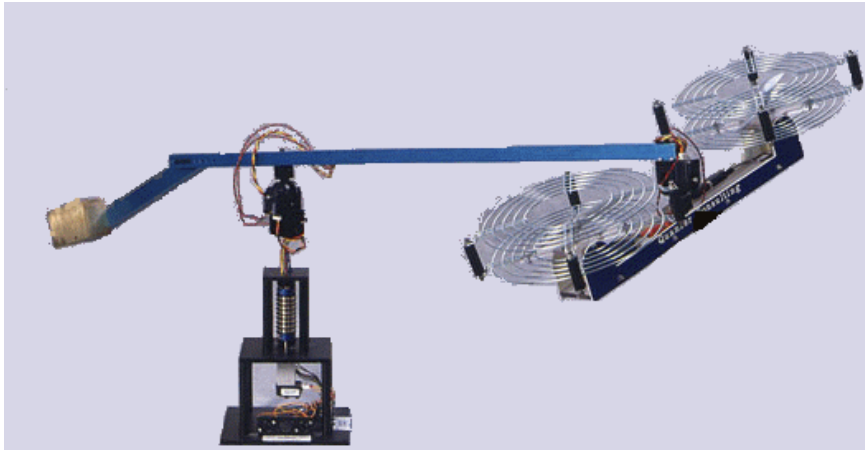


Prosjektoppgave i
TTK4115
Lineær Systemteori
Helikopterlab



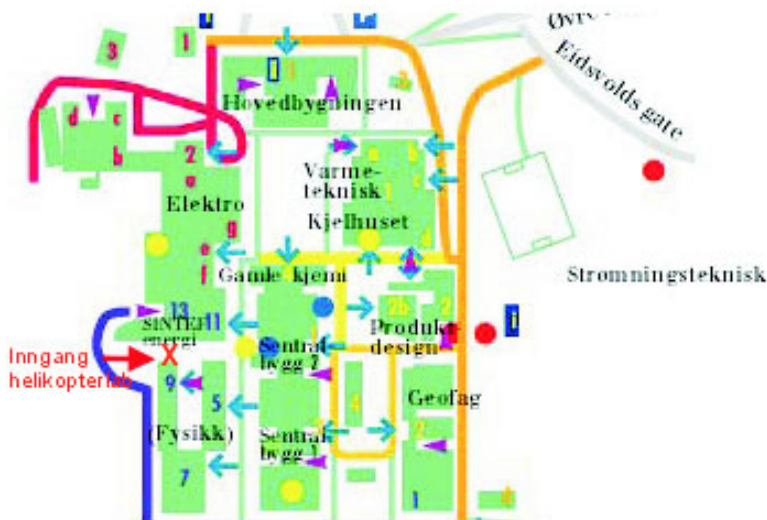
Norges teknisk- naturvitenskapelige universitet
Institutt for teknisk kybernetikk
TRONDHEIM

Versjon 3.3, August 14, 2003, JS

Tidligere versjoner:
3.2, september 2002, ESI
3.1, september 2001, ESI
3.0, juni 2001, TAJ

1 Praktisk informasjon

- Oppgaven er obligatorisk, og rapporten vil bli gitt en poengsum som teller 30% ved eksamen. Oppgaven utføres i grupper på 2 studenter, der gruppa blir gitt en felles bedømmelse. Vi forventer at hver gruppe leverer en selvstendig utført besvarelse, men det er selvsagt mulig å få tips fra andre. Det vil bli gjennomført en muntlig evaluering av hver gruppe slik at rapporten ikke trenger være så omfattende. Evalueringen vil foregå i uke 43.
- Det er beregnet ca. 40 timers arbeid på oppgaven.
- Helikoptrene (5 stk.) står i det gamle fysikkbygget (se kart nedenfor). For å få tilgang på dette bygget, trenger en et adgangskort som er aktivisert for denne laben. Dette skal være ordnet for alle som gikk 2. kyb våren 2003. Dersom du ikke gikk i 2. kyb, må du melde fra til vit.ass.
- Hver gruppe vil få 4 hele dager på laben, dvs helikopteret er reservert til din gruppe et helt døgn. En studass vil være tilstedet 2 timer hver dag.
- MATLAB 6.1 er installert på helikoptermaskinene. Simulink i MATLAB 6.1 er ikke kompatibel med senere utgaver og følgelig kan ikke modellen modifieres i andre versjoner av MATLAB.
- NB! Les alltid oppslagstavlen på fagets hjemmeside dersom du støter på problemer. All praktisk informasjon vil bli lagt ut der.



2 Formål

Formålet med denne oppgaven kan oppsummeres i følgende punkter:

- Studentene skal modellere den aktuelle prosessen, og regulere den ved hjelp av PID og multivariable regulatorer implementert i en datamaskin.
- Det skal demonstreres hvordan en kan estimere ikke-målte tilstander i systemet.
- Studentene skal bli kjent med et moderne utviklingsmiljø for sanntidsystemutvikling basert på automatisk generering av programkode fra SIMULINK blokkdiagram.

3 Labrapport

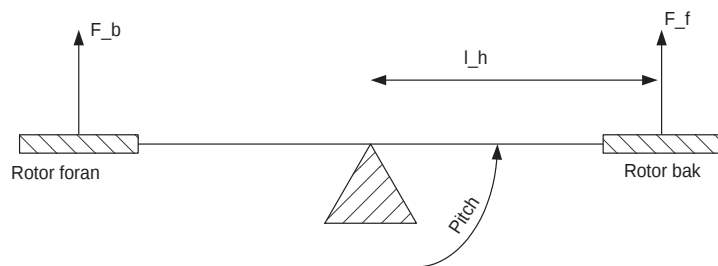
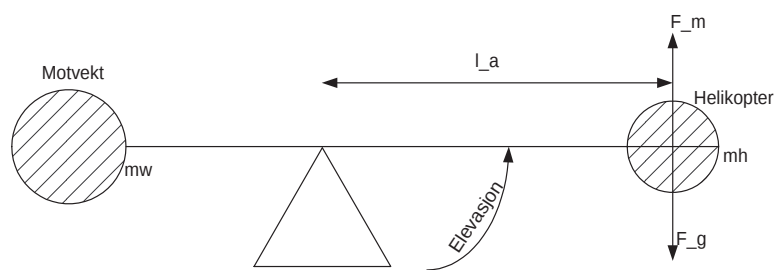
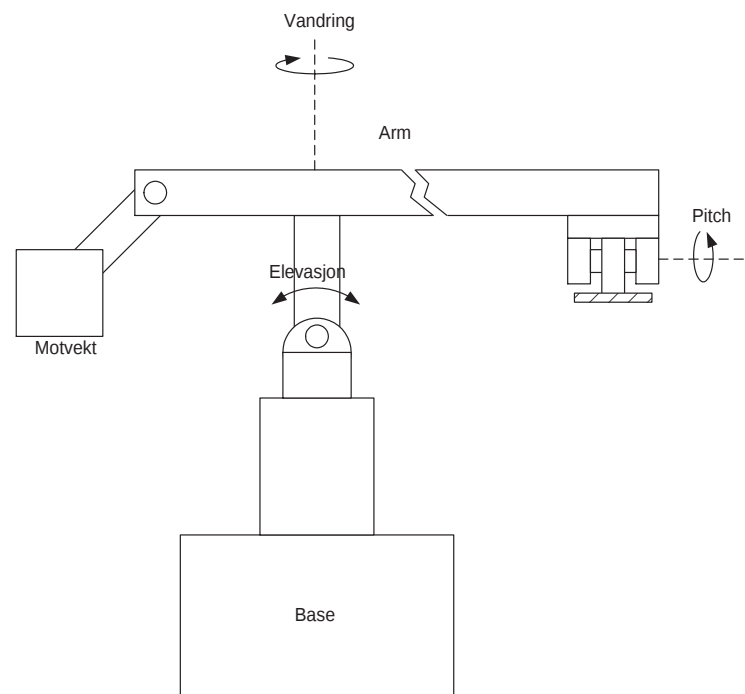
Det skal leveres en kort rapport på det som er gjort i oppgaven. For hver enkelt deloppgave skal denne inneholde

- Utskrifter av måledata fra relevante forsøk (plot).
- Relevante utledninger og beregninger som er gjort, verdier på parametre som er benyttet. Husk å oppgi enheter og skalering.
- Kort diskusjon og analyse av resultatene
- Simulink-diagram og MATLAB-kode som er brukt

4 Systembeskrivelse

Helikopteret består av en sokkel, som har en arm festet til seg, som vist i figuren nedenfor. Armen har helikopter kroppen i den ene enden, og en motvekt i den andre. Armen kan løftes opp og ned om en elevasjonsakse, og svinge rundt en vertikal akse (vandring). Selve helikopter kroppen kan dessuten bevege seg rundt en akse normalt på armen (pitch). Disse tre vinklene måles med optiske posisjonssensorer.

En har to pådrag tilgjengelig. Dette er to likestrømsmotorer, med påmontert propell. Kraften fra propellene kan antas å være proporsjonal med spenningen som settes på. Motvekten er justert slik at vekten som må løftes av propellene er ca. 70g. Dersom en setter ca. 1.5V på hver motor, vil helikopteret lette fra bakken. De fysiske dataene nedenfor er hentet fra “3D Helicopter Experiment” som følger med laben.



Fysiske data:

$l_a = 0.63m$; Lengde fra elevasjonsakse til helikopterkropp.

$l_h = 0.18m$; Lengde fra pitch-akse til motor.

$K_f = 0.5N/V$; Kraftkonstant motor.

$J_e \& J_t = 0.91 \text{ kg m}^2$; Trehetsmoment om hhv elevasjons- og vandringsakse.

$J_p = 0.0364 \text{ kg m}^2$; Trehetsmoment om pitch-akse.

$m_h = 1.15 \text{ kg}$; Helikoptermasse

$m_w = 1.87 \text{ kg}$; Motvekt

$m_g = 0.07 \text{ kg}$; Effektiv masse av helikopter (med motvekt).

$K_p = 0.686N$; Kraft som trengs for å løfte helikopter fra bakken ($g \cdot m_g$).

OBS: Enkelte av disse verdiene kan være **veldig** unøyaktige og skal derfor tas med en klype salt. F.eks. har K_f i enkelte tilfeller blitt målt til omlag $0.20N/V$. Dessuten har det i den senere tid blitt foretatt følgende nøyaktige målinger av massen til hvert enkelt helikopter (nummerert fra inngangsdøra):

Helikopter 1: 96g

Helikopter 2: 88g

Helikopter 3: 95g

Helikopter 4: 73g

Helikopter 5: 111g

Det kan forventes at de tyngre helikoptrene vil være litt vanskeligere å regulere enn de lette, under forutsetning av at pådraget er likt for alle helikoptrene. Det har ikke blitt foretatt noen nyere målinger av de andre parametrene.

Annen notasjon:

p - pitch

p_c - settpunkt for pitch

λ - vandring

r - vandringshastighet

r_c - settpunkt for vandringshastighet

e - elevasjon

e_c - settpunkt for elevasjonen

V_f - spenning, motor foran

V_b - spenning, motor bak

V_d - differanse mellom spenningene, $V_f - V_b$

V_s - sum av spenningene, $V_f + V_b$

K_{pp} , K_{pd} , K_{rp} - regulatorforsterkninger

Dokumentasjon for Simulink, Real-Time Workshop og Control Systems toolbox finnes som online-help i MATLAB.

4.1 Simulink, Wincon og Real-Time Workshop

Styreprogrammet lages med Matlab og Simulink. Det finnes egne Simulink blokker for kommunikasjon gjennom I/O-kortet. Når en har laget et blokkdiagram for regulatoren i Simulink, bruker en Wincon og Realtime Workshop for å kjøre det. Når disse programmene er installert, har en tilgjengelig en meny i Simulink, Wincon. Menyvalget Wincon → Build er det eneste som trengs i denne oppgaven. Følgende skjer når en trykker Build:

- Real-Time Workshop konverterer simulinkdiagrammet til C-kode.
- Koden kompiles og linkes med Visual C++.
- Den eksekverbare koden lastes ned til Wincon.

Etter dette er programmet klart til å kjøres. Nå kan en skru på power-modulen, og deretter starte programmet ved at en trykker "Start" i Wincon Server. Dette fører til at Wincon klient starter, og det er denne som tar seg av kjøring av programmet ved hjelp av sanntids operativsystemet RTX, som er en utvidelse av Windows NT platformen.

Det kan oppstå feil under kompilering. Det er lurt å vente med å starte helikopteret (trykke start i Wincon server), før kompilering osv. er helt ferdig. Det er også selvsagt viktig at en kompilerer med Wincon → Build, og ikke med Build som ligger under RTW-menyen.

Dersom en får uforklarlige feilmeldinger under kompilering, er det et par ting som ofte kan hjelpe. Det ene er å slette all kompilert kode, og kompilere på nytt. Dersom dette ikke hjelper er det ofte en god ide å restarte hele PCen.



Wincon Server

For å kunne foreta målinger på systemet, må en overføre sampledata til Matlab. Den vanlige "To Workspace"-blokken til Simulink fungerer ikke i sanntid, så en må bruke et sanntidsplot. Dette er det vanlige "Scope"-plottet som finnes i Simulink. For å vise sanntidsplottet under kjøring av programmet, må en før programmet starter velge det andre ikonet fra høyre i figuren over. Før en måleserie er tatt opp, må en velge følgende parametre i plottvinduet:

- Buffer size: Dette er lengden på måleserien som plottet ”husker” i sekunder.
- Frequency: Dette er samplingsfrekvensen.
- Time: Dette er tidsvinduet som vises i plottet. Velg denne lik buffer size. Dersom en krysser av for Maximize Plotting Frequency, vil dette gi raskeste samplingsfrekvens som er mulig.

Etter at en måleserie er tatt opp kan en eksportere denne til Matlab workspace, eller til en fil. Dette gjør en ved å gå på File → Save... i det aktuelle plottet. Det beste er vel å lagre data til en mat-fil, så en har dem til senere bruk.

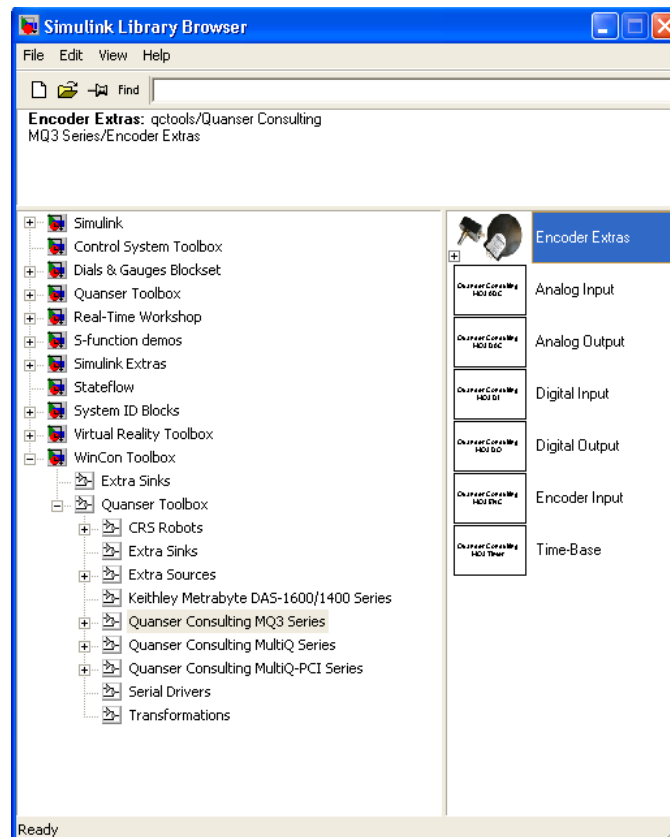
NB. Dersom det viser seg at dataene dere får overført har mye større samplingsfrekvens enn den dere valgte, så prøv å ta opp en måleserie til i det samme plottevinduet og overfør på nytt.

4.2 Hardware

4.2.1 MultiQ-3

Et MultiQ IO-kort er koplet til PC-bussen. MultiQ-kortet måler de fysiske signalene som kommer fra helikopteret, og konverterer dem til digitale signaler som kan leses av PCen. Den konverterer også signaler fra PCen, som brukes som pådrag på helikopteret.

Det finnes ferdiglagde simulink-blokker som tar seg av kommunikasjonen mellom PC og IO-kort. Disse blokkene finnes under \Wincon toolbox\Quanser Consulting MQ3 Series\



Følgende blokker er tilgjengelige:

- Analog input
- Analog output
- Digital input
- Digital output
- Time-base. IO-kortet inneholder en klokke, som er koplet til IRQ5 på PCen. En får tilgang til denne klokka ved å bruke Time-base-blokken.
- Encoder input. Denne gir ut et heltall som indikerer tilstanden til en posisjonssensor.

4.2.2 Joystick

En Joystick skal brukes til å styre helikopteret. Joysticken gir ut -4.5V til +4.5V i både x- og y-retningen.

4.2.3 Powermodul

Effekten som driver motorene kommer fra en eller to Universal Power Modules.

5 Forsiktighetsregler

De følgende punktene sier seg nok selv, men for ordens skyld:

- Hold fingrene vekk fra propellene mens de går.
- Husk å skru av power-modulen når dere forlater laben.
- Vær klar til å ta imot helikopteret når det kjøres. Systemet er i utgangspunktet nok så ustabilt, veldig dyrt og tåler lite, så unngå crashlandinger. Vær også klar til å enten skru av kraftforsyningen, eller til å stoppe programmet.

6 Utgangspunkt for programmeringen

Det er ferdiglaget et SIMULINK blokkdiagram som skal brukes som utgangspunkt for programmene deres. Dere trenger følgende filer:

- Helikopter1.mdl, Helikopter2-4.mdl eller Helikopter5.mdl - Halvferdig Simulink-diagram. (Grunnet små hardware feil og forskjeller, er det laget 3 versjoner. Helikopterne er nummerert fra inngansdøra. Sitter du ved nr.1 laster du ned Helikopter1.mdl)
- init.m - inneholder de fysiske data til helikoptrene.

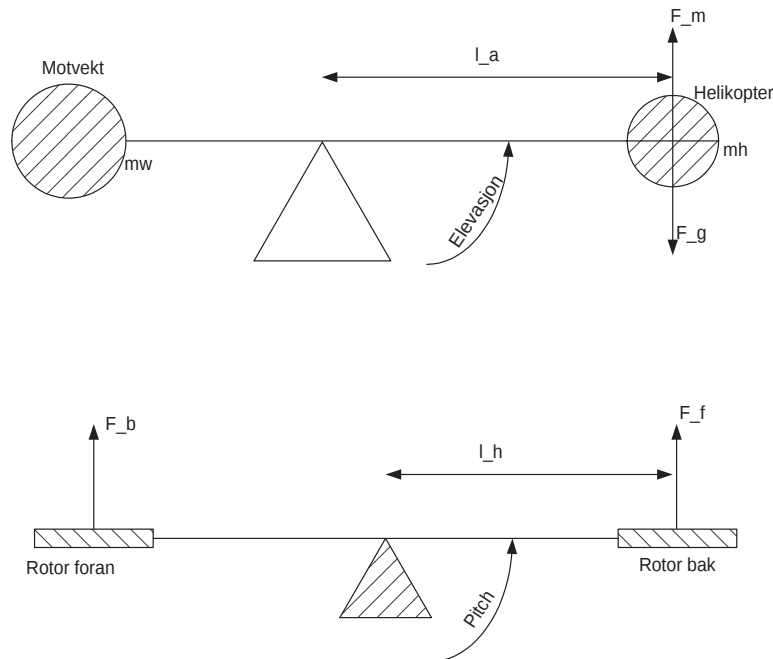
Disse filene kan lastes ned fra fagets hjemmeside (<http://www.itk.ntnu.no/fag/TTK4115/0vinger/>). Lag deres egen arbeidskatalog under C:\user\ og legg filene dit. Det blir ikke tatt backup av deres egne filer. Dette må dere sørge for selv. Det er mulig å mappe opp hjemmekatalogen deres på brutus på følgende vis: Høyreklikk my computer, velg Map Network Drive, skriv \\ brutus.itk.ntnu.no\brukernavn som path og brukernavn i Connect As-feltet. Pass på at Reconnect at Logon *ikke* er huket av. Klikk 'ok' og fyll inn passord. (Dersom brutus ikke er tilgjengelig kan stud-områdene brukes. En følger da samme fremgangsmåte, men bruker adressen: "\\sambastud.stud.ntnu.no\brukernavn").

7 Oppgaver

Oppgaven er delt inn i 4 deler:

1. Matematisk modellering. Det skal utvikles en matematisk modell av systemet som skal danne utgangspunktet for det videre arbeidet.
2. Monovariabel regulering av pitch/vandring. Det skal utvikles en regulator for en av utgangene til helikoptret. Det benyttes standard regulatorer som P, PD og PID.
3. Multivariabel regulering, dvs. regulering av flere utganger samtidig.
4. Tilstandsestimering. Systemet inneholder sensorer kun for tre tilstander, mens tilstandstilbakekopling krever at andre tilstander er kjent. Det skal derfor studeres forskjellige metoder for å estimere de ukjente tilstandene.

7.1 Del I. Matematisk modellering



De to rotorene er plassert symmetrisk i forhold til pitchaksen. Summen av kreftene $F_f + F_b$ utviklet av de to rotorene bestemmer helikopterets løft, dvs.

kraften F_m som virker vertikalt som igjen gir et rotasjonsmoment rundt elevasjonssaksen. Differansen mellom kreftene $F_f - F_b$ utviklet av de to rotorene gir et rotasjonsmoment rundt pitchaksen. Vi ønsker å regulere bevegelsen til helikopteret både horisontalt (der vil pitchvinkelen være bestemmende for kreftene) og vertikalt (der vil løftet være bestemmende for kreftene). Sammenhengen mellom spenningen på hver motor, og spenningene som vi bruker i regulatorene, kan derfor finnes slik:

$$V_s = V_f + V_b \quad (1)$$

$$V_d = V_f - V_b \quad (2)$$

Løsning av disse gir

$$V_f = 0.5(V_s + V_d) \quad (3)$$

$$V_b = 0.5(V_s - V_d) \quad (4)$$

En positiv V_d vil gi en positiv pitch-vinkel. En positiv pitchvinkel vil gi en negativ vandringsrate.

1. Sett opp differensialligninger for pitch (p), vandringshastighet (r) og elevasjon (e). Ta med luftmotstand i ligningen for vandringshastigheten. Denne modelleres vanligvis som proporsjonal med kvadratet av hastigheten. Vis at ligningene kan skrives som:

$$\ddot{p} = K_1 V_d \quad (5)$$

$$\dot{r} = -K_2 \sin p - K_1 r^2 \quad (6)$$

$$\ddot{e} = K_3 V_s - K_4 \quad (7)$$

som kan lineariseres til (ved å anta p og r liten)

$$\dot{r} = -K_2 p \quad (8)$$

Anta her at kraften fra propellene er proporsjonal med spenningen. Beregn verdier for parametrene K_1 , K_2 , K_3 og K_4 fra de oppgitte data.

2. Først skal helikopteret styres uten noen form for tilbakekopling. La signalet fra x-aksen til joysticken gå direkte inn på spenningsdifferansen V_d og signalet fra y-aksen til joysticken gå inn på summen av spenningene V_s . Motorene trenger ikke noen stor spenningsdifferanse for å øke pitchen, så det kan lønne seg å sette en liten forsterkning fra utgangen på joysticken til spenningsinngangen V_d . Det er nokså vanskelig å styre helikopteret på denne måten. Før dere prøver, bør dere tenke gjennom hva slags pådrag en trenger for å styre helikopteret. Begrunn dette med utgangspunkt i de ligningene som er satt opp i forrige oppgave. Vær forsiktig så ikke helikopteret går i bakken.

3. Finn verdien for spenningen V_s som medfører at helikopteret er i likevektstilstand. Sammenlign det du observerer fysisk med modellen utviklet ovenfor. Diskuter årsakene til avvikene og finn om mulig bedre verdier på parametrene i modellen.

7.2 Del II. Monovariabel regulering

I denne delen av oppgaven er det allerede lagt inn regulering av elevasjonen i programmet, og dere skal kun regulere pitchvinkelen eller vandringen. Dette fungerer slik at i løpet av 5 sekunder etter at programmet starter, vil helikopteret kjøres opp til en elevasjonsvinkel på 30° . (NB! Helikopterene har forskjellig vekt, så kontrolløren bør i enkelte tilfeller tunes. 30° kan også være for høyt og kan settes lavere etter ønske.) Denne regulatoren benytter summen av de to spenningene for å regulere elevasjonen. Etter dette kan dere ta over styringen av pitch/vandring. Dette skal dere gjøre ved å sette opp en differanse mellom de to spenningene på motorene. En differanse her fører til en endring av pitchvinkelen, som igjen vil endre vandringen.

1. Det skal legges inn en PD-regulator på pitchaksen. Denne er gitt ved

$$V_d = K_{pp}(p_c - p) - K_{pd}\dot{p} \quad (9)$$

der $K_{pp}, K_{pd} > 0$. Sett denne inn i differensialligningen for pitchaksen. Laplacetransformer og finn transferfunksjonen $\frac{p}{p_s}(s)$.

Finn fornuftige verdier for K_{pp} og K_{pd} . Når dere gjør dette, må dere tenke på hvor stor båndbredde det er mulig å oppnå for å styre pitchen. Målet må være en regulator som ikke gir svingninger, og som raskt regulerer pitchen til riktig verdi.

Sett PD-regulatoren inn i SIMULINK-diagrammet (velg selv om dere vil bruke blokker for kontinuerlig eller diskret tid). Koble utgangen av joysticken til settpunktet for pitchvinkelen, og prøv om styringen av helikopteret går lettere nå. Juster regulatoren slik at den fungerer bra.

2. Hastigheten til vandringen skal styres med en enkel P-regulator:

$$p_c = K_{rp}(r_c - r) \quad (10)$$

der $K_{rp} < 0$. Anta at pitch er perfekt regulert ($p = p_c$). Vis at transferfunksjonen fra referanse (r_c) til vandringsrate (r) kan skrives som

$$\frac{r}{r_c} = \frac{1}{1 + T_1 s} \quad (11)$$

Sett inn P-regulatoren i ligning (10) i Simulink-diagrammet, og velg en fornuftig forsterkning K_{rp} . Dere skal igjen prøve å styre helikopteret med joystick, men denne gangen skal utgangen på joysticken kobles til referansen for vandringshastigheten.

7.3 Del III. Multivariabel regulering

I denne delen av oppgaven skal dere styre begge motorpådragene samtidig med en multivariabel regulator slik at både pitchvinkel og elevasjonshastigheten reguleres til ønskede settpunktsverdier gitt av joysticken.

1. Sett opp ligningene for systemet (pitch og elevasjon) på tilstandsromform. Tilstandsvektoren skal være:

$$x = \begin{bmatrix} \dot{e} \\ p \\ \dot{p} \end{bmatrix} \quad (12)$$

og pådragsvektoren skal være spenningen på motorene (ikke summen og differansen som tidligere):

$$u = \begin{bmatrix} V_f \\ V_b \end{bmatrix} \quad (13)$$

2. Settpunkter for elevasjonshastigheten $\dot{e}$ og pitchvinkelen p skal gis ved hjelp av joysticken. Undersøk styrbarhet og lag en regulator for disse variablene basert på tilstandstilbakekopling og polplassering. Test ut regulatoren og finn poler som gir hurtig og nøyaktig respons.
3. Modifiser regulatoren i punktet ovenfor slik at den får integralvirkning.

7.4 Del IV. Tilstandsestimering

Resten av oppgavene vil ta for seg tilstandsestimering. Systemet inneholder sensorer for pitchvinkelen, elevasjonsvinkelen og vandringsvinkelen. I de foregående punktene i oppgaven har hastigheten til disse tre vinklene blitt beregnet ved hjelp av numerisk derivering og den resulterende støyen er redusert ved hjelp av lavpassfiltre. I denne delen av oppgaven skal det i stedet utvikles en observer basert på polplassering for å estimere de ikke målte tilstandene.

1. Sett opp ligningene for systemet på tilstandsromform. Tilstandsvektoren skal være:

$$x = \begin{bmatrix} e \\ \dot{e} \\ p \\ \dot{p} \\ \lambda \\ r \end{bmatrix} \quad (14)$$

2. Undersøk observerbarheten til systemet og lag en observer når målingene er

$$y = \begin{bmatrix} e \\ p \\ \lambda \end{bmatrix} \quad (15)$$

Prøv å regulere helikopteret (med regulatorene fra del 2, dvs pitch- og elevasjons-kontroller), men nå med tilbakekopling fra de estimerte tilstandene.

3. Vis at systemet er observerbart dersom en måler bare λ og e , men at det ikke er observerbart dersom en måler bare p og e . Lag også en observer basert på målingene

$$y = \begin{bmatrix} e \\ \lambda \end{bmatrix} \quad (16)$$

dvs. pitchvinkelmålingen benyttes ikke. Test ut estimatoren, og prøv å forklare ut fra fysiske betraktninger og modellen hvorfor enkelte estimat ikke blir så bra.