



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Master i Teknologi Kybernetikk/Signalbehandling	Vårsemesteret, 2010 Åpen / Konfidensiell
Forfatter: Ben Ove Landa (signatur forfatter)
Fagansvarlig: Tormod Drengstig Veileder(e): Tormod Drengstig	
Tittel på masteroppgaven: Utvikling og anvendelse av MPC-regulatorer på varmluftsvifte Engelsk tittel: Development and application of MPC Controllers for hot-air fan	
Studiepoeng: 30	
Emneord: MPC PI-regulator Foroverkobling Varmluftsvifte	Sidetall: 83 + vedlegg/annet: 1 Stavanger, 15/06-2010 dato/år

Sammendrag

Denne rapporten dokumenterer arbeidet forbundet med å utvikle MPC-regulatorer for varmluftsvifte. To MPC-regulatorer er utviklet for varmluftsviften. Den ene benytter informasjon om fremtidige endringer i referanse og forstyrrelse, i reguleringsystemet. Den andre MPC-regulatoren har ikke denne egenskapen.

MPC-regulatoren for regulering av viftehastigheten i varmluftsvifte er utviklet ved hjelp av MPC-verktøykassen i Matlab. Viftehastigheten er modellert ved sprang rundt valgt arbeidspunkt i både pådrag til viftemotor og spjeld. Videre er modellen verifisert mot virkelig prosess, og implementert i MPC-regulatoren.

Effekten av ulike parametervalg, samt kunnskap om fremtidig referanse- og forstyrrelsesforløp, forbundet med MPC-regulatoren er nøye testet. For å vurdere ytelsen til MPC-regulatoren, er det utviklet en sammenlignbar PI-regulator med foroverkobling. Reguleringsparameterne for PI-regulator ble bestemt med utgangspunkt i responstid og oversving fra MPC-regulering uten forhåndsinformasjon. Som mål for ytelse ble IAE-verdier (*Integral of Absolute Error*) og TV-verdier (*Total Variation*) brukt.

MPC-regulator uten forhåndsinformasjon ga i anvendelser med sinusreferanse bedre resultater enn ved PI-regulator med foroverkobling, men dårligere resultater enn PI-regulator med foroverkobling ved ulike sprang i referanse. MPC-regulator med forhåndsinformasjon benyttet denne informasjonen til å øke eller minke pådrag til viftemotor før referanseendringer ble utført for viftehastighet. Dette medførte at denne regulatoren ga klart best resultater. Ved kompensering av forstyrrelse ga PI-regulator med foroverkobling bedre resultater, enn for MPC-regulatoren. Ved endring av regulatorparameterne, kunne ytelsen ved kompensering likevel forbedres til å bli bedre enn ved PI-regulering. Dette medførte imidlertid at PI-regulator og MPC-regulatoren ikke lenger var sammenlignbare med henhold til responstid og oversving.

Forord

Oppgaven er skrevet i forbindelse med fullførelsen av mastergrad innen informasjonsteknologi ved Universitetet i Stavanger. Oppgaven ble utført på varmluftsvifte ved teknisk-naturvitenskapelige fakultet, institutt for data- og elektroteknikk.

Veileder ved oppgaven har vært Tormod Drenngstig. Jeg vil takke Tormod for konstruktiv tilbakemelding, rettskrivning og generelt god veiledning gjennom hele utførelsen av oppgaven.

Jeg vil også takke min far, Ove Landa, for hjelp med ferdigstilling av rapporten.

En stor takk må også rettes til min samboer, Anita, som i travle tider med oppgaveskriving, har stått på med hjemlige gjøremål, samt flyttearbeid, som burde vært underlagt undertegnede.

Ben Ove Landa
Stavanger, 2010

Innhold

Sammendrag	i
Forord	iii
1 Innledning.....	1
1.1 Bakgrunn for oppgaven	1
1.2 Modellprediktiv regulering	3
1.3 Rapportens inndeling og struktur.....	5
2 Utvikling av lineær modell for lufthastighet	7
2.1 Modell fra sprangrespons.....	7
2.2 Implementering i Simulink.....	9
2.3 Modellverifisering.....	12
2.3.1 Verifisering av sprangrespons.....	12
2.3.2 Frekvensrespons	12
3 Modellprediktiv regulering	15
3.1 MPC-regulatorens historiske utvikling.....	15
3.1.1 LQG	16
3.1.2 1. generasjons MPC-regulatorer.....	16
3.1.3 2. generasjons MPC-regulatorer.....	17
3.1.4 3. generasjons MPC-regulatorer.....	17
3.1.5 4. generasjons MPC-regulatorer.....	18
3.2 Beskrivelse av MPC-regulatoren.....	19
3.2.1 Kontroll- og prediksjonshorisont	20

3.2.2	Begrensninger	23
3.2.3	Vekter på inngang	24
3.2.4	Vekter på utgang	25
3.2.5	Optimeringsproblemet	25
4	MPC-regulator for styring av lufthastighet	27
4.1	Utvikling av MPC-regulatoren	27
4.1.1	Oppsett for viftescenariet	27
4.1.2	Oppsett for MPC-regulator	28
4.1.3	MPC-blokk i Simulink	34
4.2	Parametertuning for MPC-regulator uten forhåndsinformasjon	36
4.2.1	Endring av prediksjonshorison, testet på modell	36
4.2.2	Endring av prediksjonshorison, testet på virkelig prosess	38
4.2.3	Endring av kontrollhorison, testet på modell	39
4.2.4	Endring av kontrollhorison, testet på virkelig prosess	41
4.2.5	Endring av inngangsvekt, testet på modell	42
4.2.6	Endring av inngangsvekt, testet på virkelig prosess	44
4.2.7	Endring av utgangsvekt, testet på modell	45
4.2.8	Endring av utgangsvekt, testet på virkelig prosess	47
4.2.9	Endring av kontrollintervall, testet på virkelig prosess	48
4.2.10	Oppsummering av tester	50
4.3	Parametertuning for MPC-regulator med forhåndsinformasjon	50
4.3.1	Endring av prediksjonshorison, testet på modell	50
4.3.2	Endring av prediksjonshorison, testet på virkelig prosess	52
4.3.3	Endring av kontrollhorison, testet på modell	53
4.3.4	Endring av kontrollhorison, testet på virkelig prosess	55
4.3.5	Oppsummering av tester	56
5	Anvendelse av MPC-regulatorene	57
5.1	Simulinkmodell for testing av MPC-regulator	57
5.1.1	Ytelsesmål	59
5.1.2	Parametere for MPC-regulator uten forhåndsinformasjon	60
5.1.3	Parametere for MPC-regulator med forhåndsinformasjon	60
5.1.4	Parametere for PI-regulator med foroverkobling	60
5.2	Følgeegenskaper	63
5.2.1	Sprang i referanse rundt arbeidspunkt	63
5.2.2	Test av robusthet ved referansesprang	65

5.2.3	Sinusreferanse rundt arbeidspunkt	69
5.3	Kompenseringsegenskaper.....	72
5.3.1	Kompenseringsegenskaper ved endret parametervalg.....	73
5.4	Effekt av modellens nøyaktighet	76
6	Konklusjon og forslag til videre arbeid	81
6.1	Konklusjon	81
6.2	Forslag til videre arbeid	82
	Referanser.....	83
	Vedlegg A	85
A.1	Bakgrunn for feil i Simulink i Matlab versjon R2009a.....	85
A.2	Oppskrift for å rette opp feil i Matlab versjon R2009a.....	85

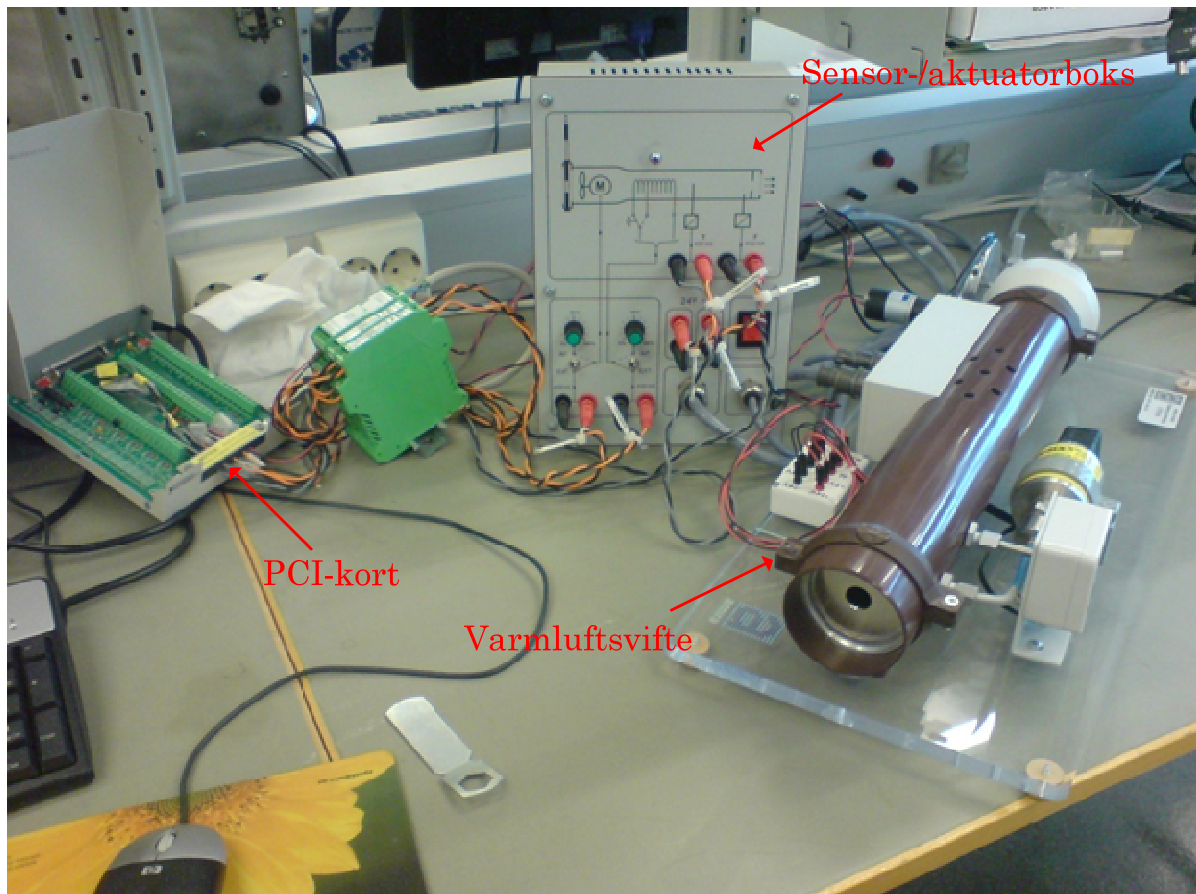
Kapittel 1

Innledning

1.1 Bakgrunn for oppgaven

Varmluftsviften er blitt brukt i forbindelse med oppgaveskriving, og laboratorieoppgaver i faget BIE240 – Reguleringssteknikk ved Universitetet i Stavanger. Varmluftsviftesystemet består av flere komponenter, se figur 1.1. Hovedkomponentene i systemet er følgende:

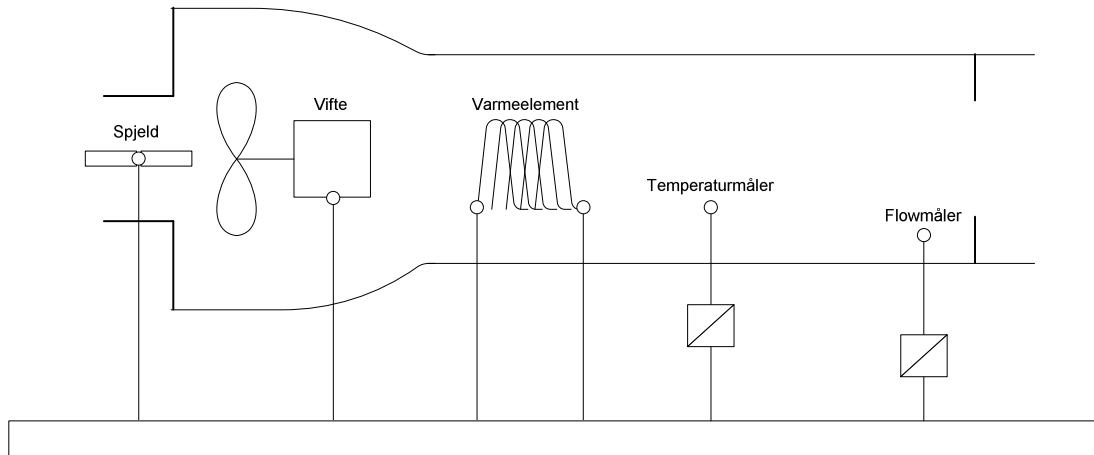
- Varmluftsviften. Viften har tre prosessinn ganger (pådrag som påføres varmluftsviften) og to prosessutganger (målinger fra varmluftsviften). Prosessinn gangene består av pådrag for viftemotor, varmeelement og spjeld. Prosessutgangene er måling av viftehastighet og temperatur i viften. Skjematisk tegning av viften er vist i figur 1.2.
- *Laboratorie-PC*. Datamaskinen som blir brukt for å implementere regulatorstruktur.
- *PCI-kort*. PCI-kortet videreformidler kommunikasjon fra laboratorie-PC til *aktuatorboks*, og fra *aktuatorboks* til laboratorie-PC.
- *Sensor-/aktuatorboks*. Sender pådragsverdier videre til prosessinn gangene på viften, og måleverdier blir sendt til PCI-kort.



Figur 1.1: Bilde av varmluftsvifte med komponenter.

Ved laboratorieoppgaver er PI- og PID-regulatorer blitt brukt for å regulere viftehastigheten i varmluftsviften. Dette er relativt enkle regulatorer som på bakgrunn av forsterkning, integral og derivasjon av reguleringsavviket gir et pådrag som minsker dette avviket. I denne oppgaven skal det utvikles og testes modellprediktive regulatorer (MPC) for varmluftsviften. MPC-regulatorene benytter en mer avansert og sofistikert algoritme for beregning av prosesspådrag.

Målet med oppgaven var å utvikle MPC-regulatorer med tanke på å regulere lufthastigheten gjennom viften, med spjeld som aktiv forstyrrelse for luftstrømmen. Videre skulle regulatorparameterne konfigureres for å oppnå god og stabil regulering. I siste del av oppgaven blir ytelsen til MPC-regulatorene vurdert opp mot ytelsen til PI-regulator med foroverkobling. Figur 1.2 viser skjematisk tegning av varmluftsviften.

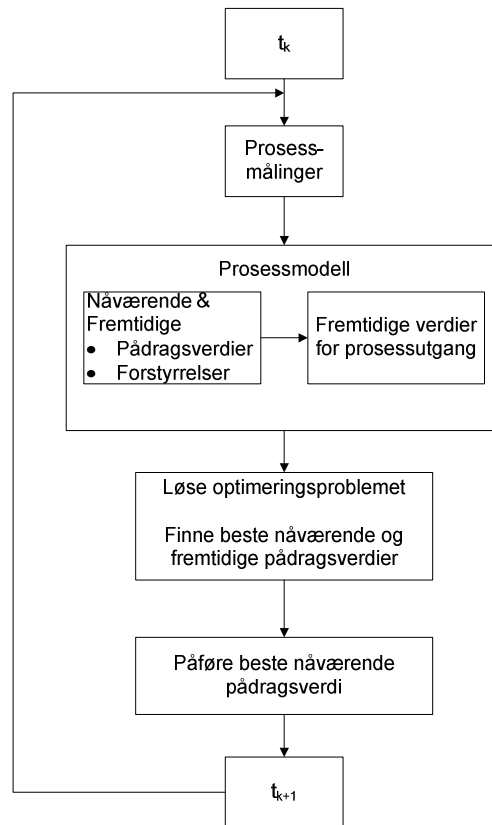


Figur 1.2: Skjematisk tegning av varmluftsvifte.

1.2 Modellprediktiv regulering

MPC-regulatorer er blitt benyttet i industrien siden midten av 1970-tallet. Siden har utvikling av datakapasitet og hard konkurranse i industrien vært med på videreutvikling av denne teknologien [1]. I dag er MPC-regulatorer blitt hyllevare, hvor to av de store aktørene er *NI LabVIEW* og *The MathWorks MATLAB* med verktøykassen "*Model Predictive Control Toolbox™*" [7].

MPC-regulatorer benytter en algoritme som genererer verdier for prosesspådrag (pådrag til viftemotor). Disse verdiene beregnes ved at algoritmen løser et optimeringsproblem. Problemet konstrueres på basis av matematisk modell av prosessen, og målinger hentet fra prosessen. Algoritmen tar i tillegg høyde for vektning av prosessens inngang og utgang. Disse vektene gir informasjon til regulatoren om prosessens restriksjoner i inngang og utgang. Aktuelle prosessmålinger og ikke-målte forstyrrelser er tilbakekoblet til MPC-regulatoren, og målte forstyrrelser er foroverkoblet fra regulatoren [1]. Flytdiagrammet i figur 1.3 viser virkemåten for en standard MPC-regulator.



Figur 1.3: Flytskjema for virkemåten til en standard MPC-regulator.

En enkel MPC-regulator har kapasitet til å håndtere flere hundre prosessinn ganger og prosessut ganger i motsetning til en vanlig PID-regulator som kun gir mulighet for å gi en pådragsverdi per prosessutgang.

1.3 Rapportens inndeling og struktur

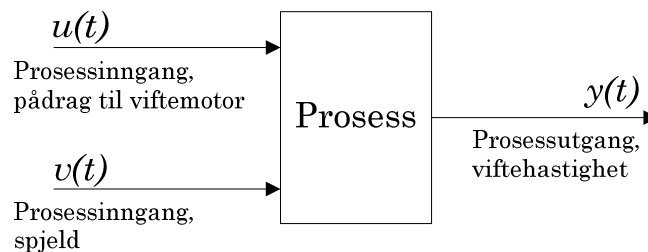
- Kapittel 1. Innledning.
- Kapittel 2. Forklarer stegvis hvordan modell for lufthastigheten er funnet og verifisert med virkelig prosess.
- Kapittel 3. Her blir sentrale begreper for MPC-regulering forklart i mer detalj, samt historisk utvikling av MPC-teknologien redegjort for.
- Kapittel 4. Implementering og testing av regulatorparametere for MPC-regulatorene, online og offline i Simulink.
- Kapittel 5. Måler ytelse for følgeegenskaper og kompenseringsegenskaper til de to MPC-regulatorene. Vurderer ytelsen opp mot ytelse for PI-regulator med foroverkobling.
- Kapittel 6. Konklusjon og forslag til videre arbeid.

Kapittel 2

Utvikling av lineær modell for lufthastighet

2.1 Modell fra sprangrespons

Selv om MPC-regulatoren kan regulere ulineære prosesser, er det et krav i Matlab at modellen som brukes i regulatoren er lineær og tidsinvariant (LTI) [2]. Figur 2.1 viser blokkdiagram for vifteprosessen. $u(t)$ er prosesspådraget som gir pådrag til viftemotor, $v(t)$ er pådraget som styrer spjeldets posisjon (åpent/lukket) og $y(t)$ er viftehastigheten i viften.

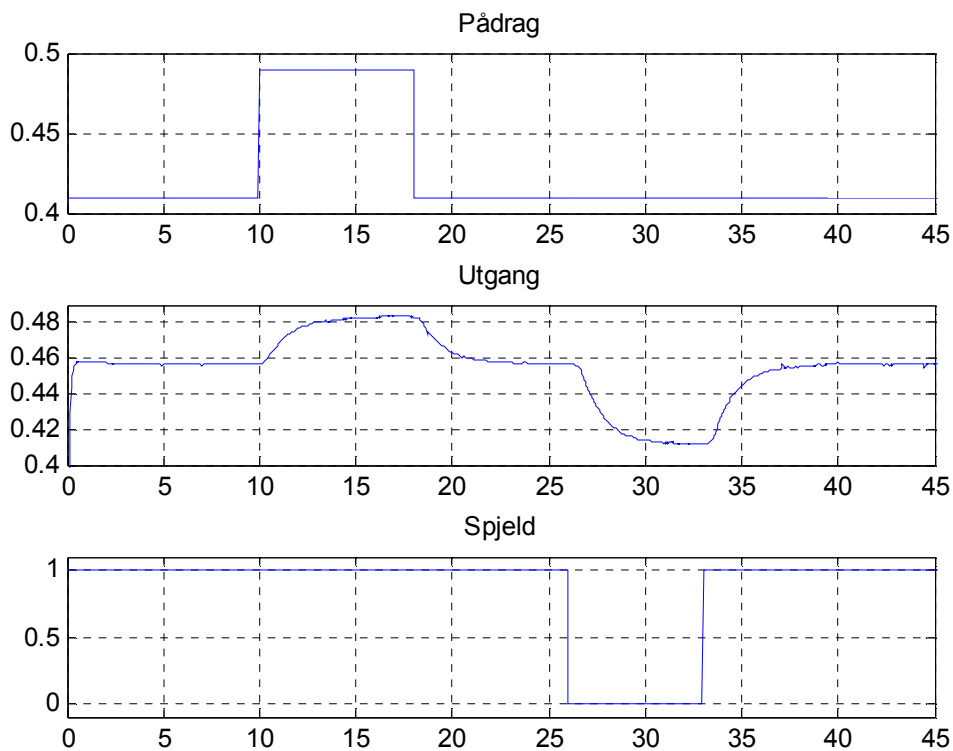


Figur 2.1: Blokkdiagram for vifteprosessen.

For å finne modell for viftehastigheten ble sprangrespons rundt valgt arbeidspunkt brukt. Valgt arbeidspunkt for pådrag til viftemotor og spjeld er henholdsvis $u_1 = 0.41$ og $v_1 = 1$ (spjeld åpent). For sprangresponsen er følgende sprang utført fra arbeidspunktet:

- $u_2 = 0.49$
- $v_2 = 0$ (spjeld lukket)

I figur 2.2 vises sprangresponsen med utgangspunkt i de valgte sprangene fra arbeidspunkt for pådrag og forstyrrelsen.



Figur 2.2: Sprangrespons for varmluftsvifte.

For å finne LTI-modell av prosessen ble transferfunksjonen benyttet. Transferfunksjonene ble bestemt ved beregninger og observasjoner fra sprangresponsen. De stasjonære utgangsverdiene før og etter sprang i pådrag og forstyrrelse var:

- $y_{u1} = 0.457$
- $y_{u2} = 0.484$
- $y_{v1} = 0.457$
- $y_{v2} = 0.412$

Dødtiden for de to sprangene ble observert til å være:

- $\tau_u = 0.06$ [s]
- $\tau_v = 0.60$ [s]

Tidskonstantene ble funnet ved å lese av tiden utgang bruker på å nå 63 % av stasjonærverdi etter sprang. Her ble de respektive dødtidene tatt hensyn til. Tidskonstantene ble beregnet til:

- $T_u = 1.52$ [s]
- $T_v = 1.07$ [s]

Forsterkningene, K_u og K_d , ble funnet ved å dividere endring i utgang på endring i inngang:

$$K_u = \frac{\Delta y}{\Delta u} = \frac{0.027}{0.080} = 0.338 \quad (2.1)$$

$$K_v = \frac{\Delta y}{\Delta d} = \frac{-0.045}{1.000} = -0.045 \quad (2.2)$$

Transferfunksjonene antas å være av 1. orden med dødtid. Med utgangspunkt i standard transferfunksjon med dødtid ble de to transferfunksjonene spesifisert:

$$H_u(s) = \frac{K_u}{T_u s + 1} e^{-\tau_u s} = \frac{0.338}{1.52s + 1} e^{-0.06s} \quad (2.3)$$

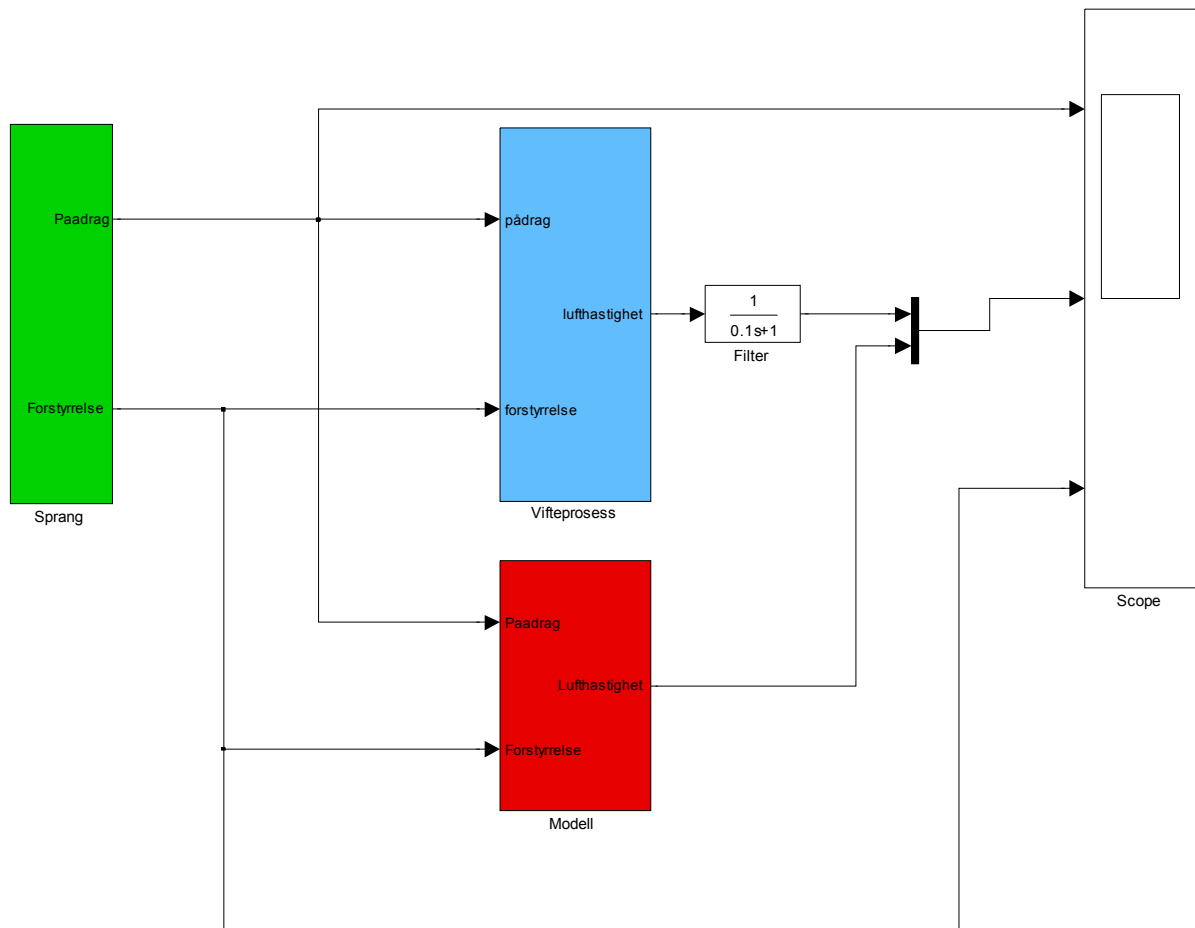
$$H_v(s) = \frac{K_d}{T_d s + 1} e^{-\tau_v s} = \frac{-0.045}{1.07s + 1} e^{-0.60s} \quad (2.4)$$

Transferfunksjonene i ligning (2.3) og (2.4) bør verifiseres før de kan brukes som modell i MPC-regulatoren, og gjelder kun i arbeidspunktet.

2.2 Implementering i Simulink

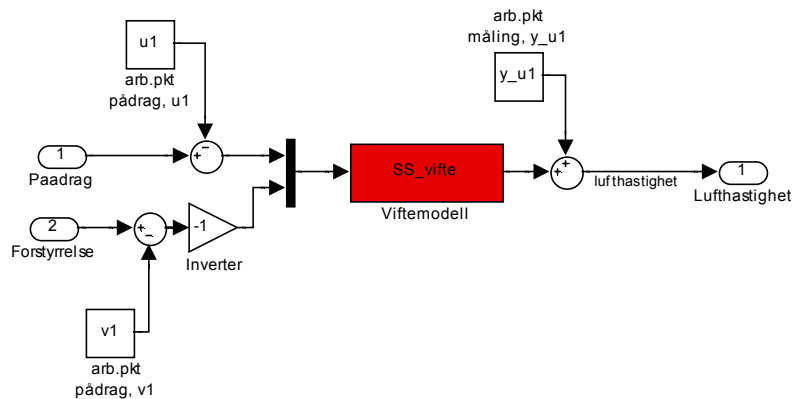
For å verifisere modellen fra transferfunksjonene, ble modellen implementert inn i parallell med virkelig prosess i Simulink¹. Figur 2.3 viser implementeringen i Simulink.

¹ Simulink er en grafisk simuleringsspakke integrert i Matlab.



Figur 2.3: Simulinkmodell for verifisering av transferfunksjonsmodell.

I grønn blokk blir de ulike sprangene i pådrag og forstyrrelse gitt etter spesifisert tidspunkt. Sprangene blir sendt videre til sine respektive innganger for vifteprosess (blå blokk), modell (rød blokk) og skopet. I blokken for vifteprosess er det implementert en analog inngang for pådrag, en digital inngang for forstyrrelsen og en analog utgang for lufthastigheten. Signalene blir sendt og mottatt ved bruk av et PCI-kort. Kortet er av typen E-6024 og produsert av National Instruments. På utgangen til vifteprosessen er det lagt til et lavpassfilter for filtrering av utgangssignalet. I blokken for modell er de to transferfunksjonene implementert. Figur 2.4 viser hvordan modellblokken er implementert.



Figur 2.4: Modellblokken.

Utgangen til modellen ble justert til verdi for arbeidspunkt ved å addere arbeidspunktverdien for utgang på utgangen, og subtrahere inngangene med pådragsverdier for arbeidspunkt, se figur 2.4 Denne blokken ble utviklet fra en eksisterende LTI-blokk i Simulink, og blir brukt til å laste inn transferfunksjoner. Blokken bruker modellen *SS_vifte* (se tabell 2.1) som ble spesifisert i en m-fil², samt verdiene på inngangen, til å gi verdi på utgang. Transferfunksjonen for forstyrrelsen ble laget slik at sprang fra 1 (åpent) til 0 (lukket), blir oppfattet som et sprang fra 0 til 1. Dette ble tatt hensyn til ved å invertere inngangen for forstyrrelsen i modellblokken.

Tabell 2.1: m-fil for generering av transferfunksjoner.

```

19 %Viftehastighet
20 % Transferfunksjon fra pådrag til utgang
21 Y_U = tf([0.338], [1.52 1], 'OutputDelay', 0.06);
22
23 % Transferfunksjon fra målt forstyrrelse til utgang
24 Y_V = tf([-0.045], [1.072 1], 'OutputDelay', 0.6);
25
26 % Setter sammen til MISO
27
28 SS_vifte = [Y_U Y_V];

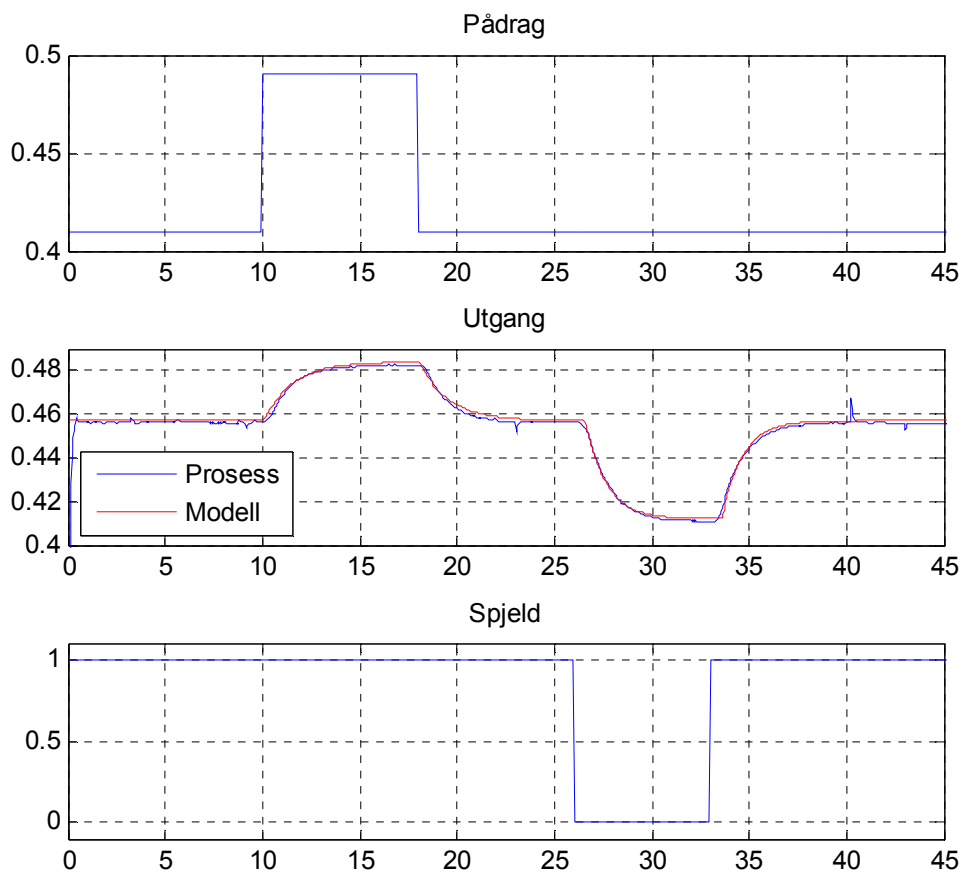
```

² m-filer er kjørbare script-filer i Matlab, hvor variabler, konstanter og Matlab-kommandoer kan spesifiseres og lastes.

2.3 Modellverifisering

2.3.1 Verifisering av sprangrespons

For at modellen skal være brukbar ved MPC-regulering er det viktig at modellen representerer prosessens hoveddynamikk. I figur 2.5 sammenlignes prosessen med modell.

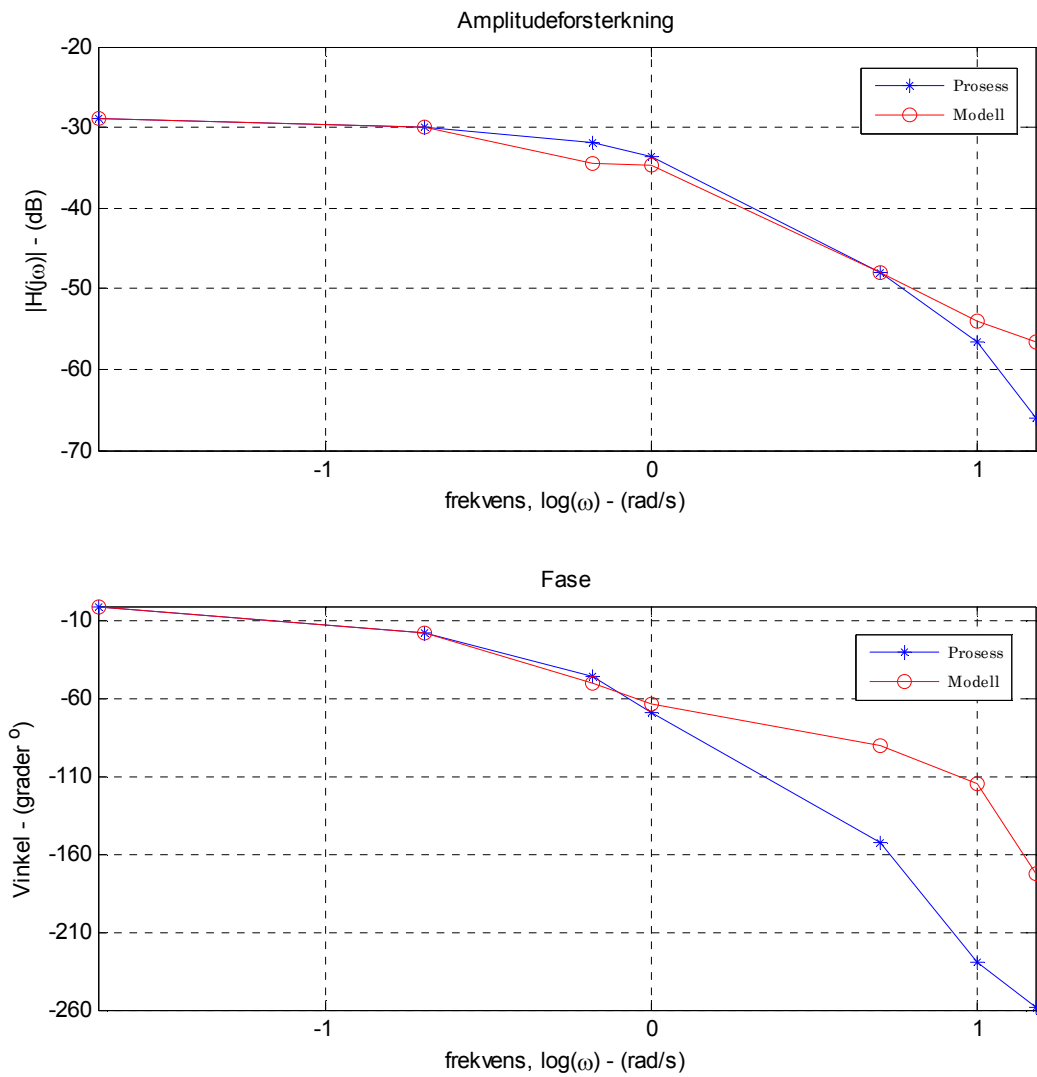


Figur 2.5: Modellverifisering av viftehastighet.

Figur 2.5 viser at respons fra modell og prosess stemmer godt overens.

2.3.2 Frekvensrespons

Figur 2.6 viser frekvensresponsen for virkelig prosess og modell, fra viftepådrag til målt viftehastighet. Frekvensen, ω (rad/s), ble variert med størrelsene [0.02 0.20 0.66 1 5 10 15].



Figur 2.6: Frekvensrespons av prosessen og modell, fra pådrag til utgang.

Figur 2.6 viser økende uoverensstemmelse ved økende frekvens. Dette er en klar indikasjon på at modellen kun er brukbar for lave frekvenser. På grunnlag av sprang- og frekvensrespons kan modell fra ligning (2.3) og (2.4) brukes for MPC-regulatorene, i arbeidspunktet.

Kapittel 3

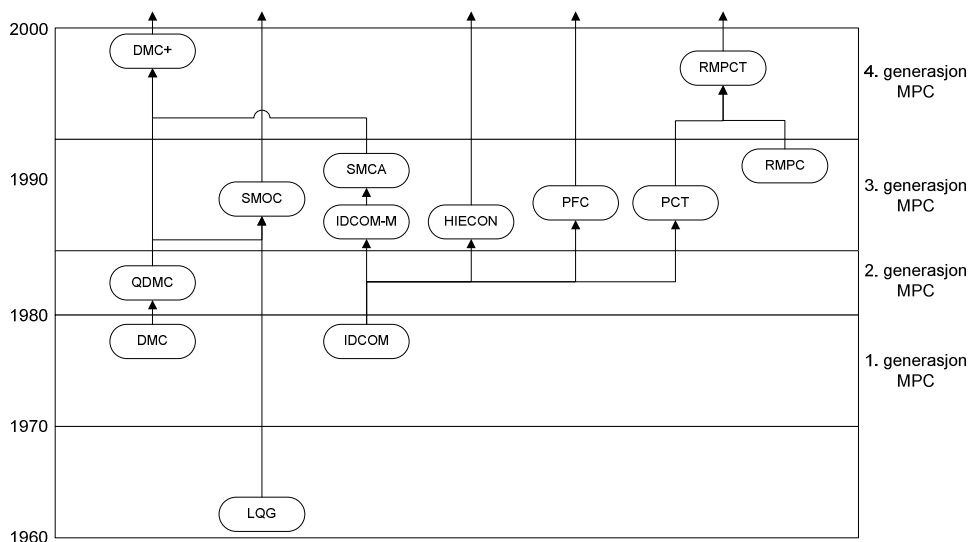
Modellprediktiv regulering

Modellprediktiv regulering (MPC) er en fellesbetegnelse for reguleringsalgoritmer som anvender modell av prosessen til prediktere fremtidige prosessutganger og innganger. Dette kapittel gir et historisk tilbakeblikk på MPC-teknologiens utvikling, og forklarer de sentrale begrepene for MPC-regulering:

- Optimeringskriteriet
- Kontroll- og prediksjonshorisont
- Begrensinger
- Vekting av inngangs- og utgangsvARIABLE

3.1 MPC-regulatorens historiske utvikling

Denne delen gir et innblikk i utviklingen som har ført fram til den moderne MPC-regulatoren. Informasjonen er hentet fra [4]. Figur 3.1 viser en kort skisse for hvordan utviklingen har foregått.



Figur 3.1: Historisk utvikling av MPC-teknologien. Figur er skissert fra [4].

3.1.1 LQG

Utviklingen av moderne MPC-regulering kan føres tilbake til tidlig på 1960-tallet. Kalman og hans medarbeidere arbeidet på denne tiden med å utvikle en reguleringsmetode som senere ble kalt LQG (*Linear Quadratic Gaussian*). Metoden går ut på å regulere en prosess beskrevet med en diskret og lineær tilstandsrommodell. For å finne pådrag til prosessen bruker metoden en funksjon som minimeres, sammen med vektmatriser, for å finne optimale pådrag. LQG metoden ble etter hvert en standardmetode for å løse problemer tilknyttet regulering. Metoden ble imidlertid aldri implementert i noen stor grad i prosessindustrien. Hovedgrunnen for dette er instrumentteknikere og kontrollingeniører på denne tiden var lite kjent med metoden eller anså den som upraktisk.

3.1.2 1. generasjons MPC-regulatorer

Den første beskrivelsen av applikasjoner for MPC-regulering ble gitt av Richalet m. fl. i 1976. Denne benyttet en metode kalt *Model Predictive Heuristic Control* (MPHC). Programvaren ble kalt *Identification and Command* (IDCOM). Metoden har følgende egenskap:

- Bruker impulsresponsmodell for prosessen
- Kvadratisk objektfunksjon over en endelig prediksjonshorisont
- Fremtidig forløp i prosessutgang blir bestemt av referansegang
- Begrensinger for prosessinnang og utgang
- Optimale pådragsverdier beregnes ved bruk av en eksperimentell iterativ algoritme

På samme tid utviklet ingeniører ved Shell Oil en egen MPC-regulator. Denne ble brukt allerede i 1973. Cutler og Ramaker viste frem algoritmen som ble brukt for reguleringen i 1979. Dette var en

ubegrenset multivariabel reguleringsalgoritme som fikk navnet *Dynamic Matrix Control* (DMC). Egenskapene tilknyttet denne metoden er:

- Bruker lineær sprangrespons til å modellere prosess
- Kvadratisk objektfunksjon over en endelig prediksjonshorison
- Fremtidig forløp prosessutgang bestemmes ved å følge referanse så godt som mulig
- Optimale pådragsverdier beregnes ved å finne løsning for et minstekvadrats problem

IDCOM og DMC er første generasjons MPC-regulatorer. De hadde stor betydning for regulering av ulike prosess i industrien, og definerer selve paradigme for industriell MPC-regulering.

3.1.3 2. generasjons MPC-regulatorer

IDCOM og DMC algoritmene ga gode resultater for regulering av ubegrensede multivariable prosesser. Håndtering av begrensinger var imidlertid fremdeles ad hoc. Ingeniører ved Shell Oil løste dette problemet ved å se på DMC algoritmen som et kvadratisk program (QP), hvor begrensinger i prosessinnnganger og utganger blir vurdert hver for seg. Cutler m. fl. introduserte algoritmen, *Quadratic Dynamic Matrix Control* (QDMC), i 1983. Metoden har følgende egenskaper:

- Bruker lineær sprangrespons til å modellere prosess
- Kvadratisk objektfunksjon over en endelig prediksjonshorison
- Fremtidig forløp prosessutganger bestemmes ved å følge referanse som godt som mulig, samtidig som begrensinger i prosess og ytelse opprettholdes.
- Optimale pådragsverdier beregnes ved å løse et kvadratisk program

QDMC algoritmen er andre generasjons MPC-regulering. Denne gjorde det muligjorde systematisk implementering av begrensinger i prosessinnang og utgang.

3.1.4 3. generasjons MPC-regulatorer

QDMC algoritmen ga en systematisk fremgangsmåte for å definere harde begrensninger i prosessinnang og utgang, men hadde ingen rutiner for å behandle en ikke-brukbar løsning. Å benytte myke begrensninger ga heller ikke noen god løsning på problemet, siden dette innebar at begrensningene ville overtreddes til en viss grad, bestemt av de relative vektene. Dette ga ideer til prioriteringsbevisste begrensninger. Noen begrensinger er viktigere enn andre, mens noen aldri kan overtreddes.

Denne problemstillingen førte til at ingeniører ved Adersa, Setpoint og Shell jobbet med å videreutvikle sine respektive MPC-algoritmer. Adersa versjonen ble kalt *Hierarchical Constraint Control* (HIECON). Setpoint versjonen ble kalt IDCOM-M, og er nesten identisk med Adersa versjonen. Egenskapene for metodene er:

- Prosess modellert fra lineær impulsrespons
- Egen rutine som fjerner uønskede løsninger
- Multi-objektfunksjon; kvadratisk objektfunksjon for prosessutgang, fulgt av en kvadratisk objektfunksjon for prosessinnngang.
- Bestemmer et sett med fremtidige punkter for hver prosessutgang, bestemt fra referansegang
- Endringer blir beregnet for hver enkelt prosessinnngang
- Begrensninger kan spesifiseres som harde eller myke, der harde begrensninger er rangert i prioritert rekkefølge

Ingeniørene ved Setpoint fortsatte å videreutvikle IDCOM-M teknologien. Dette førte til utviklingen av *Setpoint Multivariable Control Architecture (SMCA)*. SMCA gjorde det mulig å løse en sekvens av forskjellige verdier på tilstandsromform, slik at det ble enklere å inkludere flerrangerte objekter og begrensninger.

Sent på 1980-tallet utviklet ingeniører ved Shell Research i Frankrike MPC-regulatoren *Shell Multivariable Optimizing Controller (SMOC)*, som de beskrev som en bro mellom tilstandsrom og MPC-algoritmer. SMOC-algoritmen inkluderer flere av delene som i dag regnes for essensielle for en moderne MPC-regulator:

- Tilstandsrommodeller blir brukt for at hele spekteret av lineære dynamikk (stabil, ustabil og integrerende) kan modelleres
- Modell av forstyrrelse beskriver innvirkning av ikke-målte forstyrrelser.
- Kalmanfilter brukes til å estimere tilstander for prosess og ikke-målte forstyrrelser, fra måling av prosessutgang
- Skille mellom regulerte variabler i objektfunksjonen og tilbakekoblede variabler i tilstandsestimering.
- Begrensninger for prosessinnngang og utgang er oppretthold ved bruk av QP-formulering.

Tredjegerasjons MPC-regulator er representert ved IDCOM-M, HIECON, SMCA og SMOC algoritmene. PCT algoritmen levert av Profimatics Inc. og RMPC algoritmen levert av Honeywell hører også til denne generasjonen MPC-regulatorer. MPC-regulatorene fra denne generasjonen bidro med flere forbedringer av MPC-teknologien. Flere alternativer for spesifisering av begrensninger (harde, myke, prioriterte), rutiner for å behandle løsninger som er ikke-brukbare og bedre anvendelser av tilbakekobling er noen av forbedringene.

3.1.5 4. generasjons MPC-regulatorer

I siste halvdel av 1990-tallet førte økt konkurranse og fusjoner av flere MPC leverandører til store forandringer innen industriell MPC-regulering. I 1995 ble Profimatics Inc. kjøpt opp av Honeywell, og omdøpt til Honeywell Hi-Spec Solutions. RMPC algoritmen fra Honeywell ble slått sammen med PCT algoritmen til Profimatics, og fikk navnet RMPCT. I 1996 ble både Setpoint Inc. og DMC Corporation

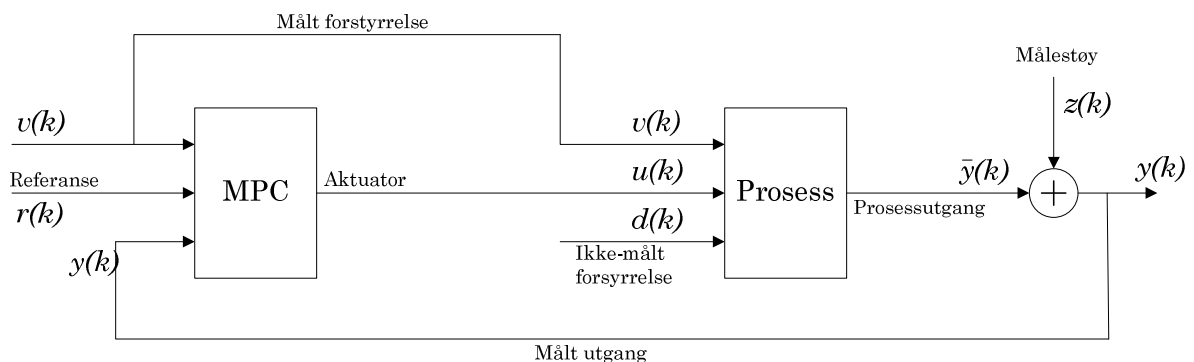
kjøpt opp av Aspen Technology Inc. Dette førte til at teknologien forbundet med SMCA og DMC ble smeltet sammen til DMC-plus. DMC-plus og RMPCT er fjerdegenerasjons MPC-regulatorer, og har følgende egenskaper:

- Windows-basert brukergrensesnitt
- Flere optimaliseringsnivå for å kunne prioritere målene for reguleringen
- Mer fleksibilitet i forhold til optimalisering av tilstandsrommodell, inkludert QP og energibesparelse
- Vurderer modellusikkerhet
- Forbedret identifiseringsteknologi

3.2 Beskrivelse av MPC-regulatoren

Under utvikling og testing av MPC-regulator er *Model Predictive Control Toolbox* fra Matlab brukt. Dette er en kraftig MPC-verktøykasse som muliggjør oppsett av MPC-regulator ved bruk av kommandoer i Matlab eller via et brukergrensesnitt. All informasjon i dette kapitlet er hentet fra [2] og [3].

Figur 3.2 viser blokkdiagram for en *Single-Input-Single-Output-prosess* (SISO-prosess) med MPC-regulering. Målet for MPC-strukturen er å holde utgangen $y(k)$, på samme verdi som referansen $r(k)$. Dette gjøres ved å justere den *manipulerbare variabelen* (aktuator el. prosesspådrag) $u(k)$. De ulike signalene er beskrevet i tabell 3.1.



Figur 3.2: Blokkdiagram for MPC-regulering av SISO-prosess.

MPC-verktøykassen er avhengig av en modell som beskriver innvirkningen $u(k)$ og $v(k)$ har på $y(k)$, for å kunne sette opp MPC-regulatoren. Modellen representeres ved tilstandsrom eller transferfunksjoner, avhengig av hva som er hensiktsmessig for aktuell prosess. Regulatoren bruker denne modellen for å beregne $u(k)$, slik at $y(k)$ er nær referanseverdi.

Tabell 3.1: Signalbeskrivelse for SISO-modell.

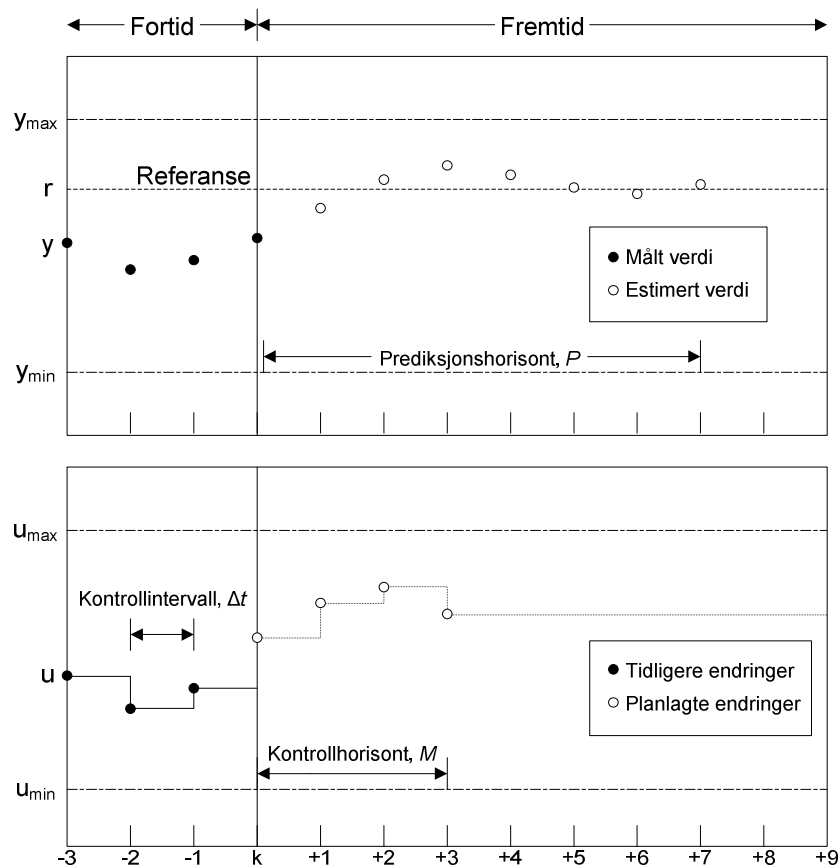
Symbol	Beskrivelse
$d(k)$	<i>Ikke-målt forstyrrelse.</i> Dette er ukjente og uforutsette forløp som forstyrrer prosessgangen. Regulatoren kompenserer for disse forstyrrelsene via tilbakekoblingen.
$r(k)$	<i>Referanse.</i> Spesifiserer verdien som er ønsket på utgang.
$u(k)$	<i>Manipulerbar variabel</i> eller <i>aktuator</i> . Signalet som brukes for å justere utgangen mot referanseverdi.
$v(k)$	<i>Målt forstyrrelse.</i> Strukturen kompenserer for denne forstyrrelsen ved en foroverkobling. Koblingen gir informasjon til regulatoren om forstyrrelsesforløpet <i>før</i> det blir målt på utgang.
$\square(k)$	<i>Utgang.</i> Verdien som reguleres etter referanseverdi.
$y(k)$	<i>Målt utgang.</i> Brukes til å estimere \square .
$z(k)$	<i>Målestøy.</i>

I MPC-verktøykassen er det mulig å spesifisere MPC-regulatoren slik at den benytter informasjon om fremtidige referanseendringer. Dette setter den i stand til å justere $u(k)$ i forkant av en referanseendring. Nøyaktigheten av denne justeringen er avhengig av prosesskarakteristikken (inkludert begrensninger), prosessmodell og størrelsesorden på referanseendringen.

Fremtidige endringer i målt forstyrrelse, $v(k)$, kan spesifiseres på samme måte. Dette forutsetter at den målte forstyrrelsen er av en slik natur at den kan forhåndsbestemmes.

3.2.1 Kontroll- og prediksjonshorisont

MPC-regulatoren optimerer over P fremtidige kontrollintervaller og beregner M pådragsverdier. Figur 3.3 viser forløp for et tenkt SISO-system uten målt forstyrrelse. Kun den første planlagte endringen i pådraget blir bruk, før regulatoren beregner nye ved neste kontrollintervall.



Figur 3.3: Regulering av SISO-prosess ved $\Delta t=k$. Figur er skissert fra [3].

MPC-regulatoren er en tidsdiskret regulator. Den utfører endringer i aktuator ved faste diskrete tidsinstanser. Disse tidsinstansene kalles *kontrollintervall*, Δt . I figur 3.3 har systemet vært operativt i flere tidsinstanser. k representerer nåværende instans. Den siste målte utgangen y_k , samt tidligere målte utganger, er representert ved fylte sirkler i figuren. De ufylte sirklene representerer estimerte utgangsverdier ved fremtidige kontrollintervall. De tidligere endringene, u_{k-3} , u_{k-2} og u_{k-1} , er markerte med fylte sirkler, og planlagte endringer vises ved ufylte sirkler.

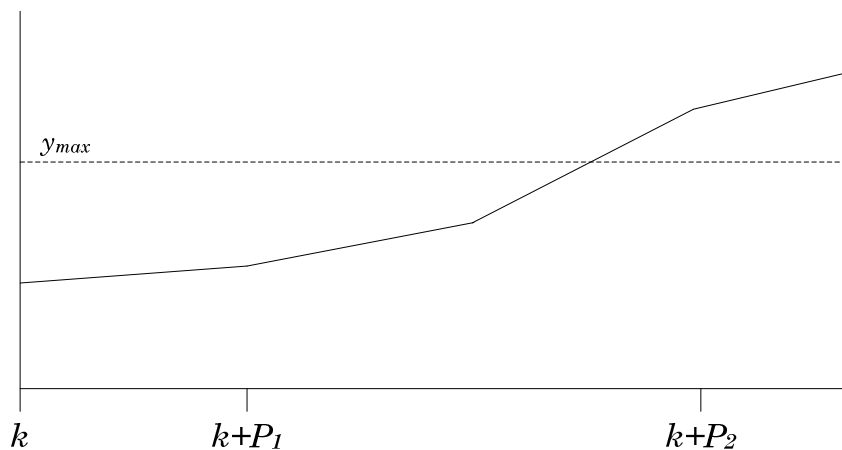
For å beregne neste endring, u_k , gjennomfører regulatoren to steg:

1. *Estimering*. For å gjøre en hensiktsmessig endring, må regulatoren vite prosessens nåværende tilstand. Dette inkluderer verdien av regulerte variabelen y_k , samt andre interne variabler som virker inn på fremtidig trend, y_{k+1} , ..., y_{k+P} . For å finne disse variablene benytter regulatoren seg av informasjon av tidligere og nåværende målinger, sammen med en gitt modell av prosessen.
2. *Optimering*. Verdier for referanse, målt forstyrrelse og begrensninger blir bestemt over en endelig horisont av fremtidige sampelinstanser, $k+1$, $k+2$, ..., $k+P$, der P er *prediksjonshorisonen*. Regulatoren beregner M endringer u_k , u_{k+1} , ..., u_{k+M-1} , der M er *kontrollhorisonten*. I det tenkte tilfellet i figur 3.3 er $P=7$ og $M=4$.

Når regulatoren har beregnet en verdi for u_k , holdes denne verdien konstant i hele kontrollintervallet Δt . Ved neste sampleinstans gjennomfører regulatoren hele beregningsprosessen på nytt. Dette gjør at regulatoren kan rette opp for eventuelle feil og uforutsette forstyrrelser.

Grunnen til at regulatoren optimerer over P fremtidige sampleperioder og beregner M framtidige endringer, når den kun benytter første endring u_k for hver syklus, er følgende:

- *Begrensninger.* Gitt at horisonten er lang nok, kan regulatoren se potensielle begrensninger og unngå dem, eller i det minste minske effekten av at prosessen går i metning. Figur 3.4 illustrerer hvorfor prediksjonshorisonten må ha en viss lengde. Regulatorens mål er å holde prosessutgangen y under en øvre grense y_{max} . Den nåværende sampleinstansen er k , og modellen predikerer en oppadgående stigning y_{k+i} . Om regulatoren ser P_1 kontrollintervaller fremover, vil den ikke ta høyde for begrensningen før mer tid har gått. Om prediksjonshorisonten er P_2 , ville regulatoren gjøre endringer for å unngå begrensningen med en gang.



Figur 3.4: Betydning av lengde på prediksjonshorisont. Figur er skissert fra [3].

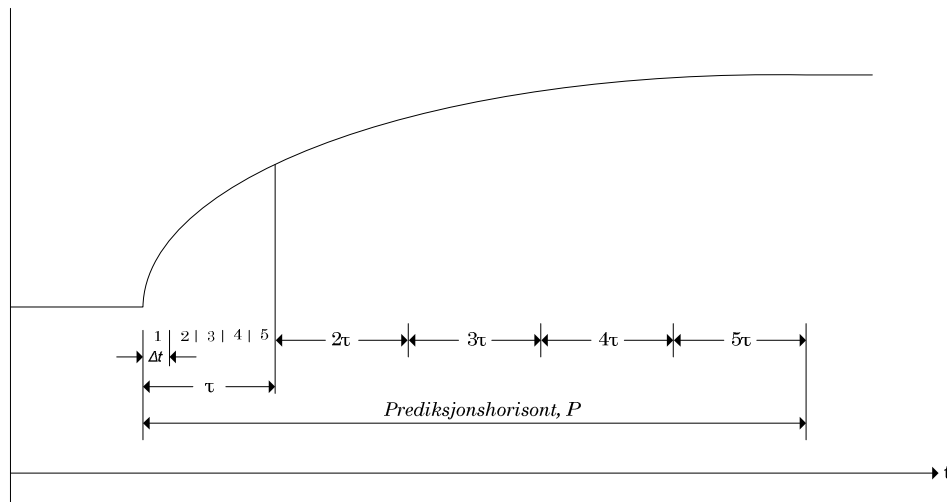
- *Prosessforsinkelser.* Sett at prosessen har en tidsforsinkelse tilsvarende D antall kontrollintervaller. Regulatorens endring, u_k , har i et slikt tilfelle ingen effekt før y_{k+D+1} . I disse tilfellene er det essensielt at $P \gg D$ og $M \ll P - D$. Med disse kriteriene oppfylt, vil regulatoren ta hensyn til full effekt ved hver pådragsendring.
- *Prosesser med inversrespons.* Ved prosess som har kortsiktig respons i en retning, men en langsiktig respons i andre retning er det viktig at P er stor nok, slik at horisonten ser det langsiktige forløpet. Om P blir satt for liten vil regulatoren gi ut feil pådragsverdier, siden den ikke tar hensyn til den langsiktige responsen.

P og M velges slik at regulatorens ytelse ikke blir påvirket av små endringer i horisontene. For å sikre dette kravet brukes følgende tommelfingerregler:

1. Kontrollintervall velges slik at prosessens stabiliseringstid i åpen sløyfe er omtrent 20-30 kontrollintervaller, eller slik at et kontrollintervall tilsvarer en femtedel av den dominerende tidskonstant.
2. Prediksjonshorisonten P velges lik antall kontrollintervall valgt i foregående punkt.

3. Velger en relativt liten kontrollhorisont M , typisk 3-5.

I figur 3.5 vises hvordan prediksjonshorisonten til en tenkt prosess velges med utgangspunkt i stabiliseringstiden til den tenkte prosessen i åpen sløyfe.



Figur 3.5: Lengde på prediksjonshorisont fra stabiliseringstid i åpen sløyfe. Figur er skissert fra [3].

I figur 3.5 er Δt kontrollintervallet og τ prosessens tidskonstant. Til denne tenkte prosessen blir prediksjonshorisonten, P , valgt til å være lik 25 kontrollintervaller.

3.2.2 Begrensninger

Begrensninger defineres som *harde* eller *myke*. En hard begrensning skal i prinsippet aldri overskrides, men i enkelte tilfeller (f. eks. ved uventet og stor forstyrrelse) er dette uunngåelig, og en realistisk regulator må tillate dette. MPC-verktøykassen løser dette ved å mykne opp alle begrensninger, slik at en hard begrensning har matematisk mulighet for å overskrides, om ekstreme tilfeller skulle inntreffe. Graden av mykhet kan bestemmes for hver enkelt begrensning, slik at en mykere begrensning vil ha større sannsynlighet for å bli overskredet enn en begrensning med mindre mykhet.

Det spesifiseres et *toleranseband* for hver begrensning. Om toleransebandet er satt til null, er begrensninger hard. Økning av toleransebandet fører til mykere begrensning. Toleransebandet er ikke en grense på hvor mye begrensningen kan overskrides, men et sammenlignbart mål for regulatoren som gjør at begrensningene vurderes relativt til hverandre.

3.2.3 Vekter på inngang

Pådragsundertrykkelse

Om regulatoren kun er opptatt av å følge referansen, vil den flere tilfeller gi store pådragsendringer til prosessen. Disse endringene kan være så store at de er vanskelige å oppnå, og kan føre til økt slitasje i prosessutstyr (ventil, motor, etc.) og/eller ustabilitet.

For å unngå dette tar MPC-regulatoren hensyn til en vektet sum av pådragsendringene, beregnet ved følgende likning:

$$S_{\Delta u}(k) = \sum_{i=1}^M \sum_{j=1}^{n_{mv}} \{w_j^{\Delta u} \Delta u_j(k+i-1)\}^2 \quad (3.1)$$

der M er kontrollhorisonten, n_{mv} er antall pådragsvariabler, $\Delta u_j(k+i-1)$ er den predikerte endringen til pådragsvariabel j ved sampleintervall $(k+i-1)$, $w_j^{\Delta u}$ er en vekt med positiv verdi eller verdi lik null.

Økning av vekten $w_j^{\Delta u}$ fører til at regulatoren gjør mindre og mer forsiktige pådragsendringer. I de fleste tilfeller fører dette til:

- Prosessens utgang følger referansen dårligere
- Regulatoren er mindre sensitiv for prediksjonsfeil, altså mer robust

Referanseverdier for pådragsvariabler

I de fleste anvendelser, er det viktig at pådragsvariabler kan endre seg fritt (innenfor satt begrensning) for å kompensere for referanseendringer og forstyrrelser. Ved å tvinge pådragsvariablene mot en bestemt verdi, forringes prosessutgangens evne til å følge referansen.

Noen prosesser har flere pådrag enn utganger. I slike prosesser vil være lønnsomt å bruke en referanseverdi for noen pådragsvariabler, slik at regulatoren kun forholder seg til et prosesspådrag per prosessutgang. For ekstra prosesspådrag, brukes følgende likning for å holde disse konstant ved en bestemt referanseverdi:

$$S_u(k) = \sum_{i=1}^M \sum_{j=1}^{n_{mv}} \{w_j^u [\bar{u}_j - u_j(k+i-1)]\}^2 \quad (3.2)$$

der \bar{u}_j referanseverdi for j -te pådragsvariabel, og w_j^u er tilhørende vekt.

$S_{\Delta u}(k)$ og $S_u(k)$ er hver sin del av optimeringsproblemet i kapittel 3.2.5, hvor vekten $w_j^{\Delta u}$ er gitt ved $w_{i,j}^{\Delta u}$, og vekten w_j^u er gitt ved $w_{i,j}^u$.

3.2.4 Vekter på utgang

Regulatoren predikerer hvor mye hver prosessutgang vil avvike fra referansen for hvert punkt langs prediksjonshorizonten. Hvert avvik multipliseres med utgangsvekten, og det beregnes en vektet sum, $S_y(k)$, bestående av kvadrerte avvik, ved hjelp av følgende likning:

$$S_y(k) = \sum_{i=1}^P \sum_{j=1}^{n_y} \{w_j^y [r_j(k+i) - y_j(k+i)]\}^2 \quad (3.3)$$

der P er prediksjonshorizont, n_y er antall prosessutganger, $[r_j(k+i) - y_j(k+i)]$ er det predikerte avviket for instansen $k+i$, w_j^y er vekten for utgang j .

Ved et tilfelle med to utganger, der $w_j^y \gg w_{i \neq j}^y$, vil regulatoren følge r_j , og ofre følgning av r_i , om nødvendig.

$S_y(k)$ er en del av optimeringsproblemet i kapittel 3.2.5, hvor vekten w_j^y er gitt ved $w_{i+1,j}^y$.

3.2.5 Optimeringsproblemet

Pådragsvariabel $u(k)$ beregnes ved å løse et optimeringsproblem. I problemet er prosessens tilstand(er) estimert fra prosessmodell, og dermed kjent. Ved tiden k , beregner regulatoren pådragsverdiene ved å løse følgende optimeringsproblem:

$$\begin{aligned} \min_{\Delta u(k|k), \dots, \Delta u(M-1+k|k), \varepsilon} & \left\{ \sum_{i=0}^{P-1} \left(\sum_{j=1}^{n_y} |w_{i+1,j}^y (y_j(k+i+1|k) - r_j(k+i+1))|^2 \right. \right. \\ & \left. \left. + \sum_{j=1}^{n_u} |w_{i,j}^{\Delta u} \Delta u_j(k+i|k)|^2 + \sum_{j=1}^{n_u} |w_{i,j}^u (u_j(k+i|k) - u_{j\text{target}}(k+i))|^2 \right) + \rho_\varepsilon \varepsilon^2 \right\} \quad (3.4) \end{aligned}$$

med begrensningene

$$\begin{aligned} u_{j\min}(i) - \varepsilon V_{j\min}^u(i) &\leq u_j(k+i|k) \leq u_{j\max}(i) + \varepsilon V_{j\max}^u(i) \\ \Delta u_{j\min}(i) - \varepsilon V_{j\min}^{\Delta u}(i) &\leq \Delta u_j(k+i|k) \leq \Delta u_{j\max}(i) + \varepsilon V_{j\max}^{\Delta u}(i) \\ y_{j\min}(i) - \varepsilon V_{j\min}^y(i) &\leq y_j(k+i+1|k) \leq y_{j\max}(i) + \varepsilon V_{j\max}^y(i) \\ i &= 0, \dots, P-1 \\ \Delta u(k+h|k) &= 0, \quad h = M, \dots, P-1 \\ \varepsilon &\geq 0 \end{aligned}$$

P og M er lengdene til prediksjonshorizonten og kontrollhorizonten. Variabler med indeks j peker til j -te komponenten til en vektor. $(k+i|k)$ refererer til verdien predikert for tiden $k+i$ basert på informasjon tilgjengelig ved k . $r(k)$ er nåværende sampel for referansen. Når referansen ikke er kjent på forhånd blir $r(k+i+1) = r(k)$ i likning 3.4.

I MPC-regulering benyttes informasjon om fremtidig referansegang der dette praktisk mulig. Dette kalles *forutsett referansegang* (eng. *look-ahead*). Samme prinsipp benyttes dersom målt forstyrrelse $v(k)$ er av en slik natur at fremtidig forstyrrelsesforløp kan bestemmes.

$w_{i,j}^{\Delta u}$, $w_{i,j}^u$, $w_{i,j}^y$ er ikke-negative vekter for tilhørende variabel. Mindre w , gjør at tilhørende variabel blir mindre viktig for løsningen av optimeringsproblemet. Tilsvarende ved økning av w , gjør at tilhørende variabel blir mer viktig for løsning av optimeringsproblemet.

$u_{j,\min}$, $u_{j,\max}$, $\Delta u_{j,\min}$, $\Delta u_{j,\max}$, $y_{j,\min}$, $y_{j,\max}$ er minimums og maksimumsverdier for tilhørende variabel. Begrensningene mykes opp ved å bruke slakkvariabelen $\varepsilon \geq 0$. Vekten ρ_ε på slakkvariabelen straffer overskridelser av begrensningene. Mer vekt på ρ_ε i forhold til vekt på inngang og utgang, fører til mer straff ved overskridelse av begrensning.

V_{\min}^u , V_{\max}^u , $V_{\min}^{\Delta u}$, $V_{\max}^{\Delta u}$, V_{\min}^y , V_{\max}^y kalles *The Equal Concern for the Relaxation* (ECR) vektorene. Vektorene har ikke-negative føringer som tar hensyn til korresponderende begrensning. Større V gir en *mykere* begrensning. $V=0$ betyr at begrensningen er *hard*, og ikke kan overskrides.

Løsningen $\Delta u(k|k)$ er den eneste som blir brukt for å beregne $u(k)$. De gjenværende samplene, $\Delta u(k+i|k)$, blir forkastet, og et nytt optimeringsproblem løses for neste sampl.

Algoritmen implementert i *Model Predictive Control Toolbox* i Matlab, bruker forskjellige prosedyrer avhengig av begrensningene i reguleringssystemet. Om alle grenser er endelige, blir slakkvariabelen ε fjernet, og problemet i likning 3.1 løses analytisk. Ellers benyttes et kvadratisk program (QP) for å løse problemet.

Kapittel 4

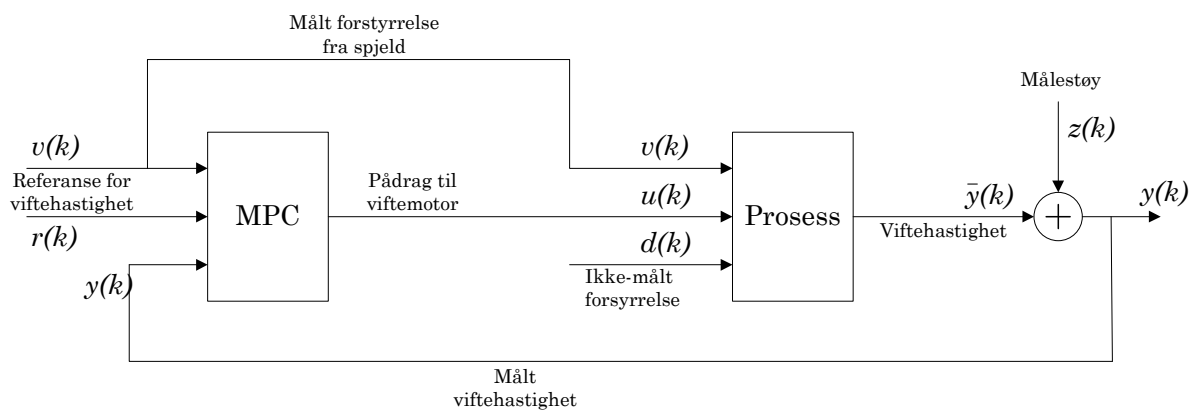
MPC-regulator for styring av lufthastighet

I dette kapittel blir MPC-regulatoren satt opp og tunet for vifteprosessen.

4.1 Utvikling av MPC-regulatoren

4.1.1 Oppsett for viftescenariet

I viftescenariet skal lufthastigheten reguleres ved hjelp av et enkelt pådrag til viftemotor. Dette er en SISO-prosess, se figur 4.1. Viften er satt opp med et spjeld som avgrenser lufttilførselen når det lukkes, og lar luften strømme fritt gjennom når det står åpent. Spjeldet forstyrrer lufthastigheten, og er i dette kapittelet satt opp som en målt forstyrrelse.



Figur 4.1: Oppsett for MPC-regulering av lufthastigheten.

4.1.2 Oppsett for MPC-regulator

MPC-regulatoren settes opp ved hjelp av et brukergrensesnitt i MPC-verktøykassen. Før regulatoren kan settes opp i brukergrensesnittet, må en modell spesifiseres. Dette gjøres ved å kjøre en m-fil.

Utarbeidelse av m-fil

Hensikten med m-filen er å konstruere transferfunksjonene i likning (2.3) og (2.4) fra kapittel 2.1, til en modell som kan brukes av MPC-regulatorene. Dette er viktig for at regulatorene skal være i stand til å utføre nødvendige estimeringer, som brukes for å løse optimeringsproblemet i kapittel 3.2.5. Tabell 4.1 viser innholdet i filen som oppretter modellen for viftescenariet.

Tabell 4.1: m-fil for generering av modell.

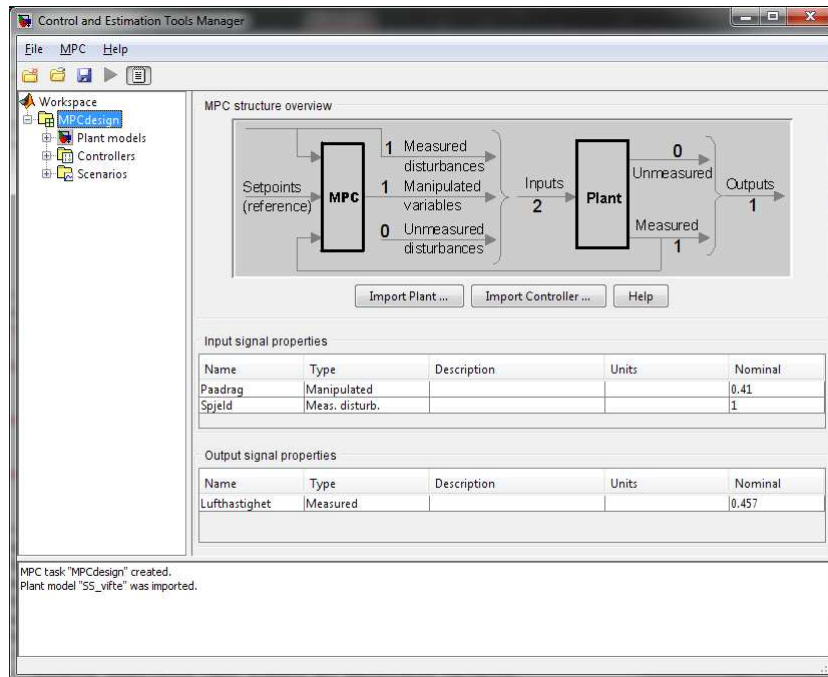
```
19 % Viftehastighet
20 % Transferfunksjon fra pådrag til utgang
21 Y_U = tf([0.338], [1.52 1], 'OutputDelay', 0.06);
22
23 % Transferfunksjon fra målt forstyrrelse til utgang
24 Y_V = tf([-0.045], [1.072 1], 'OutputDelay', 0.6);
25
26 % Setter sammen til MISO
27
28 SS_vifte = [Y_U Y_V];
29 SS_vifte.InputName = {'Paadrag', 'Spjeld'};
30 SS_vifte.OutputName = {'Lufthastighet'};
31 SS_vifte = setmpcsignals(SS_vifte, 'MD', 2);
```

Kommandoene er hentet fra [2], og er ferdigkonstruerte kommandoer som inngår i MPC-verktøykassen. Linje 21 og 24 setter opp transferfunksjonene til pådraget og forstyrrelsen. Linje 28 setter sammen de to transferfunksjonene til én enkelt modell. I linje 29 og 30 settes navnene på de ulike inngangene og utgangen. Dette er nå satt opp som en MISO-prosess (*Multiple-Input-Single-Output*), dvs. to pådrag og én utgang. Linje 31 ordner dette ved å definere den ene inngangen som målt forstyrrelse, mens den andre inngangen blir antatt til å være pådrag når m-filen kjøres.

Brukergrensesnittet i MPC-verktøykassen

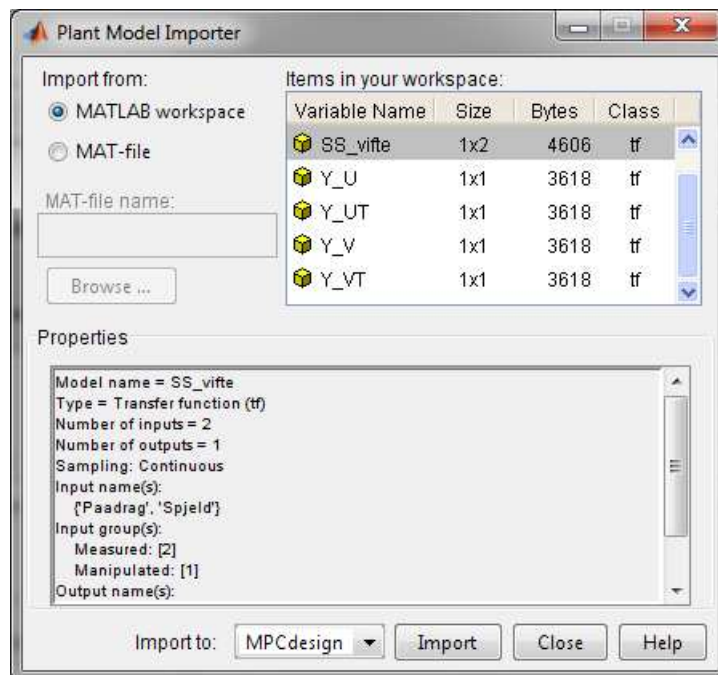
Brukergrensesnittet for å sette opp MPC-regulatoren åpnes ved å skrive kommandoen "mpctool" i kommandovinduet i Matlab.

I brukergrensesnittet, se figur 4.2, settes MPC-regulatoren opp ved å laste inn prosessmodell og spesifisere regulatorparametre.



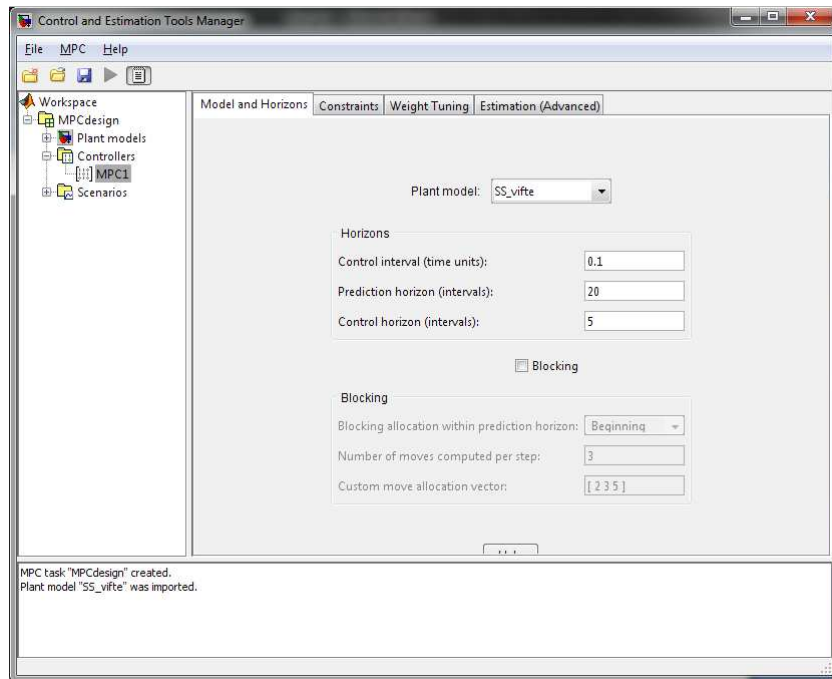
Figur 4.2: Brukergrensesnitt for oppsett av MPC-regulatoren.

Prosessmodellen lastes inn ved å trykke "Import Plant", se figur 4.3. I figur 4.2 er prosessmodellen lastet inn i grensesnittet. Strukturoversikten viser hvor mange innganger og utganger som skal brukes, og om disse er målte eller ikke-målte. Nominell startverdi spesifiseres for aktuelle innganger og utganger i dette vinduet.



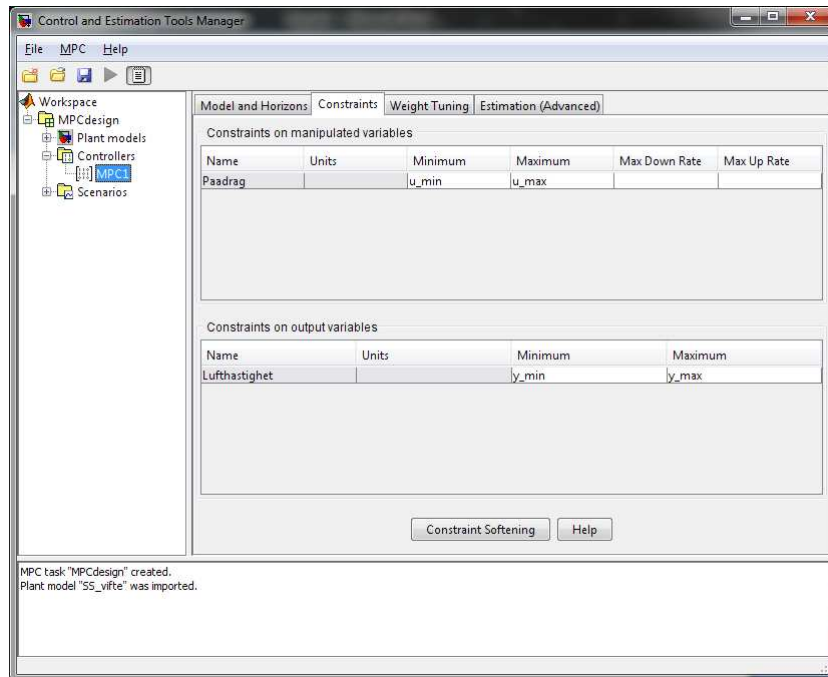
Figur 4.3: Innlasting av modellprosess.

I figur 4.3 ble modellen for viftescenariet, *SS_vifte*, lastet inn. Når modellen er lastet inn, kan regulatorens parametere (kontroll- og prediksjonshorisont, vekt på inngang, etc.) spesifiseres. Figur 4.4 viser vinduet der regulatorparameterne ble bestemt.



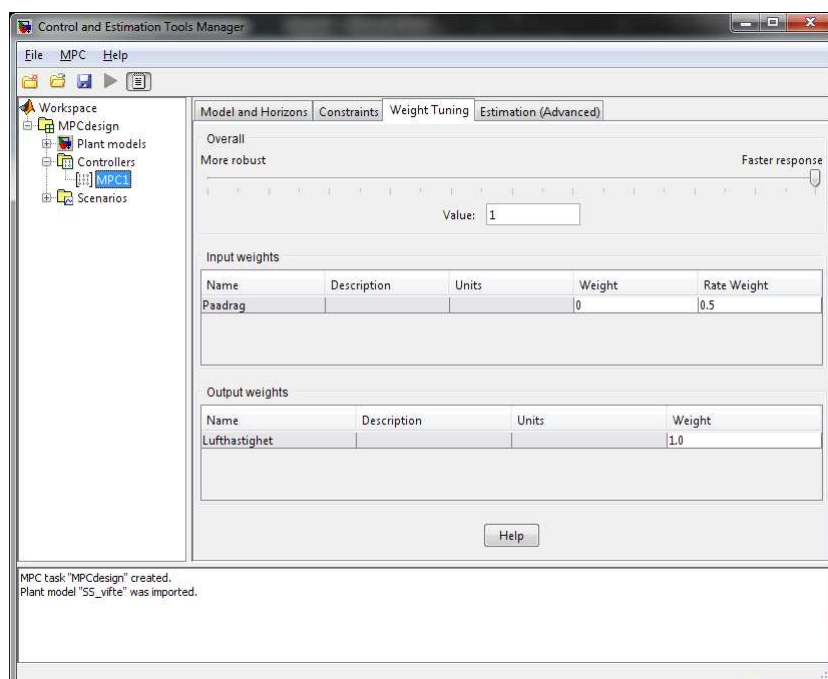
Figur 4.4: Innstilling av MPC-regulator.

Figur 4.4 viser brukergrensesnittet for spesifisering av kontrollhorisont, prediksjonshorisont og kontrollintervall. Alternativer for blokkering bestemmes også i dette vinduet. Dette alternativet er ikke utprøvd i denne rapporten. Her ble det valgt et kontrollintervall på 0.1 sekund, prediksjonshorisont med lengde tilsvarende 20 kontrollintervaller, og kontrollhorisont lik 5 kontrollintervaller. I figur 4.5 ble systemets begrensninger definert.



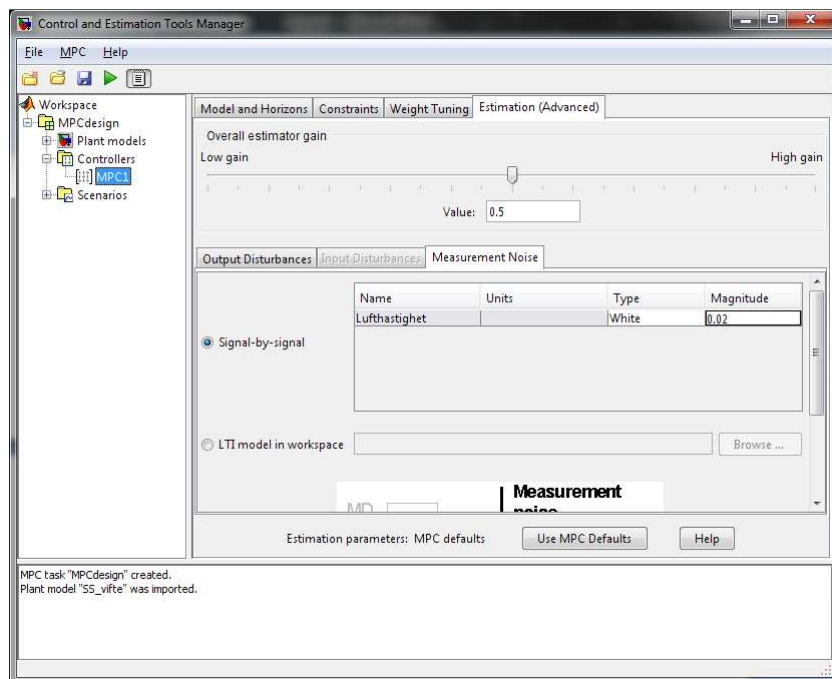
Figur 4.5: Innstillinger for systemets begrensninger.

Maksimums- og minimumsverdier for innganger og utganger ble definert ved bruk av grensesnittet i figur 4.5. Her ble begrensningene for inngang satt til u_{min} og u_{max} , og y_{min} og y_{max} for utgang. Raten for inngangsendringer kan spesifiseres i rutene "Max Down Rate" og "Max Up Rate". For å bestemme mykheten for hver enkelt begrensning benyttes knappen "Constraint Softening". Begrensningene ble satt opp som harde begrensninger. Vekt på innganger og utganger ble bestemt i figur 4.6.



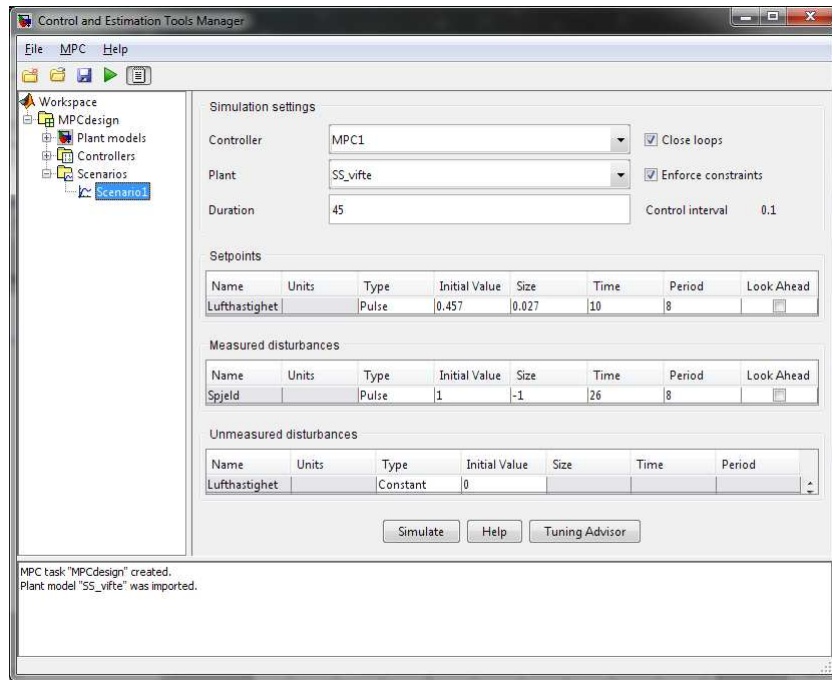
Figur 4.6: Bestemmelse av vektorer på innganger og utganger.

Vektene $w_{i,j}^{\Delta u}$, $w_{i,j}^u$, $w_{i,j}^y$ fra kapittel 3.2.5 blir satt i figur 4.6. "Input weight", $w_{i,j}^u$, bestemmer prioritering for et system med flere innganger. Høy verdi på vekt gjør at tilhørende vekt varierer lite rundt nominell verdi, lav verdi på vekt gjør at tilhørende vekt kan variere mer omkring nominell verdi. "Input weight rate", $w_{i,j}^{\Delta u}$, bestemmer hvor stor endring tilhørende inngang kan utføre per kontrollintervall. Høyverdi vekt gir mindre endringer, og tregere regulering. Lavverdi vekt gir større endringer, og raskere regulering. "Output weight", $w_{i,j}^y$, angir hvilke utganger som skal prioriteres. MPC-regulatoren vektlegger at reguleringsavviket skal være mindre for utganger med høyere verdi på vekt. Forstyrrelser og støy i systemet angis ved brukergrensesnittet i figur 4.7.



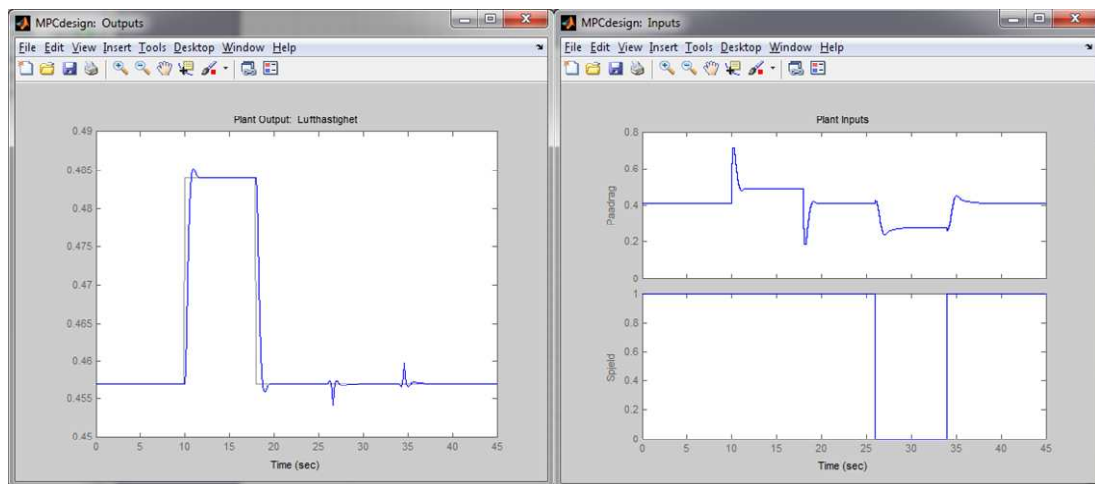
Figur 4.7: Spesifikasjon av forstyrrelse og støy i brukergrensesnitt.

Vinduet i figur 4.7 gir muligheter for å ta med forstyrrelser på utgang og målestøy, slik at dette tas hensyn til i reguleringen. Her er det antatt en hvit målestøy med amplitude 0.02. Den valgte prosessen kan testes med de valgte parametrene i grensesnittet, figur 4.8.



Figur 4.8: Simulering og testing i brukergrensesnitt.

Når ønsket prosess er lastet inn og regulatorparametre er satt, kan reguleringsystemet simuleres og testes. Scenariet settes opp i grensesnittet i figur 4.8. Her er simuleringen satt opp med et sprang i referanse ved $t = 10$, og et sprang i målt forstyrrelse ved $t = 24$. Boksene "Look Ahead" velges om regulatoren skal ha forhåndsinformasjon om sprangene, og regulere pådrag i henhold til disse. Ved å trykke "Simulate", simuleres systemet og viser plot for innganger og utganger i nye vinduer, se figur 4.9.

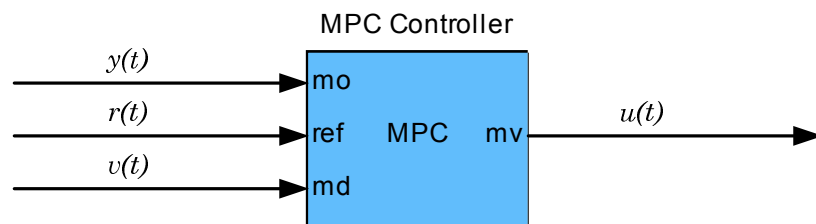


Figur 4.9: Resultat av simulering i grensesnitt.

Når ønsket respons er oppnådd, lagres innstillingen for prosess og parameterkonfigurasjon i mat-fil³ som brukes i MPC-blokk i Simulink, se kapittel 4.1.3. I figur 4.9 tolker MPC-regulatoren sprang i spjeld fra 1 til 0, som en åpning av spjeldet. Dette blir tatt hensyn til i implementering av MPC-blokk i Simulink.

4.1.3 MPC-blokk i Simulink

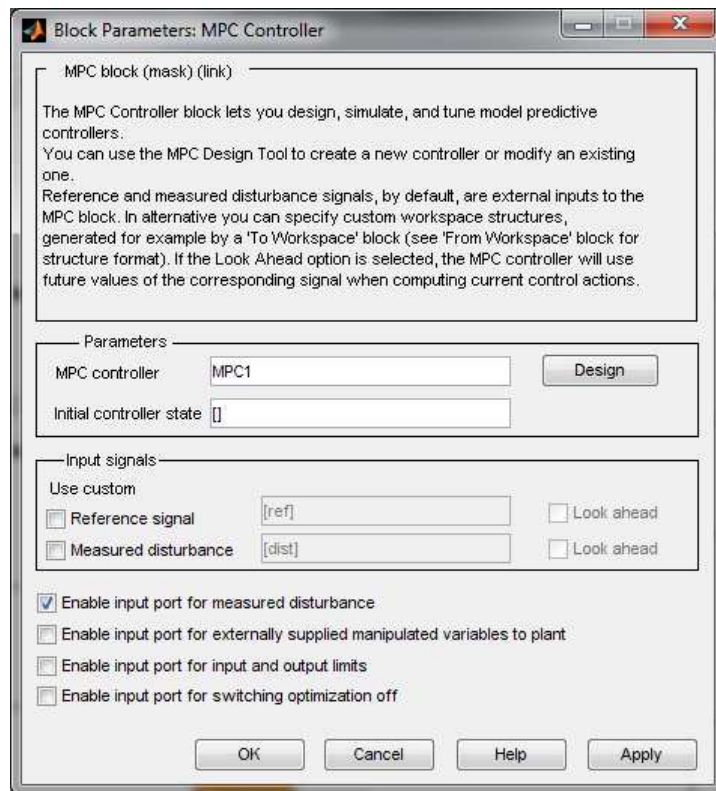
MPC-blokken representerer MPC-regulatoren i Simulink. Regulatoren blir konfigurert ved hjelp av et grafisk brukergrensesnitt, hvor modell til bruk for estimering blir lastet opp, og parametre til bruk for optimering blir satt. MPC-blokken er vist i figur 4.10.



Figur 4.10: MPC-blokk i Simulink.

Ved å dobbeltklikke på blokken vises parametrene for MPC-blokken, figur 4.11.

³ mat-filer er Matlab sitt filformat for lagring av data.



Figur 4.11: Parametre for MPC-blokk.

Prosessmodell og parameterkonfigurasjon blir definert ved å navngi mat-fil som ble lagret i det grafiske brukergrensesnittet, kapittel 4.1.2, i feltet "MPC controller". Reguleringsparameterne kan endres ved å trykke "Design".

For å bruke forhåndsinformasjon om fremtidige verdier for referanse og/eller målt forstyrrelse velges "Look ahead"-boksen tilhørende signal.

Har reguleringsystemet målt forstyrrelse, er det viktig at boksen for dette ("Enable input port for measured disturbance") er valgt. MPC-blokken, figur 4.10, vil dermed ha en inngang for målt forstyrrelse.

I utviklingen av MPC-regulatoren i Simulink ble det imidlertid oppdaget en feil i Matlab versjonen (Matlab R2009a) som ble benyttet. Dette medførte at en eldre versjon av Matlab (Matlab R2007b) ble benyttet for å utvikle MPC-regulatorene. Feilen i Matlab R2009a er dokumentert i vedlegg A.

4.2 Parametertuning for MPC-regulator uten forhåndsinformasjon

I dette kapitlet testes reguleringsystemet, uten forhåndsinformasjon om referanse- og forstyrrelsesforløp, med forskjellige verdier på følgende reguleringsparametre:

- Prediksjonshorizonten, P
- Kontrollhorizonten, M
- Inngangsvekt, $w^{\Delta u}$
- Utgangsvekt, w^y
- Kontrollintervall, Δt

Parameterendringene er testet både på modell og virkelig prosess.

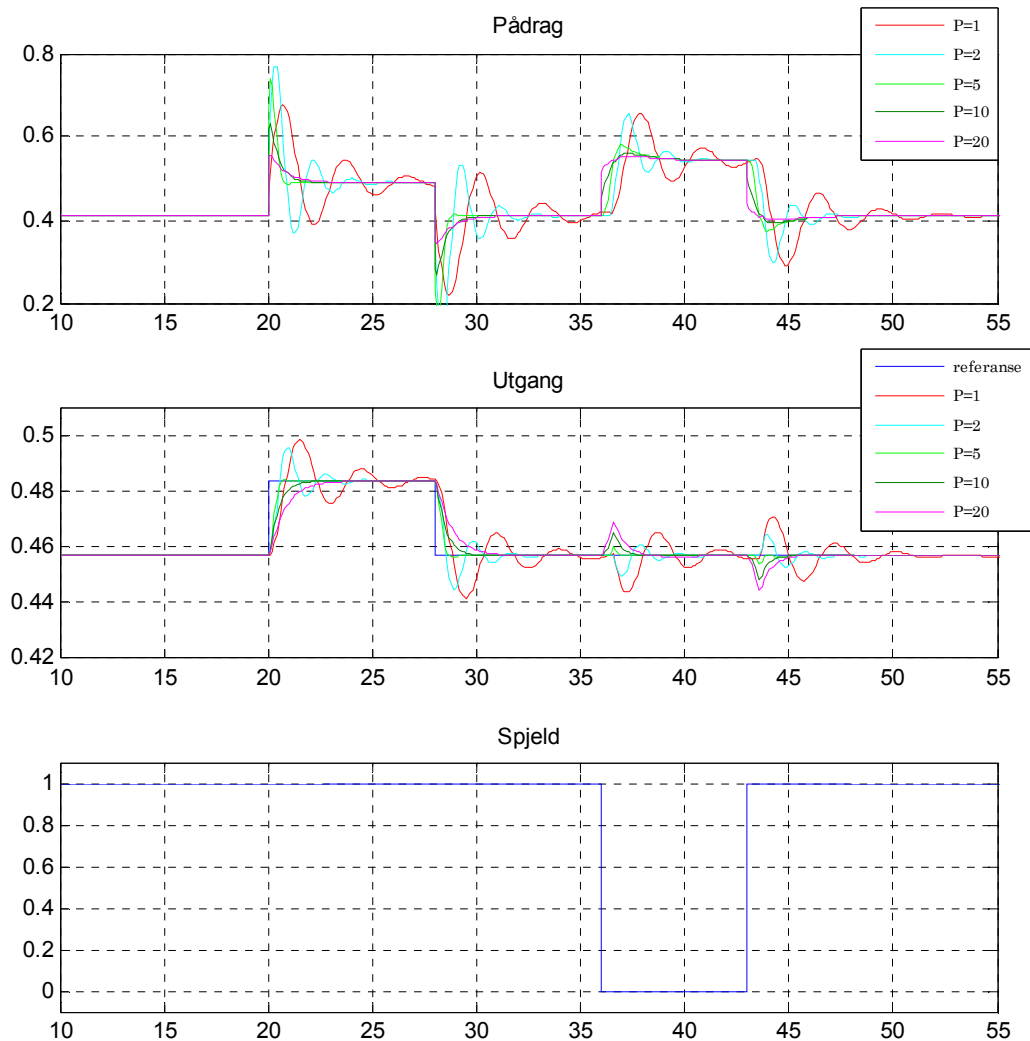
4.2.1 Endring av prediksjonshorizont, testet på modell

Modellen reguleres med parameterverdier vist i tabell 4.2. Prediksjonshorizonten varieres fra hver simulering, mens de andre parametrene like.

Plot i figur 4.12 er plottet med forskjellige verdier for prediksjonshorizonten, P . De forskjellige verdiene som er brukt er $P = [1 \ 2 \ 5 \ 10 \ 20]$.

Tabell 4.2: Parameterverdier brukt ved simulering.

Parameter	Verdi
Prediksjonshorizonten, P	Varies
Kontrollhorizonten, M	1
Inngangsvekt, $w^{\Delta u}$	0.5
Utgangsvekt, w^y	1.0
Kontrollintervall, Δt	0.1



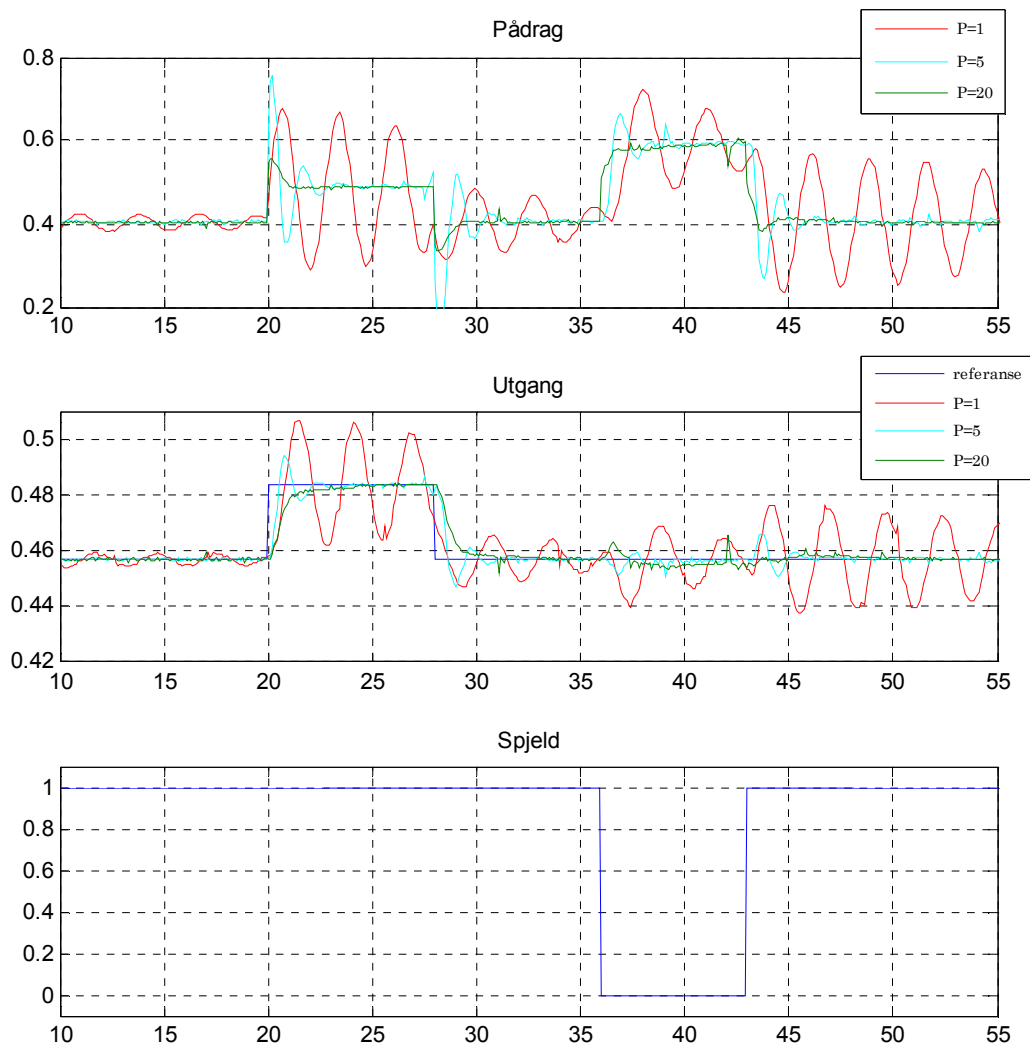
Figur 4.12: Endring av prediksjonshorison, testet på modell.

Observasjoner fra figur 4.12:

Merk at kontrollhorisonten $M = 1$, og ikke spesifisert etter tommelfingerregel i kapittel 3.2.2. Det er derfor viktig å være klar over forholdet mellom M og P i dette tilfellet. Fra figuren ses at regulering med kort prediksjonshorison ($P = 1$ og $P = 2$) har oversving og forholdsvis treg innsvingning. Dette er fordi regulatoren ikke ser langt nok frem i tid, og regulatorens pådragsendring fører til relativt store oversving i utgang. Ved motsatt tilfelle, hvor prediksjonshorison er lang ($P = 10$ og $P = 20$) har utgangen ikke noe oversving. Innsvingningen er imidlertid noe treg, som følge av at regulatoren ser for langt frem i tid, og pådragsendringen blir for slakk og forsiktig. Best regulering oppnås ved $P = 5$. Utgangens innsvingning er rask og uten oversving. Med forholdet $M = 1$ og $P = 5$, stemmer dette godt overens med tommelfingerregler fra kapittel 3.2.2. Ved forstyrrelsen kompenseres de hissige regulatorene, med kort prediksjonshorison, dårligere. Denne kompenseringen er likevel ikke betydelig dårligere enn for regulator med $P = 5$.

4.2.2 Endring av prediksjonshorizont, testet på virkelig prosess

Figur 4.13 viser responser fra variert lengde på prediksjonshorizont, der virkelig prosess reguleres. Horisonten varieres med størrelsene [1 5 20], mens de andre parametrene er som spesifisert i tabell 4.2.



Figur 4.13: Endring av prediksjonshorizont, testet på virkelig prosess.

Observasjoner fra figur 4.13:

Fra figur 4.13 ses at responser fra virkelig prosess viser tilsvarende dynamikk som ved simulering, figur 4.12. Regulator for virkelig prosess er imidlertid mer sensitiv for korte prediksjonshorisonter. $P = 1$ gir stående svingninger for virkelig prosess, mens samme prediksjonshorizont kun gir oversving for simuleringen. $P = 5$ for virkelig prosess ser ut til å tilsvare $P = 2$ for simulert forløp. I dette

tilfellet gir $P = 5$ god regulering og har gode kompenseringsegenskaper. Om innsvingningen ikke skulle hatt oversving, ville en lengde på $P = 10$, vært ideelt.

For et tilfelle hvor kontrollhorisont vil være satt slik tommelfingerregel i kapittel 3.2.2 er oppfylt (M med størrelse 3 – 5), vil en prediksjonshorisont med størrelse $P = 25$, gi god regulering.

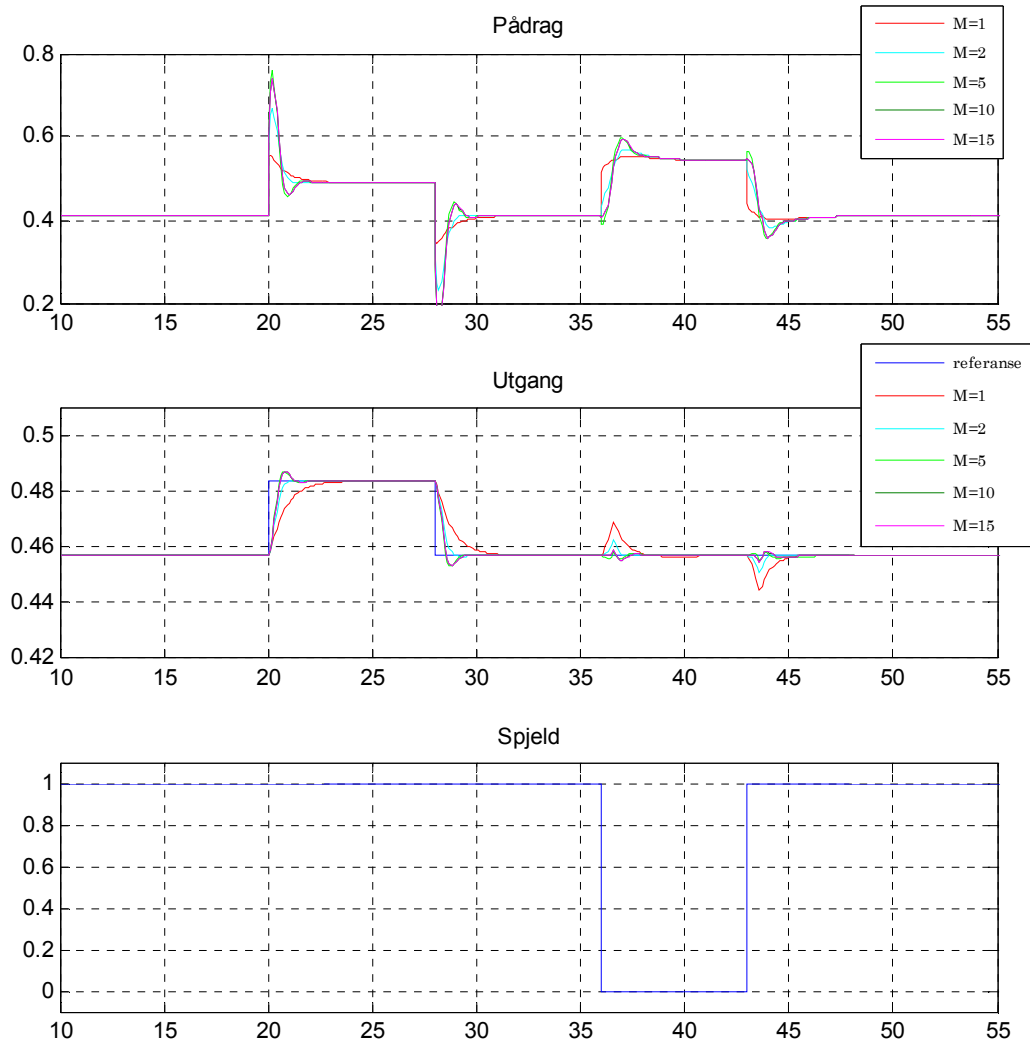
4.2.3 Endring av kontrollhorisont, testet på modell

Modellen er regulert med parameterverdier i tabell 4.3. Kontrollhorisonten varieres fra hver test, mens de andre parameterverdiene holdes konstante under alle testene.

Figur 4.14 viser simulert respons med varierende kontrollhorisont. Horisonten varieres med størrelsene [1 2 5 10 15].

Tabell 4.3: Parameterverdier brukt ved simulering.

Parameter	Verdi
Prediksjonshorisonten, P	20
Kontrollhorisonten, M	Varies
Inngangsvekt, w^{Au}	0.5
Utgangsvekt, w^y	1.0
Kontrollintervall, Δt	0.1



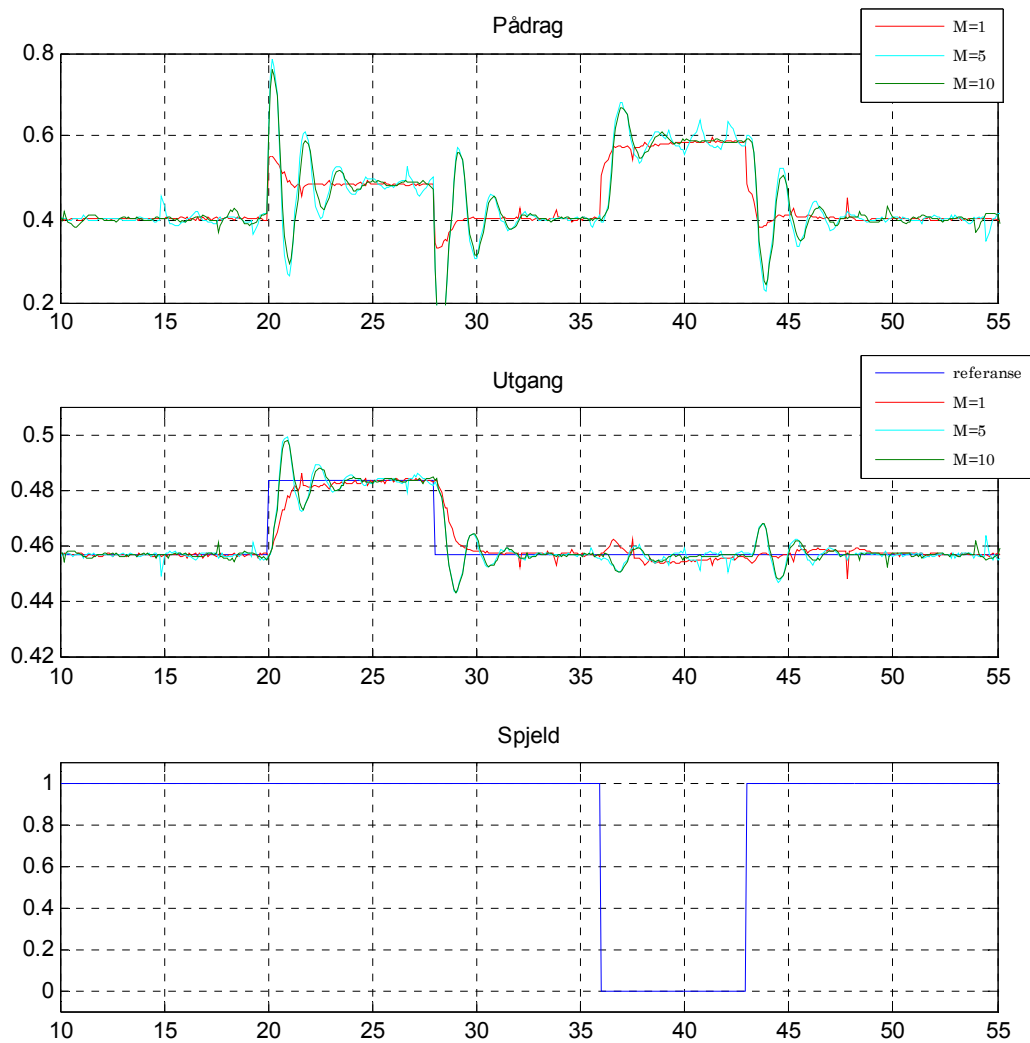
Figur 4.14: Endring av kontrollhorisont, testet på modell.

Observasjoner fra figur 4.14:

Respons fra kontrollhorisont med kort lengde, $M = 1$, bruker lang tid på å svinge seg inn mot referanse. Kompenseringsegenskapene for regulator med denne lengden er heller ikke god. For responsene med lengde $M \geq 5$ er det liten forskjell, $M = 10$ og $M = 15$ er tilnærmet identiske. Responsene har litt oversving, men følgeegenskapene er bra. Kompenseringsegenskapene er også betydelig bedre enn for $M = 1$. Regulator med $M = 2$ viser best resultat. Regulatoren følger referansen litt bedre enn for regulatorene med $M \geq 5$. Kompenseringen for forstyrrelse er nesten like god for regulatorene med $M \geq 2$.

4.2.4 Endring av kontrollhorisont, testet på virkelig prosess

Parametere fra tabell 4.3 benyttes for regulering av virkelig prosess, se figur 4.15. Kontrollhorisonten varieres med lengdene [1 5 10].



Figur 4.15: Endring av kontrollhorisont, testet på virkelig prosess.

Observasjoner fra figur 4.15:

Observerer at responsen fra virkelig prosess varierer fra simulert respons i figur 4.14. Regulator med kontrollhorisont, $M \geq 5$, har litt dårligere kompenseringsegenskaper. Følgeegenskapene er også dårligere, der utgangen bruker lengre tid for å svinge seg inn mot referanseverdi.

En kontrollhorisont med lengde $M = 3$, ville gitt bedre for regulering.

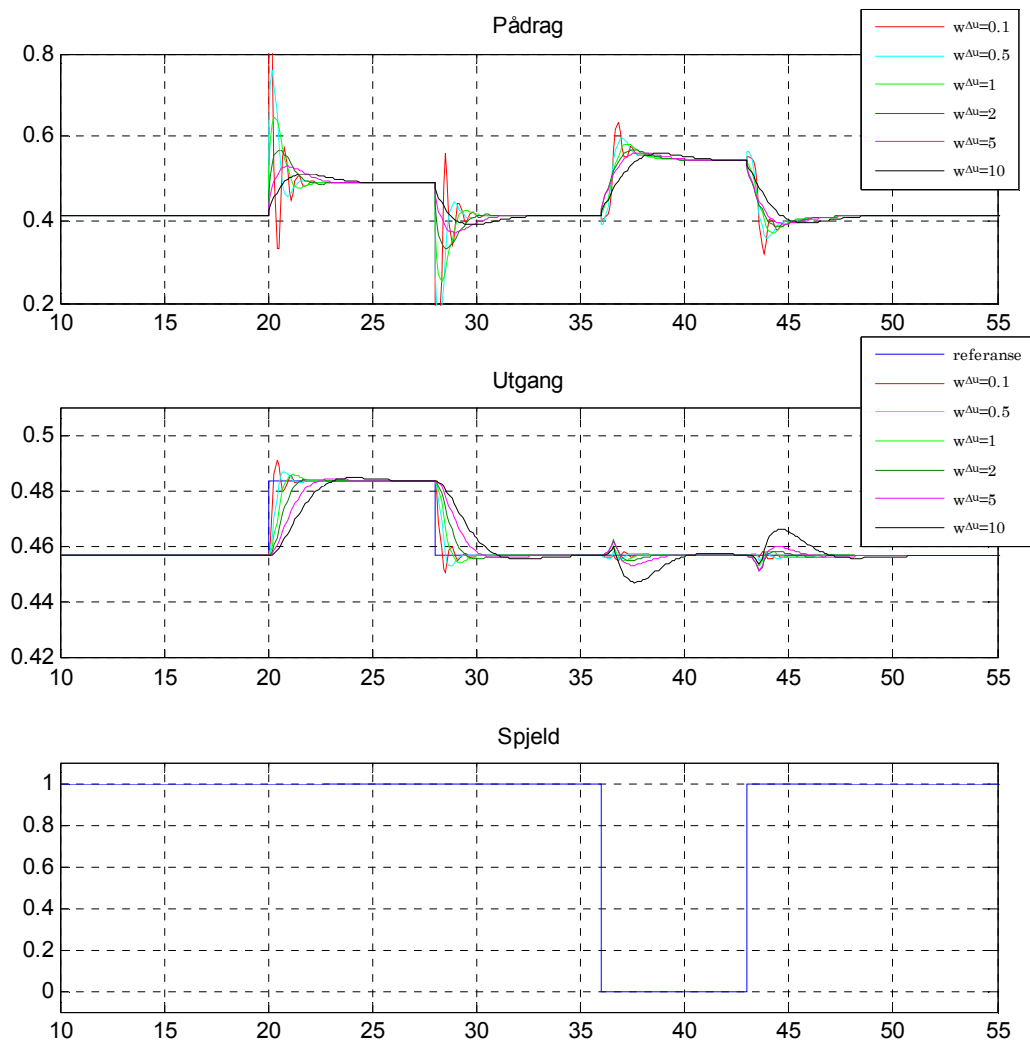
4.2.5 Endring av inngangsvekt, testet på modell

Parameterverdier i tabell 4.4 brukes for å regulere modellen. Inngangsvekt varieres, mens de andre parametrene er like for hver test.

Inngangsvektene varieres for hver test, se plot i figur 4.16. Vektene varieres med følgende størrelser, [0.1 0.5 1 2 5 10].

Tabell 4.2: Parameterverdier brukt ved simulering.

Parameter	Verdi
Prediksjonshorisonten, P	20
Kontrollhorisonten, M	5
Inngangsvekt, $w^{\Delta u}$	Varies
Utgangsvekt, w^y	1.0
Kontrollintervall, Δt	0.1



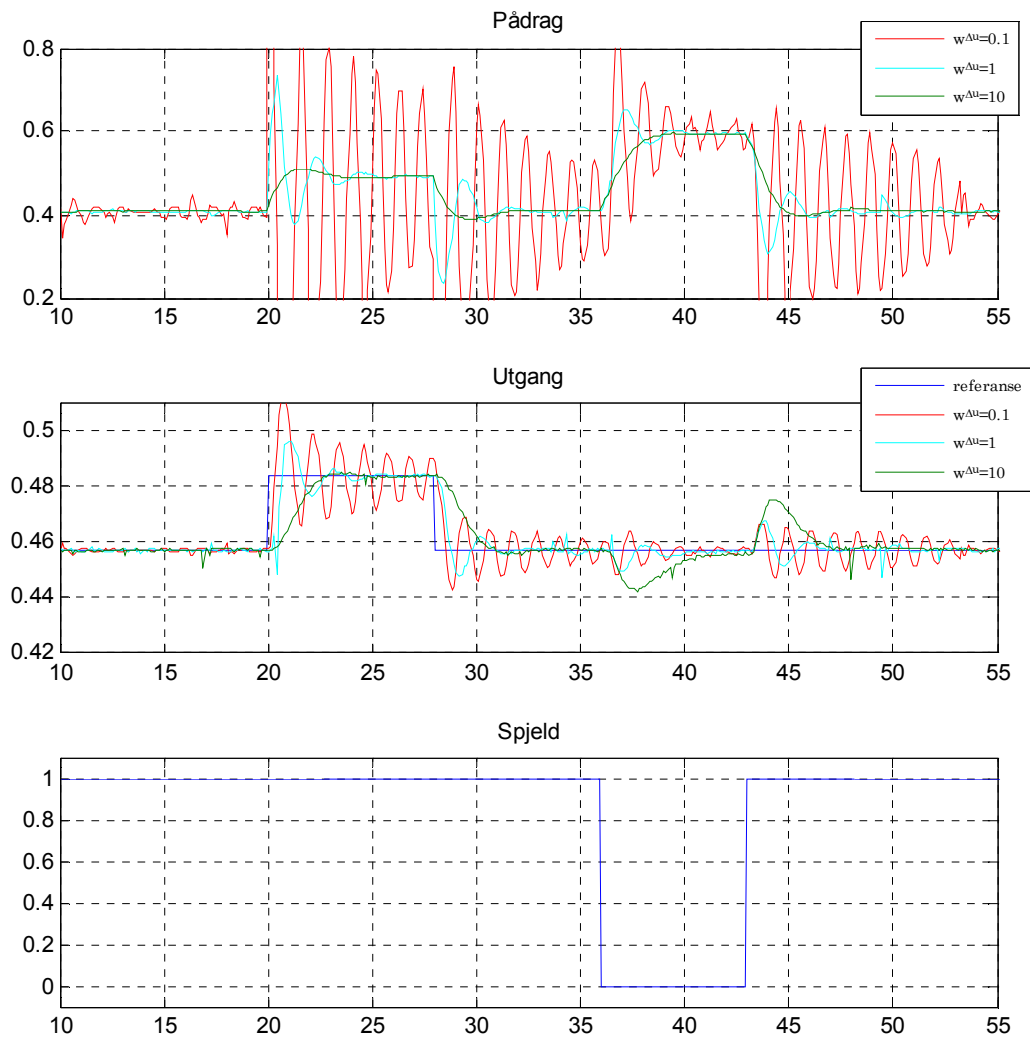
Figur 4.16: Endring av inngangsvekt, testet på modell.

Observasjoner fra figur 4.16:

For vekter med mindre verdi tillater regulatoren større inngangsendringer (pådragsendringer). Dette gjør at pådraget kan endres med større effekt, og innsvingning mot referanse og kompensering for forstyrrelse, blir bedre. Med veldig liten vekt på inngang, kan imidlertid pådragsendringer blir for store. Dette kan føre til at reguleringen blir ustabil, eller at reguleringsutstyr slites unødvendig. Fra figuren ses at selv om $w^{\Delta u} = 0.1$ har mindre vekt enn $w^{\Delta u} = 0.5$, har de to regulatorne like rask innsvingning etter sprang i referanse og forstyrrelse. Vekter med relativt høy verdi, tillater kun små pådragsendringer. Reguleringen blir tregere og kompenserer dårligere for forstyrrelse.

4.2.6 Endring av inngangsvekt, testet på virkelig prosess

I figur 4.17 er regulator med varierte inngangsvekter testet på virkelig prosess. Regulatorparametrene er som spesifisert i tabell 4.4. Inngangsvektene endres med følgende vekter [0.1 1 10].



Figur 4.17: Endring av inngangsvekter, testet på virkelig prosess.

Observasjoner fra figur 4.17:

Fra figur ses at regulering med for liten vekt, $w^{\Delta u} = 0.1$, fører til regulering med stående svingninger. Dette viser hvorfor det er viktig at vekten ikke har for lav verdi. Regulatoren kompenserer likevel godt for forstyrrelsen. Med $w^{\Delta u} = 10$ er innsvingningen for sprang i referanse og forstyrrelsen treg,

som følge av slakk pådragsendring i inngang. $w^{\Delta u} = 1$ har bedre regulering, med gode følge- og kompenseringsegenskaper.

Vekt på inngang er avhengig av vekt på utgang. Respons i figur 4.17 viser at et valg der det ikke er for stor forskjell i forhold mellom inngangs- og utgangsvekt vil være fornuftig.

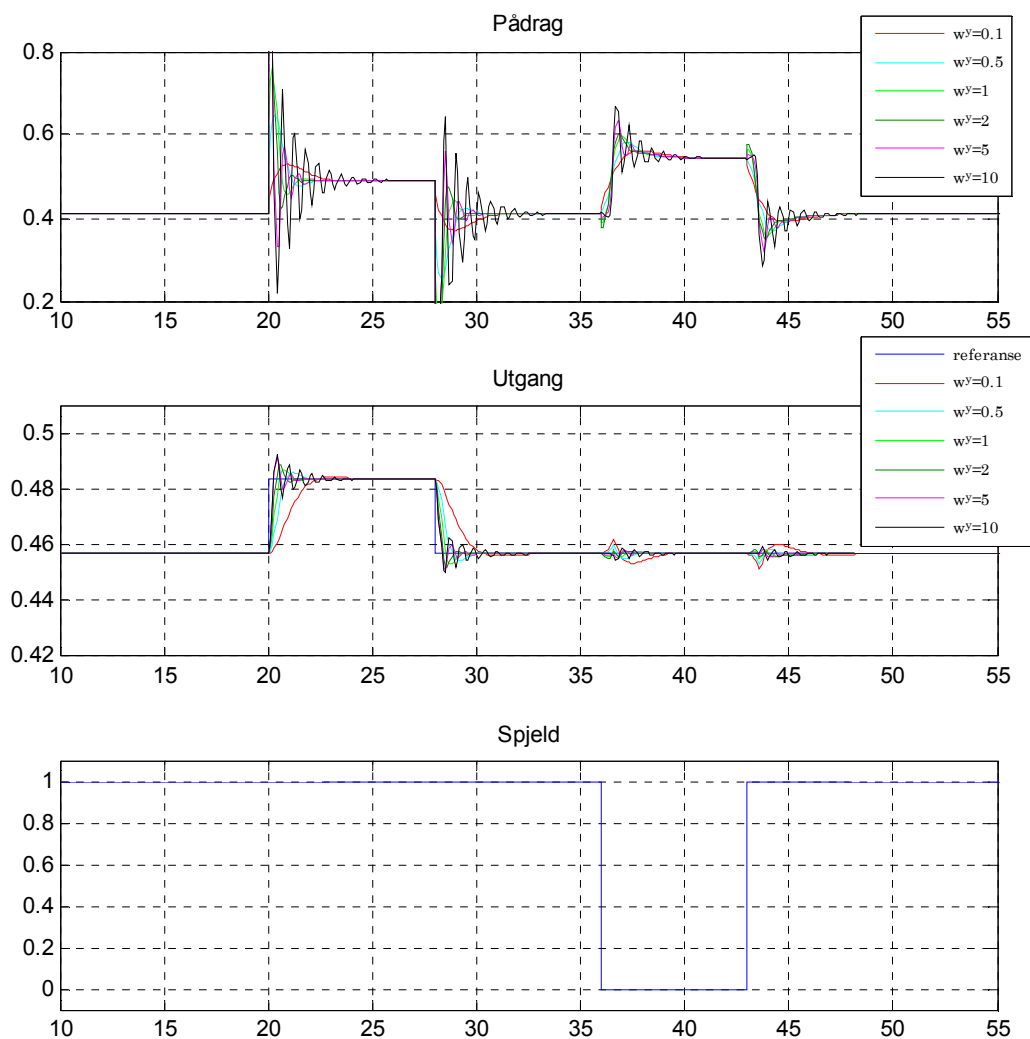
4.2.7 Endring av utgangsvekt, testet på modell

Parametrene i tabell 4.5 for å simulere regulering av prosessen. Utgangsvekten varieres med vektene [0.1 0.5 1 2 5 10].

Figur 4.18 viser resultat fra simulert regulering, hvor utgangsvektene endres.

Tabell 4.3: Parameterverdier brukt ved simulering.

Parameter	Verdi
Prediksjonshorisonten, P	20
Kontrollhorisonten, M	5
Inngangsvekt, $w^{\Delta u}$	0.5
Utgangsvekt, w^y	Varies
Kontrollintervall, Δt	0.1



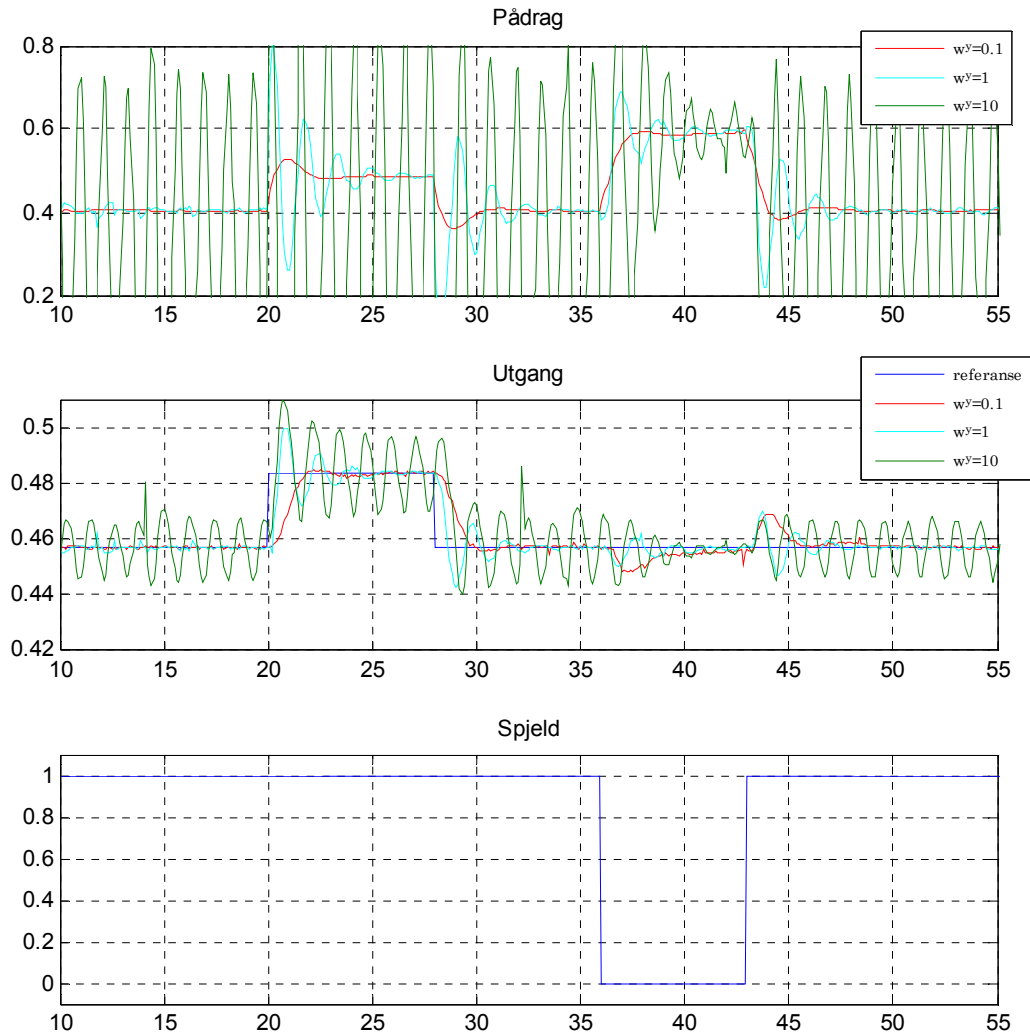
Figur 4.18: Endring av utgangsvекter, testet på modell.

Observasjoner fra figur 4.18:

For vektorer med høy verdi, prøver regulatoren å minske reguleringsavviket i større grad. Dette kan i likhet med for liten vekt på inngangsvekt, føre til ustabilitet om vekten er for stor. Det ses fra figuren at regulering med vekt $w^y = 10$, har nesten like gode følge- og kompenseringsegenskaper som regulering med utgangsvekt $w^y = 5$. For regulering lavverdi vekt blir innsvingning, som følge av sprang i referanse og forstyrrelse, tregere.

4.2.8 Endring av utgangsvekt, testet på virkelig prosess

I figur 4.19 er parametrene fra tabell 4.5 brukt til å regulere prosessen, med varierende vekt på utgang. Utgangsvektene som er brukt er [0.1 1 10].



Figur 4.19: Endring av utgangsvekter, testet på virkelig prosess.

Observasjoner fra figur 4.19:

Vekt med høy verdi, $w^y = 10$, fører til reguleringen får stående svingninger etter sprang i referanse. Kompenseringsegenskapene er likevel gode. Regulering med vekt $w^y = 1$, har bedre følgeegenskaper, og kompenserer bra for forstyrrelsen. Den beste følgeegenskapen oppnås ved regulering med $w^y = 0.1$. På bekostning av bedre følgeegenskaper, blir kompenseringsegenskapene dårligere.

Respons fra figur 4.19 viser at det vil være hensiktsmessig med et mindre forhold mellom inngangs- og utgangsvekt. Vektene $w^y = 1.1$ og $w^A = 0.7$ vil gi god regulering.

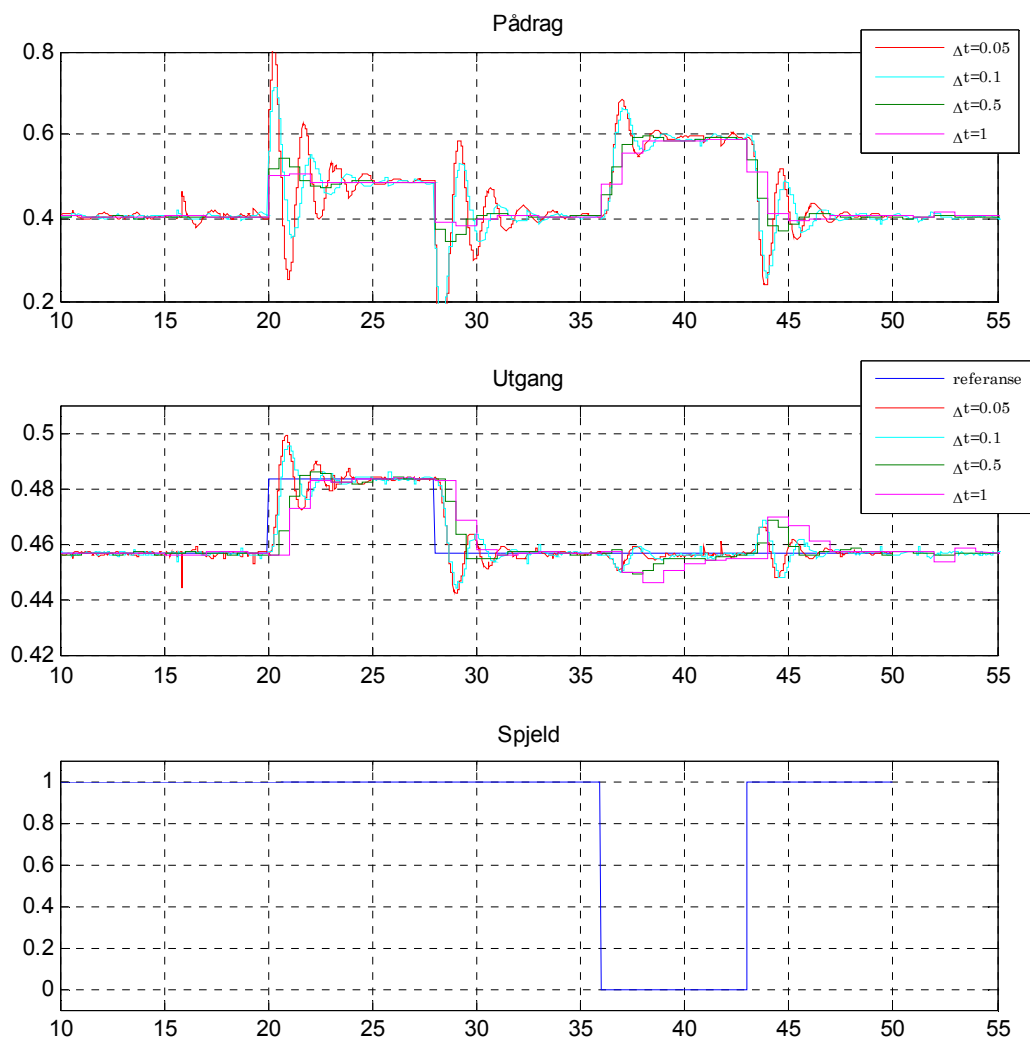
4.2.9 Endring av kontrollintervall, testet på virkelig prosess

For å se innvirkning av mindre og større kontrollintervall, er regulator med varierte kontrollintervall testet på vifteprosessen. Andre parametere er gitt i tabell 4.6.

I figur 4.20 endres kontrollintervallet med størrelsene [0.05 0.1 0.5 1].

Tabell 4.4: Parameterverdier brukt ved simulering.

Parameter	Verdi
Prediksjonshorisonten, P	20
Kontrollhorisonten, M	5
Inngangsvekt, $w^{\Delta u}$	0.5
Utgangsvekt, w^y	0.7
Kontrollintervall, Δt	Varies



Figur 4.20: Endring av kontrollintervall, testet på virkelig prosess.

Observasjoner fra figur 4.20:

Som antatt vil regulering med for stort kontrollintervall følge utgangen dårligere, på grunn av endringer ikke gjøres like ofte. På en annen side følger likevel regulering med $\Delta t = 1$, prosessen til en viss grad, med betydelig mindre ressurser enn regulering med $\Delta t = 0.1$.

Det er liten forskjell på regulering med $\Delta t = 0.05$ og regulering med $\Delta t = 0.1$. Ettersom disse har lik respons, velges kontrollintervallet som medfører minst ressursbruk, $\Delta t = 0.1$.

4.2.10 Oppsummering av tester

Testene som er utført tilsier at følgende reguleringsparametere vil gi god regulering for MPC-regulator uten forhåndsinformasjon om fremtidig referanse- og forstyrrelsesforløp:

- Prediksjonshorisont, $P = 25$
- Kontrollhorisont, $M = 3$
- Utgangsvekt, $w^{\Delta u} = 0.7$
- Inngangsvekt, $w^y = 1.2$
- Kontrollintervall, $\Delta t = 0.1$

4.3 Parametertuning for MPC-regulator med forhåndsinformasjon

I dette kapittel ble MPC-regulator, med forhåndsinformasjon om referanse- og forstyrrelsesforløp, testet med forskjellige verdier på følgende reguleringsparametre:

- Prediksjonshorisonten, P
- Kontrollhorisonten, M

Parameterendringene er testet både på modell og virkelig prosess.

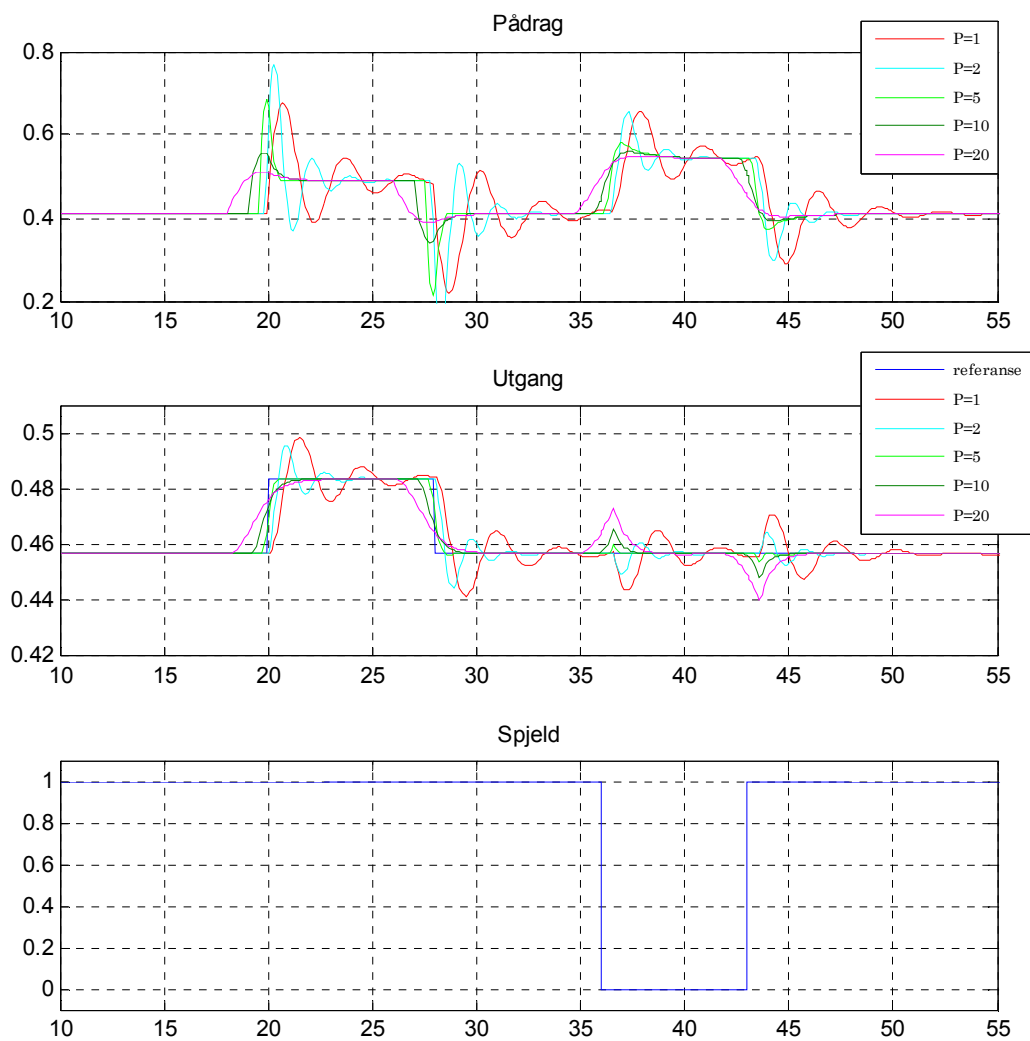
4.3.1 Endring av prediksjonshorisont, testet på modell

Modellen ble regulert med parameterverdier vist i tabell 4.7. Prediksjonshorisonten varieres fra hver simulering, mens de andre parametrene like.

Plot i figur 4.21 er plottet med forskjellige verdier for prediksjonshorisonten, P . De forskjellige verdiene som er brukt er $P = [1 \ 2 \ 5 \ 10 \ 20]$.

Tabell 4.5: Parameterverdier brukt ved simulering.

Parameter	Verdi
Prediksjonshorisonten, P	Varies
Kontrollhorisonten, M	1
Inngangsvekt, $w^{\Delta u}$	0.5
Utgangsvekt, w^y	1.0
Kontrollintervall, Δt	0.1



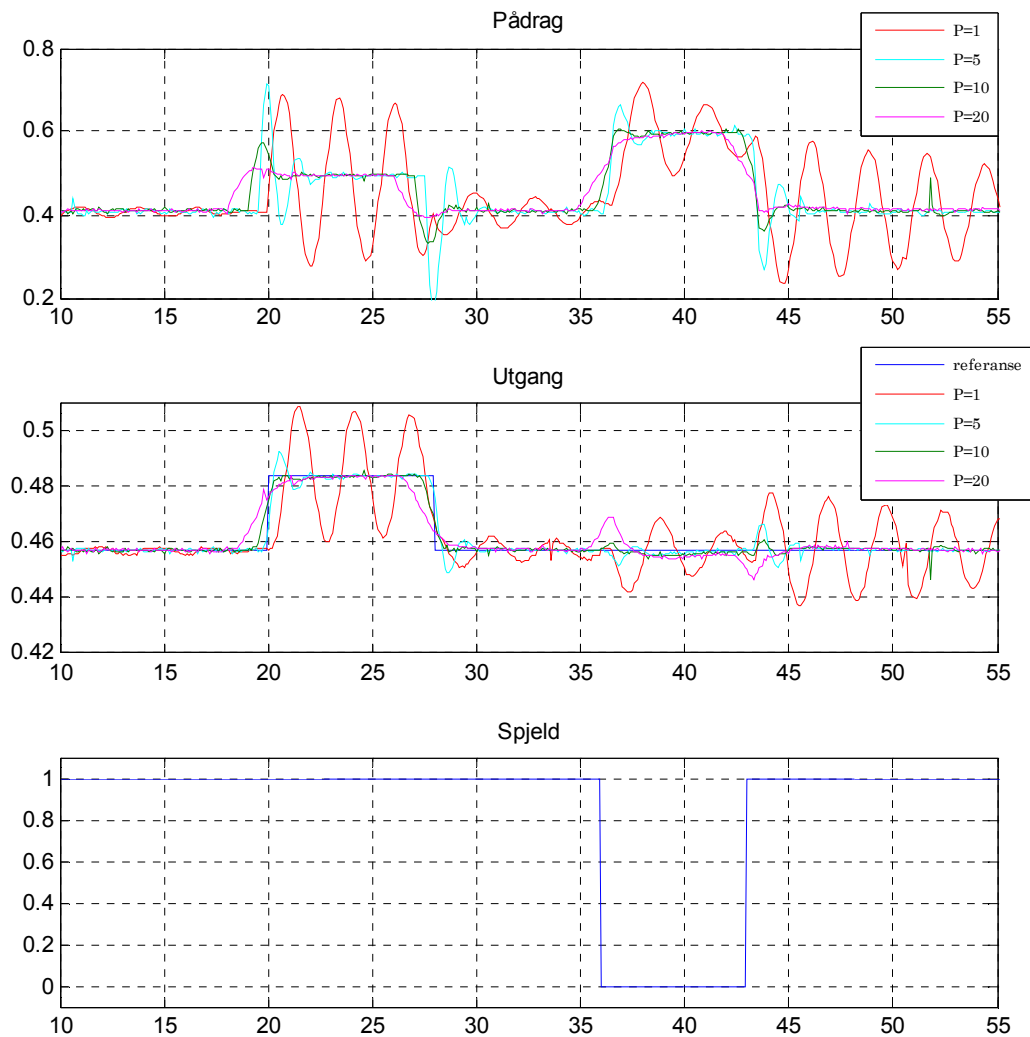
Figur 4.21: Endring av prediksjonshorisont, testet på modell.

Observasjoner fra figur 4.21:

Merk at kontrollhorisonten, $M = 1$, og ikke etter tommelfingerregel i kapittel 3.2.2. Det er derfor viktig å være klar over forholdet mellom M og P i dette tilfellet. Regulering med lang prediksjonshorisont ser sprang i referanse på et tidligere tidspunkt, og begynner derfor innsvingning til referanseverdi tidligere. Fra figuren ses at kompenseringsegenskapen er dårligere ved $P = 1$ og $P = 20$. Regulering med liten horisont, $P = 1$ og $P = 2$, evner ikke å se referanseendring før den tiltrer. Innsvingningen skjer dermed etter sprang i referanse.

4.3.2 Endring av prediksjonshorisont, testet på virkelig prosess

Figur 4.22 viser responser fra variert lengde på prediksjonshorisont, der virkelig prosess reguleres. Horisonten varieres med størrelsene [1 5 10 20], mens de andre parametrene er som spesifisert i tabell 4.7.



Figur 4.22: Endring av prediksjonshorisont, testet på virkelig prosess.

Observasjoner fra figur 4.22:

Fra figur 4.22 ses at responser fra virkelig prosess og fra modell er litt forskjellige. Til forskjell fra testing på modell, fører regulering med liten horisont, $P = 1$, til stående svingninger på utgang ved sprang i referanse.

Hvis kontrollhorisont hadde vært satt slik at tommelfingerregel i kapittel 3.2.2 var oppfylt ($3 \leq M \leq 5$), vil en prediksjonshorisont med størrelse $P = 8$, gi god regulering.

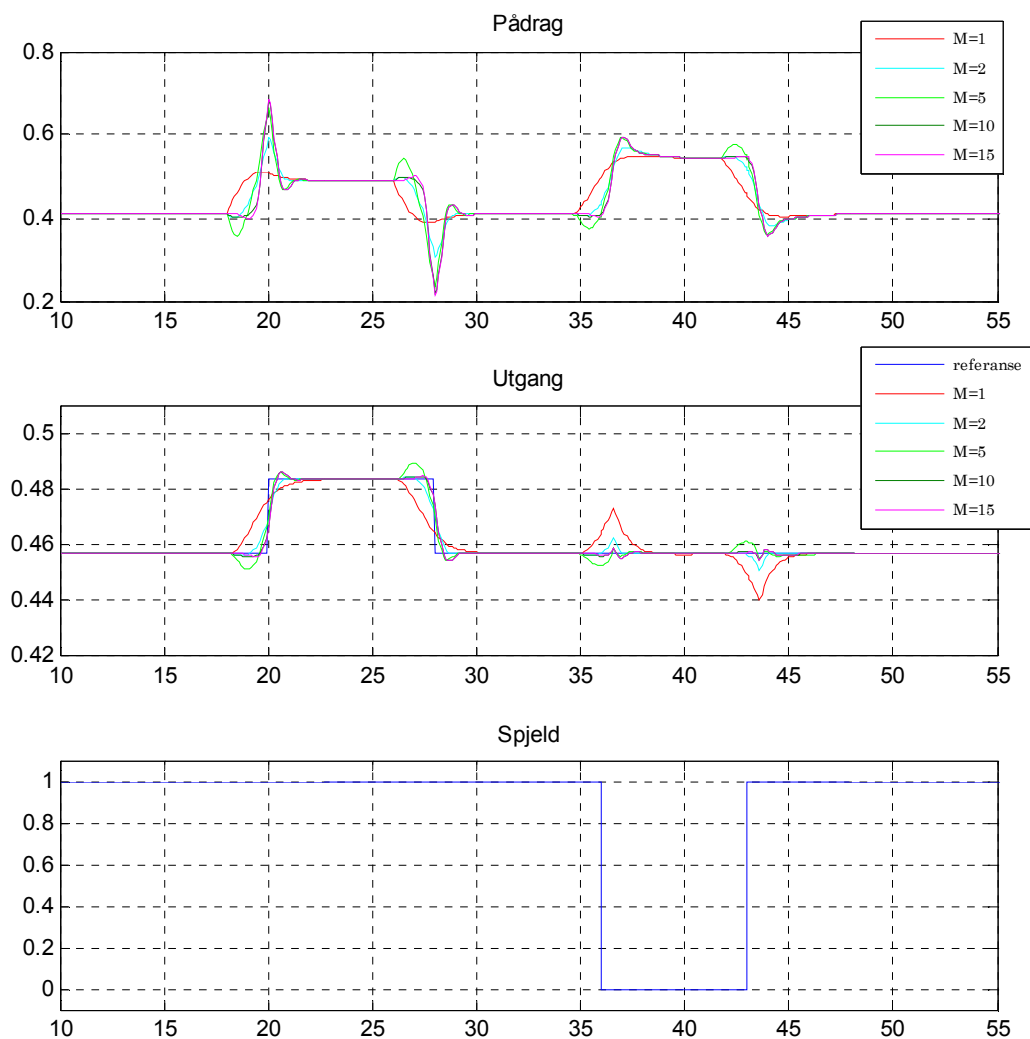
4.3.3 Endring av kontrollhorisont, testet på modell

Modellen er regulert med parameterverdier i tabell 4.8. Kontrollhorisonten varieres fra hver test, mens de andre parameterverdiene holdes konstante under alle testene.

Figur 4.23 viser simulert respons med varierende kontrollhorisont. Horisonten varieres med størrelsene [1 2 5 10 15].

Tabell 4.6: Parameterverdier brukt ved simulering.

Parameter	Verdi
Prediksjonshorisonten, P	20
Kontrollhorisonten, M	Varies
Inngangsvekt, $w^{\Delta u}$	0.5
Utgangsvekt, w^y	1.0
Kontrollintervall, Δt	0.1



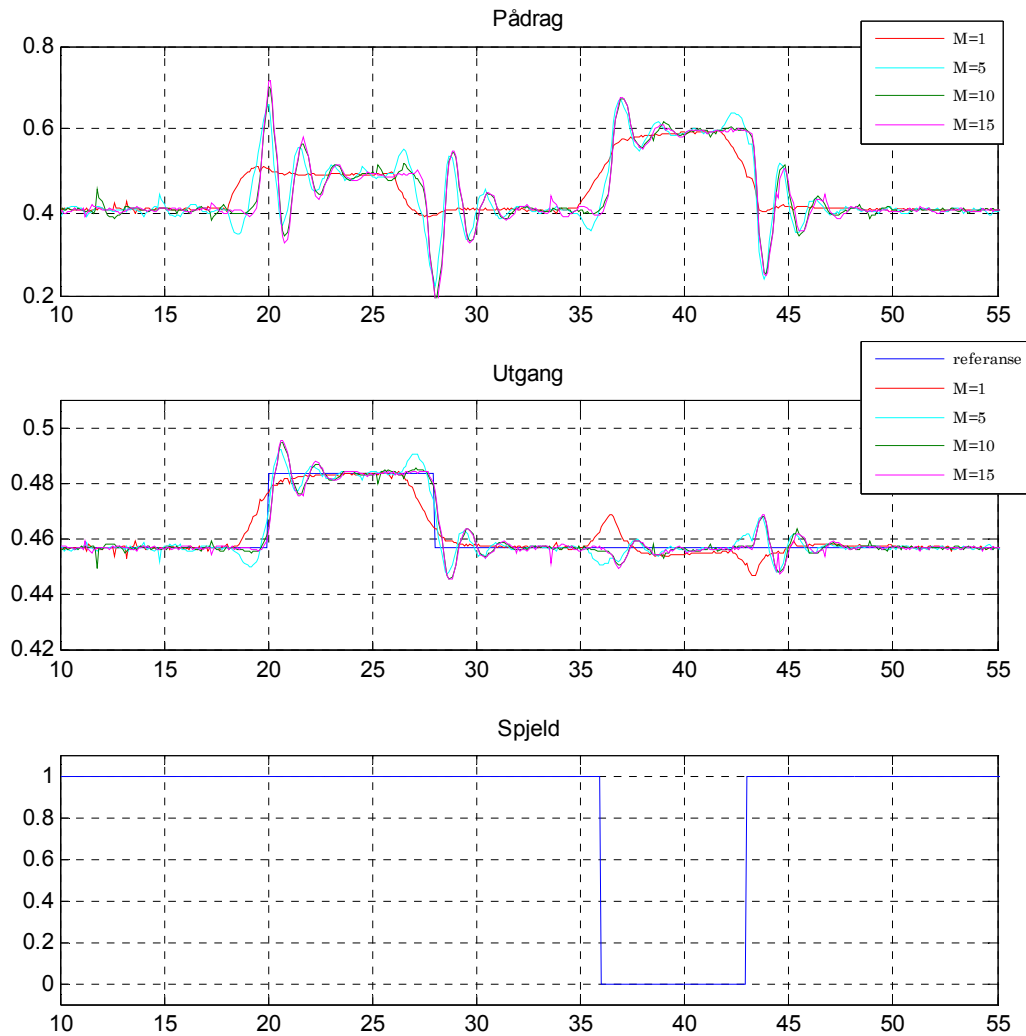
Figur 4.23: Endring av kontrollhorisont, testet på modell.

Observasjoner fra figur 4.23:

Fra figuren ses at de ulike reguleringene begynner å gjøre pådragsendringer ved samme tidspunkt, $t = 18$. Respons fra kontrollhorisont med kort lengde, $M = 1$, bruker lengst tid for å svinge seg inn mot referansen og har dårligere kompenseringsegenskaper. Horisont med $M = 5$, er et spesialtilfelle, hvor utgangen blir styrt i motsatt retning av sprang, før den styres mot referanseverdi. Dette tilfellet forekommer når forholdet mellom prediksjonshorisont og kontrollhorisont er av en bestemt størrelse, og systemets dominerende tidskonstant er relativt liten ($T < 30$ sekund). Regulering med lang kontrollhorisont har bedre følge- og kompenseringsegenskaper.

4.3.4 Endring av kontrollhorisont, testet på virkelig prosess

Parametere fra tabell 4.8 benyttes for regulering av virkelig prosess, se figur 4.15. Kontrollhorisonten varieres med lengdene [1 5 10 15].



Figur 4.24: Endring av prediksjonshorisont, testet på virkelig prosess.

Observasjoner fra figur 4.24:

Observerer at responsen fra virkelig prosess er veldig lik simulert respons i figur 4.23.

På grunnlag av observasjon ville regulering med kontrollhorisont med lengde $M = 12$, gi god regulering. Dette medfører at prediksjonshorisonten må økes for å få et større forhold mellom disse.

4.3.5 Oppsummering av tester

Testene som er utført tilsier at følgende reguleringsparametere vil gi god regulering for MPC-regulator med forhåndsinformasjon om fremtidig referanse- og forstyrrelsesforløp:

- Prediksjonshorizont, $P = 25$
- Kontrollhorizont, $M = 12$
- Utgangsvekt, $w^{\Delta u} = 0.7$
- Inngangsvekt, $w^y = 1.2$
- Kontrollintervall, $\Delta t = 0.1$

Kapittel 5

Anvendelse av MPC-regulatorene

I dette kapittel blir MPC-regulatorene (med og uten forhåndsinformasjon) anvendt i forskjellige scenarier på virkelig prosess. Resultater sammenlignes med PI-regulator med foroverkobling.

5.1 Simulinkmodell for testing av MPC-regulator

For å utføre ulike tester på viftescenariet ble det utviklet en modell i simulink, se figur 5.1. Modellen gir mulighet for å regulere vifteprosessen med forskjellige reguleringsstrukturer, samt forskjellige sprang i referanse og forstyrrelse.

5.1.1 Ytelsesmål

For å måle ytelse under ulik testing, er simulinkmodellen i figur 5.1 utstyrt med en subblokk som måler *Integral of Absolute Error* (IAE) og *Total Variation* (TV) [5].

IAE – *Integral of Absolute Error*

IAE-verdi beregnes fra likning (5.1). Her integreres reguleringsavviket fra 0 til ∞ . I praksis integreres avviket et endelig antall tidsskritt. IAE er et mål på hvor godt utgangen følger referansen. Mindre IAE-verdi betyr bedre ytelse.

$$\text{IAE} = \int_0^{\infty} |e| dt \quad (5.1)$$

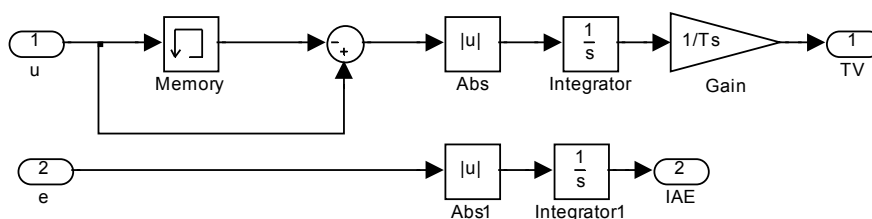
Figur 5.2 viser hvordan likningen er implementert i subblokken i simulink.

TV – *Total Variation*

TV-verdi beregnes fra likning (5.2). Verdien genereres fra endring i pådragsverdi. Mindre TV-verdi, betyr glattere pådragsbruk.

$$\text{TV} = \sum_{k=0}^{\infty} |u(t_{k+1}) - u(t_k)| \quad (5.2)$$

Det er vanskelig å lage et reguleringsystem hvor både IAE- og TV-verdi har lav verdi. Lav IAE-verdi, altså lite reguleringsavvik, fører til at pådraget må endres hyppigere og TV-verdi blir større. Liten endring av pådragsverdier, altså lav TV-verdi, fører ofte til større reguleringsavvik og dermed større IAE-verdi. I utvikling av reguleringsystem, der disse ytelsesmålene benyttes, er det viktig å være klar over dette forholdet, og bruke en fornuftig avveining mellom TV-verdi og IAE-verdi. I figur 5.2 er likning (5.1) og (5.2) implementert i subblokken i simulink.



Figur 5.2: Subblokk for ytelsesmål i simulink.

5.1.2 Parametere for MPC-regulator uten forhåndsinformasjon

I kapittel 4.2 ble MPC-regulering, uten forhåndsinformasjon om referanse- og forstyrrelsesforløp, testet med forskjellige parametervalg. På bakgrunn av resultater fra dette kapitlet, ble følgende parametere valgt for MPC-regulatoren:

- Prediksjonshorisont, $P = 25$
- Kontrollhorisont, $M = 3$
- Utgangsvekt, $w^{\Delta u} = 0.7$
- Inngangsvekt, $w^y = 1.2$
- Kontrollintervall, $\Delta t = 0.1$

5.1.3 Parametere for MPC-regulator med forhåndsinformasjon

I kapittel 4.3 ble MPC-regulering, med forhåndsinformasjon om referanse- og forstyrrelsesforløp, testet med forskjellige parametervalg. På bakgrunn av resultater fra dette kapitlet, ble følgende parametere valgt for MPC-regulatoren:

- Prediksjonshorisont, $P = 25$
- Kontrollhorisont, $M = 12$
- Utgangsvekt, $w^{\Delta u} = 0.7$
- Inngangsvekt, $w^y = 1.2$
- Kontrollintervall, $\Delta t = 0.1$

Grunnen for at kontrollhorisonten er valgt lengre en omtalte tommelfingerregler (kapittel 3.2.2), er for å unngå spesialtilfelle nevnt i kapittel 4.3.3.

5.1.4 Parametere for PI-regulator med foroverkobling

PI-regulator er satt opp med parametere i henhold til pol-plasseringsmetoden. Informasjon er hentet fra [5] og [6].

Pol-plasseringsmetoden

Metoden brukes for å finne regulatorparametere for PI-regulator med utgangspunkt i ønsket oversving og responstid for regulator. Likning (5.3) og (5.4) blir brukt for å finne forsterknings- og integralledd for PI-regulator.

$$K_p = \frac{2 \cdot \zeta \cdot \omega_0 \cdot T - 1}{K} \quad (5.3)$$

$$T_i = \frac{2 \cdot \zeta \cdot \omega_0 \cdot T - 1}{\omega_0^2 \cdot T} \quad (5.4)$$

hvor

- ω_0 er *udempet resonansfrekvens* ($\omega_0 \approx 1/T_r$, hvor T_r er responstid for regulatorsystemet)
- ζ er *relativ dekningsfaktor*
- K er forsterkning
- T er tidskonstant

Det er valgt å bruke oversvingsfaktor, $\zeta = 1$, med responstiden $T_r = 0.65$. Denne verdien for responstiden, gir $\omega_0 = 1.55$. Disse verdiene er valgt for at PI-regulatoren skal ha reguleringsdynamikk som er sammenlignbar med MPC-regulator uten forhåndsinformasjon. Forsterkning og tidskonstant leses av likning (2.3).

Med gitte og spesifiserte verdier kan K_p og T_i berenes:

$$K_p = \frac{2 \cdot \zeta \cdot \omega_0 \cdot T - 1}{K} = \frac{2 \cdot 1 \cdot 1.55 \cdot 1.52 - 1}{0.338} = 10.98 \approx 11 \quad (5.5)$$

$$T_i = \frac{2 \cdot \zeta \cdot \omega_0 \cdot T - 1}{\omega_0^2 \cdot T} = \frac{2 \cdot 1 \cdot 1.55 \cdot 1.52 - 1}{1.55^2 \cdot 1.52} = 1.02 \quad (5.6)$$

Regulatorens transferfunksjon blir da

$$H_r(s) = \frac{K_p}{T_i s + 1} = \frac{11}{1.02s + 1} \quad (5.7)$$

For å vurdere faktisk verdi av oversvingsfaktor og responstid for PI-regulatoren er følgeforholdet, $M(s)$, vurdert. Dette er gjort i likning (5.8). $H_0(s)$ er sløyfetransferfunksjonen ($H_0(s) = H_r(s)H_p(s)$).

$$M(s) = \frac{H_0(s)}{H_0(s) + 1} = \frac{2.40}{s^2 + 1.64s + 3.04} = \frac{2.40}{s^2 + 2 \cdot \zeta \cdot \omega_0 \cdot s + \omega_0^2} \quad (5.8)$$

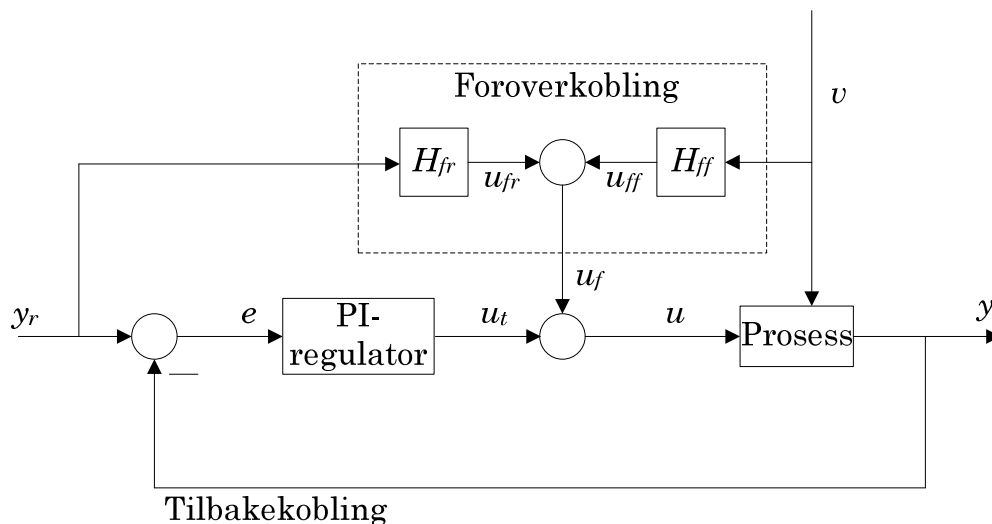
Likning (5.8) gir oversvingsfaktor, $\zeta = 0.47$, og responstid, $T_r = 1.74$, for følgeforholdet.

PI-regulator ble implementert med anti-windup (integratorbegrensning).

Foroverkobling

PI-regulatoren regulerer utgangen til å komme nær referanseverdi ved bruk av tilbakekoblingen. Når reguleringsystemet kun er regulert med tilbakekobling, kan det ikke gjøres noen pådragsendring før avviket er blitt forskjellig fra null.

Ved å implementere foroverkobling, kan pådragsendring utføres før reguleringsavviket er registret ved målingen på utgang fra tilbakekoblingen. Dette kompenserer for systemets dødtid. På denne måten vil PI-regulator oppnå tilnærmet like egenskaper som MPC-regulator uten forhåndsinformasjon, der MPC-regulator får informasjon om systemets dødtid fra modellen i regulatoren. Foroverkobling implementeres ved en kobling fra referansen og forstyrrelsen, direkte til pådraget. Pådragsendring ved foroverkobling er ikke avviksbasert, som ved tilbakekobling, men basert på kjennskap til prosessen. Figur 5.3 viser strukturen av et reguleringsystem med foroverkobling.



Figur 5.3: Reguleringsstruktur med foroverkobling.

H_{fr} og H_{ff} bestemmes fra kjennskap om referansens og forstyrrelsens innvirkning på utgangen. Disse transferfunksjonene er beregnet i kapittel og gjengitt i likning (5.9) og (5.10).

$$H_u(s) = \frac{K_u}{T_u s + 1} = \frac{0.338}{1.52s + 1} \quad (5.9)$$

$$H_v(s) = \frac{K_d}{T_d s + 1} = \frac{-0.045}{1.07s + 1} \quad (5.10)$$

u_f blir beregnet ved likning (5.11), hvor transferfunksjonene bestemmer størrelsene for H_{fr} og H_{ff} .

$$u_f(s) = \frac{1}{\underbrace{H_u(s)}_{H_{fr}(s)}} y_r(s) + \frac{-H_v(s)}{\underbrace{H_u(s)}_{H_{ff}(s)}} v(s) = H_{fr}(s) y_r(s) + H_{ff}(s) v(s) \quad (5.11)$$

H_{fr} og H_{ff} får tellerpolynom henholdsvis med høyere orden i teller enn nevner, og med samme orden i teller og nevner. Dette løses ved å sette $s = 0$, som ved likning (5.12) og (5.13).

$$H_{fr}(s) = H_{fr}(0) = \lim_{s \rightarrow 0} H_{fr}(s) \quad (5.12)$$

$$H_{ff}(s) = H_{ff}(0) = \lim_{s \rightarrow 0} H_{ff}(s) \quad (5.13)$$

H_{fr} og H_{ff} er implementert i m-fil, se tabell 5.1.

Tabell 5.1: Beregning for foroverkobling.

```

38 % Foroverkobling
39
40 Hu = tf([0.338], [1.52 1]);
41 Hv = tf([-0.045], [1.072 1]);
42 Hff = dcgain(-Hv/Hu);
43 Hfr = dcgain(1/Hu);

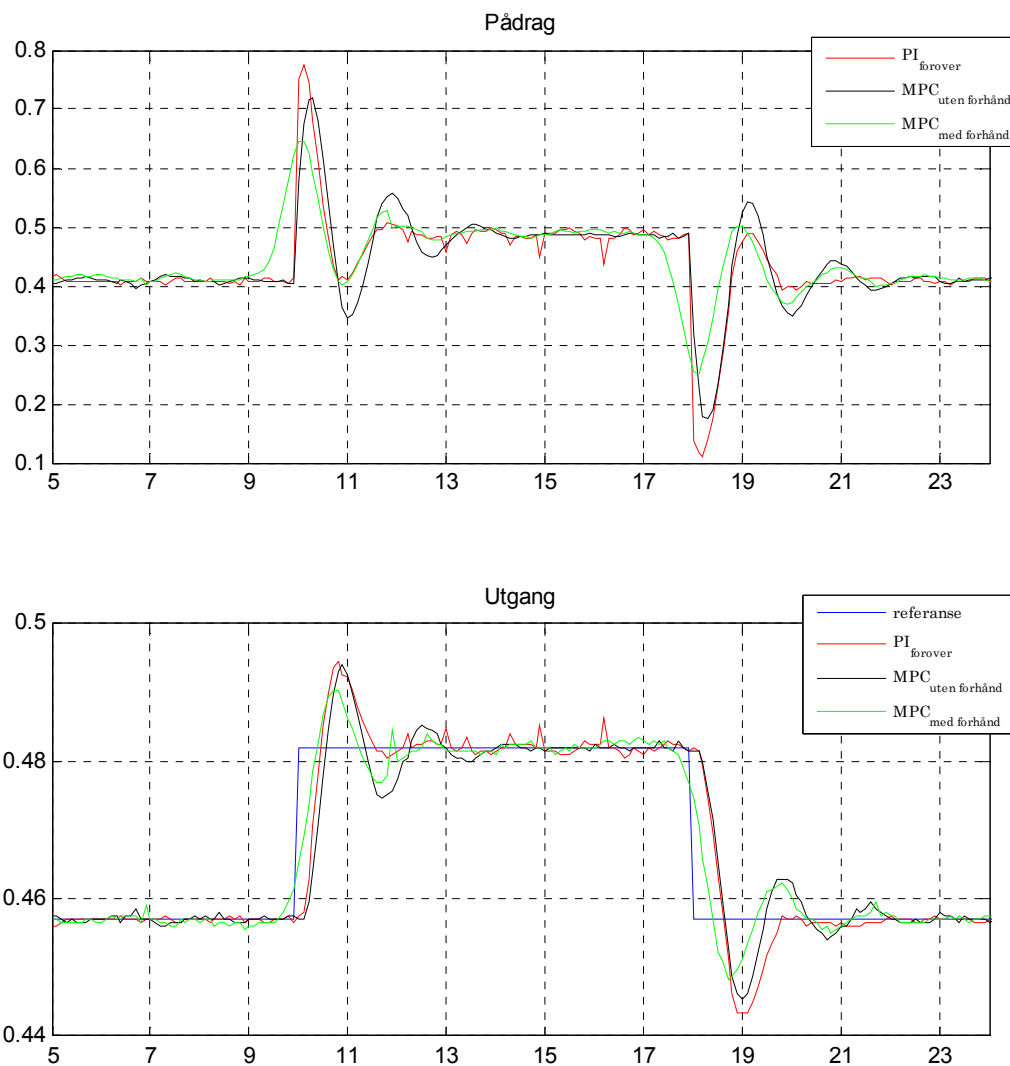
```

5.2 Følgeegenskaper

I dette kapittel blir regulatorne anvendt i tilfeller som gir grunnlag for å evaluere de respektive regulatorenes evne til å følge referansen. PI-regulatoren er i anvendelsen implementert med tidssteg som har samme lengde som MPC-regulatorenes kontrollintervall, Δt , for å kunne sammenligne de ulike regulatorstrukturene.

5.2.1 Sprang i referanse rundt arbeidspunkt

PI-regulator og MPC-regulatorne er her anvendt på et scenario hvor det utføres sprang i referanse rundt arbeidspunkt. Parameterne for hver av regulatorne er som spesifisert i kapittel 5.1.2 - 5.1.4.



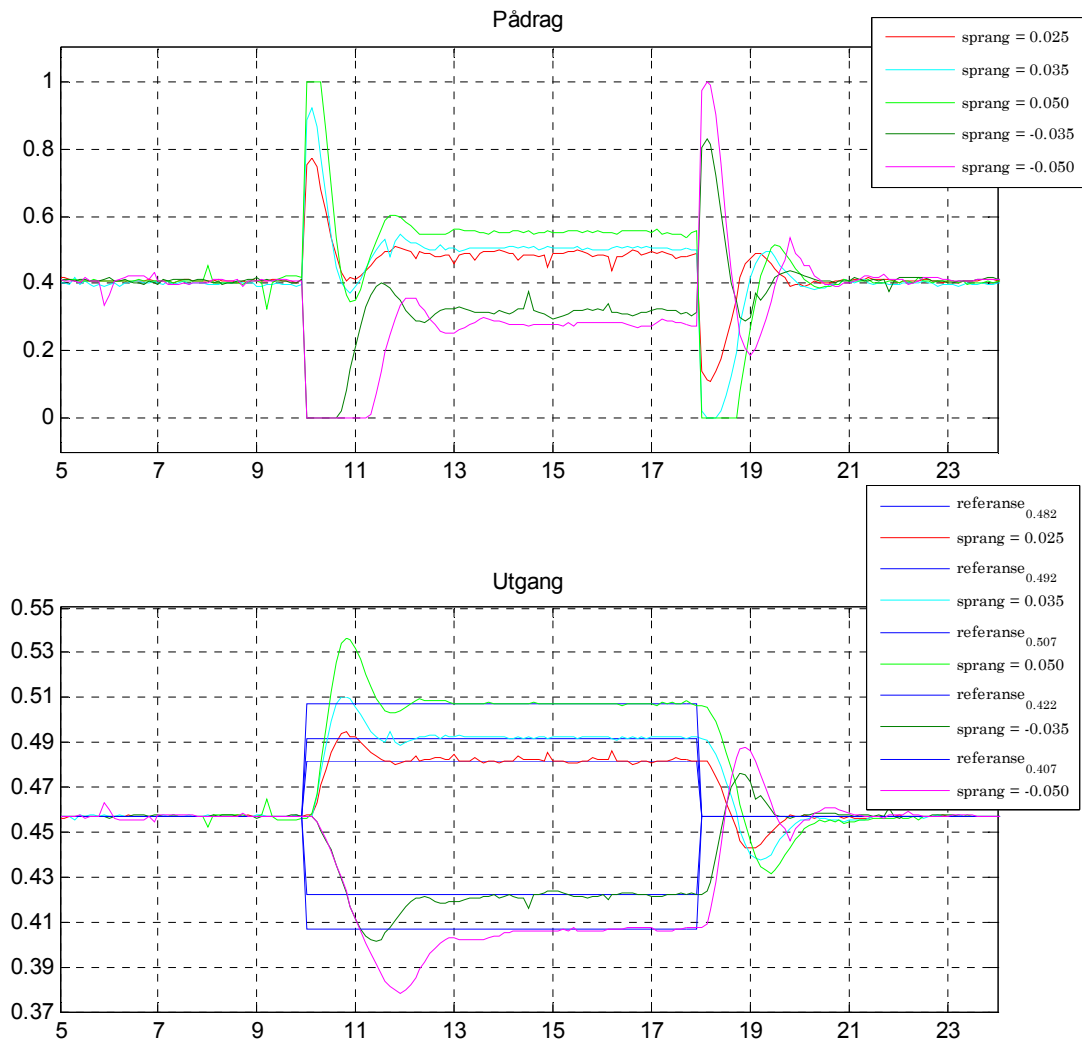
Figur 5.4: PI- og MPC-regulering ved sprang i referanse rundt arbeidspunkt.

Figur 5.4 viser at respons for PI-regulator og MPC-regulator uten forhåndsinformasjon er relativt lik. Regulatorene begynner innsvingning ved referanseendring på samme tidspunkt. PI-regulatoren har ett større over- og undersving ved endringene, men svinger seg imidlertid raskere inn mot referansen. For MPC-regulator med forhåndsinformasjon blir informasjonen brukt til å øke og redusere pådrag før referanseendring. Dette medfører at regulatoren starter innsvingning til ny referanseverdi ved et tidligere tidspunkt enn ved de andre regulatorstrukturene, og har mindre over- og undersving. Regulatoren svinger seg derimot noe senere inn mot ny referanse enn ved PI-regulering. Ytelsesmålene for responsene er vist i tabell 5.2 og 5.3 ved sprangverdi 0.025.

5.2.2 Test av robusthet ved referansesprang

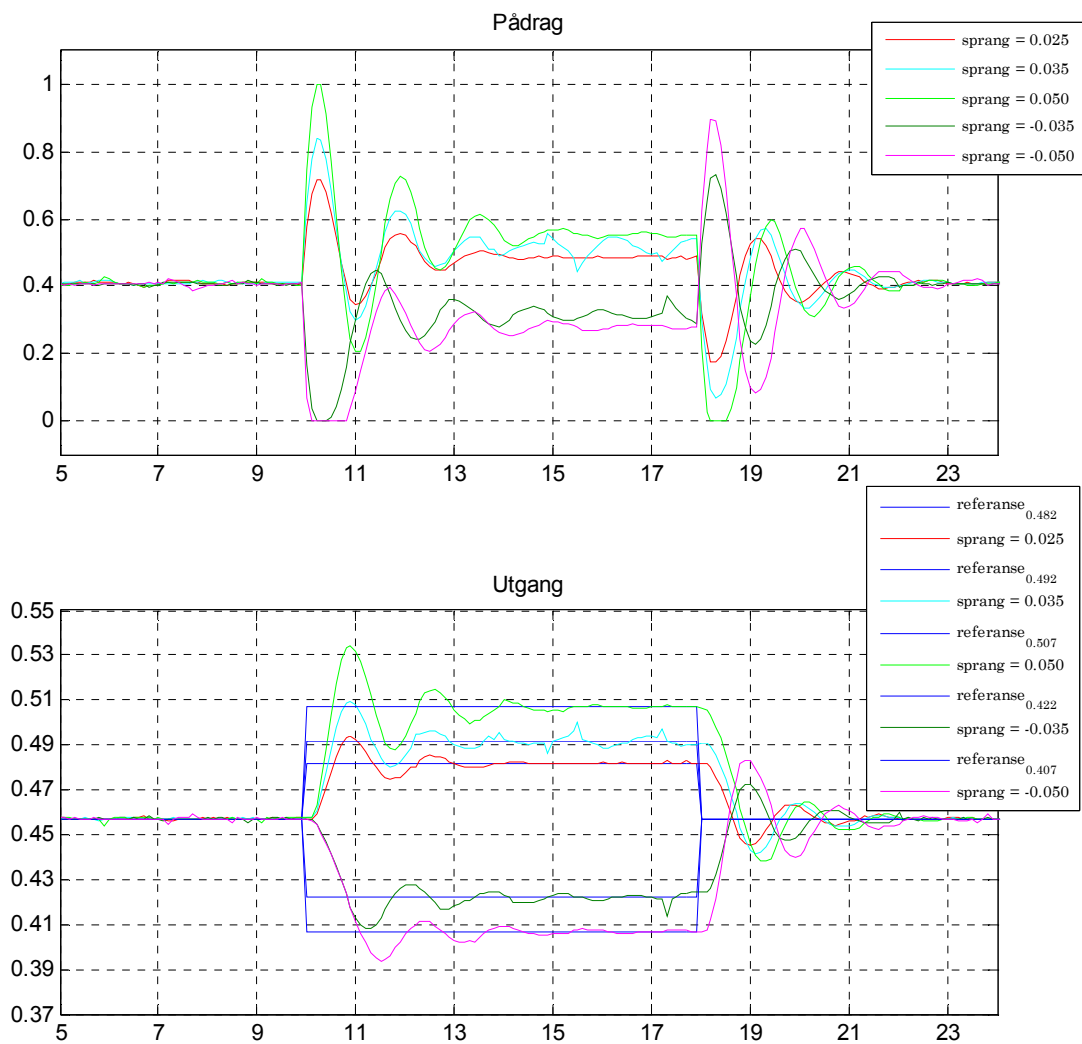
Robustheten til de forskjellige regulatorene blir testet og sammenlignet ved å utføre forskjellige sprang i referansen. Hensikten er å vurdere hvor godt regulator klarer å følge referanseverdier som ikke er innenfor spesifisert arbeidspunkt ($0.457 \leq y_r \leq 0.484$). IAE- og TV-verdier blir oppsummert i slutten av kapittelet, i tabell 5.2 og 5.3.

I figur 5.5 er PI-regulatoren med foroverkobling brukt til regulering ved forskjellige sprang i referansen.



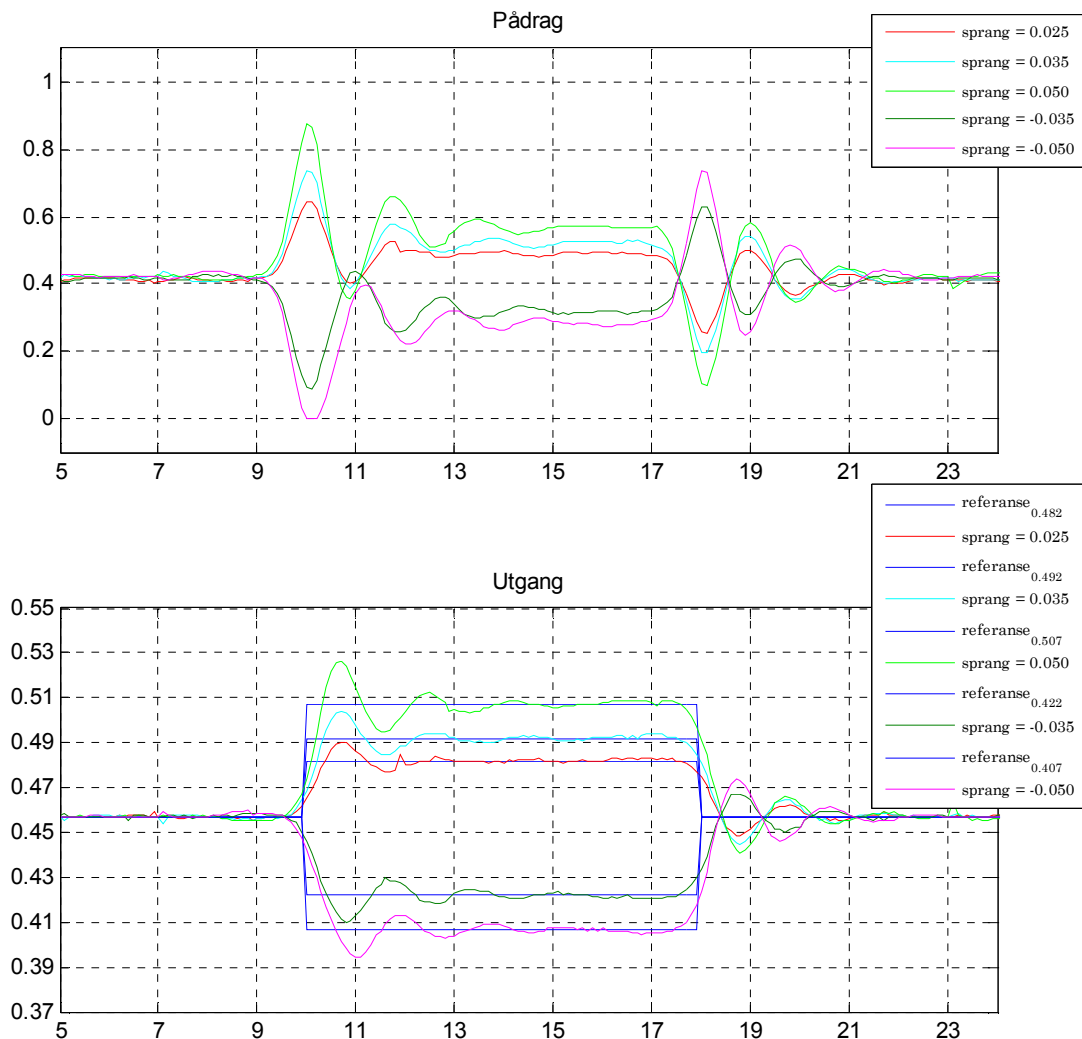
Figur 5.5: Forskjellig sprang i referanse med PI-regulator.

I figur 5.6 er MPC-regulator uten forhåndsinformasjon brukt til regulering ved forskjellige sprang i referansen.



Figur 5.6: Forskjellig sprang i referanse med MPC-regulator uten forhåndsinformasjon.

I figur 5.7 er MPC-regulator med forhåndsinformasjon brukt til regulering ved forskjellige sprang i referansen.



Figur 5.7: Forskjellig sprang i referanse med MPC-regulator med forhåndsinformasjon.

I figur 5.8 er responsene for de forskjellige regulatorne samlet i en figur. For positive sprang er oversvinget nesten likt for PI-regulator og MPC-regulator uten forhåndsinformasjon. Innsvingning til referanseverdi er bedre for PI-regulator, enn MPC-regulator uten forhåndsinformasjon, i disse tilfellene. MPC-regulator med forhåndsinformasjon har mindre oversving og utfører innsvingningen til referanse bedre enn de andre regulatorne, for positive sprang. For negative sprang har MPC-regulator uten forhåndsinformasjon mindre oversving enn for PI-regulator. MPC-regulator med forhåndsinformasjon har i dette tilfellet like stort oversving som MPC-regulator uten forhåndsinformasjon, men bedre innsvingning.

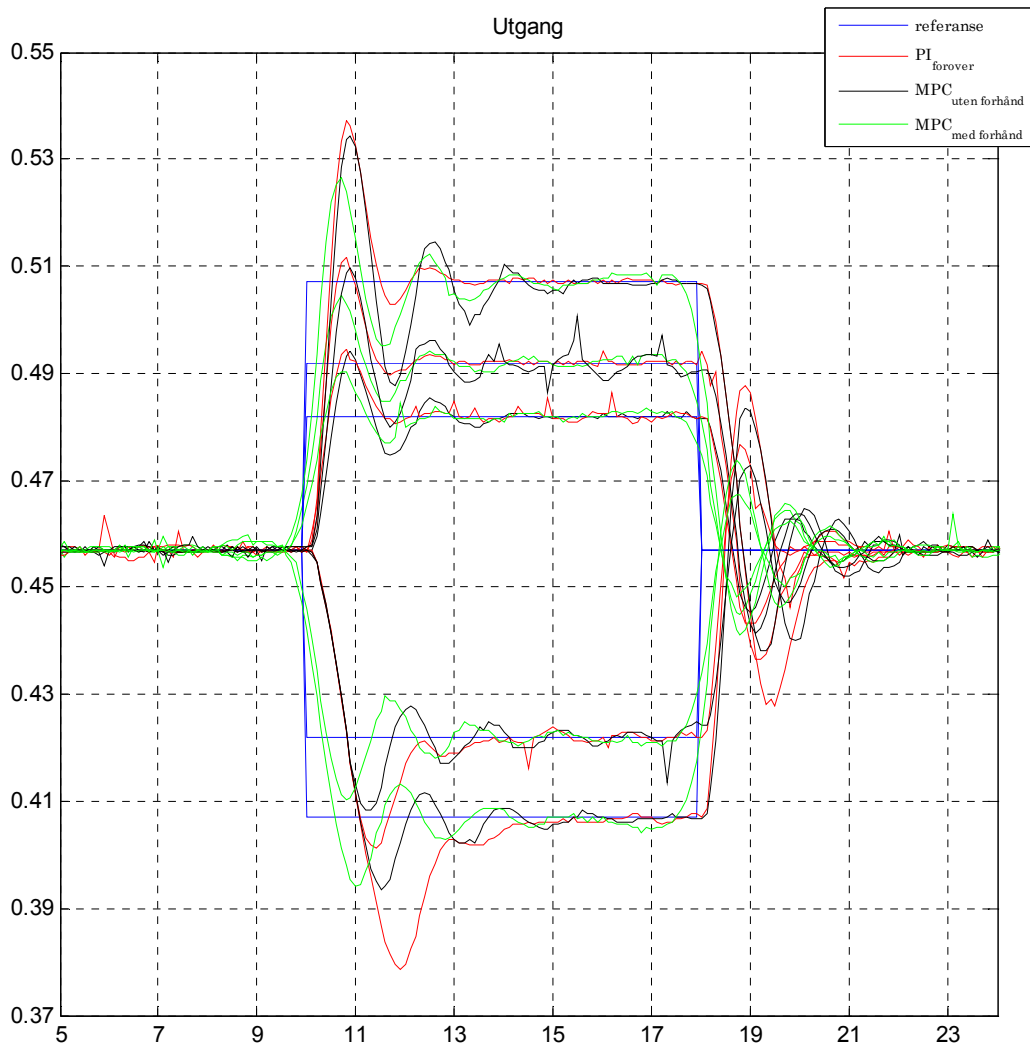
Tabell 5.2 viser reguleringsavviket (IAE-verdien) for de forskjellige regulatorne. MPC-regulator med forhåndsinformasjon viser betydelig bedre følgeegenskaper enn ved PI-regulator og MPC-regulator uten forhåndsinformasjon. Tabell 5.3 viser at pådragsbruken er mye bedre ved MPC-regulator med forhåndsinformasjon. Tabellene viser at PI-regulator har bedre følgeegenskaper og pådragsbruk enn MPC-regulator uten forhåndsinformasjon for positive og negative sprang.

Tabell 5.2: IAE-verdier for PI- og MPC-regulatorer ved forskjellige sprang i referanse.

Sprangverdi	0.025	0.035	0.050	-0.035	-0.050
IAE-PI _{med foroverkobling}	0.044	0.064	0.101	0.072	0.104
IAE-MPC _{uten forhåndsinformasjon}	0.050	0.073	0.108	0.076	0.107
IAE-MPC _{med forhåndsinformasjon}	0.033	0.047	0.068	0.046	0.064

Tabell 5.3: TV-verdier for PI- og MPC-regulatorer ved forskjellige sprang i referanse.

Sprangverdi	0.025	0.035	0.050	-0.035	-0.050
TV-PI _{med foroverkobling}	3.292	4.018	4.333	3.813	4.231
TV-MPC _{uten forhåndsinformasjon}	3.175	4.188	5.288	3.998	4.647
TV-MPC _{med forhåndsinformasjon}	1.997	3.052	4.001	2.568	3.224

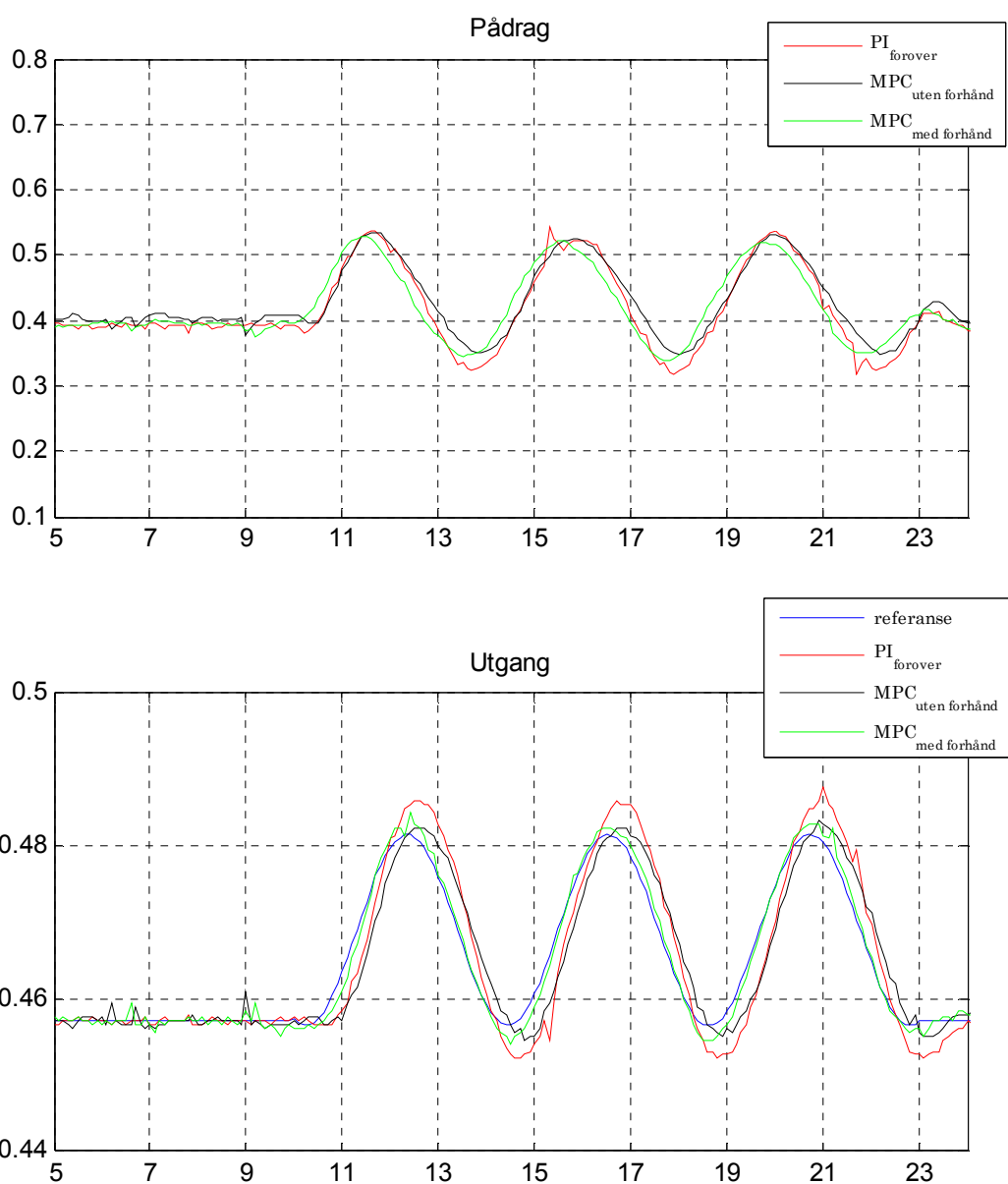


Figur 5.8: Respons fra figur 5.5-5.7 i samme figur.

5.2.3 Sinusreferanse rundt arbeidspunkt

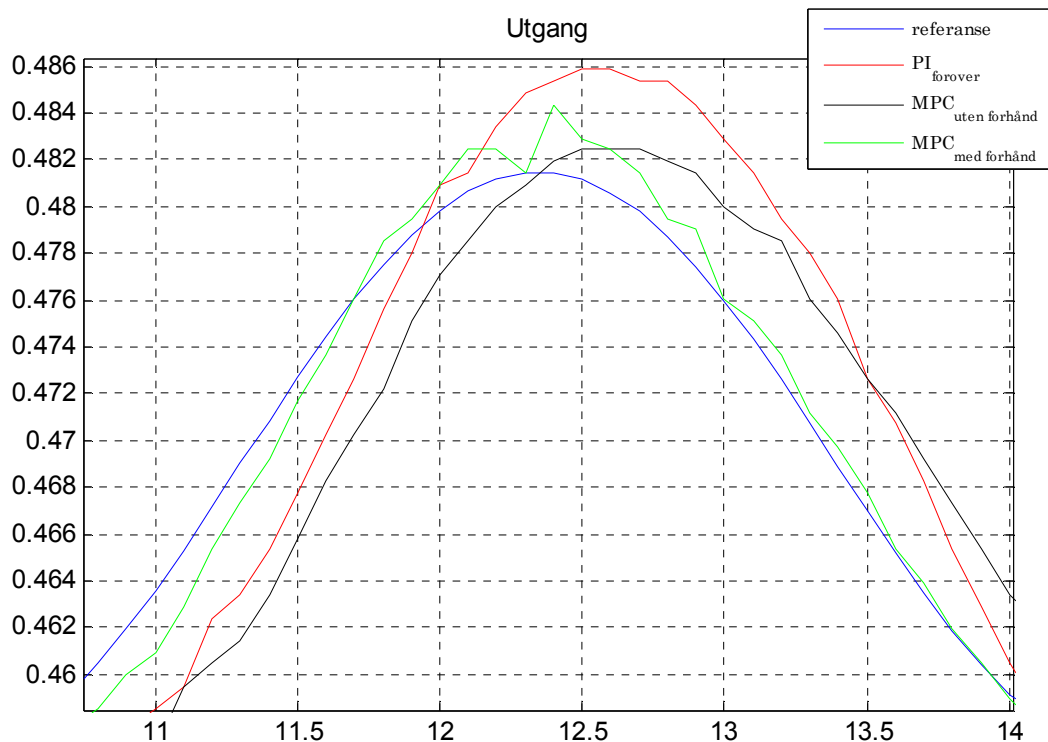
Regulatorene blir i dette kapittel anvendt i et scenario hvor referansen blir variert ved hjelp av et sinussignal. Regulatorparameterne som er benyttet er de samme som spesifisert i kapittel 5.1.2 – 5.1.4. Figur 5.9 viser respons fra regulering med de forskjellige regulatorene. Sinusreferansen som er regulatorene følger er vist i likning (5.14). Bias og amplitude er satt slik at sinusreferansen varierer innenfor arbeidspunktet. Frekvesen er satt til $\omega = 1.5 \text{ rad/s}$. Sinusreferansen har starter ved tidspunkt $t = 10$. For at sinusreferansen skal ha samme startverdi som referanse ved $t < 10$, har sinusreferansen en fase, $\varphi = 1.9 \text{ rad/s}$.

$$y_r(t) = 0.469 + 0.0125 \cdot \sin(1.5 \cdot t + 1.9) \quad (5.14)$$



Figur 5.9: Sinusreferanse med PI- og MPC-regulatorer.

For å enklere vurdere ytelsen, viser figur 5.10 et utsnitt fra figur 5.9. Utsnittet viser at MPC-regulator med forhåndsinformasjon følger sinusreferansen godt og nesten uten faseforringelse. MPC-regulatoren uten forhåndsinformasjon følger også sinusreferansen bra, men med større faseforringelse. PI-regulatoren har større oversving, samt større faseforringelse, og følger referansen dårligere enn MPC-regulatorene.



Figur 5.10: Utsnitt av figur 5.9.

Tabell 5.4 viser reguleringsavviket (IAE-verdier) for hvert av reguleringsssystemene. MPC-regulator med forhåndsinformasjon har mye bedre følgeegenskaper enn de andre regulatorene. Fra tabell 5.5 kan det ses at pådragsbruken også er betydelig bedre. MPC-regulator uten forhåndsinformasjon har bedre følgeegenskaper og pådragsbruk enn PI-regulator.

Tabell 5.4: IAE-verdier for PI- og MPC-regulatorer ved sinusreferanse.

IAE-PI _{med foroverkobling}	0.059
IAE-MPC _{uten forhåndsinformasjon}	0.053
IAE-MPC _{med forhåndsinformasjon}	0.018

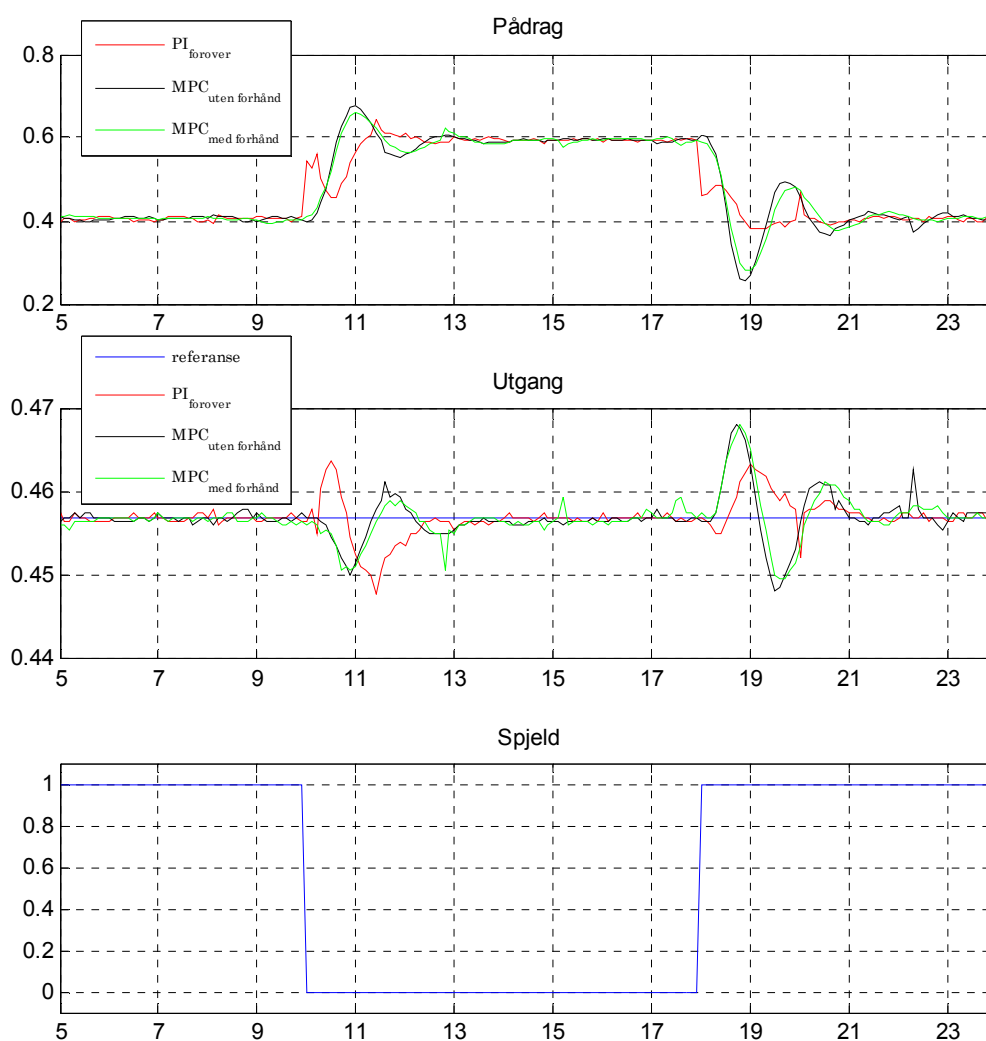
Tabell 5.5: TV-verdier for PI- og MPC-regulatorer ved sinusreferanse.

TV-PI _{med foroverkobling}	2.475
TV-MPC _{uten forhåndsinformasjon}	2.211
TV-MPC _{med forhåndsinformasjon}	2.063

5.3 Kompenseringsegenskaper

I dette kapittelet blir regulatorne anvendt i tilfeller som gir grunnlag for å evaluere de respektive regulatorenes evne til å kompensere for forstyrrelsen. PI-regulatoren er i anvendelsene implementert med samme tidssteg som MPC-regulatorenes kontrollintervall, Δt , for å kunne sammenligne de ulike regulatorstrukturene.

PI-regulator og MPC-regulatorne er her anvendt på et scenario hvor det er utført sprang i forstyrrelsen (spjeld blir lukket/åpnet). Parameterne for hver av regulatorne er som spesifisert i kapittel 5.1.2 - 5.1.4. Figur 5.11 viser respons fra ulike regulatorstrukturene.



Figur 5.11: Regulering med PI- og MPC-regulatorne ved sprang i forstyrrelse.

Figur 5.11 viser at PI-regulator kompenserer best for forstyrrelsen. Svingning i pådrag for PI-regulator ved sprang i forstyrrelse skyldes at foroverkoblingen ikke tar hensyn til forsinkelsen. Tabell 5.6 viser at kompenseringssegenskapene er best ved PI-regulator, mens det er liten forskjell på MPC-regulator med og uten forhåndsinformasjon. Fra tabell 5.7 kan det derimot ses at MPC-regulator med forhåndsinformasjon benytter mindre pådragsbruk.

Tabell 5.6: IAE-verdier for PI- og MPC-regulatorer ved forstyrrelse.

IAE-PI _{med foroverkobling}	0.021
IAE-MPC _{uten forhåndsinformasjon}	0.027
IAE-MPC _{med forhåndsinformasjon}	0.028

Tabell 5.7: TV-verdier for PI- og MPC-regulatorer ved forstyrrelse.

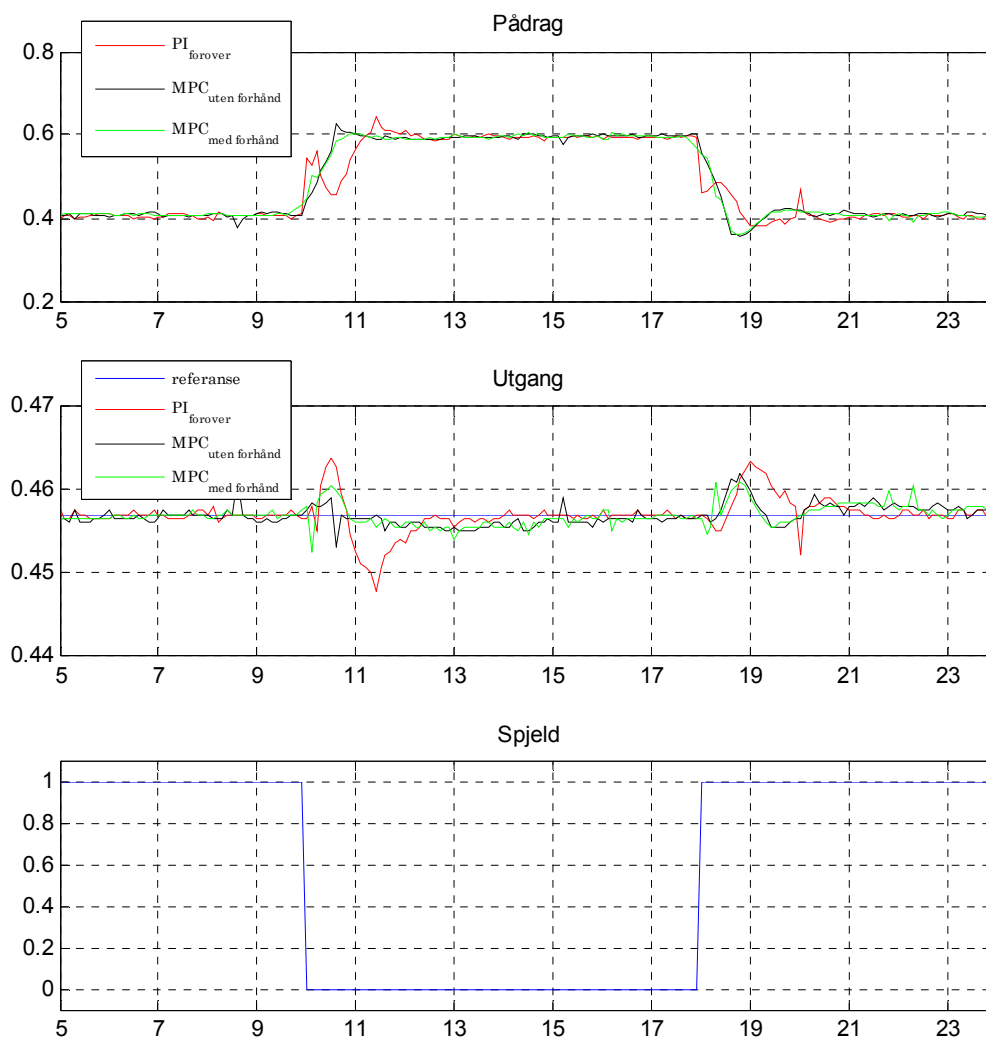
TV-PI _{med foroverkobling}	2.287
TV-MPC _{uten forhåndsinformasjon}	2.459
TV-MPC _{med forhåndsinformasjon}	2.157

5.3.1 Kompenseringssegenskaper ved endret parametervalg

For å bedre kompenseringssegenskapene ved MPC-regulering ble regulatorparameterne for MPC-regulatorene endret. Følgende parametere ble benyttet for MPC-regulering med og uten forhåndsinformasjon:

- Prediksjonshorizont, $P = 10$
- Kontrollhorizont, $M = 1$
- Utgangsvekt, $w^{\Delta u} = 0.7$
- Inngangsvekt, $w^y = 1.2$
- Kontrollintervall, $\Delta t = 0.1$

Figur 5.12 viser respons ved sprang i forstyrrelse med endret parametervalg for MPC-regulatorene. PI-regulator med foroverkobling har reguleringsparametere som spesifisert i kapittel 5.1.4.



Figur 5.12: Regulering med PI- og MPC-regulatorerne ved sprang i forstyrrelse.

I figur 5.12 er kompenseringsegenskapene for MPC-regulatorerne forbedret. Tabell 5.8 og 5.9 viser at kompenseringsegenskapene og pådragsbruken er best ved MPC-regulator med forhåndsinformasjon.

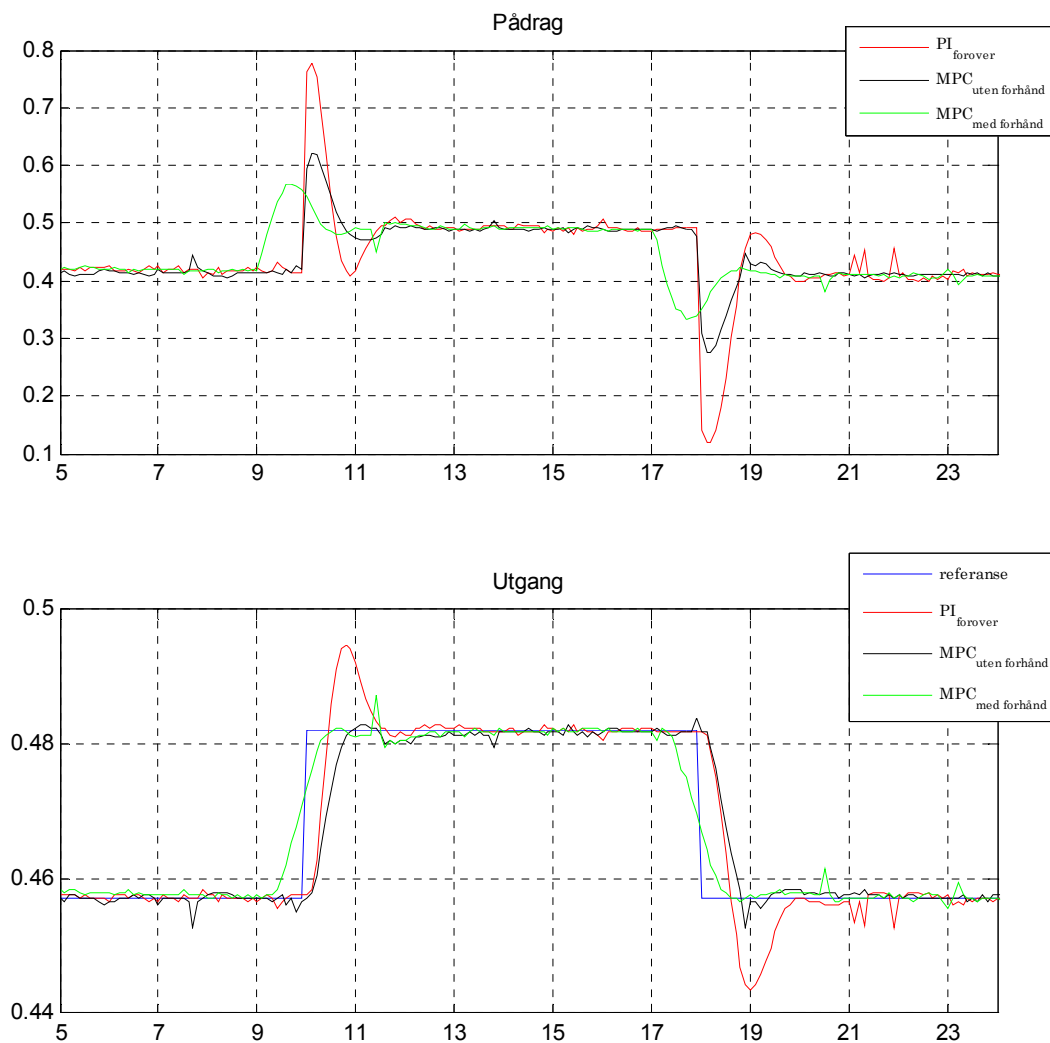
Tabell 5.8: IAE-verdier for PI- og MPC-regulatorer ved forstyrrelse.

IAE-PI _{med foroverkobling}	0.021
IAE-MPC _{uten forhåndsinformasjon}	0.018
IAE-MPC _{med forhåndsinformasjon}	0.017

Tabell 5.9: TV-verdier for PI- og MPC-regulatorer ved forstyrrelse.

TV-PI _{med foroverkobling}	2.287
TV-MPC _{uten forhåndsinformasjon}	1.869
TV-MPC _{med forhåndsinformasjon}	1.756

Disse parameterendringene for MPC-regulatorene endrer selvfølgelig regulering ved sprang i referansen. Konsekvens av dette er vist i figur 5.13. Tabell 5.10 og 5.11 viser at følgeegenskapene og pådragsbruken blir forbedret. En annen konsekvens vil være at MPC-regulator uten forhåndsinformasjon ikke lenger er sammenlignbar med PI-regulator med foroverkobling, fordi reguleringsystemenes dynamikk er blitt forskjellig, der PI-regulator med foroverkobling har oversving og MPC-regulator uten forhåndsinformasjon ikke har oversving.



Figur 5.13: Konsekvens av parameterendring vist ved sprang i referanse.

Tabell 5.10: IAE-verdier for PI- og MPC-regulatorer ved parameterendring.

IAE-PI _{med foroverkobling}	0.044
IAE-MPC _{uten forhåndsinformasjon}	0.037
IAE-MPC _{med forhåndsinformasjon}	0.023

Tabell 5.11: TV-verdier for PI- og MPC-regulatorer ved parameterendring.

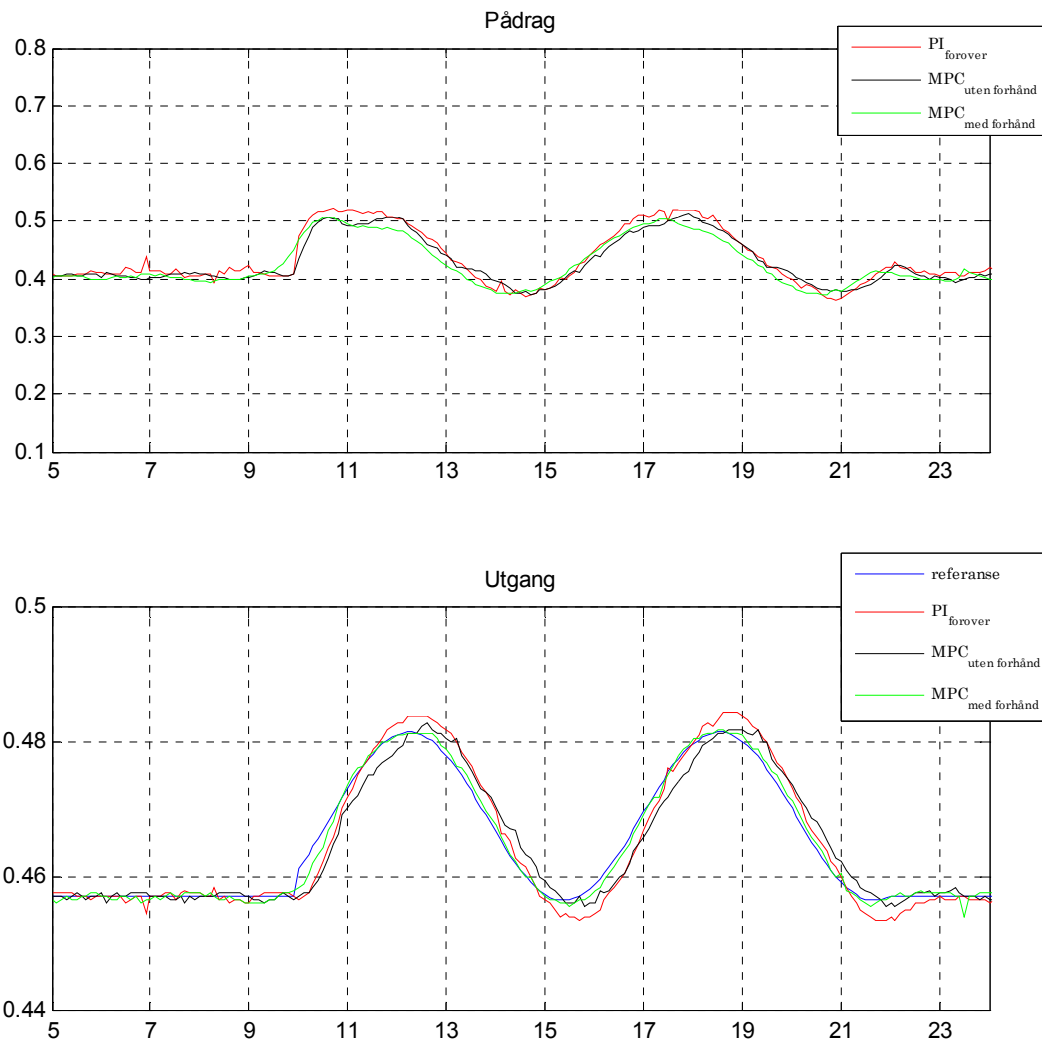
TV-PI _{med foroverkobling}	3.292
TV-MPC _{uten forhåndsinformasjon}	2.072
TV-MPC _{med forhåndsinformasjon}	1.657

5.4 Effekt av modellens nøyaktighet

I kapittel 2.3.2 ble det vist at høyere frekvenser bidrar til større avvik mellom frekvensrespons for modell og prosess. For å undersøke effekten av modellens nøyaktighet, regulerer reguleringsystemene (PI-regulator og MPC-regulatorene) sinusreferanse med frekvensene $\omega = 1$ rad/s og $\omega = 5$ rad/s. Tabell 5.12 og 5.13 viser IAE-verdier og TV-verdier for reguleringsystemene ved frekvensene.

Figur 5.14 viser respons fra sinusreferanse påført ved $t = 10$, med frekvens $\omega = 1$ rad/s, gitt i likning (5.15). Denne frekvensverdien gir lite avvik mellom frekvensrespons fra modell og prosess (se figur 2.6, kapittel 2.3.2). Fra tabell 5.12 og 5.13 kan det ses at reguleringsystemene har gode ytelsesmål ved denne sinusreferansen.

$$y_r(t) = 0.469 + 0.0125 \cdot \sin(1 \cdot t + 1.9) \quad (5.15)$$



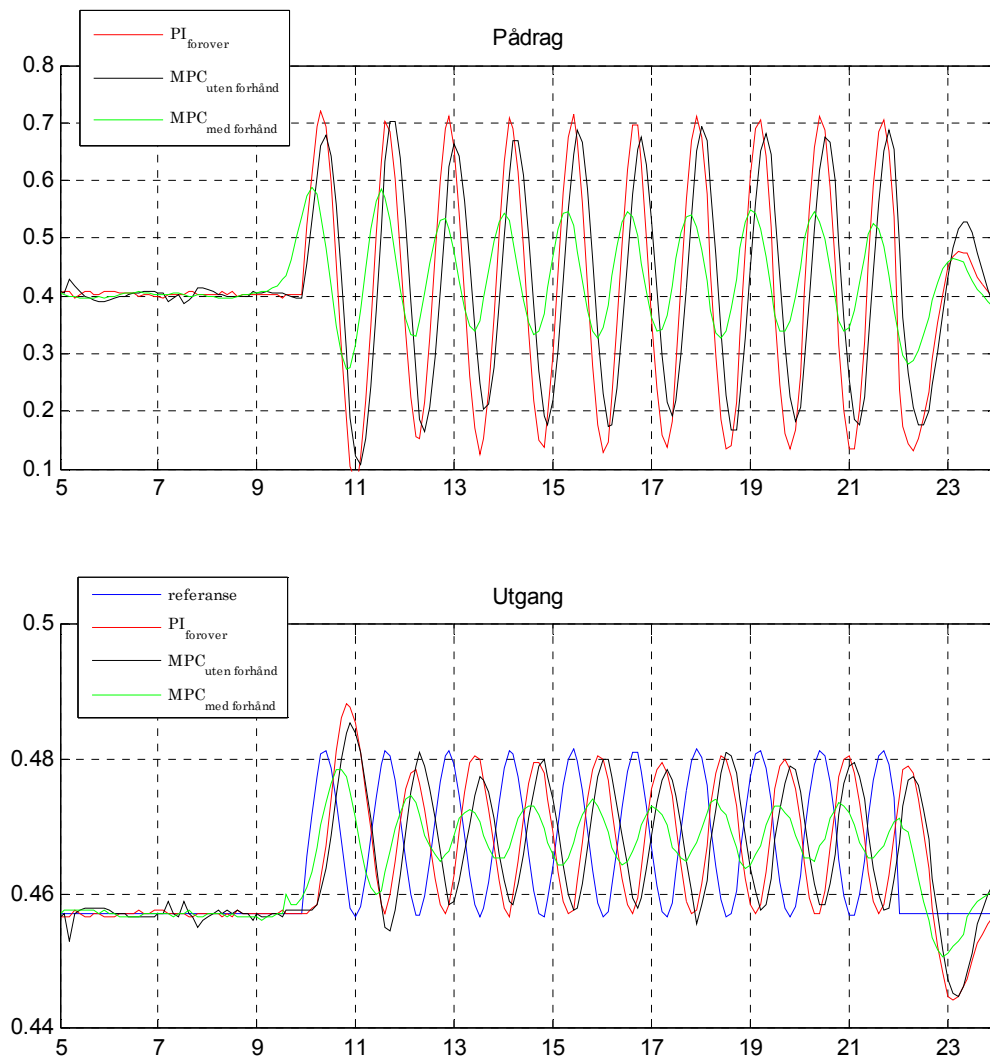
Figur 5.14: Sinusreferanse med frekvens $\omega = 1$ rad/s.

Figur 5.15 viser respons fra sinusreferanse påført ved $t = 10$, med frekvens $\omega = 5$ rad/s, gitt i likning (5.16). Denne frekvensverdien gir et større avvik mellom frekvensrespons fra modell og prosess (se figur 2.6, kapittel 2.3.2). Fra tabell 5.12 og 5.13 kan det ses at reguleringssystemene har yter dårlig ved denne sinusreferansen. PI-regulator og MPC-regulator uten forhåndsinformasjon har relativt lik respons for reguleringen, der utgangen har samme amplitude som referansen, men er faseforsinket med om lag 90 grader. Dette stemmer godt overens med frekvensresponsen i kapittel 2.3.2, der plot fra fasen også indikerer en faseforsinkelse på omtrent 90 grader mellom modell og prosess ved $\omega = 5$ rad/s.

MPC-regulator med forhåndsinformasjon har i utgangspunktet samme faseforsinkelse som PI-regulator og MPC-regulator uten forhåndsinformasjon. Forhåndsinformasjonen blir brukt slik at MPC-regulator ser endringen i sinusreferansen før den inntreffer. Dette fører til at MPC-regulator med forhåndsinformasjon har tilsynelatende mindre faseforsinkelse, noe som kommer av forminsknet

amplitude som følge av forhåndsinformasjonen. Fra tabell 5.12 og 5.13 kan det ses at MPC-regulator med forhåndsinformasjon oppnår bedre ytelse enn ved PI-regulator og MPC-regulator uten forhåndsinformasjon.

$$y_r(t) = 0.469 + 0.0125 \cdot \sin(5 \cdot t) \quad (5.16)$$



Figur 5.15: Sinusreferanse med frekvens $\omega = 5$ rad/s.

Tabell 5.12: IAE-verdier for PI- og MPC-regulatorer ved parameterendring.

Frekvens, ω	$\omega = 1$	$\omega = 5$
IAE-PI	0.034	0.205
IAE-MPC _{uten forhåndsinformasjon}	0.034	0.204
IAE-MPC _{med forhåndsinformasjon}	0.012	0.129

Tabell 5.13: TV-verdier for PI- og MPC-regulatorer ved parameterendring.

Frekvens, ω	$\omega = 1$	$\omega = 5$
TV-PI	1.920	12.70
TV-MPC _{uten forhåndsinformasjon}	1.618	11.49
TV-MPC _{med forhåndsinformasjon}	1.492	5.62

Kapittel 6

Konklusjon og forslag til videre arbeid

6.1 Konklusjon

Det er i dette arbeidet utviklet to MPC-regulatorer, en med og en uten forhåndsinformasjon om fremtidig referanse- og forstyrrelsesforløp. Formålet var å regulere vifte hastigheten til varmluftsviften. Regulatorne er testet med forskjellige parametere, både i simulert tilfelle og for virkelig prosess. Reguleringsparameterne for de to MPC-regulatorne ble valgt på basis av resultater fra parameter testingene.

PI-regulator med foroverkobling er utviklet med utgangspunkt i respons fra reguleringsystemet med MPC-regulator uten forhåndsinformasjon. Dette er gjort for å gi sammenlignbare resultater i anvendelser av MPC-regulatorne. For å måle ytelsen av reguleringsystemene er IAE-verdier (*Integral of Absolute Error*) og TV-verdier (*Total Variation*) brukt. Disse gir ytelsesmål for henholdsvis reguleringsavvik og pådragsbruk.

For ulike sprang utført i referanse var det relativt liten forskjell på regulering ved PI-regulator og ved MPC-regulator uten forhåndsinformasjon. Best resultater ble i dette tilfellet oppnådd ved MPC-regulator med forhåndsinformasjon. Denne regulatoren oppnådde også best ytelse ved sinusreferanse.

Noe overraskende var PI-regulator bedre til å kompensere for forstyrrelsen fra spjeldet. For å bedre ytelsen til MPC-regulatorne, ble parametere for disse endret til å bedre kompenseringen. Ved parameterendringen ble kompenseringsegenskapene for MPC-regulatorne bedre enn for PI-regulator. Endringen medførte i tillegg forbedrede følgeegenskaper for MPC-regulatorne ved sprang i referanse rundt arbeidspunkt. En konsekvens av denne forbedringen var imidlertid at MPC-regulatorne ikke lenger var sammenlignbar med PI-regulator i forhold til responstid og oversving.

Reguleringsystemene ga dårlige ytelsesmål ved unøyaktig modell. MPC-regulator uten forhåndsinformasjon og PI-regulator fikk samme ytelsesverdi for følgeegenskapene, mens MPC-regulatoren hadde bedre ytelsesmål for pådragsbruken. MPC-regulator med forhåndsinformasjon ga klart best resultater for ytelsesmålene ved unøyaktig modell.

6.2 Forslag til videre arbeid

Ifølge [8] er anvendelse av MPC-regulatorer på raske, eller større og komplekse systemer, en utfordring. Varmluftsviften har svært rask dynamikk ($T \approx 1.5$). Dette har ikke vist seg å være noe problem i utarbeidelsen av denne oppgaven. Det ville derfor vært interessant med et studium hvor regulering av temperatur inkluderes i systemet, slik at effekt av økende kompleksitet kan bli evaluert.

Bibliografi

- [1] **Nikolaou, Michael.** *Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs.* 2003.
- [2] **Alberto Bemporad, Manfred Morari og N- Lawrence Ricker.** *Model Predictive Control Toolbox™ 3 - Getting started Guide.* 2009.
- [3] **Alberto Bemporad, Manfred Morari og N- Lawrence Ricker.** *Model Predictive Control Toolbox™ 3 - User's Guide.* 2010.
- [4] **Qin, S. Joe & Badgewell, Thomas A.** *A survey of industrial model predictive control technology.* 2002.
- [5] **Finn Haugen.** *Praktisk reguleringssteknikk, 2. utgave.* 2003.
- [6] **Finn Haugen.** *Dynamiske systemer - modellering, analyse og simulering.* 2003.
- [7] **Finn Haugen.** *Modellbasert regulering- optimalregulering og prediktiv regulering.* 2008.
- [8] **Andrew Paice.** *Model Predictive Control: An Introduction.* 2009.

Vedlegg A

A.1 Bakgrunn for feil i Simulink i Matlab versjon R2009a

Feilen ble oppdaget ved at MPC-blokk i Simulink ga prosesspådrag til viftemotor med enorme dimensjoner (fra -10^3 til 10^3). For å rette opp i feilen ble MathWorks support, produsent for Matlab, kontaktet.

A.2 Oppskrift for å rette opp feil i Matlab versjon R2009a

Support fra MathWorks ga følgende oppskrift for å rette opp feil ved versjon R2009a av Matlab:

1. Avslutt Matlab
2. Naviger til følgende katalog:
`$MATLABROOT$\toolbox\rtw\targets\rtwin\kernel`
3. Gi nytt navn for 'rtwtkrnl.sys' til 'rtwtkrnl.sys.old'
4. Kopier 'rtwt.sys' fra vedlagt CD-disk til denne katalogen
5. Start Matlab
6. Skriv 'rtwingt -install' og reinstaller kjernen (kernel)

Feilen skal nå være rettet opp.