A_2L_3} \quad (\text{A.16})$$

$$w = \frac{K_3 \rho_{G1} v_{G1}^2}{\rho_L - \rho_{G1}} \quad (\text{A.17})$$

$$\rho_T = \alpha_{LT} \rho_L + (1 - \alpha_{LT}) \rho_{G2} \quad (\text{A.18})$$

$$\alpha_L^m = \frac{\alpha_{LT} \rho_L}{\alpha_{LT} \rho_L + (1 - \alpha_{LT}) \rho_{G2}} \quad (\text{A.19})$$

Transport ligninger

$$v_{G1} = (h_1 < H_1) K_2 \frac{H_1 - h_1}{H_1} \sqrt{\frac{P_1 - P_2 - \rho_L g \alpha_L H_2}{\rho_{G1}}} \quad (\text{A.20})$$

$$w_{G1} = v_{G1} \rho_{G1} \hat{A} \quad (\text{A.21})$$

$$m_{mix,ut} = K_1 z \sqrt{\rho_T (P_2 - P_0)} \quad (\text{A.22})$$

$$w_{G,ut} = (1 - \alpha_L^m) m_{mix,ut} \quad (\text{A.23})$$

$$w_{L,ut} = \alpha_L^m m_{mix,ut} \quad (\text{A.24})$$

Geometriske ligninger

$$H_1 = \frac{2r}{\cos(\theta)} \quad (\text{A.25})$$

$$A_1 = \frac{A_2}{\sin(\theta)} \quad (\text{A.26})$$

$$\phi = \left(\pi - \arccos \left(1 - \frac{(H_1 - h_1) \cos(\theta)}{r} \right) \right) \quad (\text{A.27})$$

$$\hat{A} = r^2 (\pi - \phi - \cos(\pi - \phi) \sin(\pi - \phi)) \quad (\text{A.28})$$

A.3 Notasjon

<i>Symbol</i>	<i>Beskrivelse</i>	<i>Benevning</i>	<i>Forklaring</i>
m_{Gi}	Masse av gass i volum i	kg	Tilstands variabel
m_L	Masse av væske	kg	Tilstands variabel
V_{Gi}	Gassvolum i	m^3	$V_{G1} = \text{konst}$
V_L	Volum okkupert av væske	m^3	
V_{LR}	Volum av væske i stigerør	m^3	
V_T	Totalt volum i stigerør	m^3	
P_i	Trykk i volum i	N/m^2	
ρ_{Gi}	Gasstetthet i volum i	kg/m^3	
ρ_L	Væsketetthet	kg/m^3	Konstant
$\bar{\rho}$	Gjennomsnittlig tetthet i stigerør	kg/m^3	
ρ_T	Tetthet oppstrøms ventil	kg/m^3	
v_{G1}	Gasshastighet ved laveste punkt	m/s	
$v_{mix,ut}$	Væskeshastighet gjennom strupeventil	m/s	
w_{G1}	Massestrøm av gass i G1	kg/s	
$w_{G,ut}$	Massestrøm av gass gjennom strupeventil	kg/s	
$w_{L,ut}$	Massestrøm av væske gjennom strupeventil	kg/s	
α_L	Gjennomsnittlig væske fraksjon i stigerør, volum basis	-	
α_{LT}	Væske fraksjon oppstrøms ventil, volum basis	-	
α_L^m	Væske fraksjon oppstrøms ventil, masse basis	-	
α_{LT}^*	Væske fraksjon oppstrøms ventil uten inng.ligning	-	
h_1	Væskeniå oppstrøms bend	m	
H_1	Kritisk væskeniå	m	Konstant
H_2	Høyde til stigerør	m	Konstant
r	Radius til rør	m	Konstant
A_1	Areal i horisontalplan, V1	m^2	Konstant
A_2	Areal av tverrsnitt til stigerør, V2	m^2	Konstant
\hat{A}	Areal tilgjengelig for gass i bunn av stigerør	m^2	
L_3	Lengde av horisontalt rør i topp	m	Konstant
θ	Føderørets vinkel mot horisontalplan	rad	Konstant
R	Gass konstant (8314)	$J/(K * kmol)$	Konstant
g	Tyngdens akselerasjon (9.81)	m/s^2	Konstant
T	Systemets temperatur	K	Konstant
M_G	Molekylvekt til gassen	$kg/kmol$	Konstant
$w_{G,inn}$	Masse til gassen tilført til systemet	kg/s	Forstyrrelse
$w_{L,inn}$	Masse til væsken tilført til systemet	kg/s	Forstyrrelse
P_0	Trykk etter strupeventil	N/m^2	Forstyrrelse
z	Ventilens posisjon	-	Inngang
K_1	Strupeventil konstant	-	Tuningsparameter
K_2	Gasstrømning konstant	-	Tuningsparameter
K_3	Friksjons parameter	-	Tuningsparameter
n	w^n i friksjonsuttrykket	-	Tuningsparameter