

Individuel pitchregulering af vindmølle

Allan Juul Larsen & Thomas Stampe Mogensen

Kongens Lyngby 2006

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Indhold

Abstract.	v
Forord	vii
Indledning	ix
0.1 Vindmølle specifikationer	xi
1 Designmodel	1
1.1 Aerodynamisk model	2
1.2 Aktuatorer	8
1.3 Samlet designmodel	15
2 Simulationsmodel	21
2.1 Vindkarakteristika	22
2.2 Tårnmodel	31

3	Kollektiv regulator strategi	35
3.1	Midområdet	36
3.2	Low- og highområdet	38
3.3	Topområdet	40
3.4	Start og stop	41
4	Kollektiv regulator design	43
4.1	PI regulator design	44
4.2	LQ regulering	48
4.3	Bumpless transfer	60
5	Test af kollektiv regulering	65
5.1	Testindhold	65
6	Modelopbygning individuel pitch	77
6.1	BEM	78
7	3D vindmodel	93
7.1	Deterministisk vindmodel	93
7.2	Stokastisk vindmodel	98
8	Individuel reguleringsstrategi	101
9	Individuel regulator design	105
9.1	Designmodel	106
9.2	PI- og P regulering	109

10 Test af individuel regulering	117
10.1 Testindhold	117
11 Konklusion	125
11.1 Kollektiv regulering	125
11.2 Individuel regulering	126
11.3 Perspektivering	127
A Litteraturliste	129
B Sybolliste	131
C Testplot	135
C.1 Verifikation af lineær vindmøllemodel	135
C.2 Bodeplot PI & P individuel	139
C.3 Kollektiv test	142
D Simulink diagrammer	147
D.1 Liste over simulink modeller:	147
E MatLab kode	157
E.1 Liste over anvendte MatLab filer	157
F HAWC	181
F.1 HAWC releterede filer	181
F.2 PASCAL DLL implementering	186

Abstract.

This project deal with individual control strategy for a 3 bladed 2MW wind turbine. The modelling is done using MatLab Simulink. The model is afterwards compared with HAWC a final element model developed and used by the wind energy department at Risø.

The first chapters of the report are related to collective pitch control. In these chapters a simple design model and a more advanced simulation model are build and tested. As collective control strategy PI and LQI controllers are investigated, and load considerations are included in the design. The individual pitch investigated is based on the Coleman transformation, and reducing the forces in yaw and tilt axis. This is made with 2 PI and 2 P regulators with different filters included.

The results achieved when combining the collective and the individual control strategy were good on both models. This shows that it is possible to design an individual controller on the simple Simulink model and implement it on the more complex model such as HAWC.

Forord

I dette projekt bliver emnet kollektiv og individuel regulering af en vindmølle behandlet. Projektet er et civilt afgangprojekt lavet af Allan Juul Larsen og Thomas Stampe Mogensen ved instituttet for matematisk modellering IMM på DTU.

Projektets primære mål har været at undersøge muligheden for at lave individuel pitch-regulering på en 3 bladet vindmølle.

I forbindelse med dette er det undersøgt, hvor effektivt forskellige kollektive regulatorer kan optimere effekt- og omdrejningsstabilitet i vindmøllen. Et andet mål har samtidig været at begrænse de belastninger som vindmøllen udsættes for og dermed forlænge møllens levetid. Den kollektive regulering arbejder primært med effekt- og omdrejningsstabilitet, og den individuelle regulering primært med belastningsreduktioner.

Vi vil gerne takke vores vejledere Niels Kjølstad Poulsen, som har været en stor hjælp gennem projektet, samt Henrik Bindner fra Risø. Desuden en stor tak til Lis Petersen og Jens Stampe Mogensen for korrekturlæsning samt Simon Juul Larsen for illustrationer.

Kongens Lyngby, marts 2006

Allan Juul Larsen

Thomas Stampe Mogensen

Indledning

Dette projekt omhandler styring af en 3 bladet variabel hastighedsvindmølle. Vindmøllen kan styres ved pitch-vinkel samt momentet leveret fra generatoren. Der opereres i fire forskellige områder alt efter, hvor meget vind, der er på et givet tidspunkt, som er betegnet low-, mid-, high- og topområderne. Der skal vælges forskellig reguleringsstrategi ud fra, hvilket område vindmøllen opererer i, da nogle af regulator kriterierne er forskellige alt efter området, der opereres i. For low- og highområderne gælder det, at omløbshastigheden skal holdes konstant. For midområdet skal den maksimale effekt tages ud af vinden, og for topområdet ønskes både omdrejninger og effekt begrænset til det nominelle niveau. Et andet kriterie for reguleringen er at minimere de strukturelle belastninger, som vindmøllen udsættes for. Dette kriterie gælder for alle de fire områder, der opereres i, og vil bevirke, at vindmøllen kan få en længere levetid, hvilket igen betyder en mere økonomisk energiproduktion.

Projektet omhandler to hovedområder: kollektiv og individuel pitch-regulering. Til hver af disse områder udvikles der indledningsvis en model af vindmøllen. Disse modeller tjener to formål; dels skal de anvendes til at designe reguleringen, og dels skal de anvendes til at verificere regulatorens ydelse efterfølgende. Modellen, der anvendes til at designe regulatorerne, ikke overraskende også kaldet designmodellen, er lavet med henblik på at indeholde alt den dynamik, der ønskes medtaget i reguleringen. Komplexiteten af designmodellen bestemmes ud fra et kompromis, da en høj kompleksitet nemt kan lede til et meget indviklet regulator design. Yderligere risikerer man at overspecialisere regulatoren til den konkrete designmodel, hvilket bevirker, at den bliver meget følsom over for modelvariation. Der er lavet to designmodeller: en for kollektiv pitch og en for individuel pitch.

Modellen anvendt til verifikation af reguleringen, også kaldet simulationsmodellen, skal derimod indeholde så meget dynamik som muligt, da det ønskes, at denne model afspejler den virkelige vindmølle.

Der er ligeledes lavet to simulationsmodeller: en for kollektiv pitch og en for individuel pitch. Den første er en udbygget version af designmodellen, hvor der er modelleret noget ekstra dynamik fra vindmøllen. For den kollektive pitch-simulationsmodel er der mulighed for at simulere et stokastisk vindfelt, hvor det kun er muligt at simulere et deterministisk vindfelt på den individuelle simulationsmodel. Ud over simulationsmodellerne, som er udviklet her i projektet, anvendes der en mere avanceret model med det beskedne navn Horizontal Axis Wind turbine Code eller HAWC. HAWC er et simuleringsværktøj udviklet af forskningscenter Risø gennem en årrække og er en yderst kompleks model af en vindmølle bygget ud fra final element teori. Denne model anvendes som det sidste led i simuleringerne og er det tætteste på en virkelig vindmølle, som reguleringen testes på.

Den kollektive pitch-regulering er det første emne, der bliver behandlet og fungerer som den primære regulering for vindmøllen. I modsætning til den individuelle pitch-regulering anvender den kollektive regulering generator momentet som styresignal, hvilket giver gode muligheder til hurtig regulering af effekten. Da alle 4 arbejdsområder bliver behandlet, kræver det, at der tages højde for, hvorledes der skal skiftes mellem de forskellige områder. Dette er nødvendigt, da det ikke er samme regulator, der virker i alle områderne. Det kan give anledning til nogle uønskede transienter i både omdrejninger og effekt, hvis skiftet mellem disse regulatorer ikke behandles på en ordentlig måde. Håndteringen af disse skift går ved navnet Bumpless transfer. Det primære område, der behandles, er topområdet, da det er her, der er flest frihedsgrader for reguleringen. Der er designet 4 forskellige regulatorer til dette område; en PI regulering samt 3 LQI regulatorer, hvor de er designet ud fra stigende kompleksitet i designmodellen. Det primære mål for den kollektive regulering er at holde omdrejningerne og effekten på det ønskede niveau, men for de to sidste LQI regulatorer er der også set på muligheden for belastningsreducering.

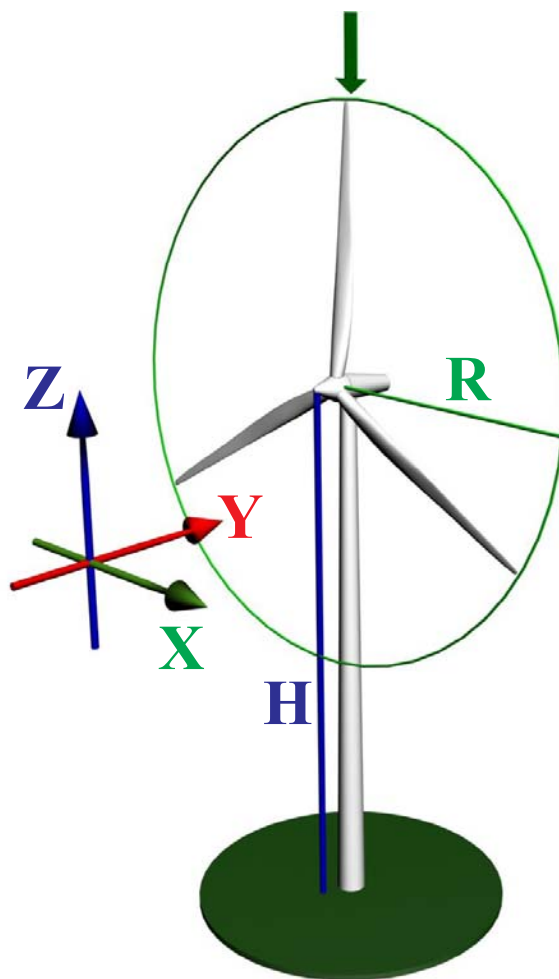
Den individuelle pitch-regulering designes med henblik på at reducere de strukturelle belastninger. Reguleringen er designet til at reducere forskellen i kraftpåvirkningen på det enkelte blad, som er forårsaget af et uhomogent vindfelt. Styresignalerne fra reguleringen adderes til styresignalerne fra den kollektive regulering. Den individuelle regulering virker således som en ekstra reguleringssløjfe, der reducerer belastningerne og kan ikke regulere vindmøllen uden den kollektive regulering. Designet af individuel regulering er kun foretaget i topområdet og er derfor også kun testet i dette område.

0.1 Vindmølle specifikationer

Vindmøllen, der arbejdes på, er en fiktiv 2MW vindmølle, hvor dataene dels er hentet fra HAWC filerne set i bilag F, som beskriver en NEG Micon NM80 mølle. Da HAWC ikke indeholder aktuatorodynamik, er realistiske data for disse fundet i samarbejde med Risø. De samlede data for vindmøllen er vist i tabel 1.

P_{nom}	=	2	[MW]	Nominel effekt.
ω_{gnom}	=	167.6	[rad/s]	Nominelt omdrejningstal ved generator.
ω_{gmin}	=	50.27	[rad/s]	Minimum omdrejningstal ved generator.
ω_{rnom}	=	1.97	[rad/s]	Nominelt omdrejningstal ved rotor.
ω_{rmin}	=	0.59	[rad/s]	Minimum omdrejningstal ved rotor.
N_g	=	1:85		Gearudveksling.
H	=	80	[m]	Højde af tårn.
R	=	40	[m]	Længde af et blad.
J_r	=	8.7e6	[kgm ²]	Inertimomet i rotoren.
J_g	=	150	[kgm ²]	Inertimomet i generatoren.
J_{tot}	=	9.78e6	[kgm ²]	Total inertimomet.
r_{top}	=	1.2	[m]	Tårnradius top.
r_{rod}	=	2.15	[m]	Tårnradius bund.
Dtr	=	3.486	[m]	Afstand fra rotor- til tårncentrum.
f_s	=	0.55	[Hz]	Fixed free resonansfrekvens for drivaksel.
f_t	=	0.3	[Hz]	Resonansfrekvens for tårn.
m_t	=	200	[T]	Ækvivalent masse til tårnmodellering.
$\dot{\theta}_{max}$	=	10	[deg/s]	Maks hastighed for pitch-ændring.
ω_n	=	8.88	[Hz]	Udæmpet egenfrekvens 'pitch-aktuator'.
ζ	=	0.9		Dæmpningsforhold for pitch-aktuator.
τ_{sv}	=	0.1	[s]	Tidskonstant i generator.

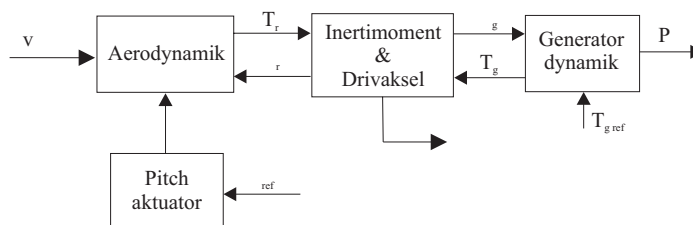
Tabel 1: Tabel over data på anvendt vindmølle



Figur 1: Koordinatsystem for vindmøllen.

Designmodel

Som udgangspunkt for design af kollektiv pitch-reguleringen laves der en simplificeret model af vindmøllen. Målet for en designmodel er at gøre den detaljeret nok til, at man kan designe en fornuftig regulator til den reelle vindmølle, men samtidig holde den så simpel, at regulator designet ikke bliver alt for komplekst. Den anvendte designmodel til kollektiv pitch-regulering vil her blive gennemgået. På figur 1.1. ses et blokdiagram over den komplette designmodel.



Figur 1.1: Blokdiagram over komplet designmodel

Som det ses består designmodellen af en aerodynamisk model, en pitch-aktuator, en generator model samt en model for drivakslen. Under beskrivelsen af den aerodynamiske model ses det, at denne del er et ulineært system. Det er derfor nødvendigt at linearisere modellen før et fornuftigt regulator design kan designes.

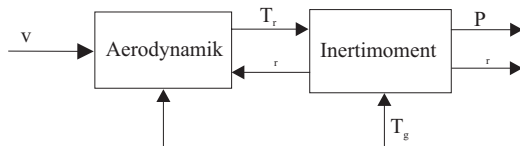
De resterende blokke vil blive modelleret med hhv. et 2.ordens system for pitch-regulatoren, et 1.ordens system for generator modellen og et 2.ordens system for dynamikken i drivakselen og gearet. Gearet antages at være ideelt. Da disse systemer allerede er lineære, vil det ikke være nødvendigt at udføre linearisering på dem.

1.1 Aerodynamisk model

Den simpleste måde at modellere en vindmølle på er ved et 1.ordens system. Her tages der kun højde for dynamik i omdrejningshastigheden, som bliver bestemt af den teoretisk tilgængelige energi i vinden samt møllens aerodynamiske effektivitet.

Vindmøllens aerodynamiske effektivitet er angivet ved en effektivitetstabel kaldet en c_p kurve. c_p kurven afhænger af pitchen på bladene samt forholdet mellem hastigheden på vinden og tip hastigheden på bladene kaldet tip speed ratio.

På figur 1.2 ses et blokdiagram over 1.ordens modellen af vindmøllen. 1.ordens modellen anvender ligningerne 1.1 til 1.3.



Figur 1.2: Blokdiagram over 1.ordens model af vindmølle.

$$J_{tot}\dot{\omega}_r = T_r - T_g N_g \quad (1.1)$$

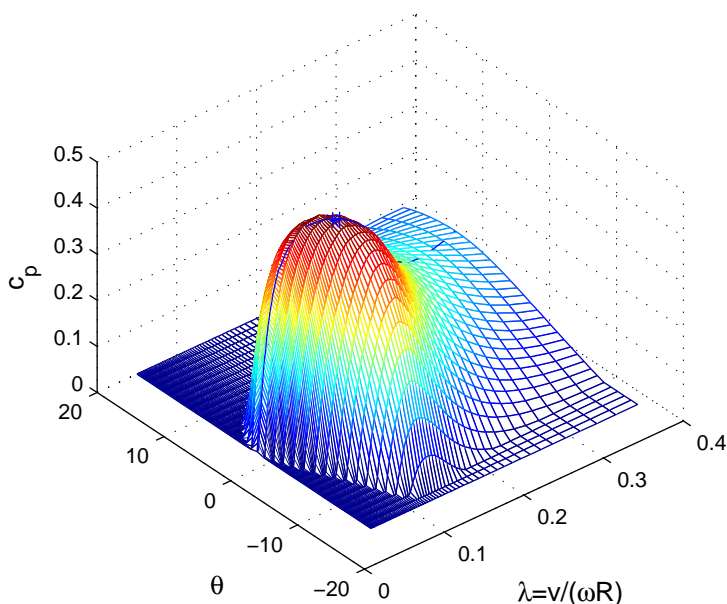
$$T_r = \frac{P_r}{\omega_r} \quad (1.2)$$

$$P_r = \frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta) \quad (1.3)$$

Ud fra disse ligninger er det nu muligt at opstille en differentilligning, der beskriver vindmøllen.

$$\dot{\omega}_r = \frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta)}{J_{tot}\omega_r} - \frac{T_g N_g}{J_{tot}} \quad (1.4)$$

Det er ud fra ligning 1.4 muligt at lave en simulink model over den simple vindmølle. Simulinkmodellen kan ses i figur 1.3.



Figur 1.4: c_p kurve for 2MW mølle

Den lineære \mathbf{A} matrice opnås ved at differentiere de enkelte differentialligninger, der beskriver systemet med hensyn til hver state. I dette tilfælde er der kun en state, nemlig omdrejningerne ω_r , og derfor kun en differentialligning.

\mathbf{B} matricen findes på samme måde. Her differentieres der blot med hensyn til alle input, som i dette tilfælde er vinden v den kollektive pitch θ og generatorens modmoment T_g . Dette giver ligning 1.8 og 1.9.

$$\mathbf{A} = \left[\frac{d\dot{\omega}_r}{d\omega_r} \right] \quad (1.8)$$

$$\mathbf{B} = \left[\begin{array}{ccc} \frac{d\dot{\omega}_r}{dv} & \frac{d\dot{\omega}_r}{d\theta} & \frac{d\dot{\omega}_r}{dT_g} \end{array} \right] \quad (1.9)$$

Differentialligningen i ligning 1.4 indsættes nu for $\dot{\omega}_r(t)$. Dette giver for \mathbf{A} :

$$\begin{aligned} \mathbf{A} &= \left[d \left(\frac{\frac{1}{2} \pi \rho R^2 v^3 c_p(\lambda, \theta)}{J_{tot} \omega_r} - \frac{T_g N_g}{J_{tot}} \right) \cdot \frac{1}{d\omega_r} \right] \\ \mathbf{A} &= \left[-\frac{\frac{1}{2} \pi \rho R^2 v^3 c_p(\lambda, \theta)}{J_{tot} \omega_r^2} + \frac{\frac{1}{2} \pi \rho R^2 v^3}{J_{tot} \omega_r} \cdot \frac{dc_p(\lambda, \theta)}{d\omega_r} \right] \end{aligned} \quad (1.10)$$

Det ses, at der stadig indgår differentialudtrykket $\frac{dc_p(\lambda, \theta)}{d\omega_r}$. Dette skyldes, at det ikke er muligt at bestemme denne koefficient analytisk. Det er derfor nødvendigt

at finde koefficienten numerisk. Dette gøres ved at tilpasse en 2.ordens funktion til kurven i lineariseringspunktet og derefter finde hældningen på denne funktion. Hældningen er fundet med matlab funktionen deriv.m, som kan ses i bilag E.

Det bemærkes, at c_p kurven er en funktion af λ og θ . Det er derfor ikke muligt at finde differentialet med hensyn til ω_r direkte. Da $\lambda = \frac{v}{\omega_r R}$ kan differentialreglen i ligning 1.11 anvendes.

$$\frac{dc_p(\lambda, \theta)}{d\omega_r} = \frac{dc_p(\lambda, \theta)}{d\lambda} \cdot \frac{d\lambda}{d\omega_r} \quad (1.11)$$

Det endelige udtryk for \mathbf{A} kan nu bestemmes og ses i ligning 1.12

$$\mathbf{A} = \left[-\frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta)}{J_{tot}\omega_r^2} + \frac{\frac{1}{2}\rho\pi R^2 v^3}{J_{tot}\omega_r} \cdot \frac{dc_p(\lambda, \theta)}{d\lambda} \cdot -\frac{v}{\omega_r^2 R} \right] \quad (1.12)$$

Hvor $\frac{dc_p(\lambda, \theta)}{d\lambda}$ findes numerisk.

Samme fremgangsmetode anvendes nu for at finde \mathbf{B} matricen. Hvilket resulterer i ligning 1.13.

$$\mathbf{B}^T = \begin{bmatrix} d\left(\frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta)}{J_{tot}\omega_r} - \frac{T_g N_g}{J_{tot}}\right) \cdot \frac{1}{dv} \\ d\left(\frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta)}{J_{tot}\omega_r} - \frac{T_g N_g}{J_{tot}}\right) \cdot \frac{1}{d\theta} \\ d\left(\frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta)}{J_{tot}\omega_r} - \frac{T_g N_g}{J_{tot}}\right) \cdot \frac{1}{dT_g} \end{bmatrix}$$

$$\mathbf{B}^T = \begin{bmatrix} \frac{1.5\pi\rho R^2 v^2 c_p(\lambda, \theta)}{J_{tot}\omega_r} + \frac{0.5\rho\pi R^2 v^3 \frac{dc_p(\lambda, \theta)}{d\lambda}}{J_{tot}R\omega_r^2} \\ \frac{\frac{1}{2}\rho\pi R^2 v^3}{J_{tot}\omega_r} \cdot \frac{dc_p(\lambda, \theta)}{d\theta} \\ -\frac{N_g}{J_{tot}} \end{bmatrix} \quad (1.13)$$

Til differentiering af c_p kurven med hensyn til vinden i $\mathbf{B}_{(1,1)}$ er differentialreglen i ligning 1.11 igen anvendt.

De analytiske udtryk for linearisering af den simple vindmøllemodel er blevet opstillet. Nu skal der blot findes et passende lineariseringspunkt. Dette bestemmes som de stationære værdier for vindmøllen ved en given vindhastighed.

Da vindmøllen kan operere i 4 forskellige områder: low, mid, high, og top, er det nødvendigt at beregne de stationære værdier på fire forskellige måder. De efterfølgende beskrevne metoder er implementeret i matlab funktionerne stpkt0.m, Prsol.m og fopt.m, som ses i bilag E.

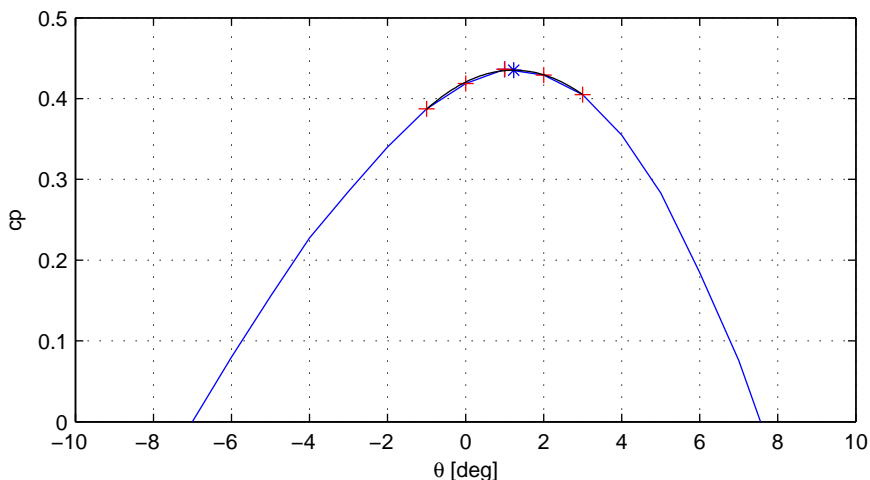
1.1.1 Low

Her ønskes vindmøllen at køre med fast omløbstal og samtidig give den maksimale effekt. Da vindhastigheden v og omdrejningshastigheden ω_{rmin} kendes, er det muligt at beregne tip speed ratio λ ved ligning 1.14.

$$\lambda = \frac{v}{\omega_{rmin}R} \quad (1.14)$$

Udtrykket for effekt i vinden kendes ud fra ligning 1.3. Det ønskes at få maksimal effekt ud af vinden, hvilket opnås ved at finde en θ værdi, der giver den maksimale c_p værdi.

På figur 1.5 ses et plot over c_p kurven som en funktion af θ ved en given λ værdi.



Figur 1.5: Maksimum på c_p kurve som funktion af θ ved en λ på 1.2618 svarende til en vindhastighed på $2m/s$.

Det ønskes altså at finde maksimalværdien på denne kurve. Den nemmeste måde at finde maksimum på kurven er selvfølgelig blot at udlæse den største værdi fra kurven. For at få et mere præcist toppunkt tilpasses en 2.ordens funktion til fem punkter omkring maksimalværdien. Sluttelig findes toppunktet på 2.ordens funktionen, hvilket giver den optimale θ værdi.

Da effekten nu kan beregnes, er det muligt at finde den sidste stationære værdi T_g ud fra ligning 1.15.

$$T_g = \frac{P}{\omega_{rmin}N_g} \quad (1.15)$$

Alle de stationære værdier for lowområdet er nu fundet.

1.1.2 Mid

I midområdet ønskes det, at vindmøllen kører med størst mulig effekt output. Dette opnås ved at finde det optimale punkt på c_p kurven både for λ og θ . Toppunktet findes næsten på samme måde som beskrevet for lowområdet. Her finder man som udgangspunkt igen et estimat af maksimum c_p som den maksimale tabelværdi. Som i lowområdet indlægges der en 2.ordens funktion omkring maksimalværdien, og toppunktet på funktionen findes. Her er det blot nødvendigt at anvende en 2.ordens flade, da c_p kurven er i tre dimensioner. θ_{nom} og λ_{nom} er kendte, og effekten kan derfor findes ud fra ligning 1.3. ω_r kan nu findes ud fra ligning 1.16.

$$\omega_r = \frac{v}{\lambda_{nom}R} \quad (1.16)$$

Når ω_r er beregnet, kan T_g findes med ligning 1.15. Hermed kan alle stationære værdier i midområdet bestemmes.

1.1.3 High

Dette område er identisk med lowområdet, og de stationære værdier beregnes derfor på samme måde. Her anvendes der blot det nominelle omdrejningstal ω_{rnom} i stedet for det minimale omdrejningstal ω_{rmin} .

1.1.4 Top

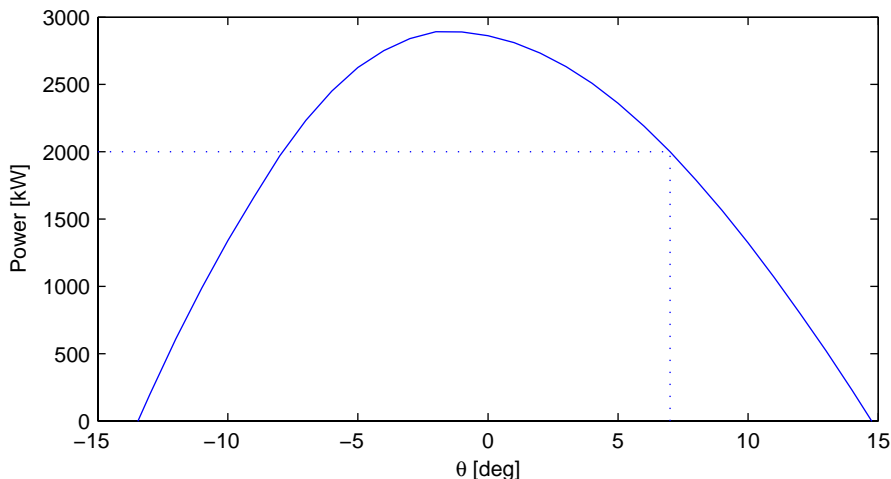
I topområdet kører vindmøllen både med det maksimale omdrejningstal ω_{rnom} og den maksimale effekt P_{nom} . T_g kan bestemmes ud fra ligning 1.17.

$$T_g = \frac{P_{nom}}{\omega_{rnom}N_g} \quad (1.17)$$

Da v og ω_{rnom} er kendt, kan λ findes ud fra ligning 1.18.

$$\lambda = \frac{v}{\omega_{rnom}R} \quad (1.18)$$

Der skal nu bestemmes en θ værdi, der giver den ønskede output-power P_{nom} . Da λ er blevet beregnet, kan θ bestemmes ud fra effekten. Dette ses på figur 1.6. Som det ses af figur 1.6 vil der være to mulige pitch-vinkler, som giver den ønskede effekt. Den mindste pitch-vinkel vil medføre, at vindmøllen staller, hvilket ikke er ønsket til dette formål. Det vil derfor i dette tilfælde altid være



Figur 1.6: Bestemmelse af pitch ud fra nominelt omdrejningstal og nominel effekt. Her vist ved en vindhastighed på $13 \frac{m}{s}$, svarende til en λ værdi på 0.1649.

den største af de to pitch-vinkler, der bliver valgt som resultat. Det er nu muligt at linearisere vindmøllen til en ønsket vindhastighed, resulterende i en lineær state spacemodel på formen, vist i ligning 1.6.

1.2 Aktuatorer

Da det ikke vil være muligt at ændre pitch-vinklen på et blad eller generatorens modmoment øjeblikkelig, er det nødvendigt at indføre aktuatorer på disse to input. Dette gøres som nedenstående beskrevet ved en 2.ordens funktion for pitch-aktuatoren og en 1.ordens funktion for generator modellen.

1.2.1 Pitchaktuator

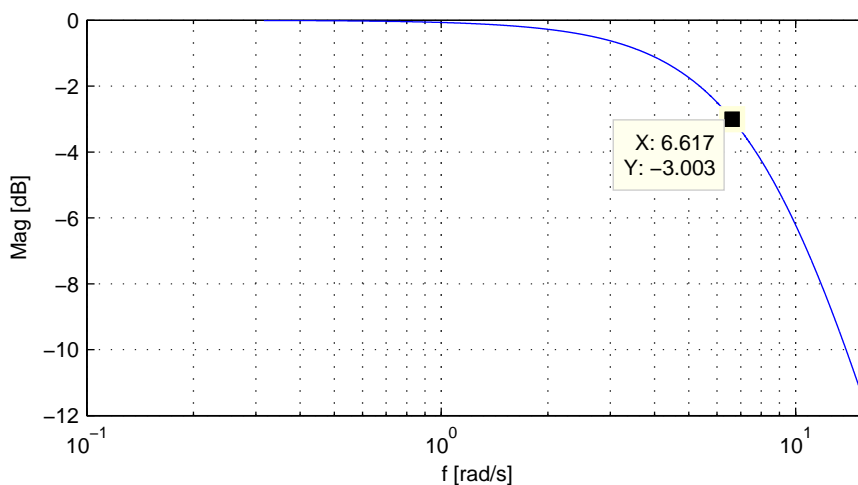
Pitchen på vindmøllen er styret af et hydraulisk servosystem. Dynamikken i aktuatoren er derfor kun beskrevet som en lineær 2.ordens model. 2.ordens modellen er vist i ligning 1.19.

$$\frac{d^2\theta}{dt^2} = \omega_n^2 \theta_{ref} - \frac{d\theta}{dt} 2\omega_n \zeta - \omega_n^2 \theta \quad (1.19)$$

Ved laplastransformation og en smule omrokering kan overføringsfunktionen for aktuatoren ses i ligning 1.20.

$$\frac{\theta}{\theta_{ref}} = \frac{\omega_n^2}{s^2 + 2\omega_n\zeta s + \omega_n^2} \quad (1.20)$$

I Risø's dobbeltmodel er den udæmpede egenfrekvens og dæmpningsforholdet opgivet til $\omega_n = 8.88$ og $\zeta = 0.9$. Dette giver aktuatoren en båndbredde, set på figur 1.7.



Figur 1.7: Båndbredde af pitch-aktuator

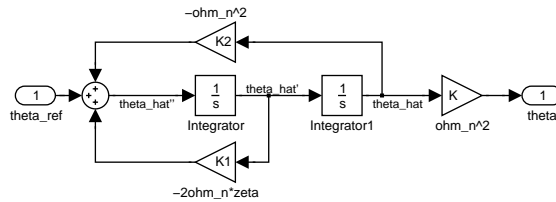
Da systemet allerede er lineært, kan man ved anvendelse af teorien i [Reg2 04] opstille state spacemodellen direkte ud fra overføringsfunktionen. State space modellen er således på companionform 1 og ses i ligning 1.21.

$$\begin{aligned} \mathbf{A}_\theta &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \\ \mathbf{B}_\theta &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \mathbf{C}_\theta &= [\omega_n^2 \quad 0] \end{aligned} \quad (1.21)$$

Hvor:

$$\bar{x} = \begin{bmatrix} \hat{\theta} \\ \dot{\hat{\theta}} \end{bmatrix} \quad \bar{u} = [\theta_{ref}] \quad \bar{y} = [\theta]$$

På figur 1.8 ses en simulink model over pitch-aktuatoren.



Figur 1.8: Simulinkdiagram over pitch-aktuator

2.ordens modellen samt de opgivne konstanter er som nævnt hentet fra Risø's model dobbelt. Det er samtidig oplyst [Risø 05], at den maksimale hastighed, pitchen på bladene kan ændres med, er $\pm 10\text{grader/s}$.

1.2.2 Generatormodel

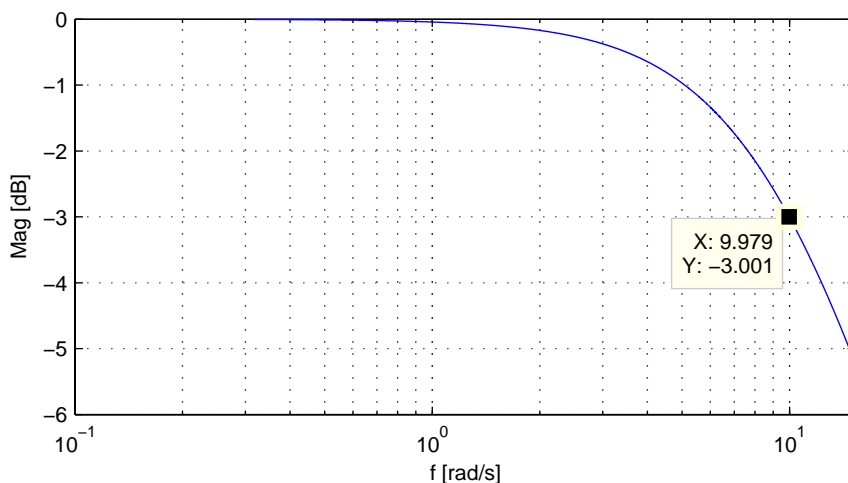
Der anvendes en asynkron generator, der kan levere et ønsket modmoment, som det er muligt at regulere vindmøllens omdrejningshastighed og effekt output med. Af [Engelen 05] ses det, at en sådan generator kan modelleres tilfredsstillende med et simpelt 1.orden system, set i ligning 1.22.

$$\frac{dT_g}{dt} = \frac{T_{gref} - T_g}{\tau_{sv}} \quad (1.22)$$

Overføringsfunktionen, som er vist i ligning 1.23, kan igen opnås ved laplace-transformation og en smule omrokering.

$$\frac{T_g}{T_{gref}} = \frac{\frac{1}{\tau_{sv}}}{1 + \frac{1}{\tau_{sv}}} \quad (1.23)$$

Tidskonstanten er igen hentet fra dobbeltmodellen $\tau_{sv} = 0.1$. Dette giver en båndbredde for aktuatoren, vist på figur 1.9.



Figur 1.9: Båndbredde af generator model

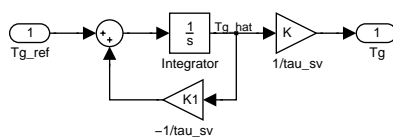
Systemet er lineært og state spacemodellen, som er vist i ligning 1.24, kan derfor stilles direkte op ud fra overføringsfunktionen.

$$\mathbf{A}_{T_g} = \left[-\frac{1}{\tau_{sv}} \right] \quad \mathbf{B}_{T_g} = [1] \quad \mathbf{C}_{T_g} = \left[\frac{1}{\tau_{sv}} \right] \quad (1.24)$$

Hvor:

$$\bar{x} = \left[\hat{T}_g \right] \quad \bar{u} = [T_{gref}] \quad \bar{y} = [T_g]$$

På figur 1.10 ses en simulink model over generator modellen.



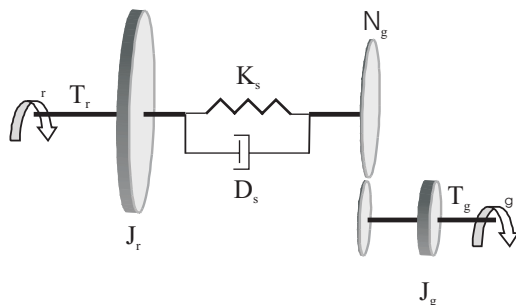
Figur 1.10: Simulinkdiagram over generator model

1.2.3 Drivaksel

Akslen, som driver gearkassen og generatoren, er ikke fuldstændig stiv. Belastningerne, der kommer på drivakslen og gearret, er afgørende for deres levetid.

For at få mulighed for at reducere belastningerne og dermed give vindmøllen en længere levetid, er dynamikken i drivakslen taget med i designmodellen.

På figur 1.11 ses et blokdiagram over, hvorledes det er valgt at modellere den mekaniske del af vindmøllen. Drivakslen er modelleret som en fjeder med en



Figur 1.11: Blokdiagram over mekanisk opbygning af drivaksel

dæmpning og en masse i hver ende. Det bemærkes, at det kun er drivakslen på rotorsiden, der betragtes som elastisk, og at gearret anses for at være tabsfrit. Dynamikken i drivakslen og strukturdelen i den simplificerede vindmølle, som ses på figur 1.2, er her blevet sat sammen til et samlet system. Der kan nu ud fra figur 1.11 opstilles en differentialligning, set i ligning 1.25, der beskriver systemet. Hvor $\omega_r = \dot{\phi}_r$ og $\omega_g = \dot{\phi}_g$.

$$T_r - T_g N_g = J_r \ddot{\phi}_r - J_g N_g \ddot{\phi}_g + \left(\dot{\phi}_r - \frac{\dot{\phi}_g}{N_g} \right) D_s + \left(\phi_r - \frac{\phi_g}{N_g} \right) K_s \quad (1.25)$$

Den mekaniske model kan, som det ses, beskrives med et 4.ordens system. Ordenen af systemet kan dog reduceres med én, da $\phi_r - \frac{\phi_g}{N_g}$ kan reduceres til $\Delta\phi$, og der herved kan spares en integration. For nemt at kunne opstille en simulink model omskrives differentialligningen på formen, vist i ligningerne 1.26 og 1.27.

$$\ddot{\phi}_r = \frac{T_r - T_g N_g + J_g N_g \ddot{\phi}_g - \left(\dot{\Delta\phi} \right) D_s - (\Delta\phi) K_s}{J_r} \quad (1.26)$$

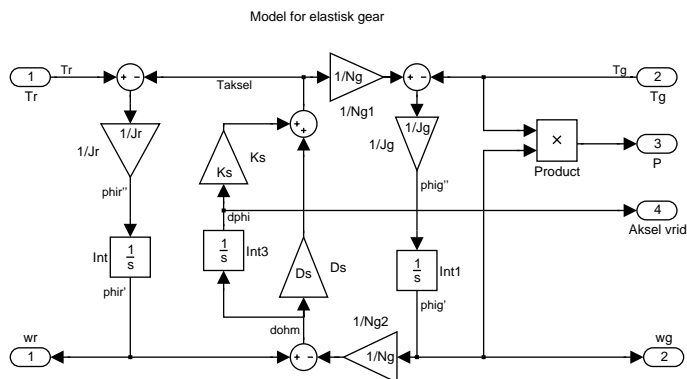
$$\ddot{\phi}_g = \frac{T_r - T_g N_g - J_r \ddot{\phi}_r + \left(\dot{\Delta\phi} \right) D_s + (\Delta\phi) K_s}{J_g N_g} \quad (1.27)$$

Her skal det bemærkes, at $T_g N_g = J_g N_g \ddot{\phi}_g$ og $T_r = J_r \ddot{\phi}_r$. Ligningerne kan derfor reduceres, som vist i ligning 1.28 og 1.29.

$$\ddot{\phi}_r = \frac{T_r - (\dot{\Delta\phi}) D_s - (\Delta\phi) K_s}{J_r} \quad (1.28)$$

$$\ddot{\phi}_g = \frac{-T_g N_g + (\dot{\Delta\phi}) D_s + (\Delta\phi) K_s}{J_g N_g} \quad (1.29)$$

Ud fra disse udtryk kan der nu opstilles en simulink model. Simulinkmodellen kan ses på figur 1.12.



Figur 1.12: Simulinkmodel over drivaksel

Der kan nu enten ud fra simulink modellen eller fra differentialligningerne opstilles en state spacemodel, som ses i ligning 1.30.

$$\mathbf{A}_a = \begin{bmatrix} -\frac{D_s}{J_r} & \frac{D_s}{J_r N_g} & -\frac{K_s}{J_r} \\ \frac{D_s}{J_g N_g} & -\frac{D_s}{J_g N_g^2} & \frac{K_s}{J_g N_g} \\ 1 & -\frac{1}{N_g} & 0 \end{bmatrix}$$

$$\mathbf{B}_a = \begin{bmatrix} \frac{1}{J_r} & 0 \\ 0 & -\frac{1}{J_g} \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C}_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & T_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.30)$$

Hvor:

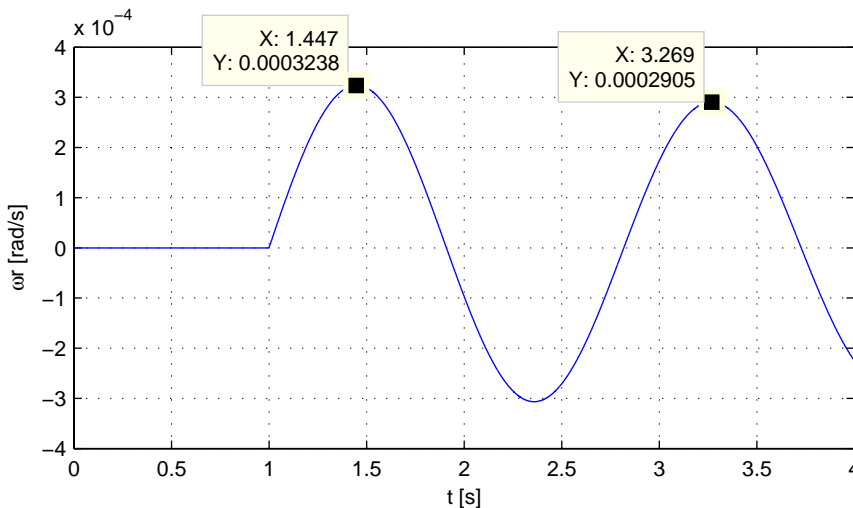
$$\bar{x} = \begin{bmatrix} \omega_r \\ \omega_g \\ \Delta\phi \end{bmatrix} \quad \bar{u} = \begin{bmatrix} T_r \\ T_g \end{bmatrix} \quad \bar{y} = \begin{bmatrix} \omega_r \\ \omega_g \\ P \\ \Delta\phi \end{bmatrix}$$

Dæmpningen D_s samt fjederkonstanten K_s er endnu ikke blevet bestemt for drivakslen. Disse konstanter bestemmes ud fra akslens resonansfrekvens f_s ud fra ligning 1.31.

$$\begin{aligned} K_s &= (2\pi f_s)^2 \cdot J_r \\ D_s &= K_s/100 \end{aligned} \quad (1.31)$$

Som det ses, bliver D_s kun groft beregnet ud fra en fingerregel om, at den skal ligge mellem en tiende- og en hundrededel af K_s . Resonansfrekvensen er taget ud fra målinger foretaget i [Risø 05], hvor den er målt med en fastholdt generator og en fri roter, der påtrykkes en impuls. Resonansfrekvensen blev her målt til 0.55Hz. Dette giver $K_s = 1.039 \cdot 10^8 N/m$ og $D_s = 1.039 \cdot 10^6 s^{-1}$.

For at verificere, at modellen med de beregnede konstanter, har den rigtige egenfrekvens, er der lavet et trinrespons på drivakselmodellen set på figur 1.13. ω_g er i testen sat til nul for at simulere en fastsat generator, og der påtrykkes et trin i rotormomentet. Trinresponsen ses på figur 1.13.



Figur 1.13: Verifikation af resonansfrekvens på drivaksel. Trin på T_r til tiden 1 sek. med en amplitude fra 0 - 10000.

Det ses at periodetiden er $3.27 - 1.45 = 1.82 \text{ sek} = 0.5495 \text{ Hz}$, hvilket anses for

at ligge tilstrækkelig tæt på de ønskede $0.55Hz$, da der er en lille unøjagtighed i aflæsningen af periodetiden.

1.3 Samlet designmodel

Alle delene i designmodellen er nu blevet beskrevet. I bilag D kan den samlede simulink model over systemet ses. Der laves nu en samlet state spacemodel for designmodellen. Da der i regulator designet, se kapitel 4, anvendes designmodeller af stigende kompleksitet, er det en fordel at lave sammenlægningen af vindmøllemodellen i to trin.

1.3.1 Drivaksel og aerodynamik

Først laves en model uden aktuatorerne indsat.

Her skal modellen, vist i ligning 1.30 og modellen, beskrevet i ligning 1.12 og 1.13, sammenlægges. Dette gøres nemmest ved at opstille differentialligningerne for figur 1.12 på formen, vist i ligning 1.32 til 1.34.

$$\dot{\omega}_r = \frac{T_r}{J_r} - \frac{K_s}{J_r} \Delta\phi - \frac{D_s}{J_r} \omega_r + \frac{D_s}{J_r N_g} \omega_g \quad (1.32)$$

$$\dot{\omega}_g = -\frac{T_g}{J_g} + \frac{K_s}{J_g N_g} \Delta\phi + \frac{D_s}{J_g N_g} \omega_r - \frac{D_s}{J_g N_g^2} \omega_g \quad (1.33)$$

$$\dot{\Delta\phi} = \omega_r - \frac{\omega_g}{N_g} \quad (1.34)$$

Fra ligning 1.2 og 1.3 kan ligning 1.35 opskrives.

$$T_r = \frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta)}{\omega_r} \quad (1.35)$$

State space modellen kan nu findes ud fra ligning 1.32 til 1.35 ved at differentiere ligningerne med hensyn til states og input. Dette giver state spacemodellen, vist

i ligning 1.36.

$$\begin{aligned}
 \mathbf{A}_{a_1} &= \begin{bmatrix} \mathbf{A}_{a_1(1,1)} & \frac{D_s}{J_r N_g} & -\frac{K_s}{J_r} \\ \frac{D_s}{J_g N_g} & -\frac{D_s}{J_g N_g^2} & \frac{K_s}{J_g N_g} \\ 1 & -\frac{1}{N_g} & 0 \end{bmatrix} \\
 \mathbf{B}_{a_1} &= \begin{bmatrix} \frac{1.5\pi\rho R^2 v^3 c_p(\lambda, \theta)}{J_r \omega_r} + \frac{0.5\rho\pi R^2 v^3 \frac{dc_p(\lambda, \theta)}{d\lambda}}{J_r R \omega_r^2} & \frac{1}{2} \frac{\rho\pi R^2 v^3}{J_r \omega_r} \cdot \frac{dc_p(\lambda, \theta)}{d\theta} & 0 \\ 0 & 0 & -\frac{1}{J_g} \\ 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{C}_{a_1} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & T_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.36)
 \end{aligned}$$

Hvor $\mathbf{A}_{a_1(1,1)}$ er givet ved ligning 1.37

$$\mathbf{A}_{a_1(1,1)} = \left[-\frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \theta)}{J_r \omega_r^2} + \frac{\frac{1}{2}\rho\pi R^2 v^3}{J_r \omega_r} \cdot \frac{dc_p(\lambda, \theta)}{d\lambda} \cdot -\frac{v}{\omega_r^2 R} - \frac{D_s}{J_r} \right] \quad (1.37)$$

\mathbf{C}_{a_1} har ikke ændret sig fra ligning 1.30. Her skal det bemærkes, at state-vektoren \bar{x} og output-vektoren \bar{y} stadig er bevaret fra ligning 1.30. Men input-vektoren \bar{u} er fra ligning 1.6.

1.3.2 Drivaksel, aerodynamik og aktuatorer

Modellen udvides nu med de to aktuatorer. Dette gøres ved at definere nye state-, input- og output-vektorer, som vist i ligning 1.38.

$$\begin{aligned}
 \bar{x}^T &= \left[\hat{\theta} \quad \dot{\hat{\theta}} \quad \hat{T}_g \quad \omega_r \quad \omega_g \quad \Delta\phi \right] \\
 \bar{u} &= \begin{bmatrix} v \\ \theta_{ref} \\ T_{gref} \end{bmatrix} \quad \bar{y} = \begin{bmatrix} \omega_r \\ \omega_g \\ P \\ \Delta\phi \end{bmatrix} \quad (1.38)
 \end{aligned}$$

Der opstilles 6 differentialligninger, som beskriver det komplette system.

$$\dot{\hat{\theta}} = \dot{\hat{\theta}} \quad (1.39)$$

$$\ddot{\hat{\theta}} = -\omega_n^2 \hat{\theta} - 2\zeta\omega_n \dot{\hat{\theta}} + \theta_{ref} \quad (1.40)$$

$$\dot{\hat{T}}_g = -\frac{1}{\tau_{sv}} \hat{T}_g + T_{gref} \quad (1.41)$$

$$\dot{\omega}_r = \frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \hat{\theta}\omega_n^2)}{\omega_r J_r} - \frac{K_s}{J_r} \Delta\phi - \frac{D_s}{J_r} \omega_r + \frac{D_s}{J_r N_g} \omega_g \quad (1.42)$$

$$\dot{\omega}_g = -\frac{\hat{T}_g}{J_g \tau_{sv}} + \frac{K_s}{J_g N_g} \Delta\phi + \frac{D_s}{J_g N_g} \omega_r - \frac{D_s}{J_g N_g^2} \omega_g \quad (1.43)$$

$$\dot{\Delta\phi} = \omega_r - \frac{\omega_g}{N_g} \quad (1.44)$$

Ud fra differentialligningerne kan der nu opstilles state-matricen \mathbf{A}_m , ved at differentiere alle differentialligningerne med hensyn til de 6 states. Inputmatricen \mathbf{B}_m findes ved differentiering med hensyn til de 3 input. Den samlede state spacemodel er vist i ligning 1.45.

$$\mathbf{A}_m = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\omega_n^2 & -2\zeta\omega_n & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{sv}} & 0 & 0 & 0 \\ \omega_n^2 \frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \hat{\theta}\omega_n^2)}{J_r \omega_r} & 0 & 0 & \mathbf{A}_{m(4,4)} & \frac{D_s}{J_r N_g} & -\frac{K_s}{J_r} \\ 0 & 0 & -\frac{1}{\tau_{sv} J_g} & \frac{D_s}{J_g N_g} & -\frac{D_s}{J_g N_g^2} & \frac{K_s}{J_g N_g} \\ 0 & 0 & 0 & 0 & -\frac{1}{N_g} & 0 \end{bmatrix}$$

$$\mathbf{B}_m = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1.5\pi\rho R^2 v^2 c_p(\lambda, \hat{\theta}\omega_n^2)}{J_r \omega_r} + \frac{0.5\pi\rho R^2 v^3 \frac{dc_p(\lambda, \hat{\theta}\omega_n^2)}{d\lambda}}{J_r R \omega_r^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{C}_m = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \omega_g \frac{1}{\tau_{sv}} & 0 & \frac{\hat{T}_g}{\tau_{sv}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.45)$$

Hvor $\mathbf{A}_{m(4,4)}$ er givet ved ligning 1.46.

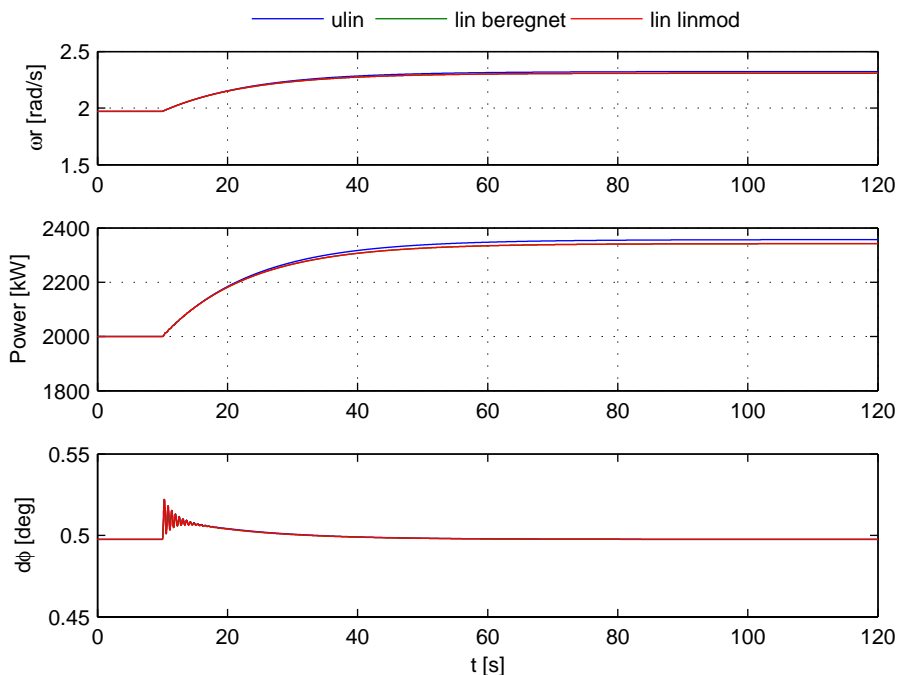
$$\mathbf{A}_{m(4,4)} = -\frac{\frac{1}{2}\pi\rho R^2 v^3 c_p(\lambda, \hat{\theta}\omega_n^2)}{J_r \omega_r^2} + \frac{\frac{1}{2}\pi\rho R^2 v^3}{J_r \omega_r} \cdot \frac{dc_p(\lambda, \hat{\theta}\omega_n^2)}{d\lambda} - \frac{v}{\omega_r^2 R} - \frac{D_s}{J_r} \quad (1.46)$$

Det bemærkes at $\mathbf{A}_{m(4,1)}$ er givet ved $\mathbf{B}_{a_1(1,2)} \cdot \mathbf{C}_{\theta(1)}$. Altså input-gainen til pitchen fra akselmodellen og output-gainen fra pitch-aktuatoren. Det samme gælder for $\mathbf{A}_{m(3,5)}$, her er det blot $\mathbf{B}_{a_1(2,3)} \cdot \mathbf{C}_{T_g(1)}$, altså input-gain til generator momentet i akselmodellen og output-gainen fra generator modellen. Effekt-outputtet er nu givet ved funktionen, vist i ligning 1.47.

$$P = \hat{T}_g \cdot \frac{1}{\tau_{sv}} \cdot \omega_g \quad (1.47)$$

Outputvektoren for $\mathbf{C}_{m(3)}$, altså effekt outputtet opnås derfor ved at differentiere ligning 1.47 med hensyn til alle states, hvor det er de to states \hat{T}_g og ω_g der giver et resultat forskelligt fra 0.

Der er nu lavet en komplet lineariseret designmodel over vindmøllen. Der kan ses et simulink diagram over modellen på bilag D. For at verificere, at den lineære model opfører sig i overensstemmelse med den ikke lineære model, laves der nu en sammenligning mellem trinrespons på de to modeller. Yderligere laves der en sammenligning med en lineær model, der er opnået ved at anvende matlab funktionen `linmod.m`. Dette gøres, da det under en designproces, hvor der hyppigt foretages ændringer i modellen, er bekvemt at kunne bruge denne matlab funktion, da den hurtigt giver et anvendeligt resultat. Verifikationen består som sagt af trinresponse på input hhv. vind-, pitch- og modmoment. Det er valgt kun at vise responset på de tre output ω_r , P og $\Delta\phi$. Vindmøllen er i de viste plot blevet lineariseret omkring en vindhastighed på 8 og 14 m/s . Amplituden for trinene er hhv. v 1 m/s , θ 1 $grad$ og T_g 100 N . Resultatet af disse simuleringer kan ses i bilag C. På figur 1.14 er verificeringen for 14 m/s med et step i vinden vist.

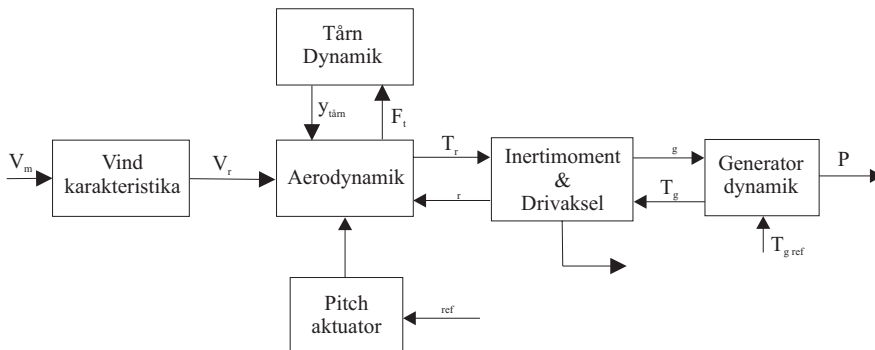


Figur 1.14: Verifikation af lineær model ved 14 m/s. Trin på vind med amplituden 1 m/s til tiden 10 sek

Det ses af plottene, at der er en god overensstemmelse mellem den lineariserede og den ulineære model. Denne overensstemmelse bliver selvfølgelig ringere, hvis trin størrelsen øges, da man vil bevæge sig længere væk fra det lineariserede område. Ved et trin på pitchen, når modellen er lineariseret omkring 8 m/s, ses det, at linmod lineariseringen giver et trin i modsat retning af de to andre. Yderligere er der dårlig overensstemmelse mellem den ulineære- og den beregnede lineære model. Dette skyldes, at en linearisering i dette område foregår på toppen af c_p kurven. Derfor vil en lille fejl i lineariseringen kunne give anledning til en stor afvigelse fra den ulineære model. Eller som for linmod's tilfælde et forkert fortegn. Dette er selvfølgelig meget uheldigt, hvis der skal designes en regulator. Det er imidlertid så bekvemt at reguleringen i dette område foregår ud fra $p\omega$ beregninger se kapitel 3, som ikke arbejder på en lineariseret model.

Simulationsmodel

Udover designmodellen er der lavet en simulationsmodel, som ses på figur 2.1.



Figur 2.1: Blokdiagram over simulationsmodel

Som det ses, er der tilføjet to blokke; vindkarakteristika og tårndynamik. Begge disse blokke kunne også være inkluderet i designmodellen. Det blev valgt at lægge disse blokke over i simulationsmodellen for at få et simpelt regulator design til den kollektive pitch. Disse blokke giver mulighed for at kunne simulere, hvordan de designede regulatorer opfører sig i et stokastisk vindfelt samt at simulere, hvor store tårnbevægelser, der kommer ved forskellige påvirkninger, og indvirkningen af denne svingning på reguleringen.

2.1 Vindkarakteristika

Det ønskes at kunne beskrive den stokastiske vindvariation, der påvirker vindmøllen. Da vindmøllemodellen kun indeholder ét input for vinden, er det nødvendigt at finde en ækvivalent værdi for variationen af vinden over hele rotorarealet. Dette gøres i to skridt. Først beskrives den stokastiske variation af vinden i et enkelt punkt, og siden udregnes den ækvivalente variation af det samlede vindfelt. Vinden vil selvfølgelig variere i alle retninger, men da dette vil blive for omfattende at modellere, vil vinden kun betragtes som værende varierende i retningen (y-retningen se figure 1), der er vinkelret på rotoren.

2.1.1 Punktvind

Der er lavet flere forslag til, hvorledes en punktvind kan beskrives/modelleres f.eks [Kaimal 72] eller [Kármán 48]. Den her anvendte modellering er baseret på [Højstrup 82], som foreslår en beskrivelse af punkt vinden med effekt tæthedsspektret, vist i ligning 2.1.

$$\frac{fS(f)}{v_{fr}^2} = \frac{0.5f_i}{(1 + 2.2f_i)^{\frac{5}{3}}} \left(\frac{z_i}{-L} \right)^{\frac{2}{3}} + \frac{105f_{ru}}{(1 + 33f_{ru})^{\frac{5}{3}}} \frac{\left(1 - \frac{H}{z_i}\right)^2}{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}} \quad (2.1)$$

Hvor: $f_i = \frac{fz_i}{v_m}$, og $f_{ru} = f \frac{H}{v_m} \frac{1}{1 + 15\frac{H}{z_i}}$.

v_{fr} er friktionshastigheden, f_{zi} er højden på den laveste inversion eller det atmosfæriske grænselag, f er frekvensen, L er Monin-Obukhov længden, H er højden på tårnet og v_m er middelvinden.

For at finde et udtryk for variansen i vinden, foretages der en integration af ligning 2.1 med frekvensen gående fra $-\infty$ til ∞ . Dette resulterer i ligning 2.2.

$$\frac{\sigma^2}{v_{fr}^2} = 0.6 \left(\frac{z_i}{-L} \right)^{\frac{2}{3}} + \frac{105}{22} \frac{\left(1 - \frac{H}{z_i}\right)^2}{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}} \quad (2.2)$$

Som det ses, er funktionen opdelt i to led. Det første led beskriver vinden ved stabile forhold. Det andet led beskriver vinden ved mere ustabile forhold, hvor $\frac{z_i}{-L}$ er en faktor, der går mod 0 ved stigende ustabilitet. Vindmøllen er placeret tæt ved jorden, hvor forholdene typisk er ustabile, hvilket bevirker, at der i modelleringen kan ses bort fra det første led. Dette svarer til, at $z_i / -L = 0$ og

giver ligningerne 2.3 og 2.4.

$$\frac{fS(f)}{v_{fr}^2} = \frac{105f_{ru}}{(1+33f_{ru})^{\frac{5}{3}}} \frac{\left(1 - \frac{H}{z_i}\right)^2}{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}} \quad (2.3)$$

$$\frac{\sigma^2}{v_{fr}^2} = \frac{105}{22} \frac{\left(1 - \frac{H}{z_i}\right)^2}{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}} \quad (2.4)$$

For at eftervise dette betragtes et konkret eksempel, hvor det vises, hvilken indflydelse den stabile og ustabile del har på det samlede spektrum. Der antages en middelvind på 12 m/s, møllehøjden er på 80 m, og det atmosfæriske grænselag ligger på 1000m taget fra [Antti 94]. Monin-Obukhov længden kan findes ud fra ligning 2.5

$$L = \frac{\rho C_P T_0 v_{fr}^3}{k \cdot g H_1} \quad (2.5)$$

T_0 er referencetemperaturen = 15°C = 258°K. ρ er luftens densitet = 1.25. k er von Kármán konstant = 0.4. C_P er specific heat at constant pressure = 1004-1030 Jkg⁻¹K⁻¹. H_1 er surface heat flux = 50-200 W/m².

Friktionshastigheden v_{fr} kan findes ud fra ligning 2.6 taget fra [Manwell 02]. Det skal hertil nævnes, at dette udtryk antager relative stabile forhold. Dette vil selvfølgelig give en vis unøjagtighed, da det ønskes at vise dominansen af den ustabile del af ligning 2.1. Dog anses denne unøjagtighed ikke for at være afgørende, da det er en tendens i funktionen, der ønskes påvist. Som det vil blive vist senere, kan der opstilles et nyt udtryk for friktionshastigheden, når simplificeringen af funktionen, som vil være gældende for ustabile forhold, er foretaget.

$$v_m = \frac{v_{fr}}{k} \log\left(\frac{H}{z_0}\right) \Leftrightarrow v_{fr} = \frac{v_m k}{\log\left(\frac{H}{z_0}\right)} \quad (2.6)$$

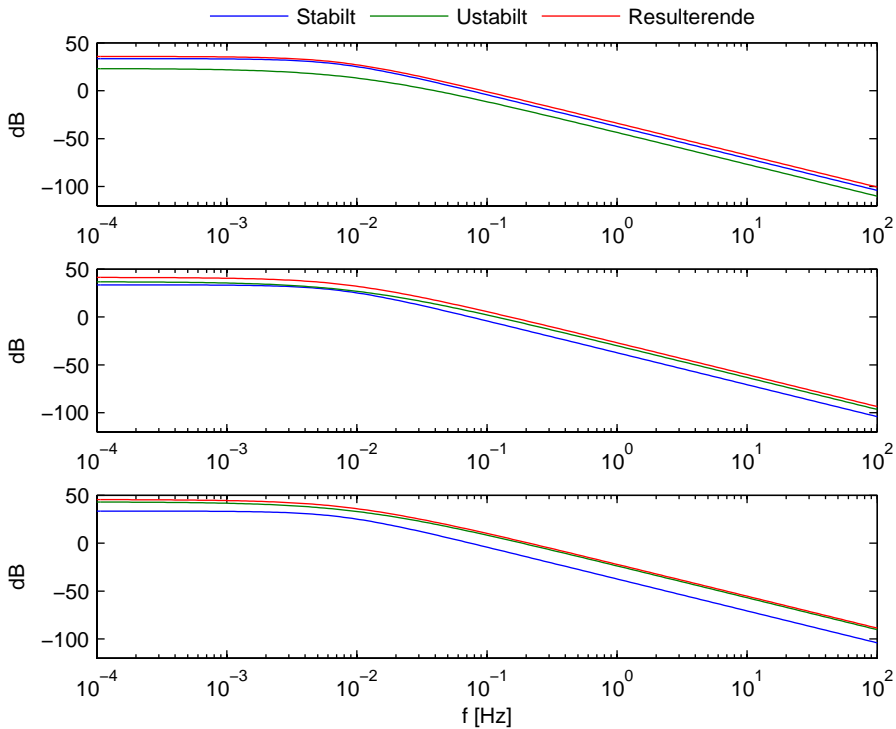
Hvor $H=80$ m. z_0 er ruhedlængden, som afhænger af terrænet omkring vindmøllen og er vist i tabel 2.1 taget fra [Manwell 02], hvor der samtidig er vist turbulensintensiteten, som kan beregnes med ligning 2.7.

$$\tau_f = \frac{1}{\log\left(\frac{H}{z_0}\right)} \quad (2.7)$$

Terrænbeskrivelse	$z_0(mm)$	$\tau_f(\%)$
Meget glat, is eller mudder	0.01	6.29
Oprørt hav	0.5	8.35
Afgrøder	50	13.55
Skov	500	19.7

Tabel 2.1: Ruhedsfaktor og turbulensintensitet ved forskelligt terræn

Det er nu muligt at beregne en værdi for Monin-Obukhov længden, hvis man lægger sig fast på en ruhedslængde. På figur 2.2 ses et plot over bidraget fra den stabile og ustabile del af Højstrup funktionen samt de resulterende spektra. Som det ses, indeholder figuren tre plot, hvor der er valgt forskellige ruhedslængder. Dette er gjort for at illustrere, hvilke dele af funktionen der er dominerende alt efter, hvor stabile forholdene er.



Figur 2.2: Stabil, ustabil og resulterende plot af Højstrup funktion. Plot1: $z_0 = 0.01mm$, Plot2: $z_0 = 50mm$, Plot3: $z_0 = 500mm$, $v_m = 12m/s$

Af figur 2.2 ses det, at den stabile del af ligning 2.1 vil være dominerende, hvis

der er en lille ruhedslængde. Dette skyldes, at der ved en lille ruhedslængde vil være meget stabile forhold i vinden. Bliver ruhedslængden større, ses det nu, at den ustabile del af funktionen er klart dominerende, dette er specielt gældende ved høje frekvenser. Plot nummer 2 er lavet ud fra de forhold, som anvendes i simuleringerne, og her ses tydeligt, at den ustabile del er den dominerende ved frekvenser over $0.1Hz$. Set ud fra et reguleringsmæssigt synspunkt vil det være frekvenser over $0.1 Hz$ der er mest interessante, da alt under denne frekvens kan anses som værende middelvindens variation. Det anses derfor som en rimelig simplificering at sætte $z_i / -L = 0$. Derfor vil ligning 2.3 og 2.4 blive anvendt til den videre analyse.

Da variansen ikke er en faktor, der umiddelbart er kendt for vinden, elimineres denne fra ligning 2.3. Dette gøres ved at udtrykke variansen ved turbulensintensiteten, som beskriver, hvor ustabile forholdene er i det pågældende område og ses af tabel 2.1. Variansen kan således udtrykkes ved ligning 2.8.

$$\sigma^2 = (\tau_f \cdot v_m)^2 \quad (2.8)$$

Indsættes dette udtryk for variansen i ligning 2.4, opnås ligning 2.9.

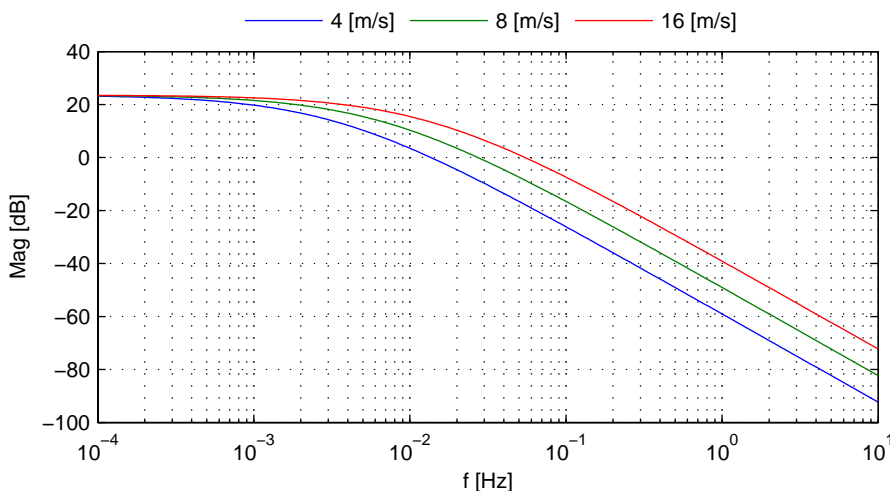
$$\begin{aligned} \frac{(\tau_f \cdot v_m)^2}{v_{fr}^2} &= \frac{105}{22} \frac{\left(1 - \frac{H}{z_i}\right)^2}{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}} \Leftrightarrow \\ v_{fr}^2 &= (\tau_f \cdot v_m)^2 \frac{22}{105} \frac{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}}{\left(1 - \frac{H}{z_i}\right)^2} \end{aligned} \quad (2.9)$$

Som det ses, giver dette samtidig et nyt udtryk for friktionshastigheden, som vil være gældende for ustabile forhold i vinden. Det endelige udtryk for punktvindspektret kan nu opnås ved at indsætte ligning 2.9 i ligning 2.3, samt udtrykket for f_{ru} . Resultatet heraf ses i ligning 2.10.

$$\begin{aligned} \frac{fS(f)}{(\tau_f \cdot v_m)^2 \frac{22}{105} \frac{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}}{\left(1 - \frac{H}{z_i}\right)^2}} &= \frac{105f \frac{H}{v_m} \frac{1}{1 + 15\frac{H}{z_i}} \frac{\left(1 - \frac{H}{z_i}\right)^2}{\left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}}}{\left(1 + 33f \frac{H}{v_m} \frac{1}{1 + 15\frac{H}{z_i}}\right)^{\frac{5}{3}} \left(1 + 15\frac{H}{z_i}\right)^{\frac{2}{3}}} \\ S(f) &= \frac{22H \frac{1}{1 + 15\frac{H}{z_i}} \tau_f^2 v_m}{\left(1 + 33f \frac{H}{v_m} \frac{1}{1 + 15\frac{H}{z_i}}\right)^{\frac{5}{3}}} \end{aligned} \quad (2.10)$$

Det opnåede udtryk for punktvindspektret kan nu bestemmes ud fra overfladeruhedslængden, middelvinden, tårnhøjden samt højden på det atmosfæriske

grænselag, hvor det normalt kun vil være middelvinden, der er en varierende størrelse, når den pågældende placering for vindmøllen er fundet. På figur 2.3 ses et plot af effekt tætheds-spektrret normaliseret med middelvinden for middelvindshastigheder på hhv. 4- 8 og 16 m/s . Turbulensintensiteten er sat til 13.55 %, hvilket svarer til en ruhedslængde på 50 mm , og svarer til en overflade bestående af afgrøder. Dette anses som værende det mest bevoksede sted, man normalt vil placere en vindmølle.



Figur 2.3: Effekttæthedspektrret normaliseret med middelvinden for middelvind på 4- 8- og 16 m/s ved en turbulensintensitet på 13.55%

Det ses af figur 2.3, at vinden vil indeholde mere energi ved de høje frekvenser ved en højere middelvind. Dette vil sige, at desto højere middelvind desto mere ustabilitet vil der være i vinden.

2.1.2 Ækvivalent vind ved rotor

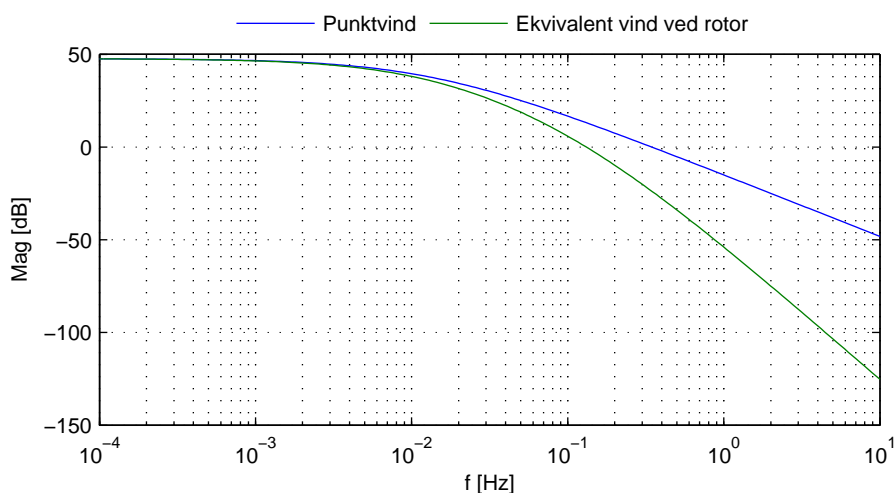
Punkt vinden er nu blevet beskrevet. For at kunne anvende vindmodellen i vindmøllemodellen er det nødvendigt at lave en beskrivelse af variationen i vinden set over hele rotorarealet. Dette kan ifølge [Knudsen 83] opnås ved at multiplicere effekt tætheds-spektrret for punkt vinden med filtret, vist i ligning 2.11.

$$F(f) = \frac{1}{\left(1 + \frac{8\sqrt{\pi}}{3} \frac{R}{v_m} f\right) \left(1 + 4\sqrt{\pi} \frac{R}{v_m} f\right)} \quad (2.11)$$

Effekttæthedsspektret for den ækvivalente vind, set over hele rotoren, kan nu bestemmes og er givet ved ligning 2.12.

$$S_{eff}(f) = \frac{22H \frac{1}{1+15\frac{H}{z_i}} \tau_f^2 v_m}{\left(1 + 33f \frac{H}{v_m} \frac{1}{1+15\frac{H}{z_i}}\right)^{\frac{5}{3}}} \cdot \frac{1}{\left(1 + \frac{8\sqrt{\pi}}{3} \frac{R}{v_m} f\right) \left(1 + 4\sqrt{\pi} \frac{R}{v_m} f\right)} \quad (2.12)$$

På figur 2.4 ses et plot for effekt tætheds-spektrret for punkt vinden samt ækvivalent vinden ved rotoren.



Figur 2.4: Effekttæthedsspektret for punkt vind samt ækvivalent vind ved rotor ved en middelvind på 16 m/s

Det ses, at den ækvivalente vind er dæmpet langt mere ved de høje frekvenser. Dette er ikke så overraskende, da det rent intuitivt giver mening, at vinden varierer langsommere set over et stort areal end på et enkelt punkt.

2.1.3 Implementering af vindmodel

For at kunne implementere vindmodellen i simulationsmodellen ønskes det at tilpasse et filter, som drevet af hvid støj på bedste vis gengiver spektret, fundet for den ækvivalente vind. Det er her valgt at anvende et 2.ordens filter som i [Østergaard 94], vist i ligning 2.13. Det skal hertil nævnes, at det antages, at den stokastiske del af vinden antages at være wide sense stationary.

$$H(s) = \frac{k}{(1 + p_1 s)(1 + p_2 s)} \quad (2.13)$$

Da det er effekt tæthed-funktionen, som det ønskes at tilpasse filtret til, findes denne for ligning 2.13 i ligning 2.14.

$$\begin{aligned}\Phi_H(\omega) &= H(j\omega)H(-j\omega) \\ \Phi_H(\omega) &= \frac{k^2}{(1+p_1^2\omega^2)(1+p_2^2\omega^2)} \\ \Phi_H(2\pi f) &= \frac{k^2}{(1+p_1^2(2\pi f)^2)(1+p_2^2(2\pi f)^2)}\end{aligned}\quad (2.14)$$

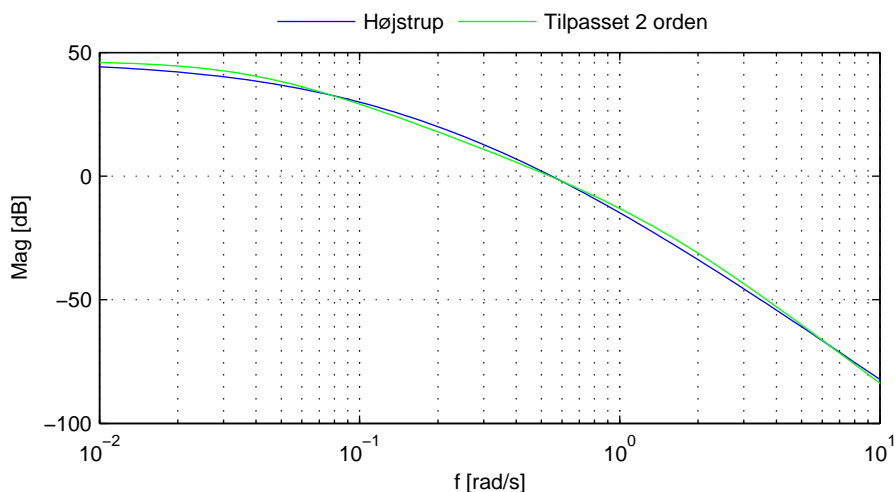
For at sikre filtret for den korrekte DC gain kan denne fastlægges inden filtret tilpasses til effekt tætheds-spektrret. Dette gøres ved at sætte k^2 lig DC gainen i spektret, som vist i ligning 2.15.

$$k^2 = 22H \frac{1}{1 + 15 \frac{H}{z_i} \tau_f^2 v_m} \quad (2.15)$$

Polerne i filtret skal nu fastlægges, så det resulterende spektrum i et valgt frekvensområde bedst muligt er tilpasset spektret fra Højstrup funktionen. Dette gøres ved numerisk minimering af fejlindekset, vist i ligning 2.16.

$$J = \int_{\omega_1}^{\omega_2} \left[\log \left(\frac{\Phi_H(\omega)}{k^2} \right) - \log \left(\frac{\Phi_{eff}(\omega)}{k^2} \right) \right]^2 d\omega \quad (2.16)$$

Her vælges ω_1 og ω_2 i det område, man ønsker at tilpasse filtret. I dette tilfælde vil det være praktisk at tilpasse vinden bedst i det frekvensområde, der kan reguleres i. Derfor vælges den øvre frekvens til den dobbelte båndbredde af generator modellen, da det er modmomentet T_g , som er det hurtigste styresignal i systemet, altså på 20 rad/s . Den nedre grænse lægges på 0.1 rad/s . Kernen i den numeriske tilpasning bliver foretaget af matlab funktionen `fminsearch` og selve implementeringen er lavet i matlab filerne `wind_model` og `spec_lin`, som ses i bilag E. Resultatet ses på figur 2.5.



Figur 2.5: 2.ordens filtertilpasning til effekt tæthedsspektrum ved 14 m/s

Det ses, at det tilpassede filter kan følge det sande spektrum rimeligt godt i det ønskede område. Når polerne i 2.ordens systemet er fundet, kan vindmodellen opstilles som en state spacemodel, da det nu er et lineært system. For at tydeliggøre, hvorledes modellen opnås, omskrives ligning 2.13, som vist i ligning 2.17.

$$\begin{aligned}
 H(s) &= \frac{k}{(1 + p_1 s)(1 + p_2 s)} \\
 H(s) &= \frac{k}{1 + (p_1 + p_2)s + p_1 p_2 s^2} \\
 H(s) &= \frac{\frac{k}{p_1 p_2}}{s^2 + \frac{p_1 + p_2}{p_1 p_2} s + \frac{1}{p_1 p_2}} \quad (2.17)
 \end{aligned}$$

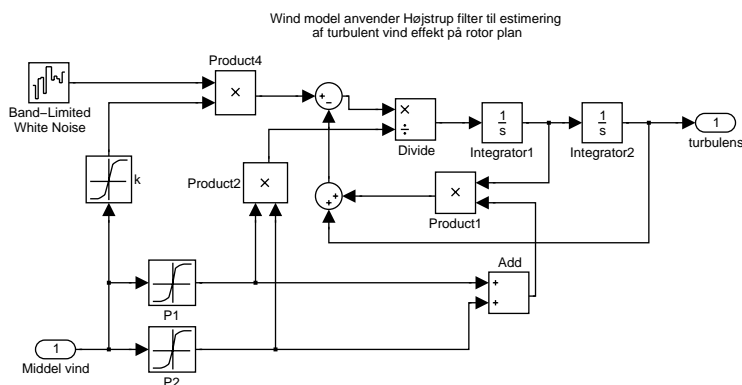
Ud fra ligning 2.17 kan state spacemodellen nu opstilles og er vist i ligning 2.18.

$$\begin{aligned}
 \mathbf{A}_v &= \begin{bmatrix} 0 & 1 \\ -\frac{1}{p_1 p_2} & -\frac{p_1 + p_2}{p_1 p_2} \end{bmatrix} \\
 \mathbf{B}_v &= \begin{bmatrix} 0 & 1 \\ -\frac{1}{p_1 p_2} & -\frac{p_1 + p_2}{p_1 p_2} \end{bmatrix} \quad \mathbf{C}_v = \begin{bmatrix} \frac{k}{p_1 p_2} & 0 \end{bmatrix} \quad (2.18)
 \end{aligned}$$

Hvor:

$$\bar{x} = \begin{bmatrix} \delta v \\ \dot{\delta v} \end{bmatrix} \quad \bar{u} = [\sigma_{vnoise}] \quad \bar{y} = [v_{turb}]$$

Som det er nævnt tidligere, er effekt tæthedsspektret afhængig af middelvinden. Det er derfor nødvendigt at lave en gain scheduling af de to poler samt DC gainen i filtret for at kompensere for ulineariteten i vinden. Denne gain scheduling er lavet ved at tilpasse filtret til vinden ved forskellige middelvindhastigheder. I simulink implementeringen af filtret, set i figur 2.6, er der lavet en interpolering mellem de beregnede poler samt DC gainen, så disse løbende vil blive ændret i forhold til den relevante middelvind.

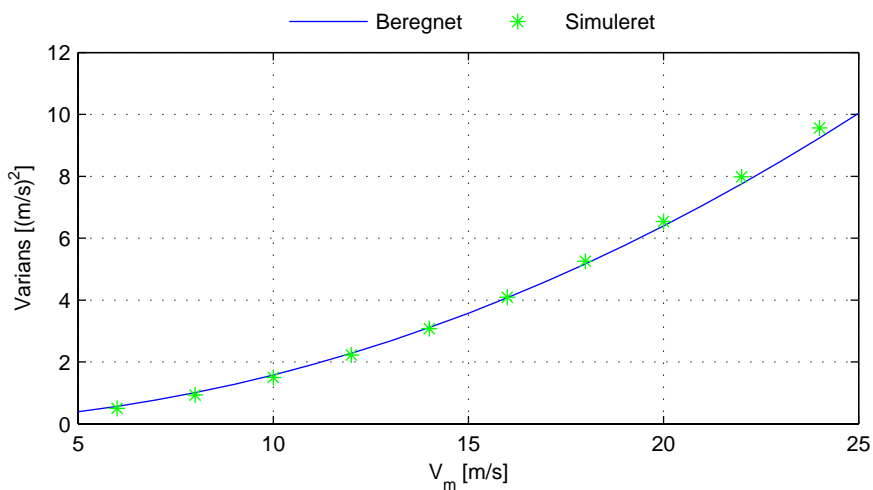


Figur 2.6: Simulinkmodel over implementering af stokastisk vindmodel med dynamisk DC gain samt poler.

Variansen af 2.ordens filtret bestemmes, så det bliver muligt at sammenligne den teoretiske varians med variansen målt på udgangen af filtret implementeret i simulink modellen. Denne sammenligning vil sikre, at modellen er korrekt implementeret, og at gain scheduling af koefficienterne virker. Variansen for et 2.ordens filter kan findes ved ligning 2.19.

$$\begin{aligned}\sigma_H^2 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_H(\omega) d\omega \\ \sigma_H^2 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{k^2}{(1 + p_1^2 \omega^2)(1 + p_2^2 \omega^2)} d\omega \\ \sigma_H^2 &= \frac{k^2}{2} \left[\left(1 + \frac{p_2^2}{p_1^2 - p_2^2}\right) \frac{1}{p_1} - \frac{p_2}{p_1^2 - p_2^2} \right] \quad (2.19)\end{aligned}$$

For at finde variansen i simulink modellen er der kørt en simulering over 5000 sek., og variansen er derefter fundet ud fra den genererede tidsrække med matlab funktionen var.m. På figur 2.7 ses et plot over den teoretisk beregnede varians som en funktion af middelvinden, sammen med variansen, fundet ud fra den simulerede tidsrække, plottet for nogle udvalgte middelvinde.



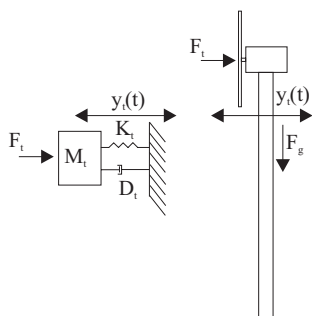
Figur 2.7: Verifikation af korrekt varians for implementering af vindmodel.

Det ses, at der er god overensstemmelse mellem variansen fra simulink implementeringen og den beregnede varians. Der er ikke fuldstændig overensstemmelse mellem de to, da dette ville kræve, at der blev lavet en simulering med uendelig høj samplefrekvens over uendelig lang tid, hvilket selvfølgelig ikke er praktisk muligt.

Der er nu lavet en model, som kan give en forholdsvis realistisk simulering af, hvorledes regulatorerne vil præstere i et stokastisk vindfelt. Det skal dog påpeges, at denne vindmodel kun kan anvendes på en vindmøllemodel, der anvender et enkelt input for vinden, og gælder kun for ustabile forhold i vinden.

2.2 Tårnmodel

For at have muligheden for at simulere hvor stort et udsving, tårnet vil være udsat for, og hvorledes dette har indvirkning på reguleringen, er der lavet en simpel model af tårnet. Det er valgt at modellere tårnet som en masse, der sidder i enden af en fjeder parallelt med en dæmpning, som vist på figur 2.8.



Figur 2.8: Blokdiagram over modelleret tårnbevægelse.

Som det ses, er det en noget simplificeret måde at modellere tårnsvingningen på. Tårnets udbøjning er kun modelleret i en retning, som er i vindretningen. Det antages, at hele udsvingningen ligger som en forskydning i vindretningen, altså bliver der ikke medregnet, at tårnet bøjer nedad.

Dette kan dog forsvares med, at tårnets udbøjning er relativ lille i forhold til tårnhøjden. Det samme gælder for, at der ikke medregnes effekten af gravitationen, da denne komponent primært vil virke som et tryk på en vertikal søjle, som ses på figur 2.8.

Ved denne simplificering kan tårnet altså modelleres som et 2.ordens system ud fra differentialligningen, vist i ligning 2.20.

$$F_t = M_t \ddot{y}_t(t) + D_t \dot{y}_t(t) + K_t y_t(t) \quad (2.20)$$

Selve tårnmodellen er et lineært system, hvor det dog skal huskes, at kraftpåvirkningen ikke er lineær.

For at kunne opstille state spacemodellen, Laplace transformeres ligning 2.20 og omroteres, som vist i ligning 2.21.

$$\begin{aligned} F_t &= M_t y_t s^2 + D_t y_t s + K_t y_t \\ \frac{y_t}{F_t} &= \frac{\frac{1}{M_t}}{s^2 + \frac{D_t}{M_t} s + \frac{K_t}{M_t}} \end{aligned} \quad (2.21)$$

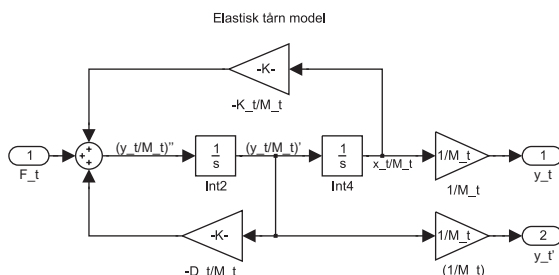
State space modellen kan nu opstilles ud fra ligning 2.21 og ses i ligning 2.22.

$$\begin{aligned} \mathbf{A}_t &= \begin{bmatrix} 0 & 1 \\ -\frac{K_t}{M_t} & -\frac{D_t}{M_t} \end{bmatrix} \\ \mathbf{B}_t &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \mathbf{C}_t &= \begin{bmatrix} \frac{1}{M_t} & 0 \end{bmatrix} \end{aligned} \quad (2.22)$$

Hvor:

$$\bar{x} = \begin{bmatrix} \frac{y_t}{M_t} \\ \dot{y}_t / M_t \end{bmatrix} \quad \bar{u} = [F_t] \quad \bar{y} = [y_t]$$

Implementeringen i simulink af tårnmodellen kan ses i figur 2.9.



Figur 2.9: Simulinkdiagram over implementering af tårnmodel

Kraften, som påvirker tårnet, kan findes ud fra udtrykket, vist i ligning 2.23

$$F_t = \frac{1}{2} \pi \rho R^2 v_m^2 c_t(\lambda, \theta) \quad (2.23)$$

c_t er en effektivitetsfaktor der ligesom c_p for rotationsmomentet er en ulineær funktion, som afhænger af tip speed ratio og pitchvinklen.

For at finde kraftpåvirkningen til den lineariserede model er det nødvendigt at foretage en linearisering for kraftpåvirkningen. Lineariseringen laves ved at differentiere ligning 2.23 med hensyn til vinden. Dette giver ligning 2.24.

$$\begin{aligned} \frac{dF_t}{dv} &= \frac{d\left(\frac{1}{2} \pi \rho R^2 v_m^2 c_t(\lambda, \theta)\right)}{dv} \\ \frac{dF_t}{dv} &= \frac{\pi \rho R^2 v_m \frac{c_t(\lambda, \theta)}{dv}}{dv} \\ \frac{dF_t}{dv} &= \frac{\pi \rho R^2 v_m \frac{c_t(\lambda, \theta)}{d\lambda} \cdot \frac{1}{\omega_r R}}{dv} \end{aligned} \quad (2.24)$$

Hældningen på c_t kurven findes ligesom for c_p kurven, altså på numerisk vis. For at gå fra et differentiale med hensyn til vinden til et differentiale med hensyn til λ er differentialreglen fra ligning 1.11 anvendt.

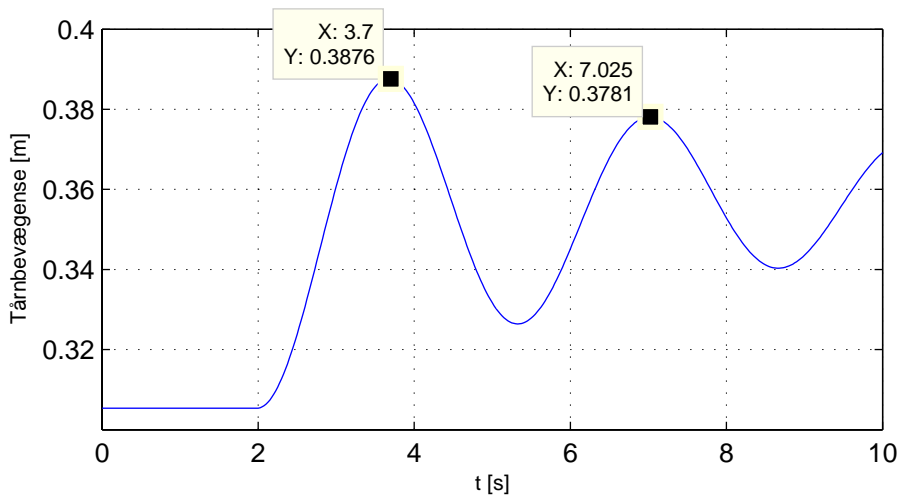
Massen af tårnet er fundet ud fra data opgivet i HAWC. Massen er antaget til at være en sum af vægten på vingerne, nacellen samt halvdelen af tårnet. Ved denne antagelse er vægten fundet til ca. 200 T.

Fjederkonstanten er som for drivakslen fundet ud fra fixed free resonansfrekvensen på tårnet f_t . Resonansfrekvensen, der er anvendt, er fundet ud fra [Risø 05] til 0.3 Hz. Fjederkonstanten kan beregnes ud fra udtrykket, der ses i ligning 2.25.

$$K_t = M_t \cdot (2\pi f_t)^2 \quad (2.25)$$

Dette giver en fjederkonstant på $K_t = 6.9484 \cdot 10^5 \text{ N/m}$. For at finde dæmpningen i tårnet er tommelfingerreglen, med at den skal ligge mellem en 10 og

100 del af fjederkonstanten, igen anvendt. Ved sammenligning af svingningen fra HAWC blev det bestemt, at en faktor $1/30$ var passende. Dette giver en dæmpningsfaktor på $D_t = 2.3161 \cdot 10^4 s^{-1}$. Ud fra de beregnede konstanter ønskes det nu at verificere, at simulink modellen, vist på figur 2.9, giver den ønskede resonansfrekvens på $0.3 Hz$. Dette gøres ved at påføre open loop vindmøllemodellen - med tårnmodellen indsat - ét trin i vinden, og derefter se på tårnsvingningen. Resultatet heraf ses i figur 2.10.



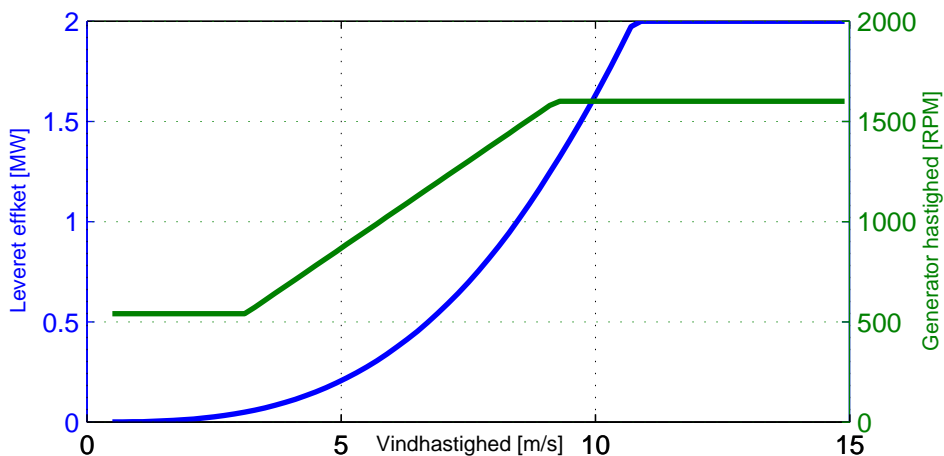
Figur 2.10: Verifikation af resonansfrekvens af simulink implementeret tårnmodel. Trin på vinden til tiden 2 sek. ved en vindhastighed på $12 m/s$. trin er på $1 m/s$.

Dette resulterer i en resonansfrekvens på $\frac{1}{7.025-3.7} = 0.3008 Hz$, hvilket stemmer overens med de forventede $0.3 Hz$, da der er en lille unøjagtighed i aflæsningen af periodetiden.

KAPITEL 3

Kollektiv regulator strategi

Den primære opgave, som regulatoren skal opfylde, er at hive så meget effekt ud af vinden som muligt. Derudover er det at sørge for, at dette sker på en måde, som varetager møllens mekaniske begrænsninger og giver mindst mulig slitage. En meget vigtig ting at huske i denne sammenhæng er, at dette sker uden at måle den faktiske vind. Havde det været muligt at måle vinden tilstrækkelig nøjagtig, ville det have givet mange andre muligheder for opbygning af regulatorer. Disse muligheder vil ikke blive betragtet nærmere. En metode, der er meget anvendt til netop dette, er $p\omega$ metoden. Denne danner grundlag for den regulering, der benyttes ved de fleste vindhastigheder, møllen bliver udsat for. Der er dog vindhastigheder, hvor metoden ikke kan anvendes. Den øvre effekt grænse for møllen er 2 MW. og den maksimale rotorhastighed er 18.8 rpm., hvilket svarer til en generator hastighed på 1600 rpm. Sammenhængen mellem vindhastigheden, den leverede effekt og generatorens omdrejningshastighed ses på figur 3.1. Ved kraftige vindforhold skal en anden metode, som sørger for at disse begrænsninger ikke overskrides, tages i brug. Derudover skal der tages hensyn til generatorens arbejdsområde, der starter ved ca. 540 rpm. For at efterkomme alle kravene kan reguleringen deles i fire forskellige områder alt efter vindforholdene; low, mid ($p\omega$), high og top. Som nævnt danner $p\omega$ grundlag for hele måden, hvorpå møllen reguleres. Derfor vil det være på sin plads at starte med at forklare dette først.



Figur 3.1: Leveret effekt og generator omdrejninger afhængig af vinden.

3.1 Midområdet

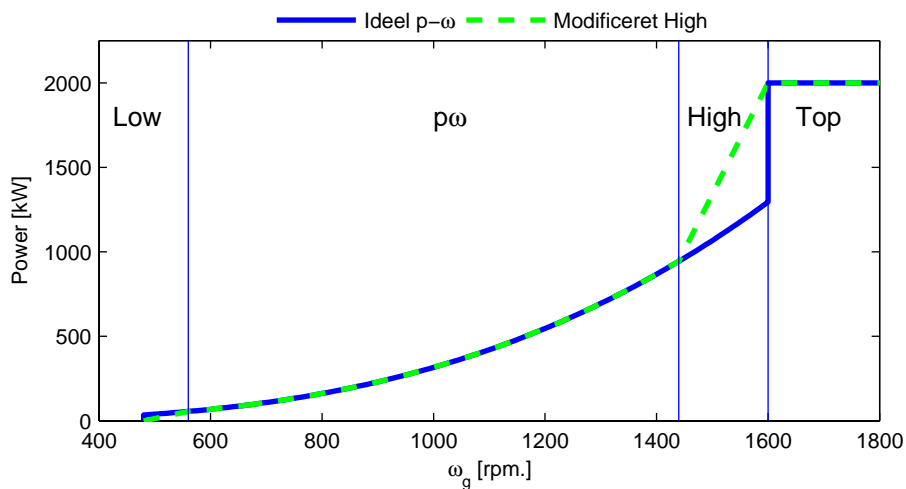
Den optimale regulering med hensyn til effektiviteten er at holde pitch-vinklen θ , og tipshastighedsforholdet λ konstant, så de hele tiden ligger på toppen af c_p kurven. c_p kurven, som er omtalt tidligere i kapitel 1, udtrykker møllens aerodynamiske effektivitet som en funktion af λ og θ . c_p kurven er afbillet på figur 1.4. Området kaldes også $p\omega$ området.

Den optimale regulering opnås ved at låse pitch-vinklen til den optimale, θ_{nom} . For at opretholde den optimale λ måles generatorens omdrejningshastighed ω_g . Når θ er låst vil ω_g kunne omregnes til den effekt, som møllen under optimale forhold kan levere. Denne referenceeffekt bruges til at sætte generatorens modmoment T_g . Hvis vinden falder vil generator hastigheden falde, referenceeffekten vil falde og modmomentet vil blive reguleret ned. Metoden vil altså søge mod den optimale λ og dermed også optimalt effekt udbytte. Og dette sker uden måling af den aktuelle vind. Generelt kan den leverede effekt bestemmes ved udtrykket i ligning 3.1.

$$P_r = \frac{1}{2} \rho R^2 v^3 c_p(\lambda, \theta) \quad (3.1)$$

Da θ og λ er optimale, betyder det, at vindhastigheden kan antages som $V = \theta_{nom} \lambda_{nom} R$, og P_{ref} findes af ligning 3.2. Generatorens modmoment beregnes ud fra referenceeffekten og generator hastigheden, se ligning 3.3

$$P_{ref} = \frac{1}{2} \rho R^2 \theta_{nom} \lambda_{nom} R c_{pnom} \quad (3.2)$$

Figur 3.2: $p\omega$ kurver

$$T_g = \frac{P}{\omega_g} \quad (3.3)$$

I regulatoren implementeres dette som et tabelopslag. Opslaget sker ud fra ω_g og bestemmer P . Når tabellen ikke er lavet, så den leverer modmomentet direkte, er det, fordi den på denne måde kan indgå som en del af reguleringen senere hen. Efter tabelopslaget bestemmes T_g altså, som det er vist i ligning 3.3. På figur 3.2 er $p\omega$ kurven vist. Området strækker sig, som det ses, fra 3-9.5 m/s .

3.1.1 Alternativ metode

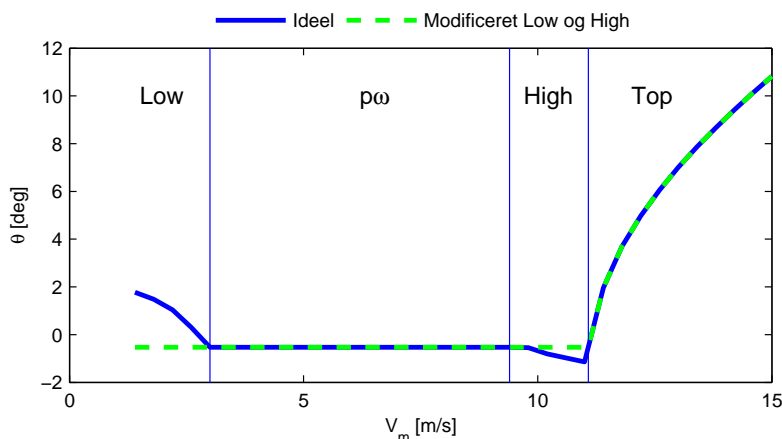
Mulighederne ved at vende $p\omega$ regulatoren på hovedet til en ωp og måle output effekten og derudfra fastsætte møllens omdrejninger blev overvejet. Fordelen ved dette ville være, at der ud fra effekten kunne fastsættes ω_g referencen. I lowområdet (under en fastsat effekt) skulle rotorhastigheden holdes konstant. I medium området følger en ωp kurve og så fremdeles. Dette blev undersøgt, og den simple strategi fungerede efter hensigten.

Problemet med denne metode er, at der ikke er en referenceeffekt, som kan følges. Dette bevirker, at den leverede effekt svinger meget. Al variation i vinden bliver overført til output effekten. Laves der en hurtig regulering på omdrejningerne, vil den faktisk bruge styresignalet T_g til at holde ω_g . Når vinden varierer, betyder det, at T_g er forstærket i forhold til vindvariationen, og variationen i effekten bliver meget stor. I den almindelige $p\omega$ metode optages denne variation

i rotorens bevægelse. Derved bruges rotorens store inertie og T_g til at dæmpe disse vindvariationer. Metoden er altså udmærket, hvis det ønskes at holde rotorens omdrejninger konstant og optage vindens variation i den leverede effekt og reguleringen ikke bliver så kraftig, at T_g varierer for meget.

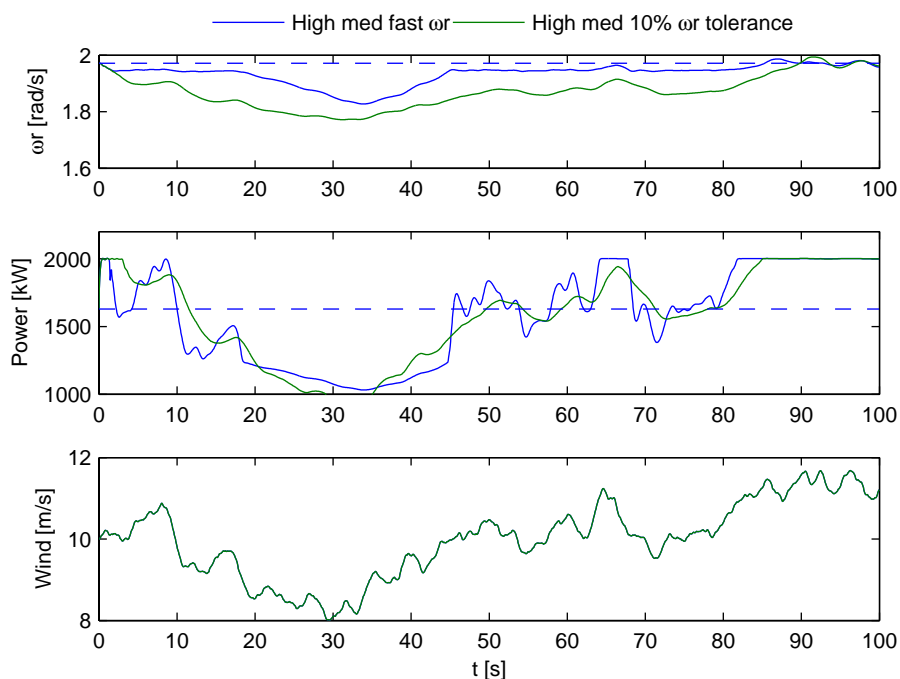
3.2 Low- og highområdet

Low (1.2-3 m/s) og high (9.5-11 m/s) områderne er meget lig hinanden. I begge områder skal effekten optimeres, mens rotorhastigheden fastholdes. Når rotorhastigheden holdes konstant og vinden varierer, vil det betyde, at λ ikke længere er optimal. Regulatorens opgave er derfor at finde den bedste kombination af θ og T_g . Af figur 3.3 ses den optimale θ værdi for varierende vindhastigheder. Værdierne er fundet ved at beregne de stationære værdier som beskrevet i kapitel 1 for en lang række vindhastigheder. Som det fremgår af figur 3.3, er det først i topområdet at pitch-vinklen for alvor bruges. I low- og highområdet er den derfor låst som i $p\omega$ området. Dette er illustreret i figuren med den grønne stiplede linje. Bekostningen er en minimal nedgang i output effekten i disse områder. Dette er hovedsageligt gjort, fordi det viste sig at være svært at lave et pænt skift mellem high og top, fordi der i highområdet pitches negativt og i topområdet positivt for stigende vind. Desuden er highområdet et meget lille område, kun fra ca. 9.5-11 m/s. Bekostningen i output effekten vil derfor i det lange løb være mindre.

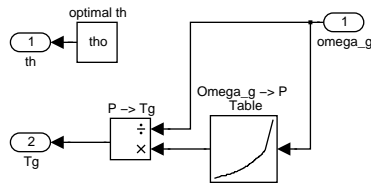


Figur 3.3: θ 's afhængighed af vinden.

Skiftet mellem $p\omega$ og highområdet sker, når generator hastigheden når 1600 rpm. Her skal omdrejningsbegrænsningen sætte ind. I hele området op til top vil omdrejningshastigheden altså skulle holdes konstant. Dette giver samme problem som nævnt for den omvendte $p\omega$ metode, nemlig at al variation i vinden slår direkte igennem på effekten. For at undgå dette er $p\omega$ kurven ændret. Ændringen består i at starte highområdet tidligere og derved have en vis variation i omdrejningerne. På figur 3.2 er grænsen flyttet til 90% af $\omega_{gnom} \sim 1440$ rpm., illustreret med den grønne stiplede linje. Problemet og resultatet kan ses på figur 3.4, hvor der er sammenlignet to forskellige grænser for skiftet mellem $p\omega$ og high. Det ses, at den blå linje med (fast ω_r) har en bedre evne til at fastholde omdrejningerne, men også tydeligt hvilken negativ indflydelse, det har på output effekten. Når grænsen flyttes til 90% er det et kompromis, og resultatet er, at den leverede effekt er langt roligere, mens rotorhastigheden varierer lidt mere. Til figuren skal det yderligere nævnes, at vinden i området mellem ca. 20-45 s. er så lav, at den falder ned i $p\omega$ området, mens der fra 80 s. er så meget vind, at output effekten bliver begrænset.



Figur 3.4: Forsøg på at holde ω_r konstant resulterer i større variation i effekten.



Figur 3.5: Blokdiagram af $p\omega$ implementering.

For de tre nævnte områder er der en fælles regulator, nemlig $p\omega$. Tilbagekoblingen sker fra den målte generator hastighed, og ved tabelopslag fastsættes det ønskede styresignal, modmomentet T_g . Implementeringen kan ses på figur 3.5. Resultatet er en relativ langsom regulering, som skyldes den store inert i rotoren. Styresignalet T_g kan nemlig først ændres, når rotorens og dermed generatorens omdrejninger ændrer sig. Egentlige test af dette ses senere i kapitel 5.

3.3 Topområdet

Topområdet bruges, når vinden er over ca. 11 m/s. Her er der så kraftig vind, at både omdrejninger og effekt skal begrænses. Topområdet adskiller sig markant fra de andre ved ikke længere at søge mod den største effektivitet. Ønsket er at holde effekt og omdrejninger konstant på møllens nominelle værdier, og effektiviteten mindskes ved at pitche bladene ud af vinden så deres effektivitet falder. Topområdet er det område, hvor reguleringen har flest frihedsgrader at arbejde på. Både θ og T_g kan varieres, så omdrejninger og effekt holdes bedst muligt. Ved hjælp af T_g kan generatorens effekt holdes meget nøjagtig, da modmomenttaktuatoren er meget hurtig. Vinklen på bladene bruges til de langsommere variationer i vindhastigheden. I dette område kan der vindes meget ved et godt regulator design, og der vil blive designet forskellige typer til sammenligning i kapitel 5.

3.4 Start og stop

Udover de nævnte områder skal der også tages højde for, hvornår og hvordan møllen starter op, når der har være vindstille. Det indebærer sandsynligvis en måling af den aktuelle vind, hvilket ikke er inkluderet i modellen og derfor ikke betragtet nærmere. Det er heller ikke overvejet nærmere, hvordan møllen stoppes, når vinden bliver så kraftig, at det ikke længere er sikkert blot at pitche ud. Et sådant stop af møllen vil muligvis kræve, at møllen drejes ud af vindretningen og måske bremses mekanisk. Dette er heller ikke muligt at simulere med vores model og bliver derfor ikke behandlet.

KAPITEL 4

Kollektiv regulator design

Ud fra designmodellen kan der nu designes forskellige reguleringer til vindmøllen. For at give en kort opsummering af, hvilke muligheder der er for at regulere vindmøllen, ses der først på, hvilke input, states og output, der kan reguleres på. Disse er vist i tabel 4.1.

Input/Styresignaler	States	Output
v	$\hat{\theta}$	P
θ_{ref}	$\hat{\theta}$	
T_{gref}	\hat{T}_g	
	ω_r	
	ω_g	
	$\Delta\phi$	

Tabel 4.1: Tabel over input/styresignaler, states og output

Dette er altså, hvad der ud fra designmodellen er til rådighed til at designe reguleringen. Regulatorerne i dette afsnit, behandler kun topområdet, hvor designkriterierne for regulatorerne primært er, at de skal kunne holde effekten og omdrejningerne fra møllen på et konstant niveau. Yderligere ønskes de strukturelle belastninger reduceret, så vindmøllens levetid forlænges. Det skal dog huskes, at pitch-vinklen maksimalt kan ændre sig med en hastighed på 10 *grader/s*. Dette tages der højde for under designet af regulatorerne ved at sørge for, at

pithhastigheden ikke overstiger det tilladte med et trin i vinden på 2 m/s . Forekommer der større trin end dette vil pitch-aktuatoren gå i begrænsning, og reguleringen vil ikke længere kunne yde, hvad den er designet til.

Der er blevet designet 4 reguleringer til kollektiv pitch-regulering af vindmøllen; en PI regulator og 3 LQI regulatorer. Disse vil i følgende afsnit blive gennemgået hver for sig, og resultatet af regulatorerne vil blive præsenteret. Der vil i dette afsnit kun være en kort præsentation af reguleringsresultaterne, da disse yderligere vil blive dokumenteret i kapitel 5. Alle LQI regulatorerne er designet ud fra samme princip men antallet af states, der tages med i designet, ændres.

4.1 PI regulator design

Den første regulering, der designes til vindmøllen, er en velkendt og forholdsvis simpel PI regulering. Fordelen ved PI reguleringen er, at den er forholdsvis hurtig at designe, når først man har bestemt en overføringsfunktion for reguleringsobjektet. Ulemperne ved PI regulering er bl.a., at det fungerer bedst til regulering af SISO systemer, da fremgangsmetoden i designfasen kun er anvendelig for sådanne systemer. Dette leder til et andet problem, nemlig at designmetoden, som foregår ud fra bodeplot og trinresponse af systemet, ikke er mulig at lave online som for nyere regulator design. Da PI regulering bliver designet ud fra en overføringsfunktion, altså fra input til output, er det heller ikke muligt at udnytte den information, man eventuelt måtte have om, hvorledes systemet opfører sig internt.

PI reguleringen vil derfor blive brugt som en reference for ydelsen af den kollektive pitch-regulering. De mere komplicerede reguleringer, der efterfølgende bliver designet, forventes at give et bedre resultat end PI reguleringen.

Da designmodellen er ulineær med hensyn til vinden, er det nødvendigt at fastlægge et lineariseringspunkt, før der kan findes en overføringsfunktion. Lineariseringspunktet er sat til 14 m/s . Hvis PI reguleringen skal virke ordentligt i hele topområdet, vil det være nødvendigt at lave en gain scheduling af regulator koefficienterne efter den målte eller estimerede vindhastighed. Dette er dog ikke blevet implementeret i projektet, hvilket medfører, at PI reguleringen vil blive dårligere, når arbejdsområdet flyttes fra lineariseringspunktet.

Når der er bestemt et lineariseringspunkt, er første trin i PI regulator designet således at bestemme open loop overføringsfunktionen for systemet. PI regulering skal ud fra de målte omdrejninger på generatoren regulere omdrejningshastigheden ved at ændre pitch-vinklen. Overføringsfunktionen kan derfor findes ud fra ligning 1.45 fra input $2 \theta_{ref}$ til output $2 \omega_g$. og ses på ligning 4.1

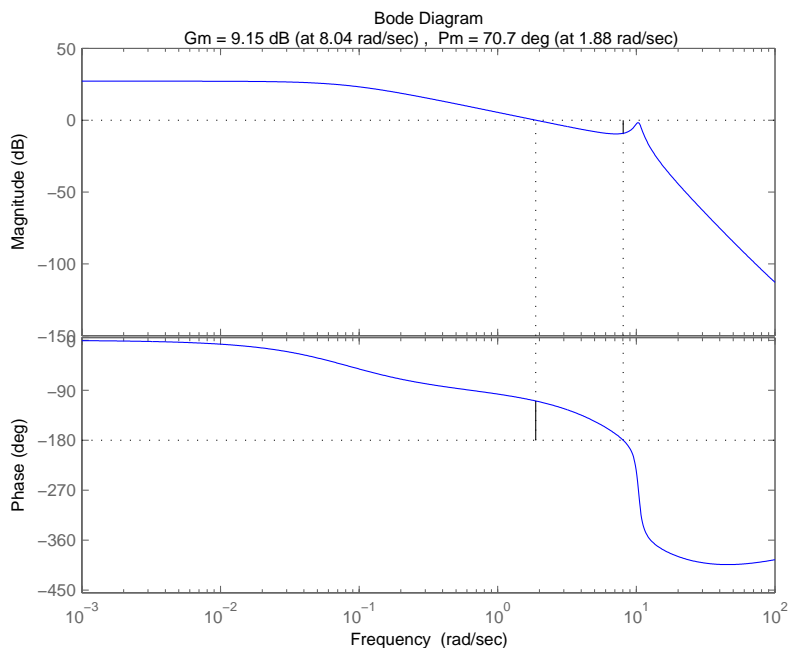
(Overføringsfunktionen er reelt fundet ud fra en linearisering med linmod.m).

$$H(s) = \frac{-106.1s^2 - 1.17 \cdot 10^4 s - 1.06 \cdot 10^5}{s^6 + 27.15s^5 + 377s^3 + 3881s^3 + 2.69 \cdot 10^4 s^2 + 8.72 \cdot 10^4 s + 6996} \quad (4.1)$$

Der kan nu anvendes standard PI design på overføringsfunktionen, altså design ud fra Bode plot og trinrespons. PI regulatoren er implementeret på formen, vist i ligning 4.2.

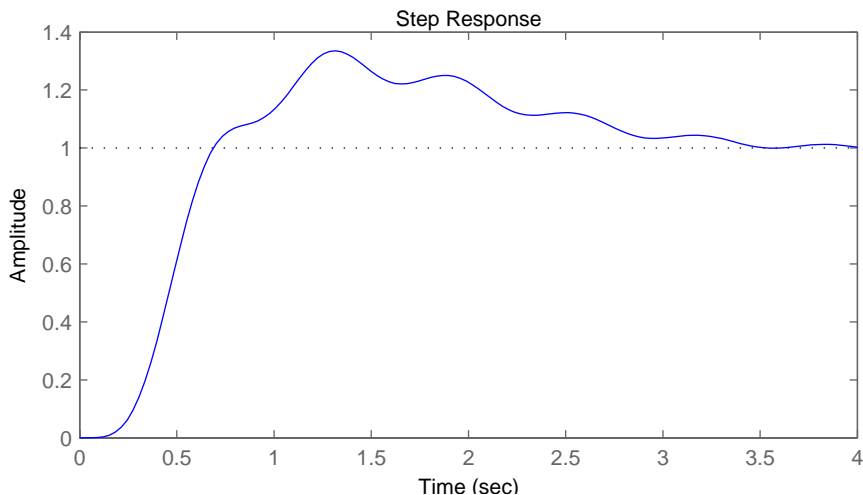
$$G(s) = K_p \frac{\tau_i s + 1}{\tau_i s} \quad (4.2)$$

For at bestemme gain- og fasemargin indsættes gainen K_p i serie med overføringsfunktionen, og den indstilles til en værdi, der giver en passende gain- og fasemargin. På figur 4.1 ses et bodeplot over open loop overføringsfunktionen med den valgte gain.



Figur 4.1: Gain- og fasemargin for open loop system med indsat gain på 1.5

Gainen K_p blev bestemt til 1.5, hvilket giver en fasemargin på ca. 70 grader. Denne relativt store fasemargin sikrer, at der ikke kommer et stort oversving i reguleringen, hvis der vælges en fornuftig værdi for τ_i . τ_i findes nu ud fra trinresponsen af lukketsløjfesystemet, hvor der skrues på værdien ind til der opnås et fornuftigt indsving. Resultatet heraf ses på figur 4.2, hvor der er valgt en τ_i værdi på 1.2.



Figur 4.2: Trinrespons på closed loop system med indsat PI regulator, hvor $K_p = 1.5$ og $\tau_i = 1.2$.

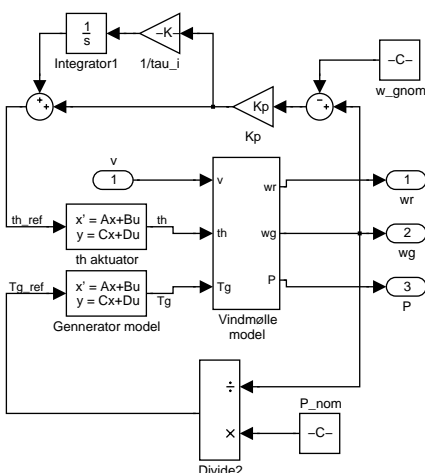
Som det ses af trinresponsen har reguleringen en indsvingningstid på ca. 4 sek. og et overshoot på ca. 40%. Det ses, at der er rippel på indsvinget. Dette skyldes dynamikken i drivakslen. Denne rippel vil være meget svær at dæmpe med PI regulering, da reguleringen som tidligere nævnt ikke kan regulere direkte på de interne states i systemet. Dette bevirker, at hvis det ønskes at mindske rippelen er det nødvendigt at gøre hele systemet langsommere, hvilket forringer hele reguleringen. Et alternativ ville være at designe yderligere en PI regulering i sløjfen fra $\Delta\phi$ til T_g . Dette er dog ikke afprøvet.

Regulatoren er kun indsat som feedback fra omdrejningerne til pitchen. Det ønskes også at regulere vindmøllen, så den ikke giver mere effekt end den nominelle. Derfor er det nødvendigt at indføre yderligere en løkke fra output effekten til modmomentet. Her er der valgt en simpel løsning, når der køres PI regulering. Modmomentet sættes blot ud fra beregningen, vist i ligning 4.3.

$$T_g = \frac{P_{nom}}{\omega_g} \quad (4.3)$$

Når modmomentet bliver sat på denne måde, bevirker det, at output effekten altid vil passe præcist, hvis der vel at mærke ikke er nogen forsinkelse fra T_{gref} til T_g , hvilket der reelt er. En momentregulering af denne type bevirker at en forøgelse af omdrejningerne resulterer i et fald i modmomentet. Dette er faktisk med til at gøre systemet ustabil, da et fald i modmomentet yderligere vil bidrage til et højere omdrejningstal. PI reguleringen på omdrejningerne vil dog sørge

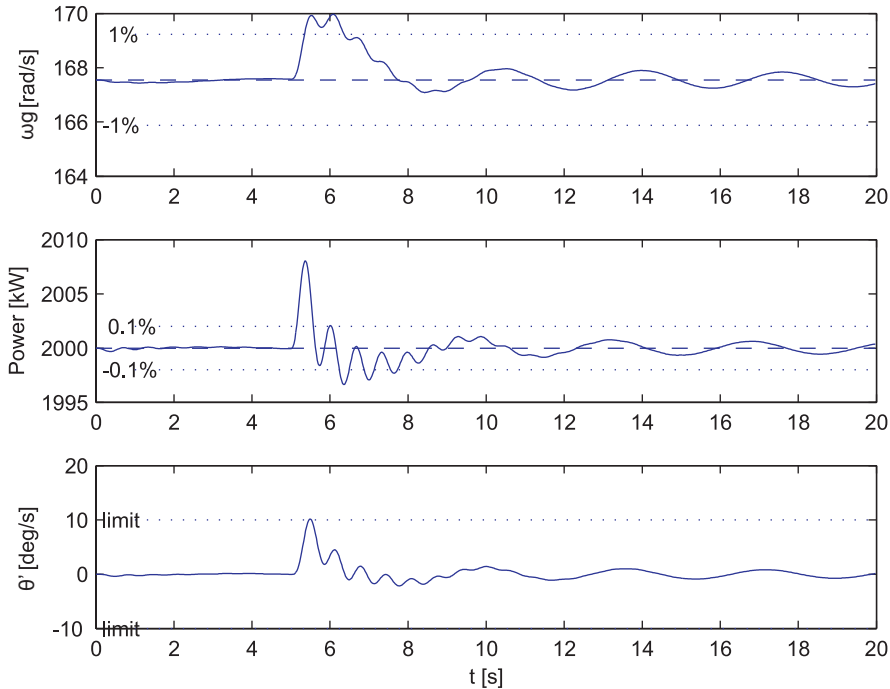
for, at omdrejningerne ikke løber løbsk, og det er derfor muligt at anvende denne beregning til at holde effekten på det ønskede nominelle niveau i stationære tilstande. På figur 4.3 ses et forenklet diagram over simulink implementeringen. Det komplette diagram kan ses på bilag D.



Figur 4.3: Simplificeret simulink diagram over implementering af PI regulering

For at give et hurtigt billede af, hvor godt PI regulatoren fungerer, vises der på figur 4.4 et plot over reguleringen indsat i simulationsmodellen, hvor der laves et trin i vinden fra 12m/s til 14m/s .

Det ses, at omdrejningerne ved generatoren har en indsvingningstid på ca. 3-4 sek. Ringningen af drivakslen slår igennem både på omdrejningerne, som det blev set under designet, men også på effekten og styresignalet. Ud over den højfrekvente ringning fra drivakslen, er der en mere lavfrekvent svingning. Denne svingning er forårsaget af tårnets bevægelse, som ikke er vist på dette plot. Sluttelig ses det, at hastigheden på pitch-aktuatoren lige præcis når de maksimale 10 grader/s , hvilket vil sige, at reguleringen er presset så meget, som er tilladt med de stillede designkriterier.



Figur 4.4: Trinrespons for PI regulering med et trin på vinden fra 12m/s til 14m/s.

4.2 LQ regulering

LQ eller Linear Quadratic reguleringen er en reguleringsform, som har flere fordele frem for PI reguleringen. For det første er det en regulering, der kan håndtere MIMO systemer, og designmetoden bygger på optimering af en kostfunktion. Designmetoden kan automatisk beregne en ny regulator til et nyt lineariseringspunkt, når der er blevet fastlagt nogle designparametre, hvilket selvfølgelig er en stor fordel, hvis man eksempelvis ønsker at lave et online design af reguleringen.

LQ anvender, som navnet antyder, en kvadratisk kostfunktion til at minimere fejlen mellem det ønskede respons og det reelle respons, som det ses af ligning 4.4.

$$J = \int_0^{\infty} (r(t) - y(t))^2 dt \quad (4.4)$$

Hvor $r(t)$ er styresignalerne og $y(t)$ er responset fra regulatoren.

Ideelt set vil den optimale løsning til dette problem være at $r(t) = y(t)$. Dette

er selvfølgelig ikke muligt i praksis, da det vil kræve uendelig hurtige ændringer i responset, hvilket igen kræver uendelig store styresignaler. Derfor indføres der et ekstra led i kostfunktionen som er kvadratisk med styresignalerne som vist i ligning 4.5. Der indføres yderligere de to vægte v_1 og v_2 , som angiver, hvordan styresignaler vægtes i forhold til output.

$$J = \int_0^{\infty} v_1 \cdot (r(t) - y(t))^2 + v_2 \cdot u^2 dt \quad (4.5)$$

Givet state spacemodellen i ligning 1.6 kan der ud fra teorien i [Reg2 04] opstilles et minimaliseringsproblem for et lineært system på en form, så kostfunktionen er givet ved ligning 4.6, her vist som værende tidsinvariant.

$$J = \int_0^{\infty} [\bar{x}^T \mathbf{R}_1 \bar{x} + \bar{u}^T \mathbf{R}_2 \bar{u} + 2\bar{x}^T \mathbf{R}_{12} \bar{u}] dt \quad (4.6)$$

Som det ses, er der her indsat de tre vægtmatriser \mathbf{R}_1 , \mathbf{R}_2 og \mathbf{R}_{12} . Disse vægtmatriser angiver, hvor stor en vægt man ønsker at påføre en state for \mathbf{R}_1 's tilfælde, et styresignal i \mathbf{R}_2 's tilfælde og krydskoblingen mellem styresignaler og states for \mathbf{R}_{12} . Vægtene kan altså styre, hvilke states man ønsker at favorisere i forhold til andre, samt hvor store styresignaler, der tillades.

Kostfunktionen er grundstenen i LQ design, da man ud fra teorien, forklaret i [Reg2 04], kan regne sig frem til en reguleringsmatrise \mathbf{K} , som ud fra de valgte vægte vil give den optimale regulering.

En ulempe ved LQ reguleringen er, at den kræver en måling eller et estimat af alle de states i systemet, som medtages i regulator designet. Dette kan give problemer, hvis designmodellen er af høj orden, da reguleringsdesignet vil indeholde lige så mange vægte som systemordenen, hvilket gør det svært at overskue valget af disse. Yderligere vil LQ i mange tilfælde kræve, at der designes en observer til reguleringsobjektet. Dette er nødvendigt i de tilfælde, hvor det ikke er muligt eller praktisk at måle en state i systemet, eller disse målinger er meget støjfyldte.

I dette projekt er der ikke blevet designet en observer til regulatoren. Det antages i stedet, at der er direkte tilgang til alle states, og at de er støjfri, hvilket selvfølgelig er en grov antagelse, men siden projektets formål primært er at undersøge mulighederne i individuel pitch, er det valgt at lave denne antagelse.

4.2.1 LQI I (simpel vindmøllemodel)

LQ reguleringen, der designes i dette projekt, er udvidet med integration for at opnå en regulering uden stationære fejl. LQI regulering indebærer, at der indføres en integration i systemet, hvor man ønsker, at den stationære fejl skal være 0. For vindmøllen i dette projekt er det på omdrejningerne ω_g og effekten

P. Integrationen på ω_g medfører også inddirekte, at ω_r stationært vil have en fejl på 0. Som udgangspunkt for LQI designet er det valgt at lave regulator designet ud fra den simpleste model. Regulator designet tager således udgangspunkt i modellen beskrevet i ligning 1.7, 1.10 og 1.13. Der er dog lavet den modifikation, at staten ω_r er flyttet, så den nu sidder på ω_g . I praksis er dette gjort ved at rykke gearet før integratoren (se figur 1.3 side 3), hvilket medfører at $\omega_r = \omega_g/N_g$. For overblikkets skyld er state space modellen vist med modifikationen i ligning 4.7.

$$\begin{aligned}
 \mathbf{A}_g &= \left[-\frac{\frac{1}{2}\pi\rho R^2 v^3 cp(\lambda, \theta) N_g}{J_{tot}\omega_g^2} + \frac{\frac{1}{2}\rho\pi R^2 v^3 N_g}{J_{tot}\omega_g} \cdot \frac{dcp(\lambda, \theta)}{d\lambda} \cdot -\frac{VN_g}{\omega_g^2 R} \right] \\
 \mathbf{B}_g^T &= \left[\begin{array}{c} \frac{1.5\pi\rho R^2 v^2 cp(\lambda, \theta) N_g}{J_{tot}\omega_g} + \frac{0.5\rho\pi R^2 V^3 \frac{dcp(\lambda, \theta)}{d\lambda} N_g}{J_{tot}R\omega_g^2} \\ \frac{\frac{1}{2}\rho\pi R^2 v^3 N_g}{J_{tot}\omega_g} \cdot \frac{dcp(\lambda, \theta)}{d\theta} \\ -\frac{N_g^2}{J_{tot}} \end{array} \right] \\
 \mathbf{C}_g^T &= \left[1/N_g \quad 1 \quad T_g \right]
 \end{aligned} \tag{4.7}$$

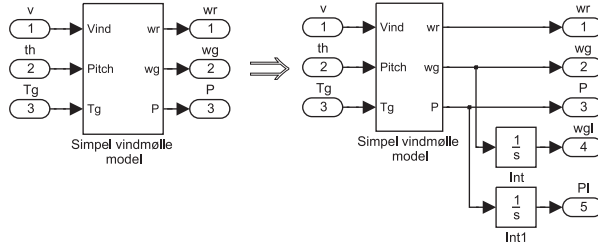
Med state-, input- og output vektorerne i ligning 4.8:

$$\bar{x} = [\omega_g] \quad \bar{u} = \begin{bmatrix} v \\ \theta \\ T_g \end{bmatrix} \quad \bar{y} = \begin{bmatrix} \omega_r \\ \omega_g \\ P \end{bmatrix} \tag{4.8}$$

Som nævnt skal der indføres integration på omdrejningerne og effekten. For at udbygge modellen med integratorerne anvendes samme metode som i kapitel 1, hvor modellen blev udvidet med to aktuatorer. Der defineres altså state- og output vektorene, vist i ligning 4.9

$$\bar{x} = \begin{bmatrix} \omega_g \\ \omega_g I \\ PI \end{bmatrix} \quad \bar{y}^T = [\omega_r \quad \omega_g \quad P \quad \omega_g I \quad PI] \tag{4.9}$$

For at visualisere processen er der på figur 4.5 vist, hvorledes integratorerne indsættes i simulink modellen. I bilag D kan den komplette simulink model ses.



Figur 4.5: Simplificeret simulink diagram over indførelse af integration i ω_g og P . Simpel designmodel.

Det ses af figuren, at input vektoren og matricen er bevaret, og at der kommer to nye states samt to nye output. Det ses yderligere, at inputtet til de to integratorstates må være givet af output $\bar{y}(1, 2)$ og $\bar{y}(1, 3)$, og at de nye states opererer uafhængigt af hinanden, hvilket vil sige, at der ikke kommer nogen krydskoblinger i den nye state matrice. De nye differentialligninger for systemet kan nu opstilles og ses i ligning 4.10 til 4.12.

$$\dot{\omega}_g = \frac{\frac{1}{2}\pi\rho R^2 v^3 cp(\lambda, \theta) N_g}{J_{tot}\omega_g} - \frac{T_g N_g^2}{J_{tot}} \quad (4.10)$$

$$\dot{\omega}_g I = \omega_g \quad (4.11)$$

$$\dot{P} I = \frac{1}{N_g} \omega_g \cdot T_g N_g \quad (4.12)$$

Den nye state space model, der nu kan opstilles, ses i ligning 4.13. Der er igen anvendt metoden med at differentiere system differentialligningerne 4.10 til 4.12 med hensyn til alle states for at opnå \mathbf{A}_{I1} matricen og differentiere med hensyn til alle input for at få \mathbf{B}_{I1} matricen.

$$\begin{aligned} \mathbf{A}_{I1} &= \begin{bmatrix} -\frac{\frac{1}{2}\pi\rho R^2 v^3 cp(\lambda, \theta) N_g}{J_{tot}\omega_g^2} + \frac{\frac{1}{2}\rho\pi R^2 v^3 N_g}{J_{tot}\omega_g} \cdot \frac{dcp(\lambda, \theta)}{d\lambda} \cdot -\frac{VN_g}{\omega_g^2 R} & 0 & 0 \\ 1 & 0 & 0 \\ T_g & 0 & 0 \end{bmatrix} \\ \mathbf{B}_{I1}^T &= \begin{bmatrix} \frac{1.5\pi\rho R^2 v^2 cp(\lambda, \theta) N_g}{J_{tot}\omega_g} + \frac{0.5\rho\pi R^2 V^3 \frac{dcp(\lambda, \theta)}{d\lambda} N_g}{J_{tot}R\omega_g^2} & 0 & 0 \\ \frac{\frac{1}{2}\rho\pi R^2 v^3 N_g}{J_{tot}\omega_g} \cdot \frac{dcp(\lambda, \theta)}{d\theta} & 0 & 0 \\ -\frac{N_g^2}{J_{tot}} & 0 & \omega_g \end{bmatrix} \\ \mathbf{C}_{I1}^T &= [1/N_g \quad 1 \quad T_g \quad 1 \quad 1] \end{aligned} \quad (4.13)$$

Med integratorerne indført i designmodellen kan der nu vælges et lineariseringspunkt, hvilket som for PI reguleringen sættes til $14m/s$. De stationære værdier for θ , T_g og ω_g kan bestemmes, som beskrevet i kapitel 1, med matlab filen

stpkt0. Når dette er gjort, er det blot at indsætte værdierne i ligning 4.13, eller at anvende linmod på modellen, vist i bilag D.1.3.

Designfasen består nu primært i at vælge nogle fornuftige vægtmatricer: \mathbf{R}_1 , \mathbf{R}_2 og \mathbf{R}_{12} , hvor den sidstnævnte vil være 0 i de følgende design, fordi alle vægte er ført direkte ud enten fra styresignaler eller states. De tre vægtmatricer kan samles til en vægtmatrice \mathbf{R}_0 ved at anvende transformationen, beskrevet i [Kjølstad 06], som ses i ligning 4.14 til 4.16, her skrevet op på en standardform.

$$\mathbf{R}_1 = \mathbf{C}^T \mathbf{R}_0 \mathbf{C} \quad (4.14)$$

$$\mathbf{R}_2 = \mathbf{D}^T \mathbf{R}_0 \mathbf{D} \quad (4.15)$$

$$\mathbf{R}_{12} = \mathbf{C}^T \mathbf{R}_0 \mathbf{D} \quad (4.16)$$

Som det ses, kan vægtmatricerne \mathbf{R}_1 , \mathbf{R}_2 og \mathbf{R}_{12} nu bestemmes ud fra en ny vægtmatrice \mathbf{R}_0 , output matricen \mathbf{C} og fremføringsmatricen \mathbf{D} . Vægtene er normalt placeret i diagonalen.

Fremføringsmatricen \mathbf{D} har ikke tidligere været vist for state spacemodellerne, da den har været 0. For at anvende transformationen er det nødvendigt at omskrive systemet så alle de styresignaler, states og output, der ønskes vægtet, føres ud som output, hvilket giver nye \mathbf{C} og \mathbf{D} matricer for systemet. For systemet i ligning 4.13 defineres nu den ønskede output vektor i ligning 4.17.

$$\bar{\mathbf{y}}^T = [\theta \quad T_g \quad \omega_g \quad \omega_g I \quad PI] \quad (4.17)$$

Nu opstilles en ny output- og fremføringsmatrice \mathbf{C}_{I11} \mathbf{D}_{I11} , vist i ligning 4.18.

$$\mathbf{C}_{I11} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{D}_{I11} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.18)$$

Der kan nu foretages et valg af vægte til designet af LQI reguleringen. Selve regulator gain matricen \mathbf{K}_1 bliver beregnet med matlab funktionen `lqr.m`, som skal have et lineært system og de tre vægtmatricer \mathbf{R}_1 , \mathbf{R}_2 og \mathbf{R}_{12} som input parametre. Som udgangspunkt for vægtningen anvendes tommelfingerreglen, vist i ligning 4.19.

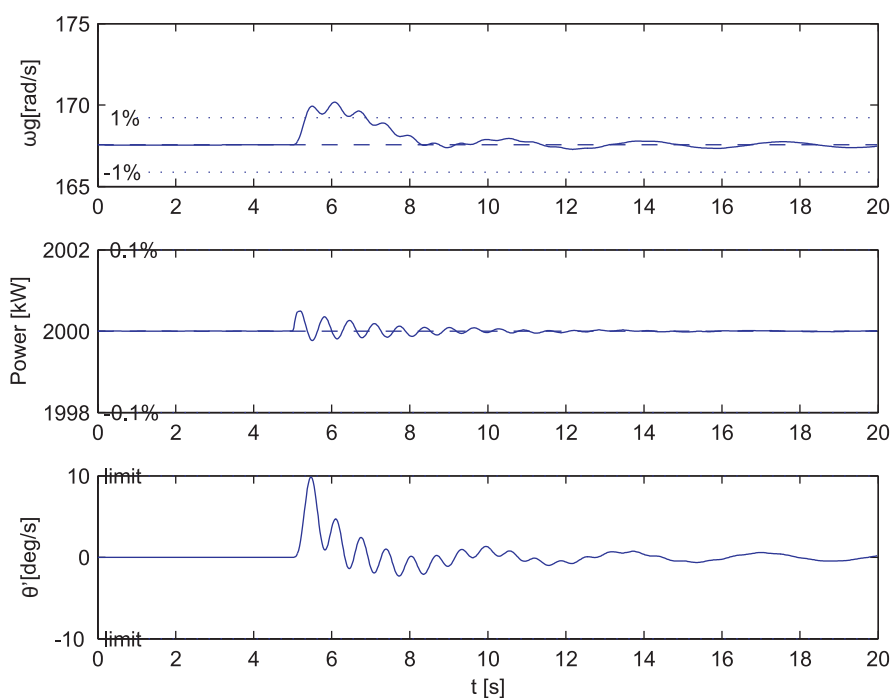
$$\mathbf{R}_1(i, i) = \frac{1}{\max(\bar{x}_i^2)} \quad \mathbf{R}_2(j, j) = \frac{1}{\max(\bar{u}_j^2)} \quad (4.19)$$

Disse vægte giver et meget ringe, men stabilt resultat. De endelige vægte er opnået ved at skrue på vægtene og derefter se på trinresponsen for omdrejningerne ω_g , effekten P og pitch-vinkelhastigheden $\dot{\theta}$. Vægtene er ikke ændret tilfældigt, men ud fra den antagelse, at de to førnævnte output skal minimeres, altså straffes i kostfunktionen. Samtidig tillades større styresignaler θ og T_g

ved at vægte disse mindre, dog skal $\dot{\theta}$ hele tiden holdes inden for de maksimalt tilladte 10 grader/s. De endelige vægte er vist i ligning 4.20

$$\mathbf{R}_0(i, i) = \begin{bmatrix} \frac{0.15}{\dot{\theta}^2_{nom}} & \frac{0.1}{T^2_{gnom}} & \frac{10}{\omega^2_{gnom}} & \frac{100}{\omega^2_{gnom}} & \frac{3000}{P^2_{nom}} \end{bmatrix} \quad (4.20)$$

Disse vægte giver en regulering, vist i figur 4.6.

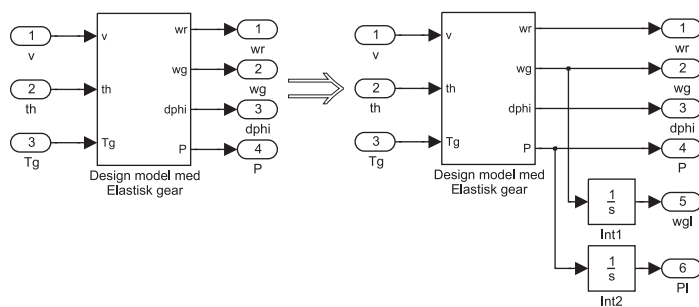


Figur 4.6: Trinrespons for simpel LQI regulator med et trin på vinden fra $12m/s$ til $14m/s$.

Det kan ses, at der stadig er ringning både fra tårnet og fra drivakslen. Dette er nu ikke overraskende, da disse ikke er taget med i designmodellen. Det ses tydeligt, at effekten nu er holdt på et meget lavere niveau, end det var muligt med PI reguleringen, og dette er opnået med omtrent samme indsvingning på omdrejningerne. Ved at vægte omdrejningerne højere vil det være muligt at lave et design, der på bekostning af mindre stabil effekt vil give et hurtigere indsving på omdrejningerne. Sluttelig ses det, at pitch-hastigheden $\dot{\theta}$ igen holdes lige på de maksimalt tilladte 10 grader/s.

4.2.2 LQI II (udvidet med drivaksel)

Regulatordesignet udvides nu, så det inkluderer drivakslen. Dette vil give mulighed for at reducere de ringninger, der ses på ω_g , og derved også reducere den dynamiske belastning af drivakslen. Som for den simple LQI regulering er det nødvendigt at udvide designmodellen med integration på effekten og omdrejningerne ved generatoren. Da fejlen på disse output stationært skal være 0. I kapitel 1 er modellen uden integration beskrevet ved ligning 1.36 og 1.37. Integratorerne indsættes, som vist i figur 4.7. Et komplet diagram over simulink implementeringen kan ses i bilag D.



Figur 4.7: Simplificeret simulink diagram over indførelse af integration i ω_g og P . Designmodel med drivaksel.

Det ses, at input vektoren \bar{u} er bevaret fra ligning 4.8. Der defineres nu state vektoren, vist i ligning 4.21.

$$\bar{x}^T = [\omega_r \quad \omega_g \quad \Delta\phi \quad \omega_g I \quad PI] \quad (4.21)$$

Modellen med integratorer vil være af 5.orden og kan beskrives ved differential-ligningerne, vist i ligning 4.22 til 4.26.

$$\dot{\omega}_r = \frac{\frac{1}{2}\pi\rho R^2 v^3 cp(\lambda, \theta)}{\omega_r J_r} - \frac{K_s}{J_r} \Delta\phi - \frac{D_s}{J_r} \omega_r + \frac{D_s}{J_r N_g} \omega_g \quad (4.22)$$

$$\dot{\omega}_g = -\frac{T_g}{J_g} + \frac{K_s}{J_g N_g} \Delta\phi + \frac{D_s}{J_g N_g} \omega_r - \frac{D_s}{J_g N_g^2} \omega_g \quad (4.23)$$

$$\dot{\Delta\phi} = \omega_r - \frac{\omega_g}{N_g} \quad (4.24)$$

$$\dot{\omega_g I} = \omega_g \quad (4.25)$$

$$\dot{PI} = T_g \omega_g \quad (4.26)$$

Statematrixen \mathbf{A}_{I2} kan nu findes ved at differentiere med hensyn til de 5 states, og input matrixen \mathbf{B}_{I2} ved differentiering med hensyn til de 3 input. Dette

resulterer i ligning 4.27.

$$\mathbf{A}_{I_2} = \begin{bmatrix} \mathbf{A}_{I_2(1,1)} & \frac{D_s}{J_r N_g} & -\frac{K_s}{J_r} & 0 & 0 \\ \frac{D_s}{J_g N_g} & -\frac{D_s}{J_g N_g^2} & \frac{K_s}{J_g N_g} & 0 & 0 \\ 1 & -\frac{1}{N_g} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & T_g & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}_{I_2} = \begin{bmatrix} \mathbf{B}_{I_2(1,1)} & \frac{1}{2} \rho \pi R^2 v^3 \cdot \frac{dcp(\lambda, \theta)}{d\theta} & 0 \\ 0 & 0 & -\frac{1}{J_g} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \omega_g \end{bmatrix} \quad (4.27)$$

Hvor $\mathbf{A}_{I_2(1,1)}$ og $\mathbf{B}_{I_2(1,1)}$ er givet ved ligning 4.28 og 4.29.

$$\mathbf{A}_{I_2(1,1)} = \left[-\frac{\frac{1}{2} \pi \rho R^2 v^3 cp(\lambda, \theta)}{J_r \omega_r^2} + \frac{\frac{1}{2} \rho \pi R^2 v^3}{J_r \omega_r} \cdot \frac{dcp(\lambda, \theta)}{d\lambda} \cdot -\frac{v}{\omega_r^2 R} - \frac{D_s}{J_r} \right] \quad (4.28)$$

$$\mathbf{B}_{I_2(1,1)} = \frac{1.5 \pi \rho R^2 v^2 cp(\lambda, \theta)}{J_r \omega_r} + \frac{0.5 \rho \pi R^2 v^3 \frac{dcp(\lambda, \theta)}{d\lambda}}{J_r R \omega_r^2} \quad (4.29)$$

Som tidligere beskrevet er det nødvendigt at definere alle input, states og output, der ønskes vægtet som output. I dette tilfælde giver det en output vektor, som ses i ligning 4.30.

$$\vec{y}^T = [\theta \quad T_g \quad \omega_r \quad \omega_g \quad \Delta\phi \quad \omega_g I \quad PI] \quad (4.30)$$

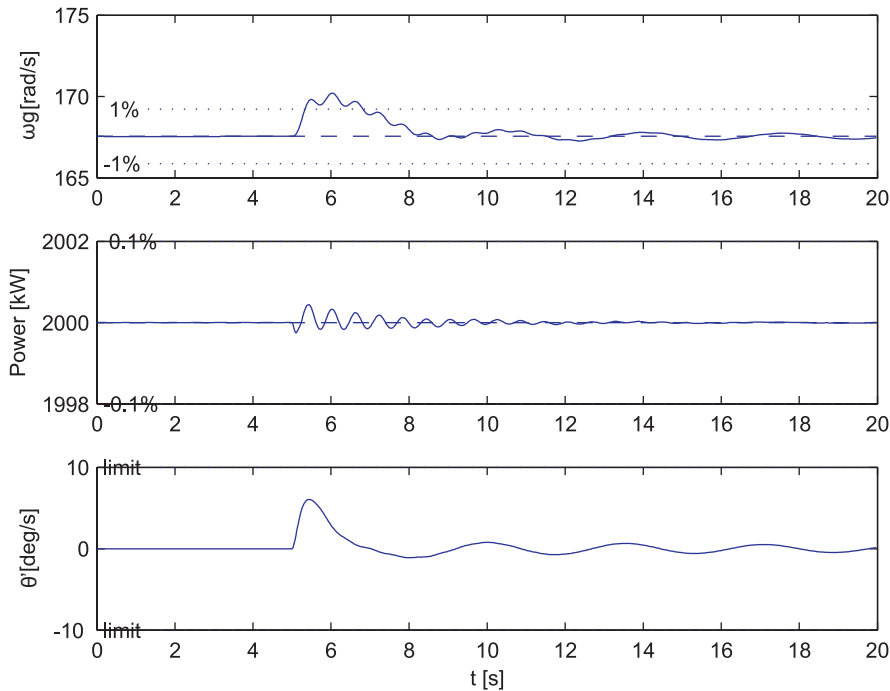
Når output vektoren er defineret, er det muligt at opstille output matricen \mathbf{C}_{I_2} og feed forward matricen \mathbf{D}_{I_2} . Disse ses i ligning 4.31.

$$\mathbf{C}_{I_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{D}_{I_2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.31)$$

Den lineære designmodel med drivaksel og indsatte integrationer er nu komplet beskrevet. Som tidligere vælges der et lineariseringspunkt på 14 m/s. Der er kommet yderligere to states at vægte: ω_r og $\Delta\phi$. For at få en regulering, der er nemt sammenlignelig med de foregående regulatorer, er det valgt som udgangspunkt at vægte disse states med 0. Vægtene er vist i ligning 4.32.

$$\mathbf{R}_0(i, i) = \left[\frac{0.15}{\theta_{nom}^2} \quad \frac{0.1}{T_{gnom}^2} \quad \frac{0}{\omega_{rnom}^2} \quad \frac{10}{\omega_{gnom}^2} \quad \frac{0}{\Delta\phi_{nom}^2} \quad \frac{100}{\omega_{gnom}^2} \quad \frac{3000}{P_{nom}^2} \right] \quad (4.32)$$

Denne regulering giver et resultat, som kan ses af figur 4.8.



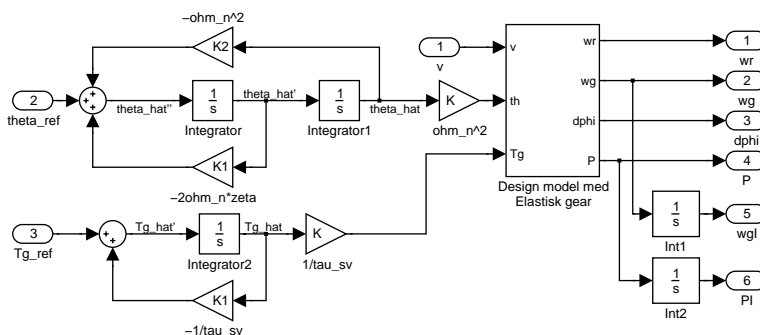
Figur 4.8: Trinrespons for LQI regulator designet med drivaksel indsat. Trin på vinden fra 12 m/s til 14 m/s .

Det ses, at responset i omdrejningerne og effekten ikke er ændret bemærkelsesværdigt fra det simple LQI design. På pitch-vinkelhastigheden ses der tydelige forbedringer. Dels er ringningen i styresignalet reduceret kraftigt, og dels er amplituden på peaket næsten halveret. Dette muliggør et nyt design af regulatoren, der uden at overskride kravet om en maksimal pitch-hastighed på 10 grader/s , kan vægtes mindre på pitch-styresignalet. Et sådant design vil kunne forbedre indsvinget i omdrejningerne. I kapitel 5 undersøges hvilke muligheder der er for at dæmpe svingningen i drivakselen ved at vægte $\Delta\phi$ højere. Vægten på ω_r vil give en mulighed for at regulere, om man vil prioritere omdrejningerne på rotor- eller generator siden.

4.2.3 LQI III (med drivaksel og aktuatorer)

Regulatordesignet udvides yderligere, så det også indeholder dynamikken fra pitch-aktuatoren og generator modellen, for at undersøge, om det er muligt at

opnå en bedre regulering. Modellen vil således blive udvidet med tre states, som vist i figur 4.9. Et komplet simulink diagram kan ses i bilag D.



Figur 4.9: Simulinkdiagram over indførsel af integration i ω_g og P . Designmodel med drivaksel og aktuatorer

Den nye state vektor kan ses i ligning 4.33.

$$\bar{x}^T = \left[\hat{\theta} \quad \dot{\hat{\theta}} \quad \hat{T}_g \quad \omega_r \quad \omega_g \quad \Delta\phi \quad \omega_g I \quad PI \right] \quad (4.33)$$

Som det ses, er modellen nu af 8.orden. De 8 differentialligninger, der beskriver systemet ses i ligning 4.34 til 4.41.

$$\dot{\hat{\theta}} = \dot{\hat{\theta}} \quad (4.34)$$

$$\ddot{\hat{\theta}} = -\omega_n^2 \hat{\theta} - 2\zeta\omega_n \dot{\hat{\theta}} + \theta_{ref} \quad (4.35)$$

$$\dot{\hat{T}}_g = -\frac{1}{\tau_{sv}} \hat{T}_g + T_{gref} \quad (4.36)$$

$$\dot{\omega}_r = \frac{\frac{1}{2}\pi\rho R^2 v^3 cp(\lambda, \hat{\theta}\omega_n^2)}{\omega_r J_r} - \frac{K_s}{J_r} \Delta\phi - \frac{D_s}{J_r} \omega_r + \frac{D_s}{J_r N_g} \omega_g \quad (4.37)$$

$$\dot{\omega}_g = -\frac{\hat{T}_g}{J_g \tau_{sv}} + \frac{K_s}{J_g N_g} \Delta\phi + \frac{D_s}{J_g N_g} \omega_r - \frac{D_s}{J_g N_g^2} \omega_g \quad (4.38)$$

$$\dot{\Delta\phi} = \omega_r - \frac{\omega_g}{N_g} \quad (4.39)$$

$$\dot{\omega}_g I = \omega_g \quad (4.40)$$

$$\dot{PI} = \frac{\hat{T}_g}{\tau_{sv}} \omega_g \quad (4.41)$$

Som tidligere findes state matricen \mathbf{A}_{I3} og input matricen \mathbf{B}_{I3} ved at differenciere ligning 4.34 til 4.41 med hensyn til states og input. Resultatet se i ligning

4.42.

$$\begin{aligned}
\mathbf{A}_{I3} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\omega_n^2 & -2\zeta\omega_n & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{sv}} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{A}_{I3(4,1)} & 0 & 0 & \mathbf{A}_{I3(4,4)} & \frac{D_s}{J_r N_g} & -\frac{K_s}{J_r} & 0 & 0 \\ 0 & 0 & -\frac{1}{J_g \tau_{sv}} & \frac{D_s}{J_g N_g} & -\frac{D_s}{J_g N_g^2} & \frac{K_s}{J_g N_g} & 0 & 0 \\ 0 & 0 & 0 & 1 & -\frac{1}{N_g} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{\omega_g}{\tau_{sv}} & 0 & \frac{\hat{T}_g}{\tau_{sv}} & 0 & 0 & 0 \end{bmatrix} \\
\mathbf{B}_{I3} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{1.5\pi\rho R^2 v^2 cp(\lambda, \hat{\theta}\omega_n^2)}{J_r \omega_r} + \frac{0.5\pi\rho R^2 v^3 \frac{dcp(\lambda, \hat{\theta}\omega_n^2)}{d\lambda}}{J_r R \omega_r^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.42)
\end{aligned}$$

Hvor $\mathbf{A}_{I3(4,1)}$ og $\mathbf{A}_{I3(4,4)}$ er givet ved ligning 4.43 og 4.44.

$$\mathbf{A}_{I3(4,1)} = \omega_n^2 \frac{\frac{1}{2}\rho\pi R^2 v^3}{J_r \omega_r} \frac{dcp(\lambda, \hat{\theta}\omega_n^2)}{d\hat{\theta}\omega_n^2} \quad (4.43)$$

$$\mathbf{A}_{I3(4,4)} = -\frac{\frac{1}{2}\pi\rho R^2 v^3 cp(\lambda, \hat{\theta}\omega_n^2)}{J_r \omega_r^2} + \frac{\frac{1}{2}\rho\pi R^2 v^3}{J_r \omega_r} \cdot \frac{dcp(\lambda, \hat{\theta}\omega_n^2)}{d\lambda} \cdot -\frac{v}{\omega_r^2 R} - \frac{D_s}{J_r} \quad (4.44)$$

Som tidligere defineres output vektoren nu med alle input, states og output, som ønskes vægtet i reguleringen. Denne ses i ligning 4.45.

$$\bar{y}^T = \left[\theta \quad T_g \quad \hat{\theta} \quad \dot{\hat{\theta}} \quad \hat{T}_g \quad \omega_r \quad \omega_g \quad \Delta\phi \quad \omega_g I \quad PI \right] \quad (4.45)$$

Ud fra output vektoren og simulink diagrammet i bilag D kan \mathbf{C}_{I3} og \mathbf{D}_{I3} opstilles, som vist i ligning 4.46.

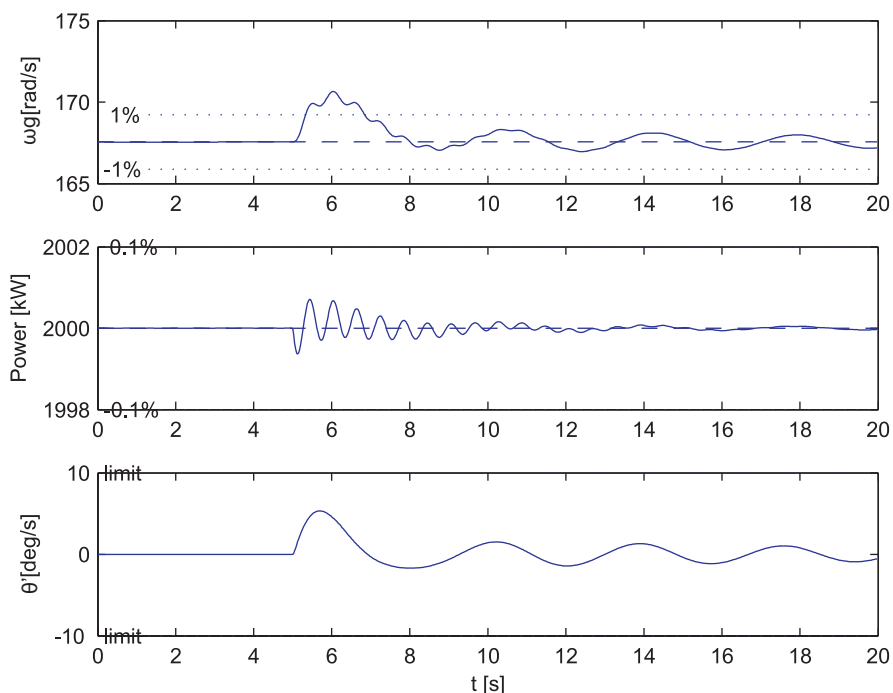
$$\begin{aligned}
\mathbf{C}_{I2} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{D}_{I2} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.46)
\end{aligned}$$

For igen at få et regulator design, der nemt kan sammenlignes med de foregående regulatorer, er de nye states $\hat{\theta}$, $\dot{\hat{\theta}}$ og \hat{T}_g indledningsvis vægtet med nul, hvilket

giver vægtmatricen, vist i ligning 4.47.

$$\mathbf{R}_0(i, i) = \begin{bmatrix} \frac{0.15}{\theta_{nom}^2} & \frac{0.1}{T_{gnom}^2} & \frac{0}{\theta_{nom}^2} & \frac{0}{\dot{\theta}_{nom}^2} & \frac{0}{\dot{T}_{gnom}^2} & \dots \\ \frac{0}{\omega_{rnom}^2} & \frac{10}{\omega_{gnom}^2} & \frac{0}{\Delta\phi_{nom}^2} & \frac{100}{\omega_{gnom}^2} & \frac{3000}{P_{nom}^2} \end{bmatrix} \quad (4.47)$$

Resultatet af reguleringen vises i figur 4.10.



Figur 4.10: Trinrespons for LQI regulator designet med drivaksel og aktuatorer indsat. Trin på vinden fra $12m/s$ til $14m/s$.

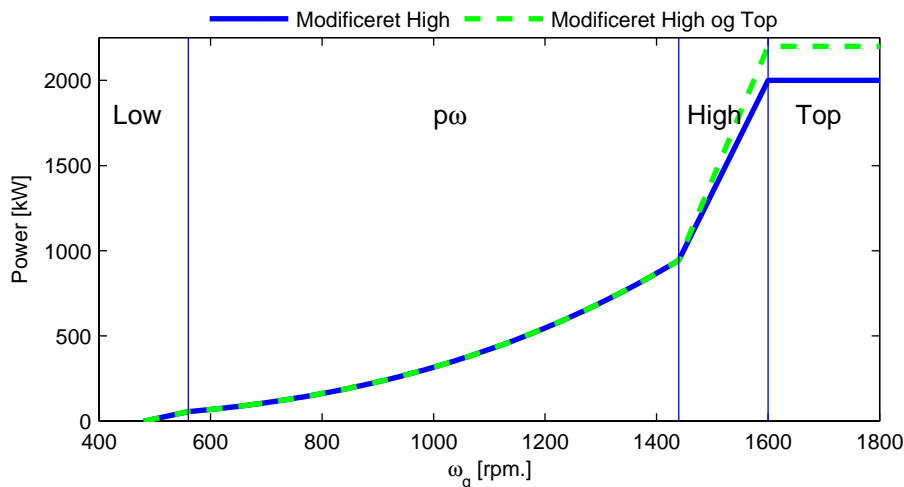
Som det ses, er der ikke nogen forbedring af regulator designet uden aktuatorer. Der er nærmere en forringelse, da indsvinget i både omdrejninger og effekt har større amplitude og er blevet lidt langsommere. Det ses yderligere, at den lavfrekvente ringning fra tårnsvingningen er længere tid om at dø ud. Den eneste umiddelbare forbedring ligger i, at hældningen på pitch-hastighedsindsvinget er blevet mere fladt, hvilket svarer til en mindre acceleration. Accelerationen har dog ikke været en designparameter i dette projekt, men kunne tænkes at være en væsentlig designfaktor. Der er blevet lavet test med flere forskellige vægtkombinationer, men resultatet har hele tiden været, at der kunne opnås en bedre regulering ved at designe uden aktuatorer. Det var forventet, at inkluderingen

af aktuatorerne ville give en forbedring i reguleringen, hvilket umiddelbart ikke syntes at være tilfældet. Designmetoden er dog ikke blevet helt afskrevet, da det ikke kan udelukkes, at der vil kunne opnås bedre resultater ved dybere undersøgelse. Der er ikke blevet gravet dybere i at finde forklaringen på det uventede resultat. Reguleringen, designet med aktuatorerne, vil derfor ikke blive testet yderligere.

4.3 Bumpless transfer

Som det fremgår af de to tidligere afsnit, er der behov for at kunne skifte mellem $p\omega$ og topreguleringen, hvilket selvfølgelig skal ske så smertefrit som muligt. Et emne, som kaldes for Bumpless transfer. Dette emne er vigtigt, da de to regulatorer ikke nødvendigvis er enige om styresignalernes størrelse på det tidspunkt, hvor skiftet sker. Hvis skiftet foretages i et sådant tilfælde, vil det medføre et kraftigt indsving af den nye regulator, da styresignalet ændres abrupt. I værste fald medfører dette indsving, at der skiftes tilbage til den første regulator med muligheden for, at systemet går i selvsving. Reglen siger at: 'Udgangen på den inaktive regulator skal til en hver tid være lig det aktive styresignal'. Dette er vigtigt, hvis der skal kunne laves regulator skift til et vilkårligt tidspunkt. I vindmøllens tilfælde er dette ikke nødvendigt. Skiftet af regulator skal ske på et bestemt tidspunkt, nemlig når vinden bliver så kraftig, at effekt begrænsning er nødvendig. Dette sker ved stationære vindhastigheder på over ca. 11 m/s. Selv hvis det var muligt at måle vinden, vil det ikke være hensigtsmæssigt at bruge den til at skifte efter. Grunden ligger i møllens store inert, som bevirker, at output effekten først vil komme væsentligt senere end en stigning i vinden. Det vil derfor være langt mere hensigtsmæssigt at bruge output effekten til at bestemme skiftetidspunktet ud fra. Systemets to styresignaler θ og T_g skal være ens i det øjeblik, skiftet sker. I $p\omega$ området er der et næsten fast forhold mellem T_g og den leverede effekt. Det er derfor valgt at skifte til topregulatoren i det øjeblik $p\omega$ regulatorens modmoment er større end det nominelle, altså $T_{gp\omega} \geq T_{gnom}$. For at sikre at $p\omega$ regulatorens modmoment forbliver over T_{gnom} , mens topregulatoren er aktiv, er $p\omega$ kurven endnu engang ændret lidt. Ved at ændre effekt niveauet for highområdet til at ligge over 2MW sikres det, at mindre variationer i ω_g ikke forårsager et utilsigtet skift. Derfor ser den anvendte $p\omega$ kurve ud, som vist på figur 4.11.

For pitch-vinklen θ er det ikke noget problem at skifte. Den vil nemlig være lig med θ_{nom} i highområdet. Dette er sikret ved de tidligere modifikationer beskrevet i afsnit 3.2 og illustreret på figur 3.3. topreguleringen vil gå mod θ_{nom} , når den nærmer sig highområdet.



Figur 4.11: $p\omega$ kurve med ekstra stor effekt reference i high sikrer mod utilsigtede skift.

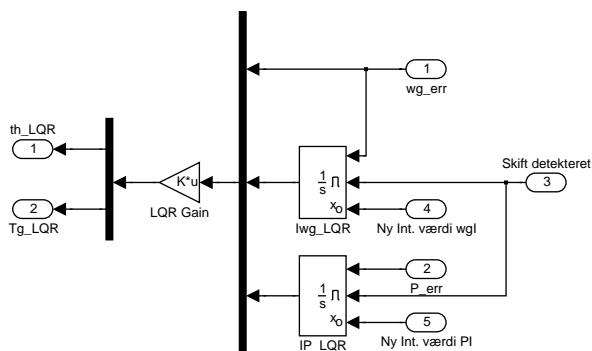
Nu er skiftegrundlaget altså bestemt, og det vides, hvordan $p\omega$ regulatoren opfører sig, når den er inaktiv. Der skal yderligere tages højde for at topreguleringens styresignaler er korrekte i skifteøjeblikket. Dette gøres ved at resette integratorerne i regulatoren en sample før skiftet sker.

Der er i alt betragtet to forskellige regulator typer i topområdet, PI og LQI.

Da PI regulatoren beregner sit styresignal ud fra den samme reference som $p\omega$ regulatoren, nemlig ω_g , vil styresignalerne også passe i skifteøjeblikket. Anderledes er det for de tre LQI regulatorer. I det følgende er skiftemekanismen for disse beskrevet.

LQI regulatoren har to integratorer; en på w_{gerr} og en på P_{err} . Styresignalerne beregnes således ud fra de to integratorværdier samt det aktuelle fejlsignal ω_{gerr} . Når regulatoren skiftes, og styresignalerne skal passe sammen, skal det altså ske ved at bestemme de rette integratorværdier ud fra den afvigelse, der er på ω_g . På den måde vil styresignalerne passe i skifteøjeblikket. Efter skiftet vil der komme et indsving, da den nye regulator har andre forstærkninger og derfor vil minimere de forskellige fejlsignaler på en anden måde. Men det, der er det

vigtige, er, at der ikke er noget abrupt skift i styresignalerne. De nye reset-



Figur 4.12: LQR Topregulator kan resettes med værdier fundet ud fra den aktuelle ω_{gerr} .

værdier bliver beregnet løbende, og hvis der bliver detekteret et skift, resettes integratorerne med disse nye værdier. Et simpelt diagram illustrerer princippet på figur 4.12. Udfra den aktuelle ω_{gerr} kan de to integratorværdier beregnes, når man husker på de to styresignaler er θ_{nom} og T_{gnom} . Dette gøres v.h.a. ligning 4.48 og 4.49.

$$I\omega_{reset} = \frac{\theta_{nom} - K_{(1,1)}\omega_{gerr} + K_{(1,2)}I\omega_{gerr}}{K_{(1,3)}} \quad (4.48)$$

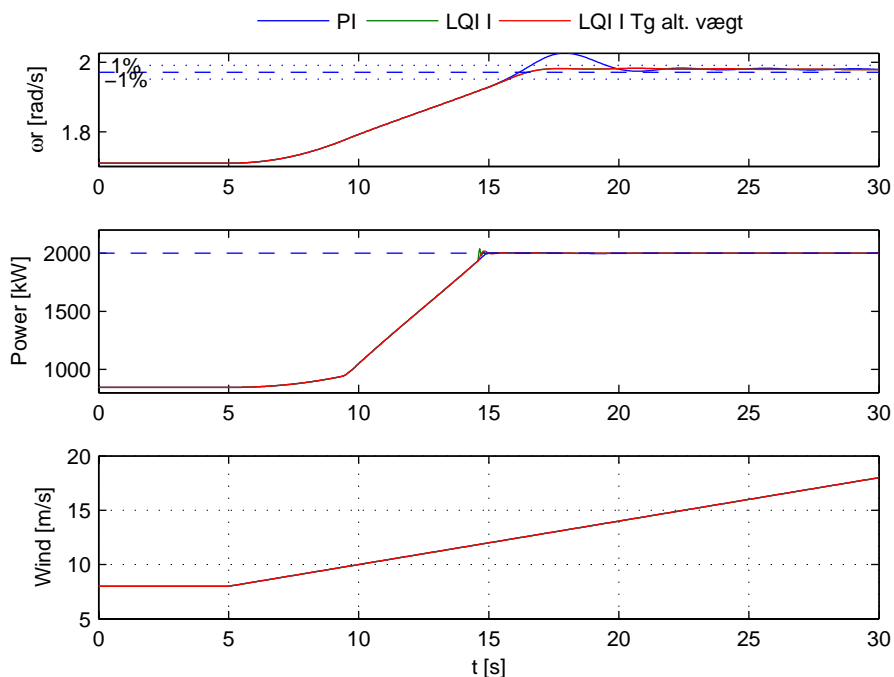
$$IP_{reset} = \frac{T_{gnom} - K_{(2,1)}\omega_{gerr} + K_{(2,3)}IP_{gerr}}{K_{(2,3)}} \quad (4.49)$$

Ligningerne løses symbolsk i matlab for hver regulator design, og for hver sample beregnes der derved nye reset-værdier. For LQI II og III er det det samme princip. Her skal de to integratorværdier dog beregnes på baggrund af henholdsvis 3 og 6 ikke integrerede states, i stedet for blot at være afhængig af ω_{gerr} . Implementeringen af de i alt 4 regulatorer kan ses i bilag D.

Bumpless metoden er afprøvet både med vind, der stiger og falder som rampe, trin og med den stokastiske vindmodel. Det vil være for omfattende at bringe alle plot her i selve rapporten, derfor henvises til kapitel 5 og bilag C hvor der er yderligere test.

Her er det valgt at vise et skift for PI- og LQI regulatoren, når vinden stiger som en rampe. Vinden starter på 8 m/s og stiger jævnt 0.4 m/s² fra tiden 5 sek. Skiftet til topregulering sker ca. til tiden 15 sek. Vindinput til simulationen samt output effekt og rotoromdrejningshastighed ses på figur 4.13. Styresignalerne θ og dennes afledte $\dot{\theta}$ samt T_g er vist i figur 4.14.

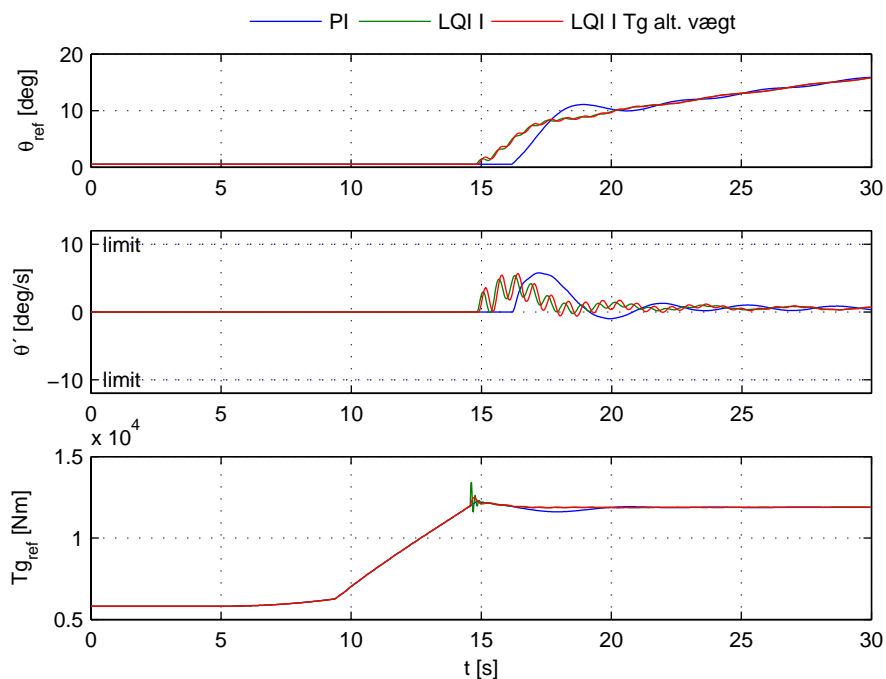
På figur 4.14 nederst ser det ud til at styresignalet T_g ikke skifter glat som forventet. Dette skyldes topregulatorens kraftige forstærkninger, som giver et



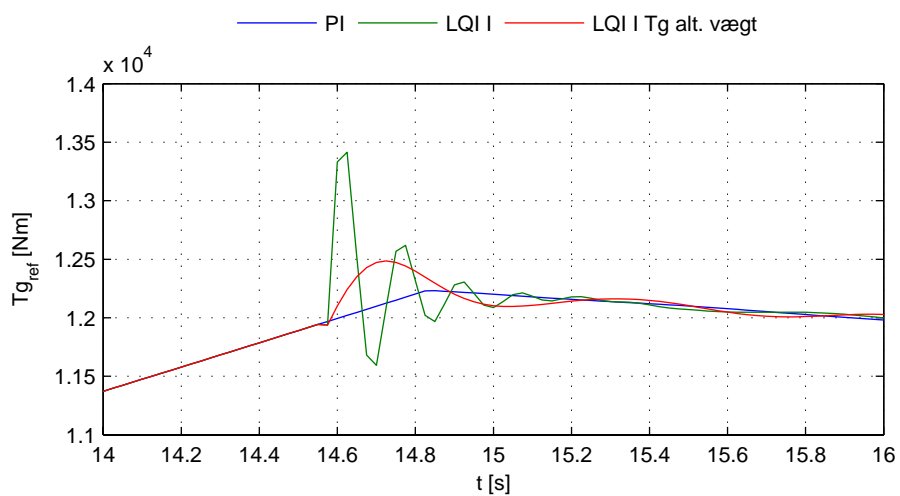
Figur 4.13: Skift fra $p\omega$ til topregulator, vind stiger som rampe.

indsving med overshoot, når den aktiveres. Ved at begrænse T_g styresignalet i regulator designet kan skiftet gøres glattere. På figur 4.15 er der vist, hvordan T_g ser ud i skiftet. Det er her tydeligt at se, at der ikke er tale om abrupt skift, når regulatoren skifter, men faktisk et indsving. Desuden ses det her, at den 'mildere' regulator, hvor T_g er vægtet mere (100 gange), giver mindre overshoot og dermed et glattere skifteforløb.

Det vil altså være muligt at designe et system med pænere skift ved at ændre på regulatoren i et tidsrum efter skiftet eller helt vælge et mildere design. Da det ikke er målet for dette projekt at designe en optimal Bumpless regulator, vil der ikke blive arbejdet videre med denne ide. Til vores formål virker den viste metode med de normale vægte fuldt ud tilstrækkeligt.



Figur 4.14: Regulatorskiftet sker ca. til tiden 14.6 s. Det ses at T_g ringer en del ved skiftet.



Figur 4.15: T_g styresignalet bliver kraftigt reguleret, når topregulatoren skiftes ind. Kan mindskes med andet regulator design.

KAPITEL 5

Test af kollektiv regulering

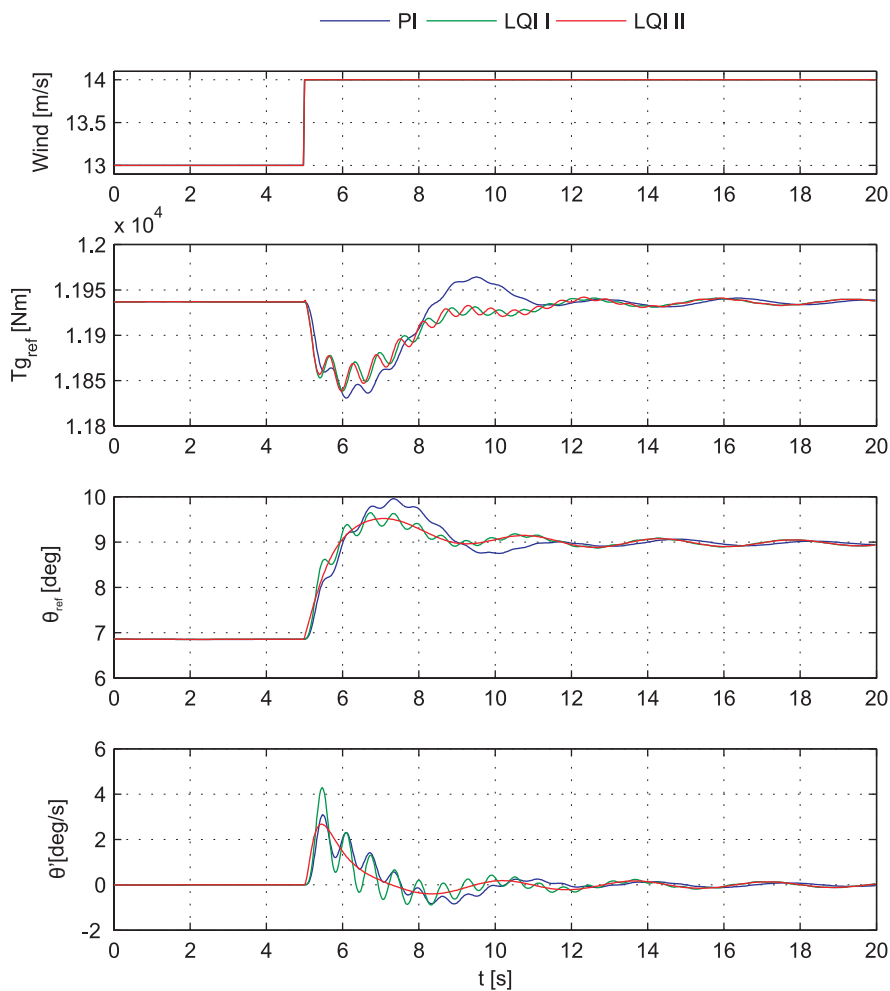
5.1 Testindhold

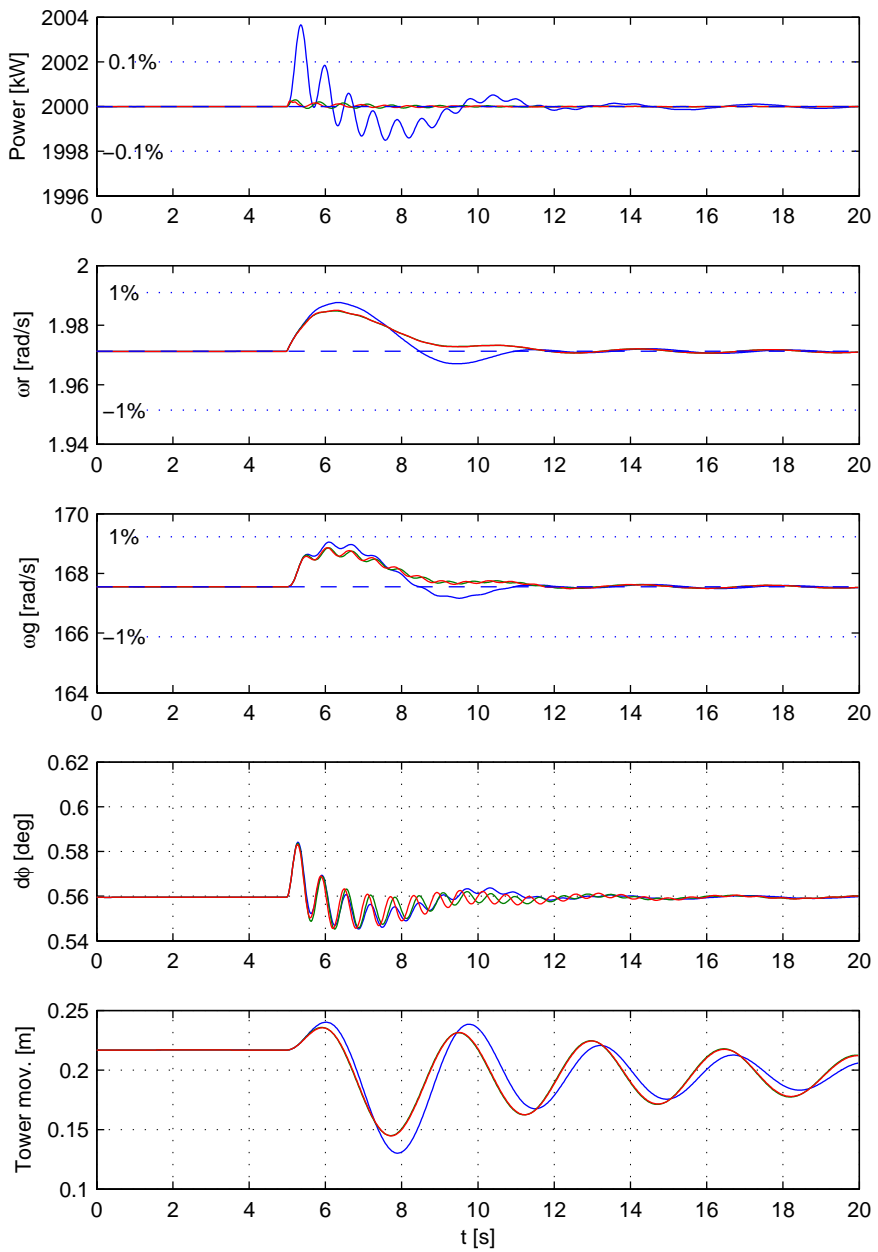
Test plot i dette kapitel:

- 1 Trin; PI, LQI I, LQI II i Simulink
- 2 Trin; PI (korrigeret koef) i Simulink og HAWC
- 3 Trin; LQI II Simulink vægtning af $d\phi$
- 4 Trinsekvens; PI, LQI I, LQI II i Simulink
- 5 Stokastisk vind TOP; PI, LQI I, LQI II i Simulink

5.1.1 PI, LQI I og LQI II

Test af simulink modellen med 3 regulatorer. Trinpåvirkning i vinden til tiden 5 sek. Skift fra 13-14 m/s .. Observer hvor stor forbedring de to LQI regulatorer har på effekt indsvinget. Indsvingtid og amplitude på ω_r er omtrent ens for de tre regulatorer. Det er desuden værd at bemærke LQI II regulatorens "glatte" indsving på θ , hvilket skyldes at drivakslens dynamik er medtaget i regulatoren.

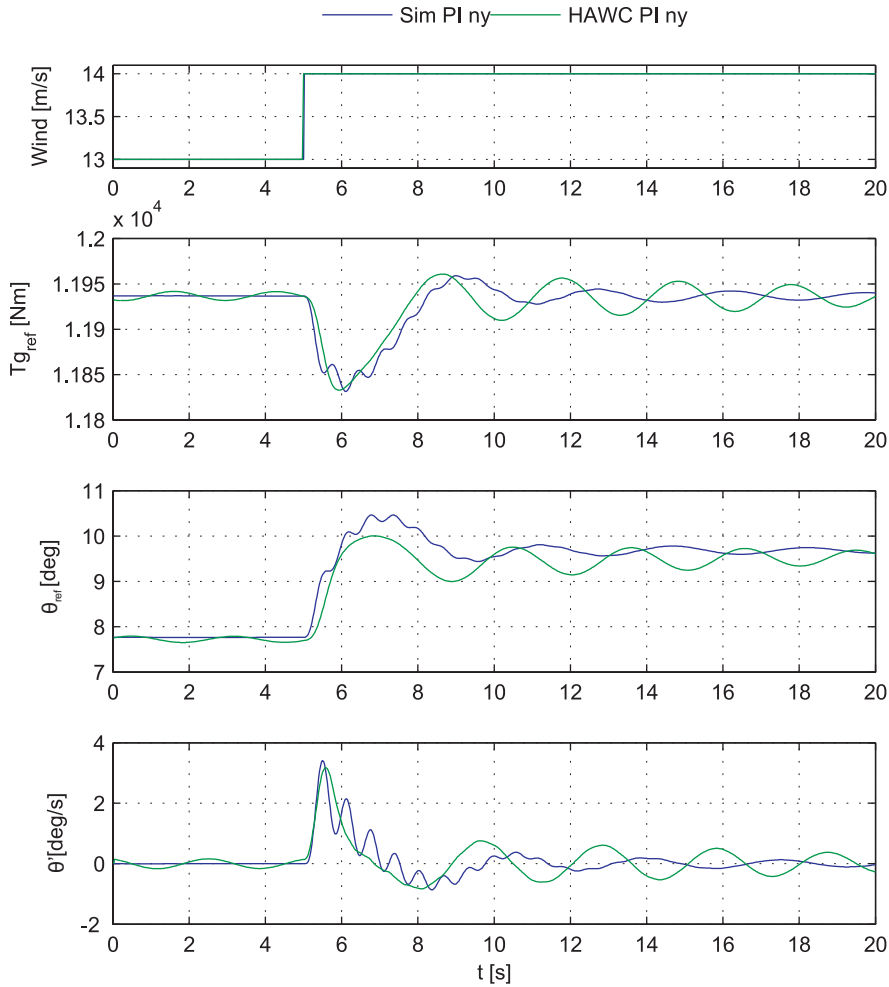


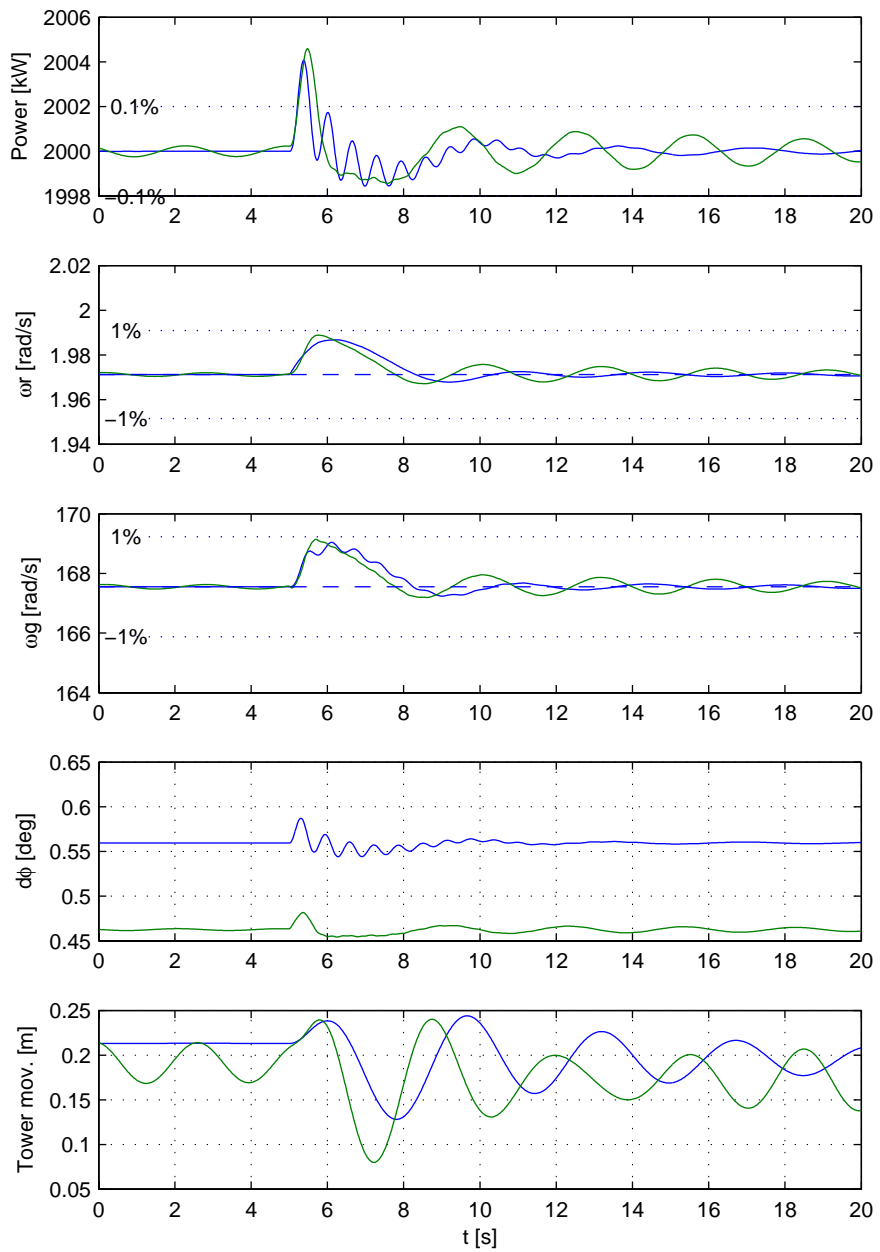


5.1.2 PI i Simulink og HAWC

Testen sammenligner et trinrespons simuleret i HAWC og med den opbyggede simulink model. Den PI regulator, der var designet i simulink, viste sig at være for voldsom til HAWC. Derfor er denne test lavet med en PI regulator med mindre forstærkninger. Regulatorcoeffcienterne er således rettet fra $K_p = 1.5, \tau_i = 1.2$ til $K_p = 1, \tau_i = 0.77$.

HAWC har haft 100 sek. til indsving inden det viste trin i vinden fra 13-14 m/s kommer. Der observeres en stationær afvigelse på akselvridet.



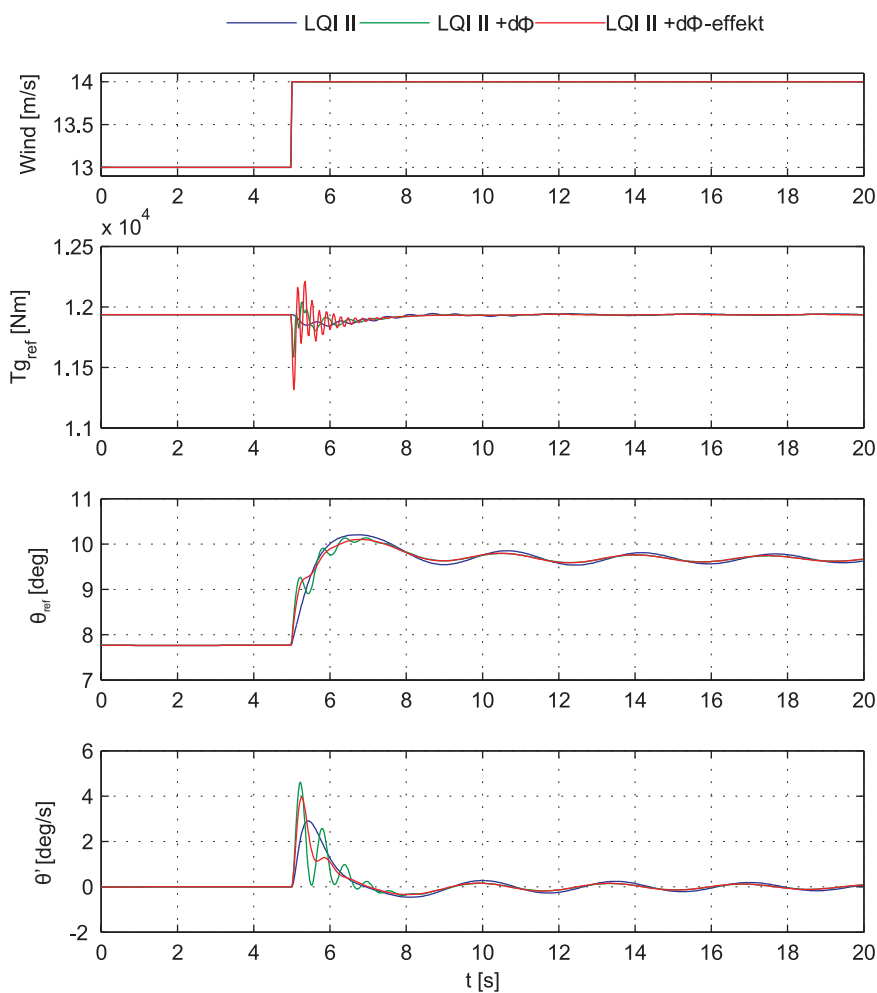


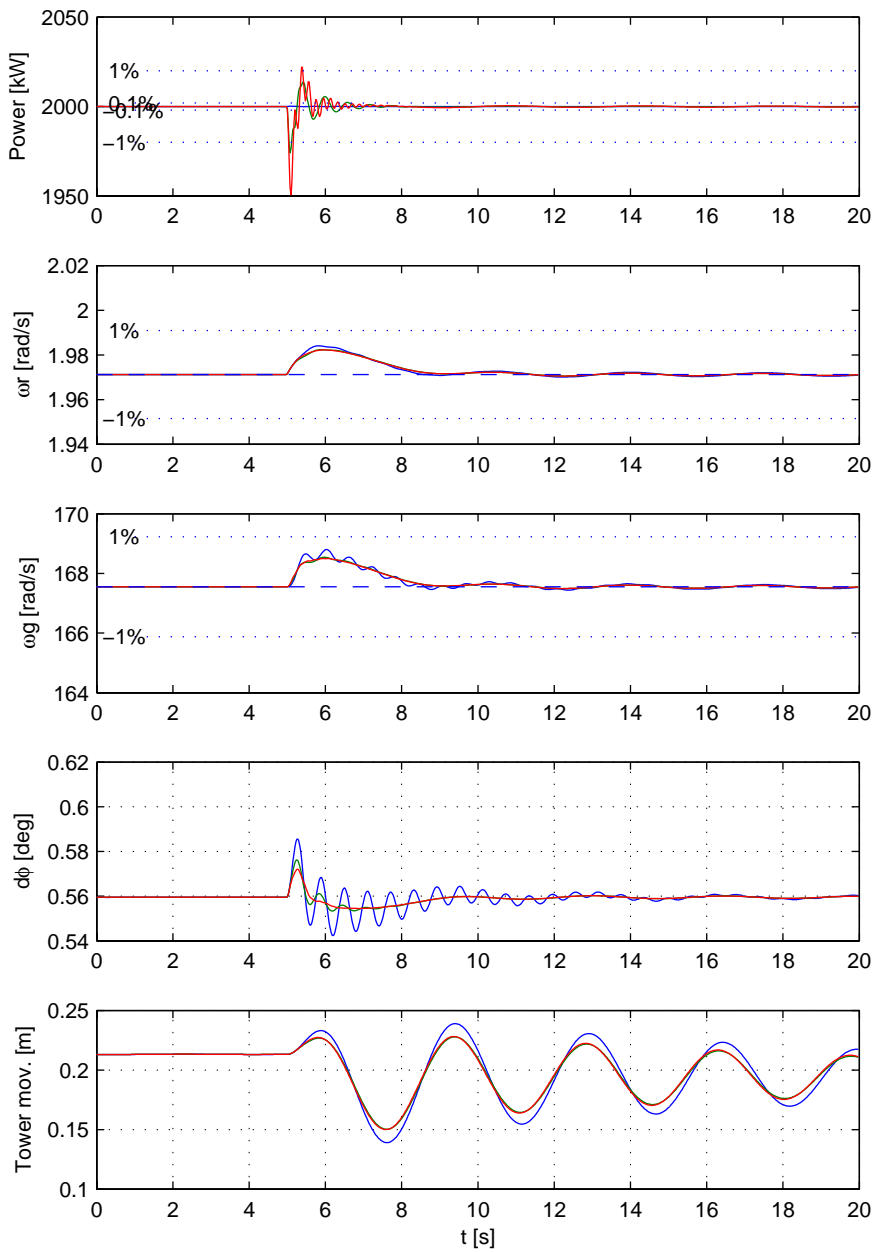
5.1.3 LQI vægtet med hensyn til akselvrid.

Ved at vægte $d\phi$ højere (fra 0-10) kan akselvridet mindskes som vist i testen. I den sidste test er vægten på effekten desuden reduceret (fra 3000-1000). Vægtmatricen med begge ændringer er vist i ligning 5.1.

$$\mathbf{R}_0(i, i) = \begin{bmatrix} \frac{0.15}{\theta_{nom}^2} & \frac{0.1}{T_{gnom}^2} & \frac{0}{\omega_{rnom}^2} & \frac{10}{\omega_{gnom}^2} & \frac{10}{\Delta\phi_{nom}^2} & \frac{100}{\omega_{nom}^2} & \frac{1000}{P_{nom}^2} \end{bmatrix} \quad (5.1)$$

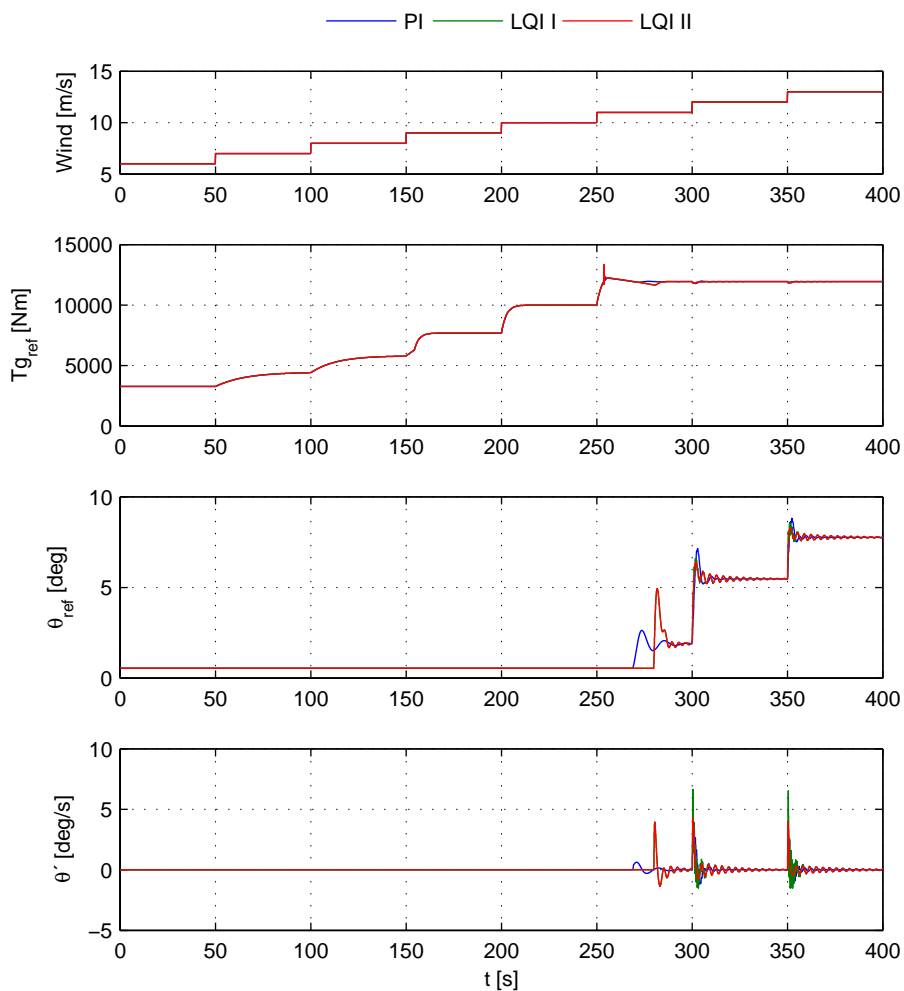
Resultatet er et roligere indsving af $d\phi$, og amplituden er reduceret kraftigt.

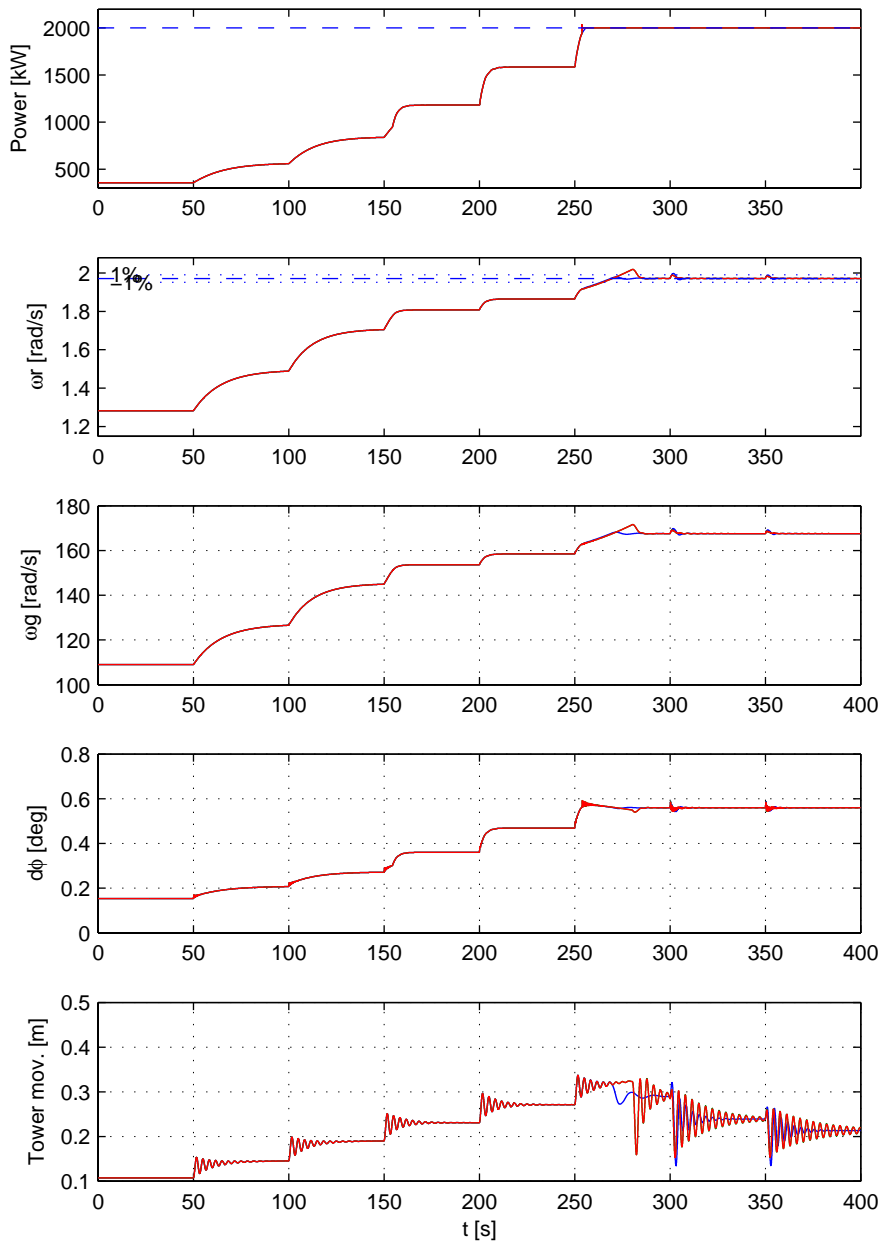




5.1.4 Trinsekvens fra 6-13 m/s

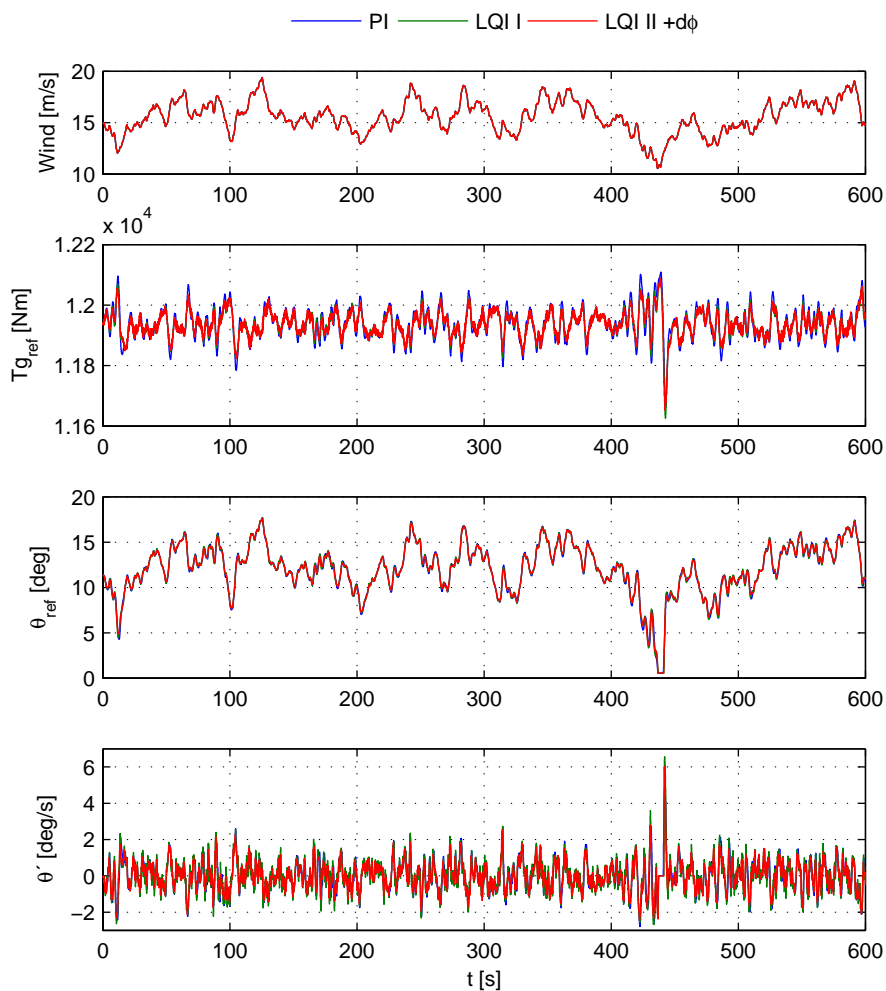
Af trinsekvensen ses det, hvordan møllen reagerer forskelligt i de forskellige områder, og hvordan den skifter mellem disse. De to første trin (6-7 og 7-8 m/s) sker med meget lange indsvingningstider. Årsagen er, at møllen arbejder i $p\omega$ området. Midt i det 3. trin bliver indsvinget hurtigere, da møllen nu er i highområdet. Der bemærkes et oversving, når LQI I og LQI II regulatorerne når op i topområdet. Dette kan, som beskrevet i kapitel 4.3, afhjælpes ved en anden vægtning af T_g , når topreguleringen skiftes ind.

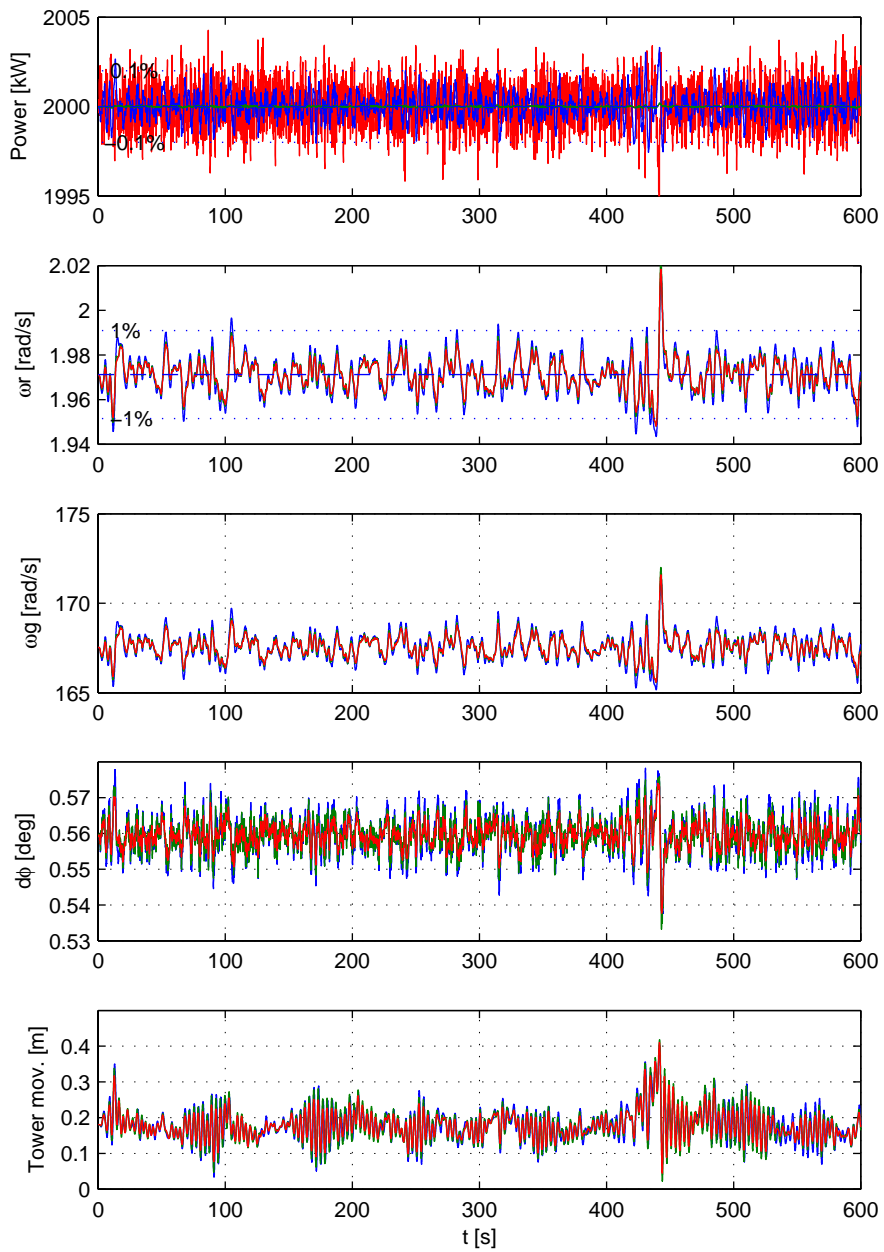




5.1.5 Test med stokastisk vind

Testen er lavet med stokastisk vind genereret ud fra den implementerede vind-model. Middelvinden er 15 m/s, turbulensintensiteten 13.55% genereret med seeden 75243. Testen sammenligner PI LQI I og LQI II med $\Delta\phi$ vægtet med 10. Det, der er værd at bemærke er, hvor godt LQI I regulatoren holder en konstant effekt. Derudover fremgår det, at vægtningen af $\Delta\phi$ har haft den forventede virkning. Akseludbøjningen er således reduceret og bekostningen er en større variation i effekten. De første 60 sek. af denne test er også printet i bilag C.





5.1.6 Testkommentarer

Den vigtigste kommentar at knytte til disse test er den gode overensstemmelse mellem HAWC og den opbyggede simulink model. Indsving på effekt og omdrejninger passer godt overens. Tårnsvingning og drivakslens udbøjning passer ikke helt så godt, specielt de stationære værdier kunne have passet bedre. Disse modellen kan med rette undersøges og bringes til bedre overensstemmelse. Desuden ses at belastningen på akslen reduceres ved at inkludere den i regulator designet.

KAPITEL 6

Modelopbygning individuel pitch

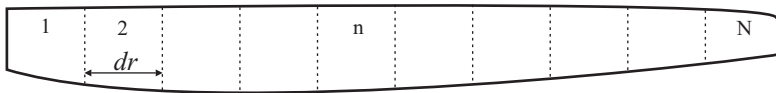
Den hidtil anvendte model har baseret alt aerodynamisk på tabelopslag, som udtrykker hele møllens egenskaber. Modellen har brugt to tabeller c_p og c_t til at finde den optagede effekt og den kraft, som rotoren påvirkes med. Denne model er anvendt med gode resultater også sammenholdt med HAWC, som er en langt mere kompliceret model.

Denne c_p/c_t baserede model kan dog ikke anvendes, når man begynder at kigge på individuel pitch, da den ikke betragter hvert blad individuelt. c_p/c_t tabellerne er beregnet for alle 3 blade i ét homogent vindfelt. En mulighed ville altså være at tage 1/3 af tabelværdierne og bruge dem på hvert blad. På den måde vil det være muligt at angive en pitch for hvert blad og beregne effekten og kraften for hver af dem. Begrænsningen ved dette er, at der kun kan bruges én vindhastighed for hvert blad. Det skal altså antages, at der er en middelvind for hvert blad, som så rammer bladet f.eks. 1/3 ude. Hvor brugbar denne metode er, er ikke blevet undersøgt, i stedet er der gået et skridt videre og kigget på de beregninger, der ligger til grund for c_p og c_t tabellerne.

6.1 BEM

Blade Element Momentum metoden eller bare BEM bruges til at beregne kræfterne på hvert blad og kan gøre det for forskellige vindforhold ud fra bladets længde. Fra referencemodellen HAWC er det muligt at finde alle de aerodynamiske og geometriske data, som er nødvendige for at kunne anvende BEM. Ud fra BEM metoden kan der desuden beregnes en ny c_p kurve, og ved at sammenligne den med den oplyst for møllen, kan det verificeres om beregningerne ser rigtige ud. BEM er en sammenfatning af flere forskellige teorier, som vil blive gennemgået i de følgende afsnit. Teknikken er vidt udbredt og derfor beskrevet i mange kilder hvoraf de vigtigste kan nævnes indledningsvis; [Manwell 02], [Risø 80], [Hansen 00] og [Ahlström 05].

Grundideen er at betragte bladet som en række elementer med en endelig længde dr , se figure 6.1. Hvert element kan påvirkes med en specifik vind, og kræfterne kan findes. Metoden antager, at hvert element betragtes individuelt og uden at have nogen indflydelse på de andre elementer. Desuden antages det, at kræfterne er konstante over hele bladelementets længde. BEM beregningerne kan lade

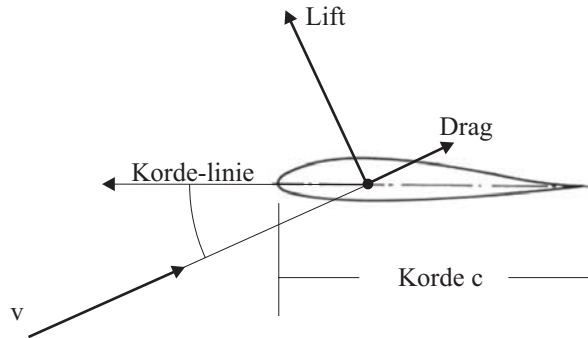


Figur 6.1: Bladet inddeles i N lige store bladelementer, hver med længden dr .

sig gøre, fordi der opstår en balance imellem de kræfter, der beregnes ud fra 2D aerodynamik, og en balance mellem aksialt og tangentielt moment i momentteorien [Ahlström 05]. For hvert bladelement ønskes det at bestemme kraften i rotationsretningen dF_x og vindretningen dF_y . Disse kræfter summeres op over hele bladets længde, og på den måde bestemmes den totale påvirkning, som vinden har på bladet og endeligt på rotoren.

6.1.1 Bladelementteori

Først betragtes bladets aerodynamiske forhold i 2 dimensioner. På figur 6.2 ses lift- og drag-kræfterne på bladet. Liftet er den ønskede kraft, der virker til at dreje rotoren rundt, mens drag er den kraft, der påvirker bladet direkte i vindretningen. Denne kraft ønskes selvfølgelig så lille som mulig, da den giver anledning til uønsket påvirkning af blade og tårn. Det aktuelle lift og drag for et bladelement beregnes ud fra udtrykkene i ligning 6.1 og 6.2. c_l og c_d er lift- og drag-koefficienterne og er konstanter specifikke for den anvendte bladtype.



Figur 6.2: 2D aerodynamik, som det kendes fra flyvinger med lift og drag.

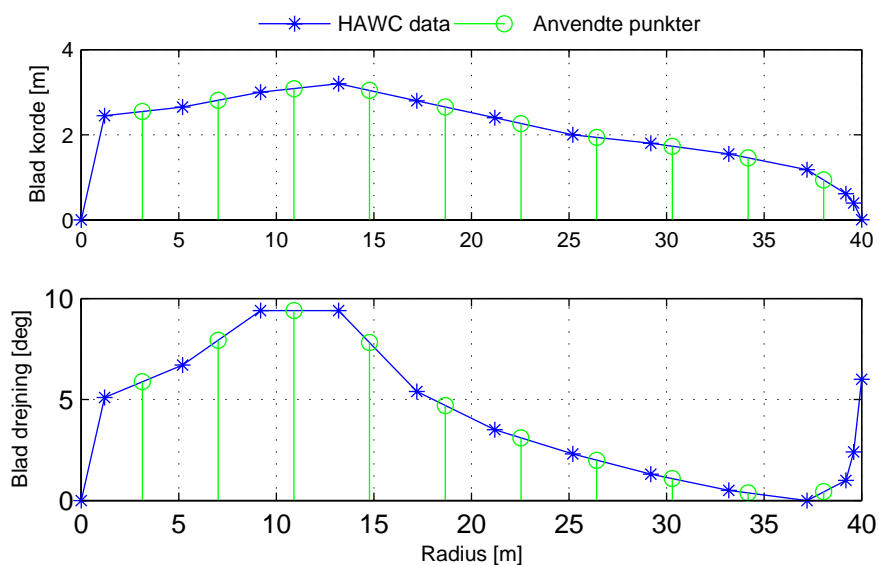
Begge er afhængige af bladets angrebsvinkel α , som er vinklen mellem bladets korde c og vindretningen, angivet med α i figur 6.2, hvor bladets korde c også er vist.

$$L = \frac{1}{2}\rho v^2 c c_l \quad (6.1)$$

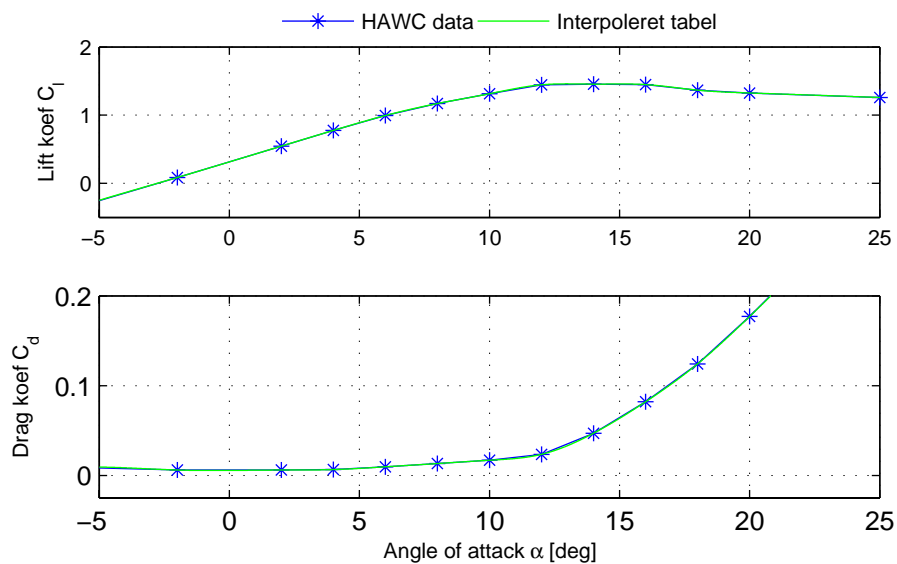
$$D = \frac{1}{2}\rho v^2 c c_d \quad (6.2)$$

Bladprofilen, der bruges på denne mølle, er af typen NACA 63415. Bladets lift- og drag-koefficienter ligger i HAWC_ pc.001, og bladets fysiske dimensioner; korde c og drejning θ_0 ligger i HAWC_ ae.001. Begge filer er vist i bilag F. Bladets drejning θ_0 , også kaldet smiget, svarer til et pitch-vinkel offset. Det er nødvendigt at interpolere disse værdier, da BEM modellen ikke bruger de samme bladelementinddelinger som HAWC, og da lift- og drag-koefficienterne skal bruges for vilkårlige α værdier. Bladets korde og drejning skifter ud langs bladets radius. På figur 6.3 er HAWC data samt de interpolerede værdier for bladets dimensioner vist.

Bladets lift- og drag-koefficienter er ligeledes interpoleret. Det skal nævnes, at den egentlige implementering ikke anvender en interpolation, men i stedet en meget tæt inddelt tabel med de interpolerede værdier. Grunden er, at netop α værdien bruges til beregninger i en indlejret løkke og derfor skal bruges mange gang. Matlabs egen interpolationsmetode `interp1.m` er meget langsom, så i stedet for at kalde denne inde i løkken er det implementeret som et tabelopslag med en tilstrækkelig præcision. Den anvendte tabel er plottet i figure 6.4 sammen med de værdier, der er opgivet i HAWC datafilen, se bliag F.

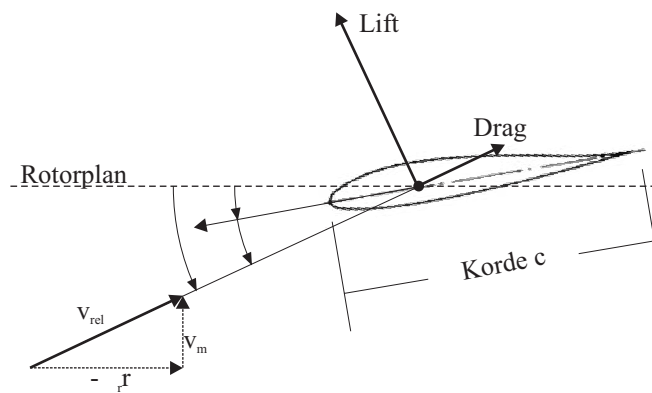


Figur 6.3: Bladkorde c og bladdrejning θ_0 for det anvendte blad.



Figur 6.4: Lift- og drag-koefficienter for NACA 63415 bladprofil, som anvendes på møllen.

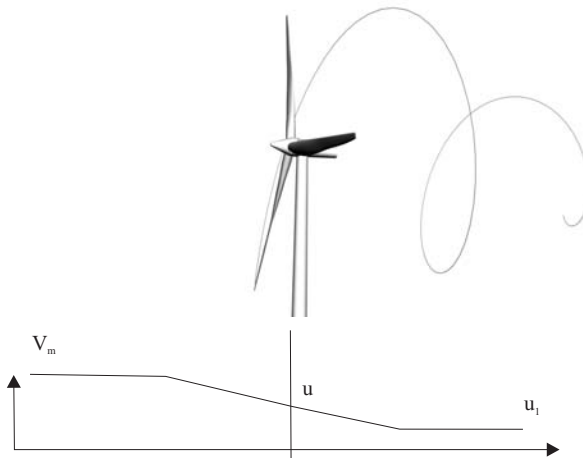
Bladet på møllen bevæger sig i en roterende bevægelse og vil derfor påvirke de lokale vindhastigheder alt efter, hvor langt fra centrum et bladelement befinder sig. Da mølletypen er en såkaldt hurtigløber, betyder det, at rotorhastigheden ved tippen er meget større end den indkomne vind. Det betyder, at den effektive vindretning, som hvert bladelement ser, er afhængig af den lokale radius r og omdrejningshastighed ω_r . Dette er illustreret i figur 6.5, hvor den effektive vindretning er betegnet v_{rel} . Vinklen ϕ er altså den samlede vinkel mellem rotorplanet og den vindretning, som bladet ser v_{rel} .



Figur 6.5: Angrebsvinklen α er vinklen mellem v_{rel} og bladets korde.

6.1.2 Momentteori

Bladets profil ligger, som beskrevet, til grund for de aerodynamiske kræfter, og når bladet gives en roterende bevægelse, ændres de lokale vindretninger og dermed den effektive angrebsvinkel. Dette er ikke så svært at forholde sig til, men det, der yderligere sker, er, at bladene påvirker vinden både foran og bagved møllen, som igen påvirker bladets effektive angrebsvinkel α . Dette er hvad momentteorien beskriver. Når luften strømmer omkring møllen, bliver den bremsset og derved optages energien fra vinden. Jo større opbremsning, der er af vinden, jo mere kinetisk energi er der taget ud af vinden og overført til møllen. Møllens effekt kan derfor udtrykkes ud fra vindhastighederne v.h.a. ligning 6.3 [Hansen 00]. v_m er vinden langt foran møllen, u er vinden i mølleplanet og u_1 vindhastigheden langt bag møllen. Se også figur 6.6, som viser vindhastighederne foran, omkring og bag møllen.



Figur 6.6: Vindhastigheden aftager allerede foran møllen og fortsætter med at falde efter.

$$P_r = \frac{1}{2} \rho u (v_m^2 - u_1^2) R^2 \pi \quad (6.3)$$

Opbremsningen af vinden sker altså ikke brat i det øjeblik luften rammer bladene, men starter faktisk foran mølleplanet, hvilket er illustreret på kurven nederst på figur 6.6. Netop det at vinden, der rammer bladet, ikke har hastigheden v_m , men en lavere hastighed u , se ligning 6.4, har indvirkning på bladets angrebsvinkel α , da v_{rel} bliver en anden. Den faktor, som vinden er bremsset med, når

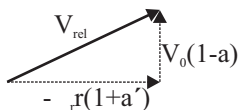
den rammer mølleplanet, kaldes for den aksiale induktionsfaktor a .

$$\begin{aligned} u &= (1 - a)v_m \\ a &= \frac{v_m - u}{v_m} \end{aligned} \quad (6.4)$$

Udover opbremsning af vinden er der et andet fænomen, der har betydning og som også er indeholdt i momentteorien. Da møllen ikke er ideel, vil den afgive en del af sin roterende bevægelsesenergi til vinden. Det betyder, at vinden bag møllen vil have en roterende bevægelse modsatrettet rotorens omdrejninger. Vinden vil altså bevæge sig i en spiralform bag møllen. Dette fænomen betyder, at vinden i rotorplanet, som rammer bladene, vil have en hastighed u_t modsatrettet bladenes omløbsretning ω_r . Altså en hastighed tangentielt med vindretningen. Størrelsen af denne hastighed udtrykkes ved den tangentielle induktionsfaktor a' , udtrykt som i ligning 6.5. Det kan nævnes, at den tangentielle hastighed u_t er 0 lige foran bladet og $2\omega_r r a'$ lige bag bladet. Altså ikke aftagende på samme langsomme måde, som vindens hastighed [Risø 80].

$$\begin{aligned} u_t &= \omega_r r a' \\ a' &= \frac{u_t}{\omega_r r} \end{aligned} \quad (6.5)$$

Med disse to nye størrelser a og a' inde i billedet er det nødvendigt igen at se på, hvordan den relative vind v_{rel} bliver for hvert bladelement. Dette er illustreret i figur 6.7.



Figur 6.7: v_{rel} ændres pga. induktionsfaktorerne a og a' .

Den endelige vinkel mellem rotorplanet og den relative vind kan således bestemmes af ligning 6.6.

$$\phi = \text{atan} \left(\frac{(1 - a)v_m}{(1 + a')\omega_r} \right) \quad (6.6)$$

Der huskes på, at lift- og drag-kræfterne virker parallelt og tangentielt på vindretningen v_{rel} . Når nu den relative vind ikke er vinkelret på rotorens plan, er det derfor nødvendigt at omregne disse kræfter via cosinus og sinus. Dette sker ud fra ligning 6.7 og 6.8.

$$c_y = c_l \cos(\phi) + c_d \sin(\phi) \quad (6.7)$$

$$c_x = c_l \sin(\phi) - c_d \cos(\phi) \quad (6.8)$$

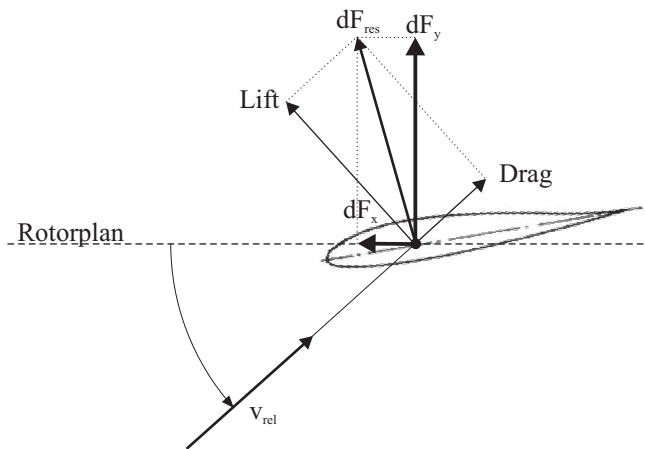
For hvert bladelement kan kraften i vindretningen dF_y og i rotorplanet dF_x således findes ud fra ligning 6.9 og 6.10. Drejningsmomentet på akslen fra ét

bladelement dQ bestemmes ud fra kraften dF_x multipliceret med bladelementets radius, ligning 6.11. Kræfterne er illustreret på figur 6.8.

$$dF_y = \frac{1}{2} \rho v_{rel}^2 c_y c dr \quad (6.9)$$

$$dF_x = \frac{1}{2} \rho v_{rel}^2 c_x c dr \quad (6.10)$$

$$dQ = F_x r \quad (6.11)$$



Figur 6.8: Kræfterne dF_x og dF_y i rotorplanet findes ud fra lift og drag.

For at finde kræfterne, der påvirker hele rotoren, skal alle bladelementernes kræfter summeres. Hvis b angiver bladnummeret og n bladelementnummeret, kan det totale drejningsmoment bestemmes af ligning 6.12 og kraften i vindretningen v_m af ligning 6.13

$$Q_t = \sum_1^B \sum_1^N dQ_{(b,n)} \quad (6.12)$$

$$F_t = \sum_1^B \sum_1^N dF_{y(b,n)} \quad (6.13)$$

Det er disse kræfter, der er interessante. Ud fra dem kan modellen nemlig opbygges. Der opstår dog et problem. For at kunne beregne a og a' værdierne er det nødvendigt at vide, hvor meget effekt der hives ud af vinden og dermed, hvordan den opbremses og afbøjes. Men for at vide, hvor meget effekt der optages, skal a og a' kendes! Dette bider altså sig selv i halen.

De to induktionsfaktorer kan, udover ligning 6.4 og 6.5, beregnes som funktioner af c_x og c_y [Risø 80] og [Hansen 00], som ses i ligning 6.14 og 6.15.

$$a = 1 / \left(\frac{4 \sin^2(\phi)}{\sigma c_y} + 1 \right) \quad (6.14)$$

$$a' = 1 / \left(\frac{4 \sin(\phi) \cos(\phi)}{\sigma c_x} - 1 \right) \quad (6.15)$$

Hvor σ er defineret som ligning 6.16 og er et udtryk for den brøkdæl af det totale vindfelt, som rammer i en given radius.

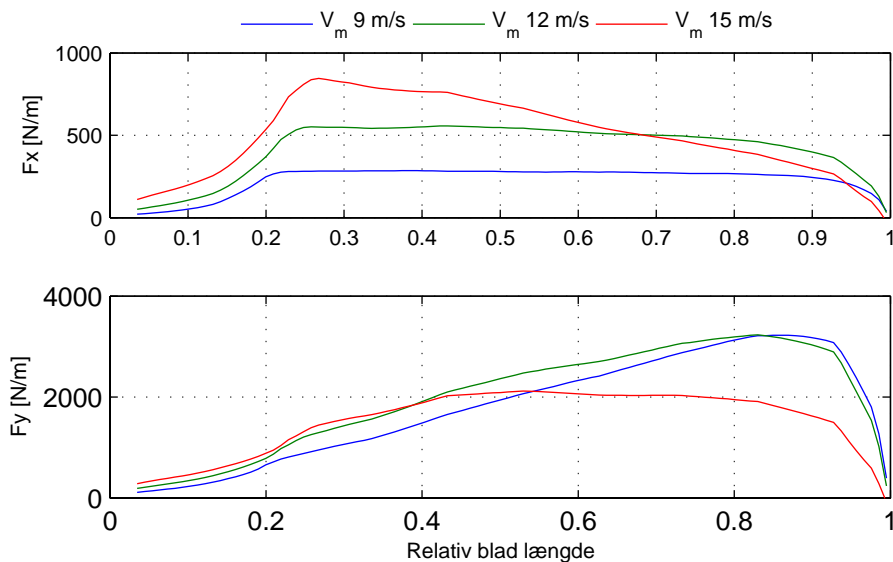
$$\sigma = \frac{cB}{2\pi r} \quad (6.16)$$

Ved at gætte på et sæt induktionsfaktorer kan man iterativt løse dette problem og finde induktionsfaktorer og deraf også kræfterne på bladet. Denne proces er beskrevet i [Risø 80], [Hansen 00], [Ahlström 05] og [Manwell 02] og kan opremses således:

1. Trin: Initialiser a og a' , evt. til $a = a' = 0$
2. Trin: Beregn vinklen mellem bladkorde og relativ vind ϕ v.h.a. ligning 6.6
3. Trin: Beregn lokale angrebsvinkel $\alpha = \phi - \theta$
4. Trin: Aflæs lift- og drag-koefficienter c_l og c_d fra tabelværdier.
5. Trin: Beregn nye a og a' værdier ud fra ligning 6.14 og 6.15.
6. Trin: Hvis a og a' har ændret sig mere end én tolerance, gå til trin 2.
7. Trin: Beregn kræfterne for det enkelte bladelement.

For at verificere om disse beregninger er korrekte, er bladets kræfter i første omgang sammenlignet for tre forskellige vindforhold. Hvis bladets aerodynamik er beregnet korrekt, vil det for moderate vindforhold give en kraft F_x , som er nogenlunde jævnt fordelt over hele bladets længde. Kraftfordelingen i F_x og F_y er vist for 9, 12 og 15 m/s i figur 6.9. Her ses en overensstemmelse med det forventede, og sammenholdes kraftfordelingen med andre datablade f.eks. fra [Ahlström 05], ser det også korrekt ud.

Det er disse beregninger, der gennemføres i kald fra simulink. For hvert blad kendes der 10 vindhastigheder, som giver 10 sæt induktionsfaktorer og kræfter. Til sidst summeres kræfterne F_y og dQ over bladet og returneres til simulink. Implementeringen ligger i BEM.m som ses i bilag E.



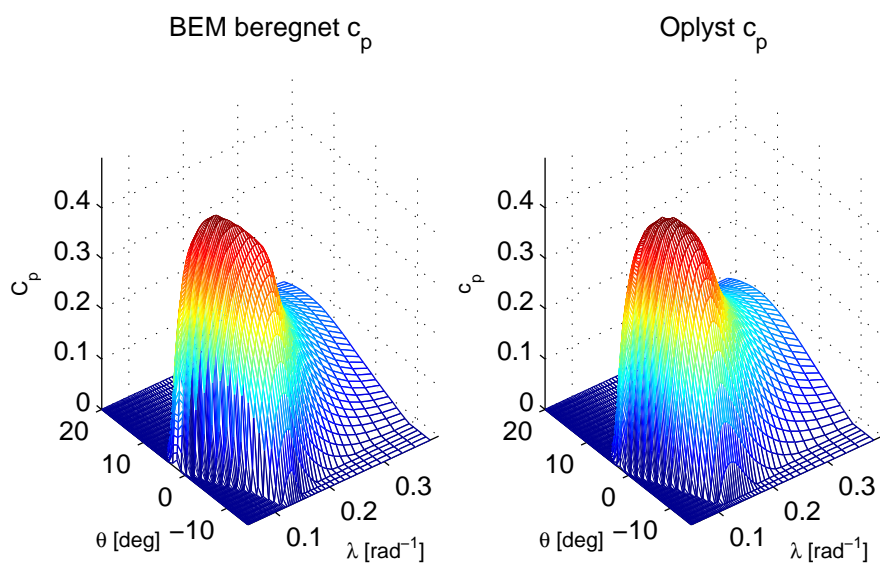
Figur 6.9: Kraftfordelingen over bladets længde ved forskellige vindhastigheder.

6.1.3 BEM beregnet c_p og c_t

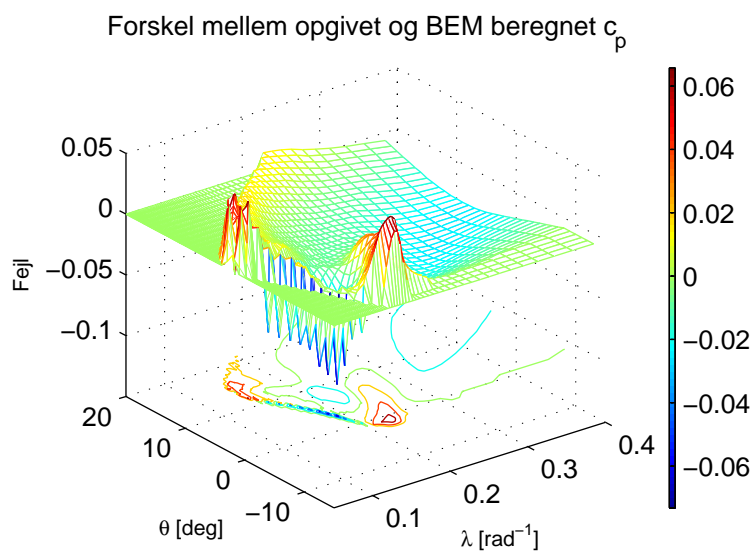
Ved at beregne den optagede effekt for variationer af λ og θ kan en ny c_p tabel beregnes v.h.a. BEM. Dette gøres for at kunne verificere, om beregningerne passer i alle de forskellige forhold, som bladene udsættes for. Det er desuden muligt at beregne en c_t tabel. Denne er ikke oplyst for den aktuelle mølle og er derfor også meget interessant. I figur 6.10 ses til venstre c_p kurven beregnet med den implementerede BEM metode, til højre den oplyste for møllen. Der er stor overensstemmelse mellem de to kurver. I tabel 6.1 ses nøgletallene for de to c_p kurver. Den optimale pitch-vinkel θ er den, der afviger mest fra den oplyste værdi. Dette passer godt med, hvad der ses på figur 6.10, hvor det kan ses, at toppen på kurven er bredere i θ -aksen for den BEM beregnede end den oplyste.

	BEM	Oplyst
c_{pnom}	0.4787	0.4714
λ_{nom}	0.1117	0.1235
θ_{nom}	0.7660	-0.5329

Tabel 6.1: Tal, der beskriver c_p kurven



Figur 6.10: God overensstemmelse mellem BEM c_p og den oplyste.



Figur 6.11: Fejlen er størst for turbulente forhold ved lave λ og negativ θ .

De to kurver er meget svære at sammenligne ved blot at se på dem. Derfor er fejlen mellem de to vist i figur 6.11. På fejlfladen bemærkes specielt, at de

største fejl ligger omkring en skrå linje ved lave λ og θ værdier. Netop i dette område begynder der at opstå turbulens langs bladet, og det er derfor sværere at beregne de nøjagtige kræfter her.

6.1.4 Korrektionsfaktorer

Der gøres mange antagelser for at få BEM til at fungere. Nogle af disse kræver, at der korrigeres på forskellig vis. Der er to korrektionsfaktorer, som er anvendt i BEM beregningerne, men som endnu ikke er beskrevet. Det drejer sig om tiploss og korrektion ved turbulente forhold. Tiplosskorrektionen er indført for at tage højde for det tab, der vil være for enden af hvert blad. Her vil luften bevæge sig over bladets spids fra tryksiden til vakuumsiden og derved introducere et tab [Manwell 05]. Faktoren modelleres ofte med Prandtl's korrektion F , som antager, at faktoren er 0 ved bladets tip og asymptotisk bliver 1 ind mod rotorens midte. Prandtl's tiploss er vist i ligning 6.17 og 6.18

$$F = (2/\pi)\cos^{-1}(e^{-f_{ti}}) \quad (6.17)$$

hvor :

$$f_{ti} = -\frac{N_b}{2} \left(1 - \frac{r}{R}\right) \frac{1}{\sin(\phi_r)} \quad (6.18)$$

Tiplosskorrektionsfaktoren har indvirkning på de kræfter, som findes for hvert bladelement. Derfor indføres den, så ligning 6.9 og 6.10 kommer til at se ud som ligning 6.19 og 6.20.

$$dF_y = F \frac{1}{2} \rho v_{rel}^2 c_y c dr \quad (6.19)$$

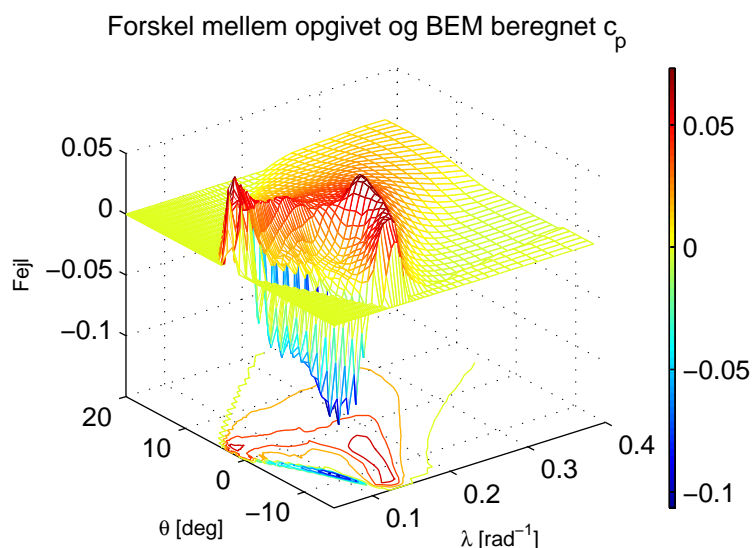
$$dF_x = F \frac{1}{2} \rho v_{rel}^2 c_x c dr \quad (6.20)$$

Ligeledes ændres udtrykkene, der bruges til at beregne de to induktionsfaktorer a og a' , så udtrykkene i ligning 6.14 og 6.15 bliver erstattet med ligning 6.21 og 6.22.

$$a = 1 / \left(\frac{4F \sin^2(\phi)}{\sigma c_y} + 1 \right) \quad (6.21)$$

$$a' = 1 / \left(\frac{4F \sin(\phi) \cos(\phi)}{\sigma c_x} - 1 \right) \quad (6.22)$$

For at illustrere effekten af tiploss korrektionen er der i figur 6.12 vist fejlen mellem BEM og den oplyste c_p kurve, når tiploss ignoreres. Det er tydeligt at se, at dette har en indvirkning på den opnåede effektivitet i et meget stort område. Det kan oplyses, at c_{pnom} uden tiploss er 0.5206. De to andre nøgleværdier har



Figur 6.12: Fejlen bliver forværret, når tiploss korrektionen fjernes.

ikke ændret sig nævneværdigt.

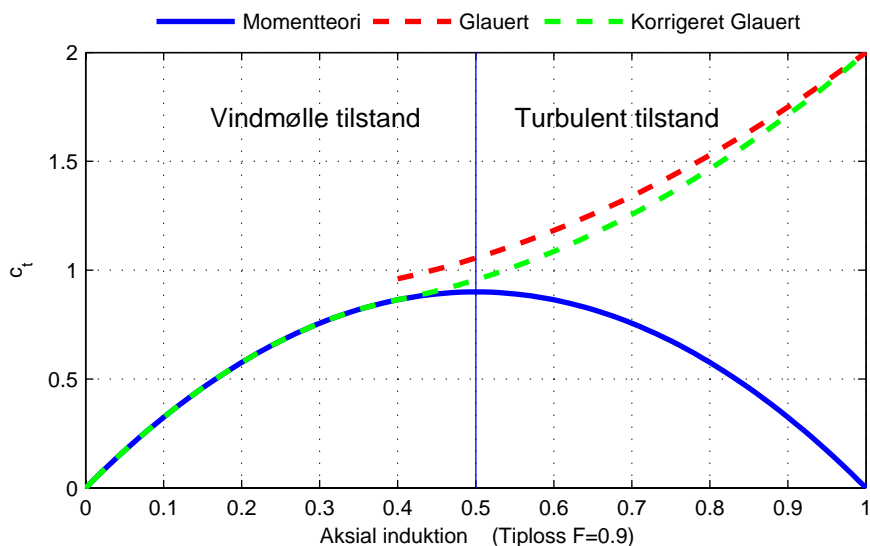
Udover tiploss er det nødvendigt at tage højde for, at der under visse forhold opstår turbulens langs bladet. Når dette sker, kan de almindelige momentteori-betragtninger ikke længere anvendes. Det siges, at møllen arbejder i en turbulent tilstand. Den turbulente tilstand opstår i det tilfælde, at induktionsfaktoren a bliver over 0.5. Når dette sker, betyder det, som det kan ses af figur 6.6, at vinden langt bag ved møllen må stoppe helt op. Dette giver forståeligt nok et problem. Problemet er, at induktionsfaktoren kan blive helt op til 1, uden at det er forkert, men dette kan ikke beskrives med momentteori. Det er a 's indvirkning på c_t koefficienten, der ikke passer længere. For den almindelige momentteori beregnes c_t ud fra ligning 6.23

$$c_t = 4a(1 - a) \quad (6.23)$$

Sammenhæng mellem a og c_t er vist med den fuldtoptrukne blå linje i figur 6.13, hvor det ses, at den fra 0.5 igen falder mod 0, hvilket ikke er korrekt. Glauert's metode, som ofte anvendes, er lavet ud fra målinger på forskellige møller i turbulent tilstand. Det viser sig, at c_t går imod 2, når a går mod 1 på en måde, der passer godt med en parabel. Glauert's parabel er vist i figur 6.13 og givet ved ligning 6.24.

$$c_t = 0.889 - \frac{0.0202 - (a - 0.143)^2}{0.6247} \quad (6.24)$$

Der kan dog opstå nogle numeriske kvaler, når BEM beregningerne løses ite-



Figur 6.13: Almindelig momentteori kan ikke beskrive den turbulente tilstand. Bemærk tiploss $F=0.9$.

rativt, og der er taget tiploss med i beregningerne. Glauert's parabel er nemlig lavet så den netop tangerer den almindelige momentteori i punktet 0.4. Dette sker bare ikke, hvis der medregnes tiploss, så ligger c_t længere nede, og den iterative BEM proces får problemer med at konvergere. Problemet kan tydeligt ses på figur 6.13, hvor den røde stiplede linje ikke rammer i 0.4, fordi der er indført tiploss på 0.9. Dette problem er indgående beskrevet i [Marshall 05], hvor udtrykket erstattes med det viste i ligning 6.25. En parabel, der ganske som Glauert's tangerer i 0.4 og slutter i 2, men som medtager tiploss faktorens, indvirker på c_t . Denne er også illustreret i figur 6.13.

$$c_t = \frac{8}{9} + \left(4F - \frac{40}{9}\right)a + \left(\frac{50}{9} - 4F\right)a^2 \quad (6.25)$$

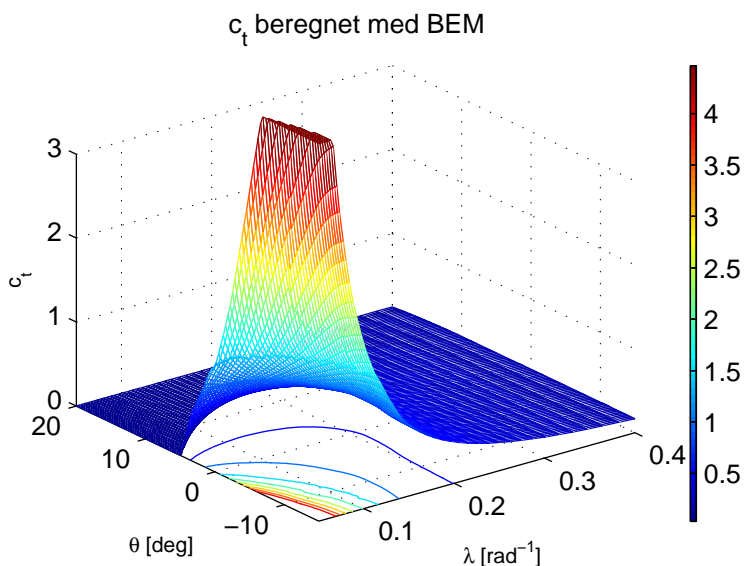
Når Glauert's metode anvendes, betyder det altså, at den nye ligning til at beregne a , ligning 6.21, kun kan bruges for a værdier under 0.4. Over 0.4 skal der findes et nyt udtryk baseret på den nye c_t beregning. Dette udtryk [Hansen 00] findes ved at sætte ligning 6.25 lig udtrykket for c_t fra bladelementteorien, som det ses i ligning 6.26

$$\frac{(1-a)^2 \sigma c_n}{\sin^2 \phi} = \frac{8}{9} + \left(4F - \frac{40}{9}\right)a + \left(\frac{50}{9} - 4F\right)a^2 \quad (6.26)$$

Her skal a isoleres, hvilket er løst symbolsk i MatLab.

Indtil nu er det kun c_p kurven, der er blevet kigget på, men via BEM kan en c_t

kurve også beregnes. Kurven er vist i figur 6.14 og er et udtryk for, hvor stor en kraft der påvirker møllen i vindretningen, ganske som c_p er den plottet for variationer af θ og λ . Da der ikke er en referencekurve at sammenligne med, er det svært at verificere korrektheden af c_t værdierne. Det observeres, at c_t stiger til over 2 i det turbulente område, og at skiftet hertil fra det ikke turbulente sker glat.



Figur 6.14: .

6.1.5 Implementering af BEM

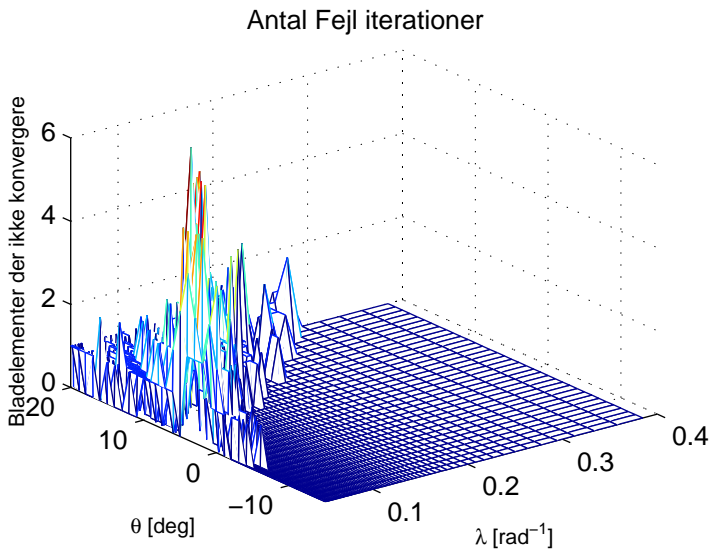
BEM beregningerne er implementeret i matlab funktionen BEM.m, som ses i bilag E, kaldes fra simulink med: bladets aktuelle azimuthvinkel, vinden for de 10 bladelementer, pitch-vinklen, ω_r og den aktuelle simulationstid. En vigtig parameter for implementeringen er, at beregningerne ikke må være for beregningstunge, da de foretages 3 gange for hver simulink sample. Det er allerede nævnt, at der derfor er brugt tabelopslag i indlejrede løkker i stedet for interpolation. Andre faktorer, der betyder meget for beregningshastighed, er antallet af iterationer, der foretages, hvilket kan styres ved at justere tolerancen for, hvornår den iterative proces stoppes. Antallet af bladelementer pr. blad N har også stor betydning og er i BEM.m sat til 10, hvilket ikke er meget, men efter forsøg viste det sig tilstrækkeligt. Antallet kan nemt varieres, og til beregning

af bladkræfterne i figur 6.9 blev der brugt 100 inddelinger.

Når der også er lagt stor vægt på, at BEM beregningerne skal konvergere uden for det pitch-område, der betragtes som normalt, skyldes det, at der med et stokastisk vindfelt på rotoren vil opstå mange lokale vindforhold, som kan give angrebsvinkler, som er uden for det normalt anvendte område. Der er dog stadig beregninger, der ikke konvergerer, og derfor er der i BEM.m lavet en log over disse. Loggen bliver skrevet ud, når simulationen er færdig. På den måde er det muligt at se data for de bladelementer, der ikke konvergerede.

For beregningerne af c_p og c_t , som vist i de foregående sider, er der også beregninger, som ikke konvergerer. For at give et overblik over hvor mange og under hvilke forhold, det sker, vises i figur 6.15 antallet af bladelementer, som ikke konvergerer. Der er 10 elementer på et helt blad.

De beregninger, som her ikke konvergerer, ligger alle i et område hvor c_p er nul eller meget tæt herpå.



Figur 6.15: .

KAPITEL 7

3D vindmodel

Den tidligere beskrevne vindmodel i kapitel 2 beskriver et ækvivalent vindfelt over rotoren. Den vil derfor ikke kunne bruges til at simulere et 3D vindfelt, som er nødvendigt.

Der er dog en deterministisk del af vinden, der er meget karakteristisk, og som kan beskrive meget af det, som hvert blad udsættes for. Den stokastiske del er langt mere kompliceret at beskrive. I afsnittet her beskrives den anvendte vindmodel.

7.1 Deterministisk vindmodel

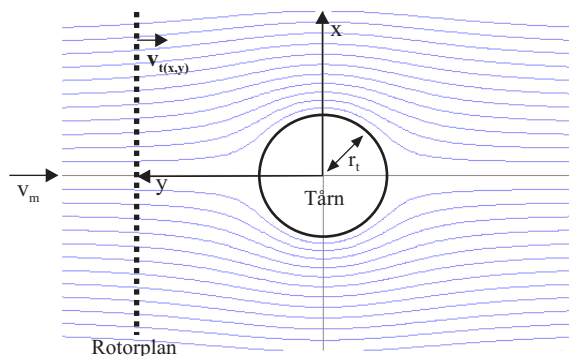
Vinden, som rammer møllens rotor, har nogle deterministiske egenskaber, som er vigtige, når vindindfaldet på hvert blad betragtes. Der er to fænomener, som betragtes: vindshear og tårnskygge. Vindshear er et produkt af, at jordens overflade ikke er helt glat, og derfor vil vinden bevæge sig langsommere tættere ved jordens overflade. Tårnskygge er, som det ligger i ordet, tårnets indvirkning på den vind, der rammer hvert blad.

7.1.1 Tårnskygge

Først og fremmest skal tårnets indvirkning på vinden betragtes. Tårnet på denne mølle er et cylinderformet tårn med varierende diameter. Andre design benytter gitteropbygninger, som har en mindre indvirkning på vinden. Årsagen til at vælge cylinderløsningen fra kan være på grund af udseendet eller møllens store masse. Selvom rotoren er placeret foran tårnet, har det stadig en meget stor indvirkning på den vind, som rammer bladene. Fordi denne påvirkning kun opstår, når bladene passerer tårnet, giver det en meget ujævn belastning på møllen. Da den yderligere er meget kortvarig, er den desuden også sværere at minimere. Tårnskyggen kan modelleres ud fra potentiel flowteori omkring en uendelig lang cylinder [Sørensen 02] og [Rosas 03]. Modellen, som her er anvendt, er beskrevet i ligning 7.1

$$v_t = v_m r_t^2 \frac{x^2 - y^2}{(x^2 + y^2)^2} \quad (7.1)$$

v_t er vinden i y retningen i et vilkårligt punkt (x,y) omkring tårnet. Hvor r_t er tårnets radius. Disse størrelser er illustreret på figur 7.1 som viser et vandret snit af tårnet for en given højde. Det skal bemærkes, at modellen af møllen ikke inkluderer rotorens tilt, som ellers er 6° . Dette betyder, at afstanden fra rotorplanet til tårnets centrum y er konstant for alle højder, nemlig 3.48 meter. Vinden i det beregnede punkt vil ikke kun have en retning i y retningen, den vil være afbøjet og derfor også indeholde en x komponent. Denne ses der bort fra, da det antages, at bidraget til rotormomentet vil være minimalt.



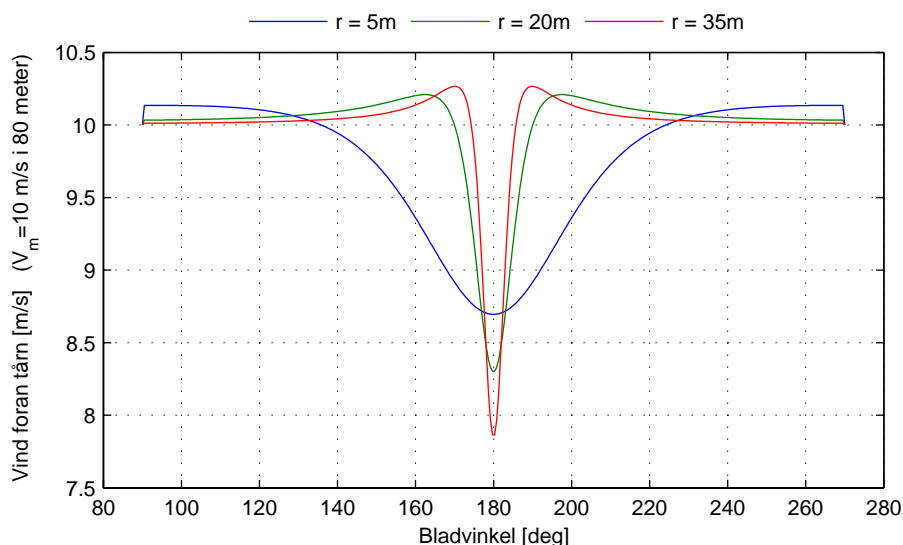
Figur 7.1: Flow omkring tårnet, x-y akse for én højde.

Tårnets radius r_t varierer derimod alt afhængig af højden. I tårnets top er radius 1.2 meter og ved roden 2.15 meter. Det antages, at tårnets radius ændres lineært med højden, og kan derfor til en vilkårlig højde bestemmes ud fra ligning 7.2.

$$r_t(h) = \left(\frac{r_{top} - r_{rod}}{H} h \right) + r_{rod} \quad (7.2)$$

Det er ud fra ligning 7.1 og 7.2 muligt at beregne tårnskyggens påvirkning af vinden i y retningen til et vilkårligt punkt foran eller omkring møllen i en vilkårlig højde.

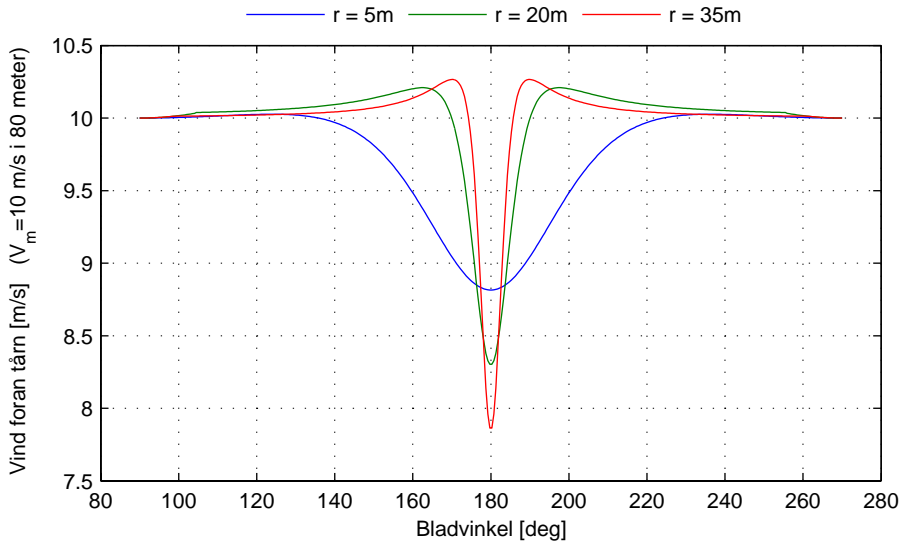
Det forventes, at tårnskyggen vil give et dyk i vindhastigheden, når et blad er lige foran tårnet. Faktisk sker der yderligere det, at vinden stiger, før bladet når ind foran tårnet. Det ser altså ud som om vinden, der bliver presset uden om tårnet, giver anledning til en højere vindhastighed ved siden af tårnet. Dette er illustreret i figur 7.2, hvor tårnskyggen for ét bladelement er vist for hele den nedre halvcirkel af et blads bevægelse.



Figur 7.2: Vindhastigheden som ét element ser foran tårnet, her vist for 3 steder på bladet.

Det ses desuden, at når bladet går ind i den nedre halvcirkel (vinkler fra 90° – 270°) vil der være et abrupt spring i vinden, hvilket skyldes tårnets abrupte afslutning. Dette ses især for de små radier. Dette er altså et produkt af, at modellen af tårn og nacelle er for simpel. Flowet omkring nacellen kunne være inkluderet og ville nok give et pænere resultat.

Da det er dykket i vinden foran tårnet, der er den markante egenskab ved tårnskyggen, som ønskes i modellen, er problemet løst ved at lade tårnet afslutte med en kegle på toppen. For de sidste 5 meter af tårnet antages altså, at radius går mod nul. Dette vil give et pænt forløb, når et blad bevæger sig ind i tårnskyggen og ikke påvirker det dyk, der ligger i vinden foran tårnet. Resultatet ses i figur 7.3 for de samme bladradier som i figur 7.2.



Figur 7.3: Det abrupte spring, når bladet bevæger sig ind i tårnskygge, er fjernet ved at afslutte tårnet med en kegle.

7.1.2 Vindshear

Friktionen mellem vind og jord bevirker, at vinden bevæger sig langsommere nær jordens overflade. Vindhastigheden får derfor en vertikal profil, som her er forsøgt beskrevet ved en simpel model. Vindshear er i meget stor grad afhængig af den placering, møllen har. Hver slags jordoverflade beskrives i modellen med et tal, ruhedslængden, som fortæller, hvor meget friktion jordoverfladen har. Ruhedslængden, z_0 er brugt tidligere til at beskrive turbulensforholdene og her gengivet i tabel 7.1 for nogle forskellige typer terræn.

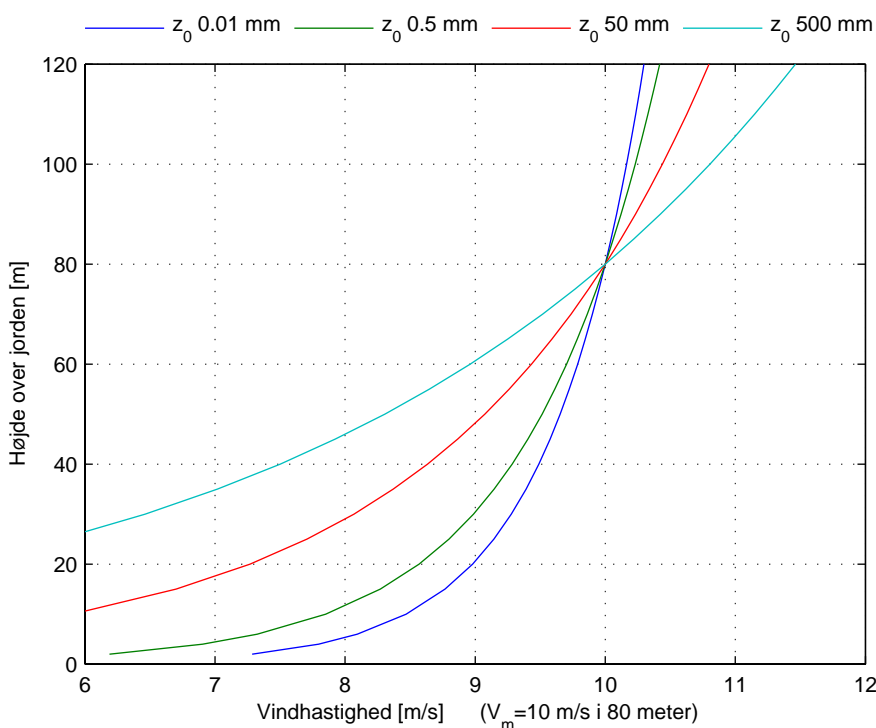
Terrænbeskrivelse	$z_0(mm)$
Meget glat, is eller mudder	0.01
Oprørt hav	0.5
Afgrøder	50
Skov	500

Tabel 7.1: Ruhedsfaktor ved forskelligt terræn

Modellen, der anvendes, kaldes en logaritmisk profil og er en af de mere brugte modeller [Manwell 05]. I sin grundform beskriver den forskellen i vinden for to højder. Dette er vist i ligning 7.3

$$\frac{v_m(H + h_1)}{v_m(H + h_2)} = \frac{\ln\left(\frac{H+h_1}{z_0}\right)}{\ln\left(\frac{H+h_2}{z_0}\right)} \quad (7.3)$$

Hvor h_1 og h_2 er de to højder, der ønskes sammenlignet. For de 4 ruhedslængder (vist i tabel 7.1) er vindhastigheden beregnet fra jordens overflade og op til 120 meter, som er det højeste punkt bladene rammer. Dette er vist i figur 7.4 for en middelvind v_m i 80 meter på 10 m/s.



Figur 7.4: Vindprofiler for forskellige ruhedslængder.

En af grundene til at anvende netop denne shearmodel er også, at den er implementeret i HAWC. Dette giver mulighed for at simulere under samme betingelser. Derudover har det givet mulighed for at verificere modellen i forhold til HAWC's implementering. Dette kan gøres ved at udlæse vindhastigheden fra HAWC på et givent punkt i rotorplanet, når den stokastiske vindmodel og tårnskygge er slået fra. Sammenligningen viste, at de to modeller passede overens

indtil omkring 4. decimal.

Vindshear vil bevirke, at bladene i top konsekvent vil blive påvirket af en kraftigere vind end nede. Da påvirkningen er fordelt jævnt over hele rotorplanet, vil det ikke have en lige så voldsom indvirkning på møllen som tårnskyggen. Men som det ses senere i kapitel 8, giver det anledning til nogle uens kræfter, som forventes at kunne påvirke møllens holdbarhed på længere sigt.

7.1.3 Implementering

Både tårnskyggen og shearprofilen er implementeret i modellen umiddelbart før BEM beregningerne. Her kendes hvert blads vinkel, og dermed via simpel geometri kan alle bladelementers koordinater bestemmes. For hvert bladelement beregnes shear- og tårnskyggens påvirkning. Dette er implementeret i m-filen DVM.m se bilag E. Denne kaldes med en azimut-vinkel samt de radier på bladet, som ønskes beregnet. Resultatet er en gain for hvert bladelement, der kan multipliceres på den oprindelige vind.

7.2 Stokastisk vindmodel

Modelleringen af den stokastiske variation i vinden, set for et givent antal punkter i rotorplanet, er blevet overvejet og forsøgt implementeret. Det skal dog understreges, at implementeringen af den her foreslåede model ikke er blevet færdiggjort p.g.a. manglende tid. Det er dog alligevel muligt at give en kort gennemgang af det teoretiske grundlag for modelleringen.

Som ved modelleringen af stokastisk vindvariation for en ækvivalent vind er udgangspunktet for modelleringen af vinden i et givent antal punkter et udtryk for vindvariationen af en punkt vind. Her kan der igen anvendes udtrykket, foreslået af [Højstrup 82], set i ligning 2.1. Det kunne være fristende at modellere vinden i et givent antal punkter blot ved at beregne punktvindsvariationen for hver af de ønskede punkter. Dette vil imidlertid være en oversimplificering, fordi vinden i et givent punkt er korreleret med vinden i et andet punkt, der ligger i en given afstand fra punktet. Graden af korrelation afhænger af afstanden mellem de to punkter som ifølge [Winkelaar 92] og [Veers 89] er givet ved ligning 7.4 og kaldes korrelationsfunktion.

$$\lambda_{(i,j)}^2 = \exp\left(\frac{-A_{(i,j)}\Delta r_{(i,j)}f}{v_m}\right) \quad (7.4)$$

$A_{(i,j)}$, kaldet decay coefficient, er ifølge [Winkelaar 92] afhængig af retningen og højden, som vinden måles i, men ifølge [Veers 89] kan den bestemmes som en

konstant på 7.5. $\Delta r_{(i,j)}$ er afstanden mellem de to punkter, som afhængighedsfaktoren ønskes bestemt for. f er frekvensen og v_m er middelvinden. Metoden til at beregne den stokastiske vindvariation for et givent antal punkter er beskrevet i [Winkelaar 92] og [Veers 89], som i meget korte træk beregner vinden for hvert punkt som værende afhængig af vinden i alle de andre punkter, der betragtes i vindfeltet. Problemet med denne metode er, at den er meget beregningstung, hvilket dog delvist kan afhjælpes ved at anvende metoden beskrevet i [Winkelaar 92]. Da denne model ikke er blevet implementeret, er der ikke yderligere gået i dybden med emnet, men det vil være et oplagt område at lave videre undersøgelser på.

En anden mulighed ville være at få skrevet vindsekvensen fra HAWC ud til en fil. Gøres det for et passende antal punkter vil de kunne bruges som input til simulink modellen. HAWC indeholder også mulighed for at udlæse vinden i et vilkårligt punkt i rotorplanet til output filen. Ud fra HAWC kan der umiddelbart kun udlæses sekvensen for ét punkt pr. simulation. En åbenlys fordel ved at hente vindsekvensen fra HAWC vil desuden være at de to modeller så vil være nemme at sammenligne.

Individuel reguleringsstrategi

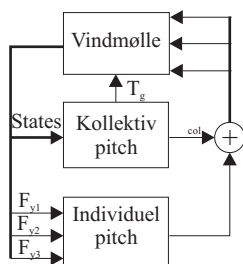
Indtil nu er reguleringen af vindmøllen blevet foretaget ved at pitchen på alle de tre rotorblade justeres ens, altså kollektiv pitch-regulering. Som nævnt i kapitel 6, varierer vindfeltet over rotorplanet, hvilket betyder, at de enkelte blade er udsat for forskellige vindfelter og dermed forskellige kraftpåvirkninger. Det vil derfor være en naturlig udvidelse i reguleringen at se på, om det er muligt at regulere de enkelte blade. Denne regulering kunne enten foretages ud fra en måling af vindfeltet på bladene eller ud fra måling af den kraftpåvirkning, bladene udsættes for.

Vinden, på hvert blad, er imidlertid problematisk at måle. Dels fordi bladet selv roterer, hvilket bevirker, at vindmålingen skal være meget robust med hensyn til retningsbestemmelse af vindfeltet, og dels skal der laves mere end en måling for hvert blad, da vindfeltet kan variere ud langs bladet. Fordelene ved at regulere ud fra måling af vinden vil dels være, at det kunne være muligt at måle vinden lidt foran bladet og derved muliggøre en feed forward regulering, og dels at der ikke er noget delay i målingen, som der vil være i en måling af kraften på bladet, grundet bladets fleksibilitet.

Målingen af kraften vil til gengæld kunne foretages relativt nemt, da den eksempelvis kan laves ved at placere en måler inde i hvert blad. Til design af individuel pitch-regulering er det således valgt at antage at kraften på hvert blad kan måles sammen med rotorpositionen i form af azimut-vinklen. Der er ikke taget højde for det førnævnte delay i målingen, da bladet antages at være stift. kraften måles i flapvis retning og ikke kantvis, da reguleringen kun vil

blive designet med henblik på at reducere den flapvise kraftpåvirkning. Det er selvfølgelig også antaget, at de tre blade har en selvstændig pitch-servo for at muliggøre individuel regulering.

Den individuelle regulering er implementeret som en selvstændig sløjfe, der kører parallelt med den kollektive regulering, som illustreret på figur 8.1.



Figur 8.1: Principdiagram over samspil mellem individuel- og kollektiv pitch-regulering

Dette bevirker, at den individuelle regulering kommer med et styresignal til det enkelte blad, som lægges til det kollektive pitch-signal. Det giver den fordel, at designet fra den kollektive pitch-regulering kan bevares, så den individuelle regulering kan designes uafhængigt af denne.

Det skal nu fastlægges, hvilke mål der ønskes opnået med den individuelle regulering, samt hvilken strategi reguleringen skal designes ud fra. Da et af hovedmålene med reguleringen er at begrænse de strukturelle belastninger møllen udsættes for, vil det være naturligt at se på, hvorledes disse kan reduceres med individuel regulering. I den kollektive pitch-regulering var det muligt at reducere belastningen på drivakslen i rotationsretningen. Her er det dog ikke muligt at kompensere for de belastninger bladene, tårnet og rotorlejet udsættes for i yaw- og tiltretningen, hvilket der er flere årsager til. Den kollektive pitch reguleres ud fra målinger på rotationens hastigheden og effekten. Disse giver ikke nogen god indikation af forskellen i kræfterne på de enkelte blade, men nærmere en resulterende værdi af de tre momenter. Da pitchen reguleres med et fælles styresignal er det heller ikke muligt at kompensere for forskelle i momenterne. Sluttelig er designmodellen lavet ud fra en antagelse af, at vinden er ens over hele rotorplanet.

Dette er ikke tilfældet for den individuelle pitch-regulering, der som nævnt tager udgangspunkt i kraften målt på det enkelte blad i flapvis retning og har muligheden for at regulere bladene hver for sig. Det er valgt at lave reguleringen ud fra en betragtning om, at hvis kraften holdes ens på alle tre blade,

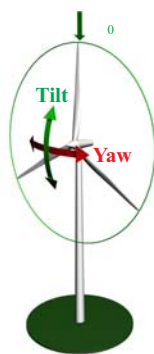
vil der være opnået en regulering, der fuldstændig kompenserer for det ujævne vindfelt. En sådan regulering vil naturligvis ikke være mulig, men den giver et reguleringsgrundlag, der siger, at kraften på alle bladene skal være ens. Ud fra dette grundlag er der overvejet to løsninger, hvoraf den sidste er blevet implementeret. Den første går på at designe en regulator til hvert blad, der bruger middelfraften på de tre blade som reference. Der er ikke blevet foretaget dybere undersøgelser af, hvorledes dette kan implementeres.

Princippet i den anvendte metode er at minimere kræfterne i yaw- og tiltakserne, se figur 8.2. Denne metode bygger på en såkaldt Coleman transformation [Coleman 57], vist i ligning 8.2, samt den inverse transformation i ligning 8.1.

$$\begin{bmatrix} F_1^c \\ F_2^c \\ F_3^c \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3}\sin(\psi_1) & \frac{2}{3}\sin(\psi_2) & \frac{2}{3}\sin(\psi_3) \\ \frac{2}{3}\cos(\psi_1) & \frac{2}{3}\cos(\psi_2) & \frac{2}{3}\cos(\psi_3) \end{bmatrix} \cdot \begin{bmatrix} F_{y1} \\ F_{y2} \\ F_{y3} \end{bmatrix} \quad (8.1)$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} 1 & \sin(\psi_1) & \cos(\psi_1) \\ 1 & \sin(\psi_2) & \cos(\psi_2) \\ 1 & \sin(\psi_3) & \cos(\psi_3) \end{bmatrix} \cdot \begin{bmatrix} \theta_1^c \\ \theta_2^c \\ \theta_3^c \end{bmatrix} \quad (8.2)$$

Hvor F_{y1-3} er kraften målt på hvert blad, θ_{1-3} er det totale pitch-styresignal til hvert blad, θ_1^c er den kollektive pitch, θ_2^c er pitchen i yaw-aksen, θ_3^c er pitchen i tilt-aksen, F_1^c er en kraftkomponent, der ikke anvendes, F_2^c er kraften i yaw-aksen, F_3^c er kraften i tilt-aksen og ϕ er azimut-vinklen. Bemærk at sinus komponenten giver yaw-aksen, og cosinus komponenten tilt-aksen, dette skyldes at azimut-vinklen er defineret som værende 0, når et blad står lodret op, se figur 8.2.



Figur 8.2: Yaw- og tiltkræfternes retninger, samt Azimuth nulpunkt.

Den inverse Coleman transformation transformerer de tre kræfter målt på bladene som en funktion af azimut-vinklen til kræfter i yaw- og tiltretningerne samt en tredje komponent, der ikke anvendes. Fordelen ved at anvende denne transformation er, at man har to stabile referencepunkter for yaw- og tiltkræfterne, som

begge ligger i 0. Når reguleringen er foretaget i de to akser, kan styresignalerne fra den individuelle regulering så transformeres tilbage med Colman transformationen til tre individuelle styresignaler for de enkelte blade, og det kollektive styresignal lægges til hver af de tre styresignaler. Individuel regulering ud fra Coleman transformationen har dog en ulempe, som vil blive beskrevet senere. Reguleringsstrategien er altså at designe en regulering, som minimerer kræfterne i yaw- og tiltretningen. Efter dette er gjort, skal det undersøges, hvilken indvirkning denne regulering eventuelt har på den kollektive regulering.

Individuel regulator design

Reguleringen, der designes til den individuelle pitch, består af to PI- og to P regulatorer. P reguleringen vil blive begrundet senere i afsnittet. Valget faldt på PI regulering, dels fordi designet forholdsvis hurtigt kan laves, og dels fordi man ved Coleman transformation opnår en afkobling af de to akser i h.h.v. yaw og tilt, så reguleringen i hver akse kan designes uafhængigt af hinanden. Dette gælder dog kun, hvis vindfeltet er homogent, hvilket selvfølgelig ikke er tilfældet. Der vil derfor reelt være en krydskobling mellem yaw- og tiltreguleringen, men designet ser bort fra denne. Dette bringer ulempen ved at anvende Coleman transformationen på banen. Det er nemlig nødvendigt at antage et homogent vindfelt for effektivt at kunne anvende transformationen, fordi den transformerede model ellers bliver meget kompleks. Dette er en grov antagelse, især når man tager i betragtning, at formålet med den individuelle pitch-regulering netop er at kompensere for et ikke homogent vindfelt.

Simplificeringen er dog blevet anvendt, hvilket kan forsvares med at reguleringen anvender den målte kraft på de enkelte blade som input. Disse målinger vil være afhængige af vindfeltet på det enkelte blad, hvilket bevirker at reguleringen reelt vil virke som individuel regulering, selvom der under designet af selve regulatoren er antaget et homogent vindfelt. Der er dog også en fordel ved at antage et homogent vindfelt. Det ville ellers være nødvendigt at lave et regulator design, som var afhængig af azimut-vinklen, hvilket ville kræve at reguleringen implementeres med en gain scheduling i forhold til azimut-vinklen. Dette er ikke nødvendigt, når vindfeltet under designet antages for værende homogent, hvilket

simplificerer designet og implementeringen af regulatoren. Det anses derfor som et godt udgangspunkt for individuel pitch-regulering. Det vil selvfølgelig være interessant at undersøge, hvilken regulering der kan opnås ved at designe ud fra ikke homogene vindfelter, men dette emne er ikke behandlet i dette projekt.

9.1 Designmodel

Modellen, PI reguleringen designes ud fra, er primært en gain, bestemt ved linearisering af BEM beregningerne i det ønskede lineariseringspunkt, samt overføringsfunktionen fra pitch-aktuatoren, dog vil der også som beskrevet under PI- og P regulering blive indført nogle filtre for at opnå en tilfredsstillende regulering. Det er valgt at lave denne simple betragtning af designmodellen, fordi det vil være de primære faktorer i designet, da designkriteriet udelukkende går på at eliminere kræfterne i yaw- og tiltakserne og ikke omdrejnings- og effekt begrænsning. Det ville være relevant at tage tårndynamikken med i designet, da denne har en direkte indvirkning på kræfterne, vingerne udsættes for. Indvirkningen heraf er ikke blevet undersøgt, men anses for at være en relevant udvidelse af designet, som er værd at undersøge.

Kræfterne i flapretningen kan findes ud fra BEM beregningerne, så designmodellen kan opskrives, som vist i ligning 9.1 og 9.2.

$$\sum_{i=1}^3 F_{yi} = \sum_{i=1}^3 BEM(v_{mi}, \theta_i, \omega_{ri}) \quad (9.1)$$

$$\theta_i = \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \cdot \theta_{iref} \quad (9.2)$$

Modellen kan nu lineariseres, som vist i ligning 9.3 og 9.4.

$$\sum_{i=1}^3 \frac{\partial F_{yi}}{\partial(v_{mi}, \theta_i, \omega_{ri})} = \sum_{i=1}^3 \frac{\partial BEM(v_{mi}, \theta_i, \omega_{ri})}{\partial(v_{mi}, \theta_i, \omega_{ri})} \quad (9.3)$$

$$\theta_i = \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \cdot \theta_{iref} \quad (9.4)$$

Designmodellen opdeles nu, som vist i ligning 9.5.

$$\sum_{i=1}^3 \Delta F_{yi} = \left[k_{v_m} \sum_{i=1}^3 \Delta v_{mi} + k_{\theta} \sum_{i=1}^3 \Delta \theta_i + k_{\omega_r} \sum_{i=1}^3 \Delta \omega_{ri} \right] \quad (9.5)$$

Hvor k_{v_m} , k_{θ} , k_{ω_r} er hældningen i lineariseringspunktet med hensyn til v_m , θ , ω_r . Det bemærkes, at disse hældninger er sat uden for sumationen, hvilket kun kan

gøres i tilfælde af et homogent vindfelt. Da reguleringen kun kan styre pitch-vinklen, er Δv_{mi} , $\Delta\theta_i$, $\Delta\omega_{ri}$, givet ved ligning 9.6.

$$\Delta v_{mi} = 0 \quad \Delta\theta_i = \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \cdot \Delta\theta_{iref} \quad \Delta\omega_{ri} = 0 \quad (9.6)$$

Modellen kan derfor reduceres til ligning 9.7 og 9.8.

$$\sum_{i=1}^3 \Delta F_{yi} = k_\theta \sum_{i=1}^3 \Delta\theta_i \quad (9.7)$$

$$\sum_{i=1}^3 \Delta F_{yi} = k_\theta \sum_{i=1}^3 \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \cdot \Delta\theta_{iref} \quad (9.8)$$

For at kunne anvende designmodellen i yaw- og tiltakserne skal den transformeres med Coleman transformationen. Til dette opskrives først nogle regler, vist i ligning 9.9 til 9.11, som simplificerer processen p.g.a. bladernes symmetriske placering, og der antages et homogent vindfelt.

$$\sum_{i=1}^3 \theta_i = 3\theta_1^c \quad (9.9)$$

$$\sum_{i=1}^3 \sin(\psi_i)\theta_i = \frac{3}{2}\theta_2^c \quad (9.10)$$

$$\sum_{i=1}^3 \cos(\psi_i)\theta_i = \frac{3}{2}\theta_3^c \quad (9.11)$$

Transformationen i ligning 8.2 laves nu, så designmodellen udtrykkes ved de tre kræfter F_{1-3}^c , set i ligning 9.12 til 9.14.

$$\begin{aligned} \Delta F_1^c &= \frac{1}{3} \sum_{i=1}^3 \Delta F_{yi} \\ \Delta F_1^c &= \frac{1}{3} k_\theta \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \sum_{i=1}^3 \Delta\theta_{iref} \\ \Delta F_1^c &= k_\theta \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \Delta\theta_{1ref}^c \end{aligned} \quad (9.12)$$

Kraften fundet i ligning 9.12 vil ikke blive anvendt i reguleringen.

$$\begin{aligned}
 \Delta F_2^c &= \frac{2}{3} \sum_{i=1}^3 \sin(\psi_i) \cdot \Delta F_{yi} \\
 \Delta F_2^c &= \frac{2}{3} k_\theta \frac{\omega_n^2}{s^2 + \zeta \omega_n s + \omega_n^2} \sum_{i=1}^3 \sin(\psi_i) \cdot \Delta \theta_i \\
 \Delta F_2^c &= k_\theta \frac{\omega_n^2}{s^2 + \zeta \omega_n s + \omega_n^2} \Delta \theta_2^c
 \end{aligned} \tag{9.13}$$

$$\begin{aligned}
 \Delta F_3^c &= \frac{2}{3} \sum_{i=1}^3 \cos(\psi_i) \cdot \Delta F_{yi} \\
 \Delta F_3^c &= \frac{2}{3} k_\theta \frac{\omega_n^2}{s^2 + \zeta \omega_n s + \omega_n^2} \sum_{i=1}^3 \cos(\psi_i) \cdot \Delta \theta_i \\
 \Delta F_3^c &= k_\theta \frac{\omega_n^2}{s^2 + \zeta \omega_n s + \omega_n^2} \Delta \theta_3^c
 \end{aligned} \tag{9.14}$$

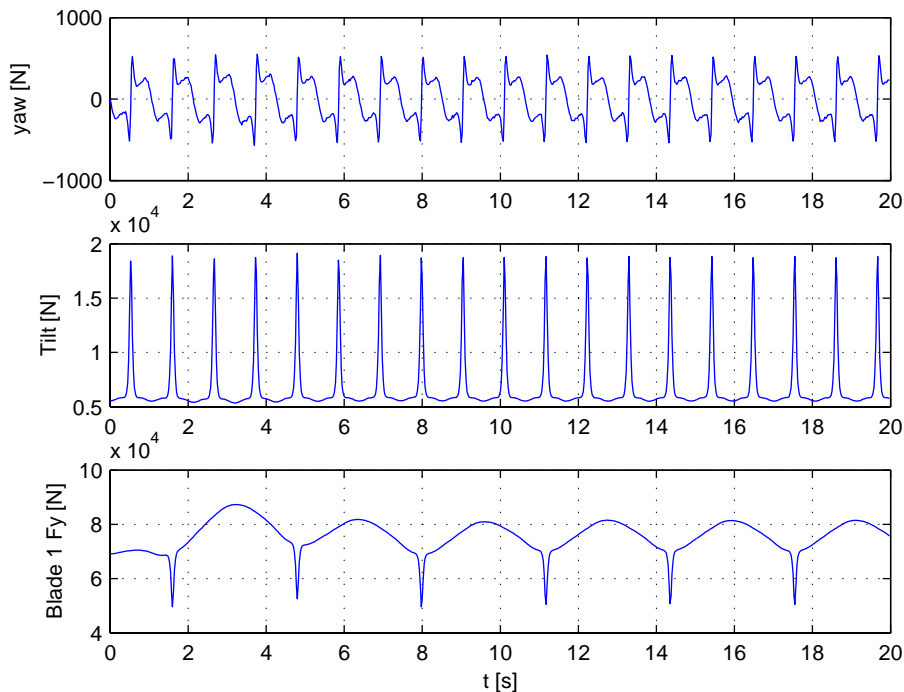
Som nævnt findes gainen k_θ ved at linearisere BEM beregningerne med hensyn til pitch-vinklen. Så længe vindfeltet antages som værende homogent, ville resultatet også kunne opnås ved at linearisere $1/3 c_t$ kurve, som gjort i [Engelen 05]. Fordelen ved at anvende BEM beregningerne ligger i, at det her vil være muligt at udbygge designmodellen, så der bliver taget højde for, at vindfeltet reelt ikke er homogent.

Selve lineariseringen af BEM beregningerne foretages numerisk med matlab funktionen `lin_BEM.m`, ses i bilag E. Muligheden for at finde et analytisk udtryk for hældningerne er undersøgt, men da en del af BEM beregningerne bestemmes ved en iterativ proces, gør det analytisk differentiering af BEM beregningerne problematisk, og koefficienterne bestemmes derfor kun numerisk.

9.2 PI- og P regulering

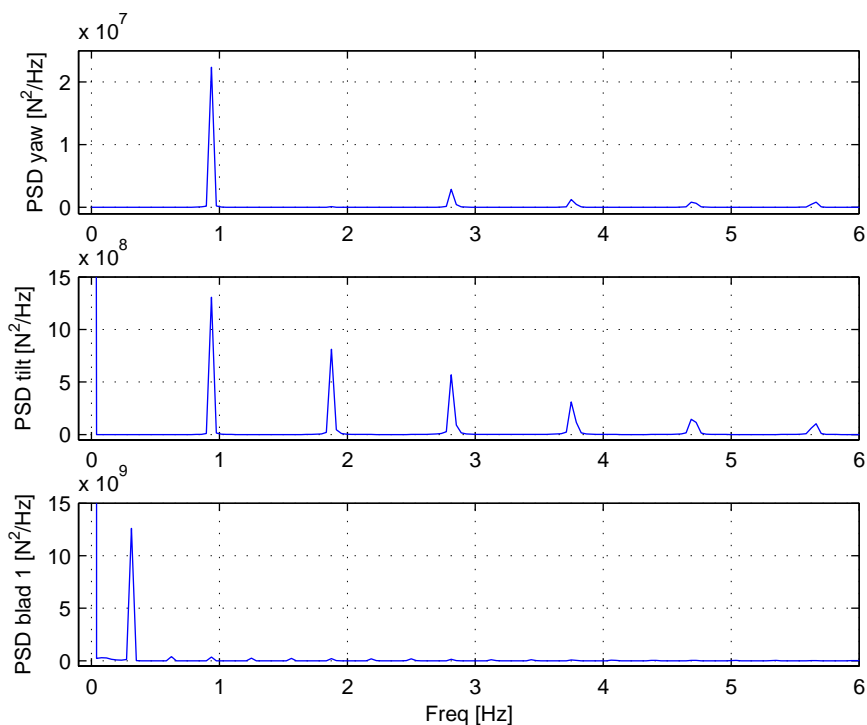
Inden selve designet af PI- og P reguleringen foretages er det nødvendigt at se på, hvorledes vindfeltet reelt ser ud for at få en ide om, hvad regulatoren forventes at kompensere for. Som det blev gennemgået i kaptiel 7, indeholder simulationsmodellen for individuel pitch, udviklet i dette projekt, kun en deterministisk beskrivelse af variationen af vindfeltet. Det vil derfor ikke være muligt for denne simulationsmodel at se, hvilken indflydelse den stokastiske del af vinden har på reguleringen. Det er dog muligt at teste dette med HAWC, men dette skal gøres med visse forbehold, da HAWC ikke har de samme systemparametre som simulationsmodellen fra dette projekt.

På figur 9.1 ses et plot over kræfterne i yaw- og tiltakserne samt det rotationssamplede vindfelt set fra et af bladene ved en vindhastighed på $12m/s$. Det er udelukkende effekten af det deterministiske vindfelt, der ses, altså shear- og tårnskygge. Den kollektive regulering er LQI II reguleringen. Der er i disse test anvendt en ruhedslængde på $500mm$ svarende til skov, for at tydeliggøre indvirkningen af shear i yaw- og tiltakserne.



Figur 9.1: Kræfter på vindmølle med deterministisk vindfelt ved en middelvind på $12m/s$, uden individuel regulering. Kollektiv regulering er LQI II, ruhedslængden er $500mm$.

På det enkelte blad ses det, at vindshear er skyld i, at der kommer en lavfrekvent svingning, som har frekvensen $1p$, altså vindmøllens rotationshastighed, hvilket for nominelt omløbstal vil være $0.314Hz$. Yderligere er der et højfrekvent dyk i kraften, som er forårsaget af tårnskyggen. Dette dyk kommer også med et interval på $1p$, men selve dykket vil være mere højfrekvent. På tilt-aksen er vindsher årsag til et offset i kraften, hvilket skyldes, at vinden er kraftigere i den øverste del af rotorplanet. Det ses også tydeligt, at tårnskyggen slår igennem til tilt-aksen ved at give nogle store peeks hver gang et blad rammer skyggen, altså med $3p$. Yawaksen har ikke noget offset, hvilket heller ikke er forventet, da der ikke er nogen shear i denne retning. Det ses, at kraften svinger med en grundfrekvens på $3p$, men der er også mere højfrekvente svingninger i kraften. For at undersøge mere præcist, ved hvilke frekvenser svingningerne i kraften ligger for de to akser samt på bladet, laves der en PSD af kræfterne, som vist på figur 9.2. PSD'en er lavet ud fra en simulering på 400 sek., hvor de første 100 samples er smidt væk p.g.a. indsving. Der er brugt en FFT, med 2^{10} punkter.



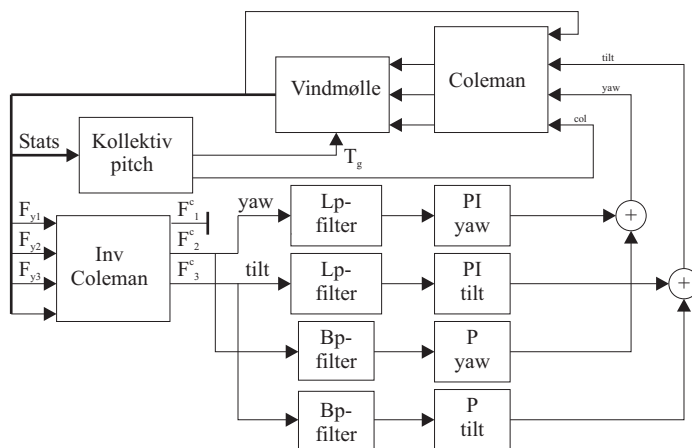
Figur 9.2: PSD for kraften i yaw- og tiltakserne samt for blad 1.

Den primære variation i kraften for yaw-aksen ligger klart ved en frekvens på $3p$, men der er også variationer ved nogle af de højere harmoniske frekvenser. For tilt ses en klar peak ved $3p$, men der er også aftagende, men stadig kraftige peaks ved de overharmoniske $6p$, $9p$ og opefter. Yderligere ses en kraftig $0p$ komponent, hvilket stemmer overens med, hvad der kunne ses af tidsplottet på figur 9.1. For bladet ligger den primære variation i kraften som forventet på $1p$, også her er der en $0p$ komponent.

Ved design af PI reguleringen er det nødvendigt at indsætte et filter, der filtrerer den højfrekvente del af tilt- og yawkræfterne væk. Dette skyldes, at pitch-reguleringen ikke er hurtig nok til at kompensere for effekten af tårnskyggen. Hvis der ikke indsættes et LP-filter før reguleringen af tilt- og yawkræfterne, vil denne forsøge at kompensere for tårnskyggen, hvilket kun har en negativ effekt grundet den faseforskyning, der er mellem målt kraft og resulterende pitch-vinkel.

For både yaw- og tiltkraften indsættes en P regulering i serie med et BP-filter, der har en pasfrekvens på $3p$. Denne regulering giver en ekstra gain i netop det område, hvor kraften varierer aller mest, hvilket gav gode resultater, se i kapitel 10. På figur 9.3 ses et blokdiagram over implementeringen af reguleringen, og

en komplet implementering i simulink kan ses på bilag D.



Figur 9.3: Blokdiagram over implementering af individuel og kollektiv pitch-regulering

LP-filtrene før tilt- og yawaksernes PI regulering er et 4.ordens Butterworthfilter. For tilt-aksen sættes en knæfrekvens på 1 Hz , altså lige over $3p$ frekvensen, da det ønskes at dæmpe de kraftige overharmoniske svingninger ved $6p$ og $9p$. For yaw-aksen er LP-filtret ikke så kritisk, da de overharmoniske svingninger ikke er så kraftige i denne akse, og knæfrekvensen sættes derfor til 2 Hz . De to båndpasfiltre før P reguleringen er ens og designes som 2.ordens Butterworthfiltre med en pasfrekvens på 0.85 til 1.1 Hz , altså omkring $3p$ frekvensområdet. Disse filtre skal tages med i overføringsfunktionen til design af regulatorerne. Dette giver en designmodel for tilt PI reguleringen, vist i ligning 9.16, yaw PI reguleringen i ligning 9.15, og for P reguleringen i tilt og yaw i ligning 9.17.

$$\Delta F_2^c = LP_{yaw} \cdot k_\theta \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \Delta\theta_3^c \quad (9.15)$$

$$\Delta F_3^c = LP_{tilt} \cdot k_\theta \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \Delta\theta_3^c \quad (9.16)$$

$$\Delta F_{2-3}^c = BP \cdot k_\theta \frac{\omega_n^2}{s^2 + \zeta\omega_n s + \omega_n^2} \Delta\theta_{2-3}^c \quad (9.17)$$

Filtrene er designet ud fra et nominelt omdrejningstal og er også kun testet i dette område. Det er klart, at hvis reguleringen skal blive ved med at fungere optimalt ved et lavere omdrejningstal, er det nødvendigt at designe nogle filtre og regulatorer, der følger de harmoniske områder.

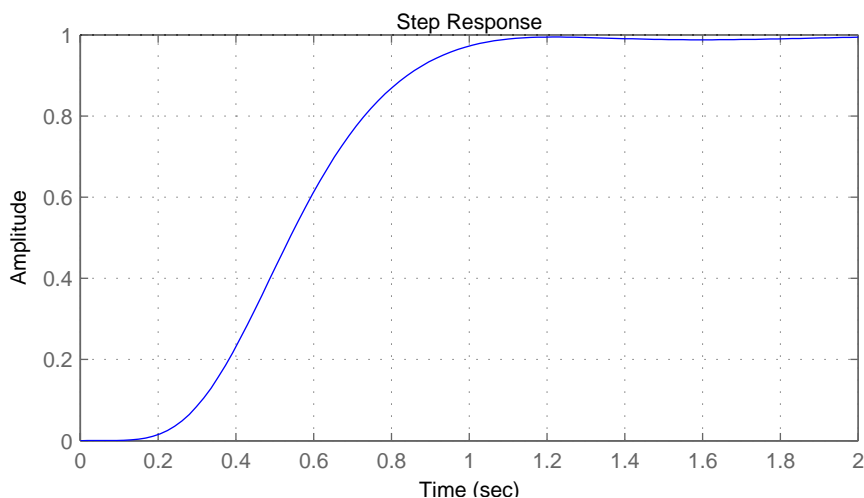
9.2.1 PI regulator design

Da overføringsfunktionerne for designmodellen er fundet for yaw-aksen ved ligning 9.15 og for tilt-aksen med ligning 9.16, kan der nu designes en PI regulering til de to akser ud fra bodeplot og trinrespons. Begge PI reguleringer er implementeret med overføringsfunktionen, vist i ligning 9.18.

$$G(s) = K_p \frac{\tau_i s + 1}{\tau_i s} \quad (9.18)$$

Den konventionelle fremgangsmåde, som i kapitel 4, er blevet testet til design af PI regulatorerne. Dette giver et design, der har overshoot på indsvinget, hvilket viser sig at give et ringe resultat ved implementering. Der er derfor lavet et design, hvor der er et meget lille overshoot på indsvinget, altså er K_p gjort mindre end designmetoden tillader, og τ_i kan derfor gøres større.

Resultatet af designet for yaw-regulatoren blev $K_p = 0.000045$ og $\tau_i = 0.18$, hvilket giver trinresponsen, vist i figur 9.4. Bodeplottet kan ses på bilag C.



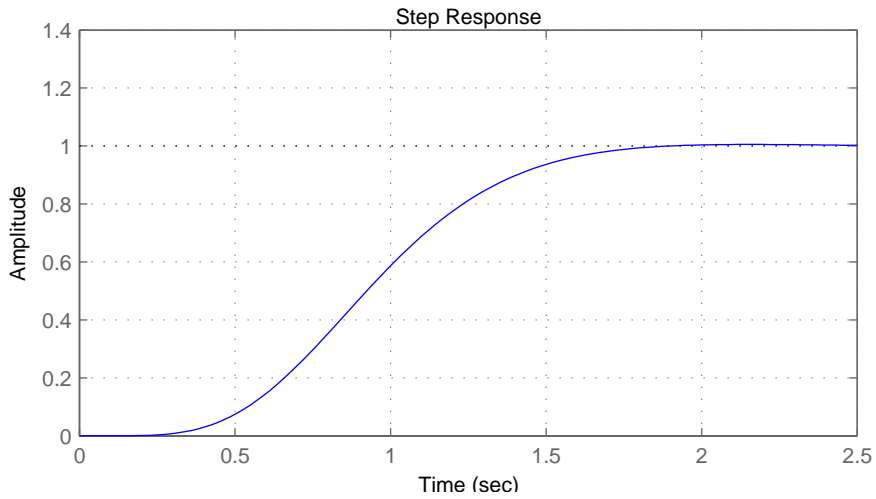
Figur 9.4: Stepresponse for PI design i yaw-akse.

Samme fremgangsmetode er brugt for tilt-aksen, hvilket resulterede i $K_p = 0.00003$ og $\tau_i = 0.2$. Trinresponsen herfra ses på figur 9.5, og bodeplottet kan ses på bilag C.

Trinresponsen i yaw-aksen er hurtigere end for tilt-aksen p.g.a., at det lavpasfilter, der blev indsat, har en højere knækfrekvens.

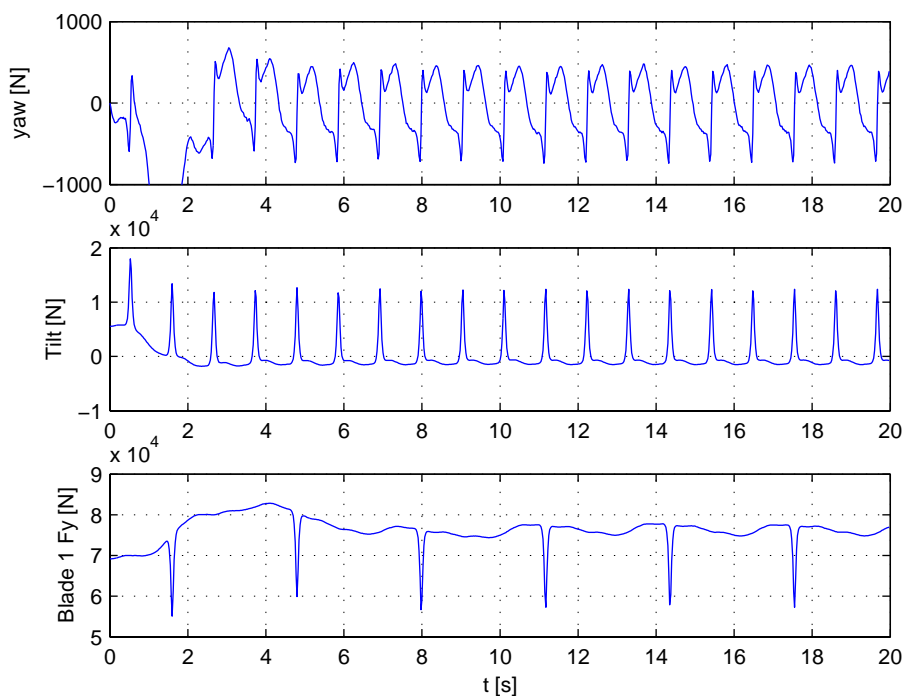
På figur 9.6 ses resultatet af den individuelle PI regulering i yaw- og tiltakserne.

Reguleringen får trukket variationen i tilt-kraften ned at ligge omkring 0, og



Figur 9.5: Stepresponse for PI design i tilt-akse.

1p kraften på bladet er dæmpet betydeligt. Der er dog kommet større udsving på kraften i yaw-aksen. For en mere dybdegående undersøgelse af reguleringen henvises til kapitel 10.

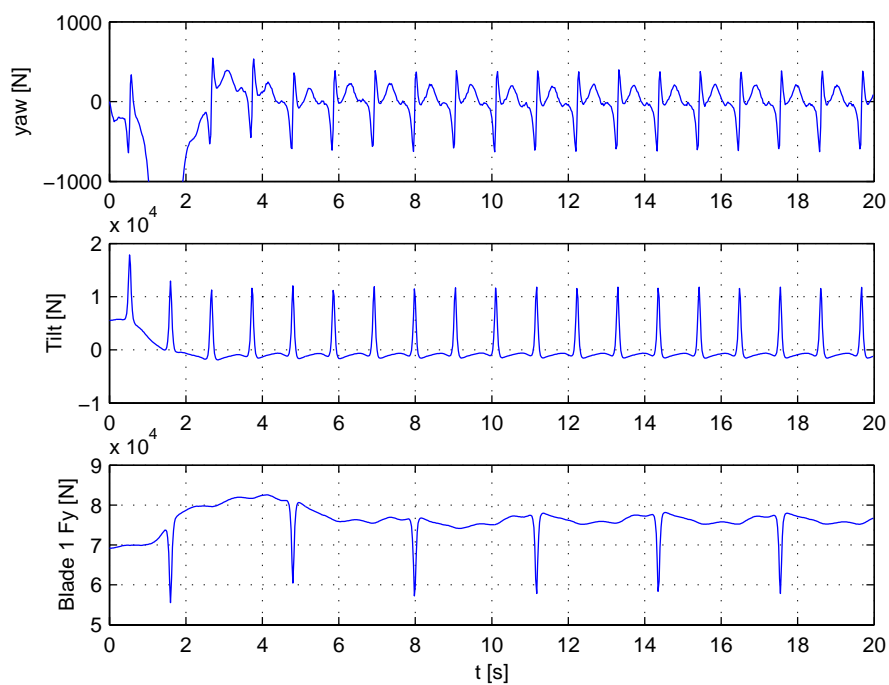


Figur 9.6: Kræfter på vindmølle med deterministisk vindfelt ved en middelvind på 12 m/s , med individuel PI regulering i yaw- og tiltakse. Kollektiv regulering er LQI II, ruhedslængde 500 mm .

9.2.2 P design

Som nævnt har det vist sig at være en fordel at indsætte en gain direkte i 3p frekvensområdet. Denne gain er ens for de to akser og designes på simpel vis ud fra bodeplottet set i bilag C. Der blev opnået et godt resultat med en gain på $P = 0.0001$, hvilket giver en gain margin på 10.8 dB . Dette kan umiddelbart virke som en stor margin, men ved test på det samlede system blev det fundet, at en større gain ikke gav anledning til en forbedring af resultatet. På figur 9.7 ses resultatet af den individuelle regulering med både PI- og P regulering indsat.

Der ses en forbedring i udsvinget i yaw-akse kraften. Igen henvises til kapitel 10 for en mere dybdegående undersøgelse af reguleringen.



Figur 9.7: Kræfter på vindmølle med deterministisk vindfelt ved en middelvind på 12m/s , med individuel regulering PI- og P regulering i yaw- og tiltakse. Kollektiv regulering er LQI II, ruhedslængde 500mm .

KAPITEL 10

Test af individuel regulering

10.1 Testindhold

Test plot i dette kapitel:

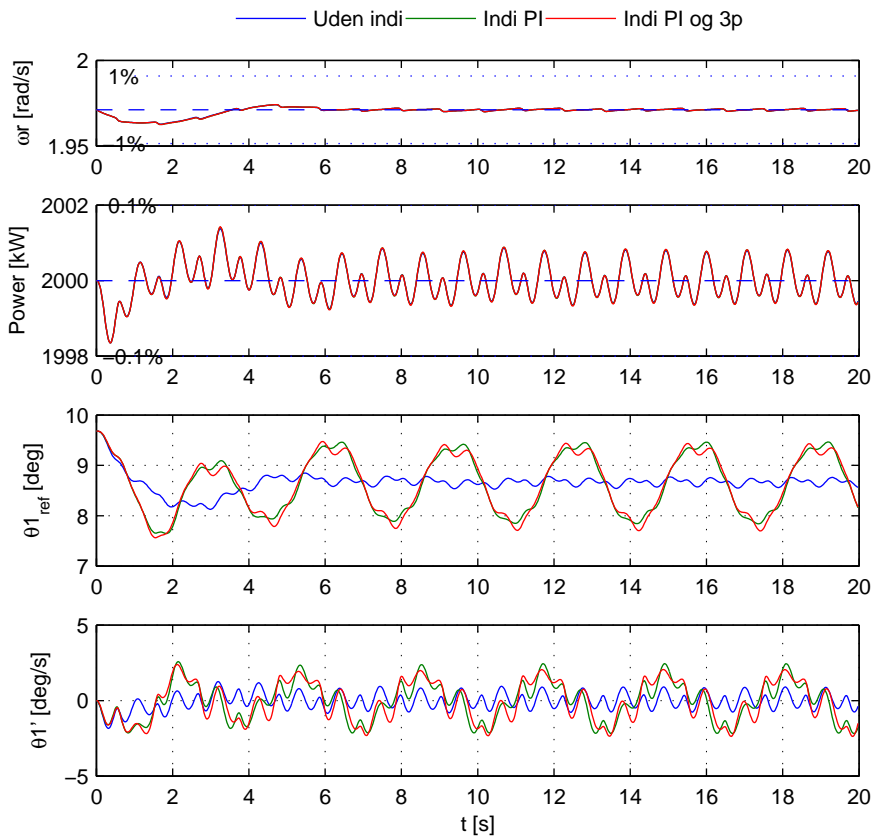
- 1 Individuel pitch i simulink
- 2 Individuel pitch i HAWC
- 3 Sammenligning af simulink og HAWC

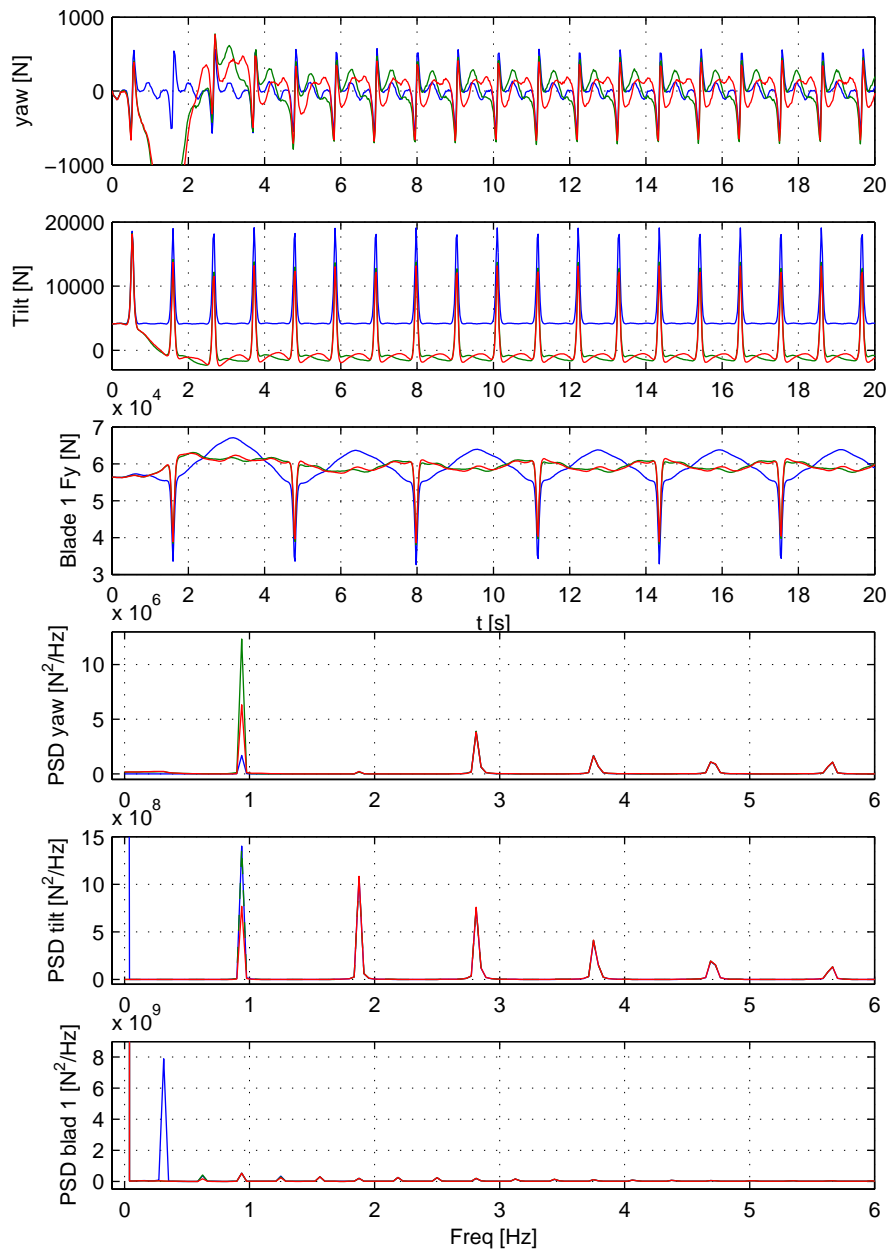
Formålet med denne test er at give et bedre indblik i, hvor meget den individuelle regulering kan reducere belastningerne i hhv. yaw- og tiltakserne samt kraften på bladene. Testen består af et plot som viser kræfterne i disse retninger både i tid og i frekvens. Samtidig vises et plot over ω_r , effekt, $\theta_{1_{ref}}$ og $\dot{\theta}_1$, hvor de to sidstnævnte er styresignalet til pitch-vinklen på et enkelt blad og hastigheden af pitchen, også set på et enkelt blad. Dette plot laves for at vise, hvilken effekt den individuelle pitch har på den kollektive regulering. Testen består af tre trin. Først vises kræfterne uden individuel regulering, derefter med individuel PI regulering i både yaw- og tiltakserne, og sluttelig indsættes P reguleringen som virker på 3p frekvensen. Reguleringen er både implementeret i simulink og i HAWC og derfor laves den samme test for begge implementeringer. Sluttelig laves en sammenligning mellem de to simulationsmodeller med den komplette individuelle regulering indsat.

Testen er udført med kollektiv PI regulering ved en vindhastighed på 14 *m/s* med en ruhedslængde på 50mm.

10.1.1 Individuel pitch i simulink

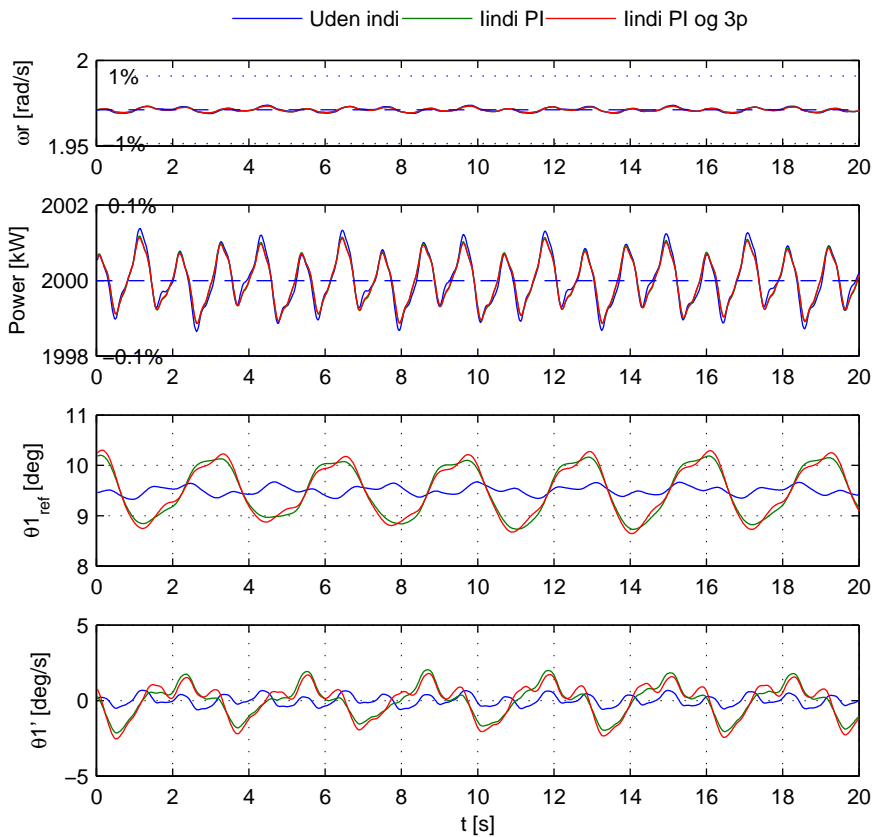
Det ses, hvorledes hhv. ω_r , effekt, θ_{1ref} og $\dot{\theta}_1$ ændres ved indførsel af først PI regulering af individuel pitch, og derefter tilføjes P regulering på 3p frekvensen. Det ses tydeligt at både ω_r og effekten kun påvirkes minimalt. θ_{1ref} , som er set på et enkelt blad, og $\dot{\theta}_1$ bliver mere aktiv. Yderligere ses øverst kræfterne på hhv. yaw- og tiltakserne samt kraften på et enkelt blad. Nederst ses en PSD af disse kræfter. Det ses, at PI reguleringen effektivt reducerer 0p komponenten i tilt-aksen samt 1p komponenten på bladet, men forstærker 3p komponenten i yaw-aksen. Denne komponent bliver igen reduceret med P reguleringen i 3p.

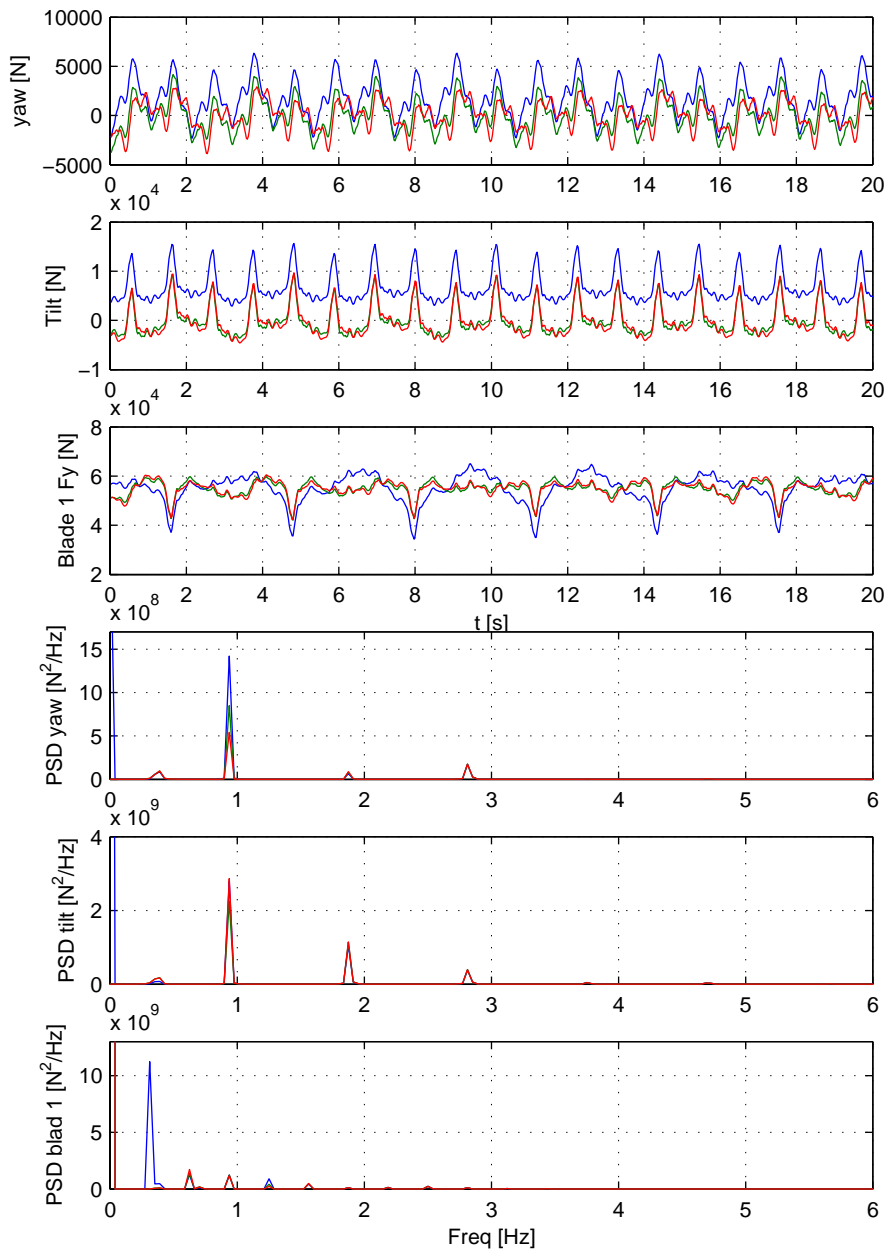




10.1.2 Individuel pitch i HAWC

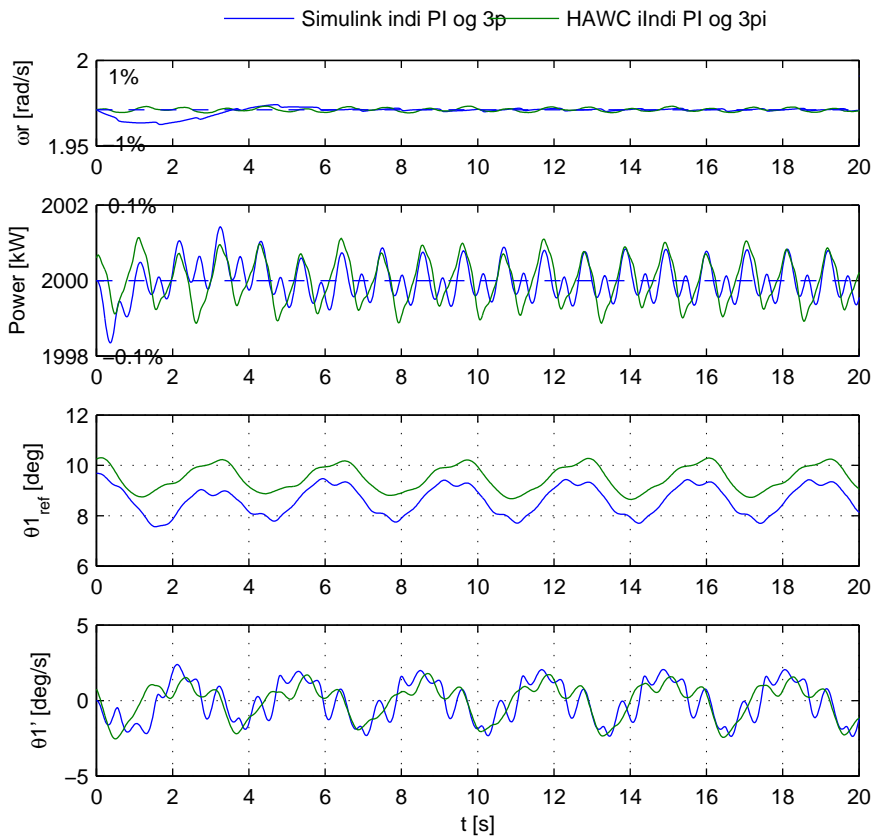
Samme test, som der blev lavet for individuel regulering af simulink modellen, laves nu for reguleringen implementeret i HAWC. Igen ses det, at der ikke er nogen bemærkelsesværdig ændring i ω_r og effekten. Ligeledes ses det, at θ_{1ref} og $\dot{\theta}_1$ bliver mere aktive. Yderligere bemærkes, at den individuelle regulering effektivt reducerer 0p komponenterne på yaw- og tiltakserne, og reducerer 3p komponenten i yaw-aksen. Det skal bemærkes, at HAWC også indeholder en kraftig 0p komponent på yaw-aksen, som bliver kraftigt reduceret af den individuelle regulering.

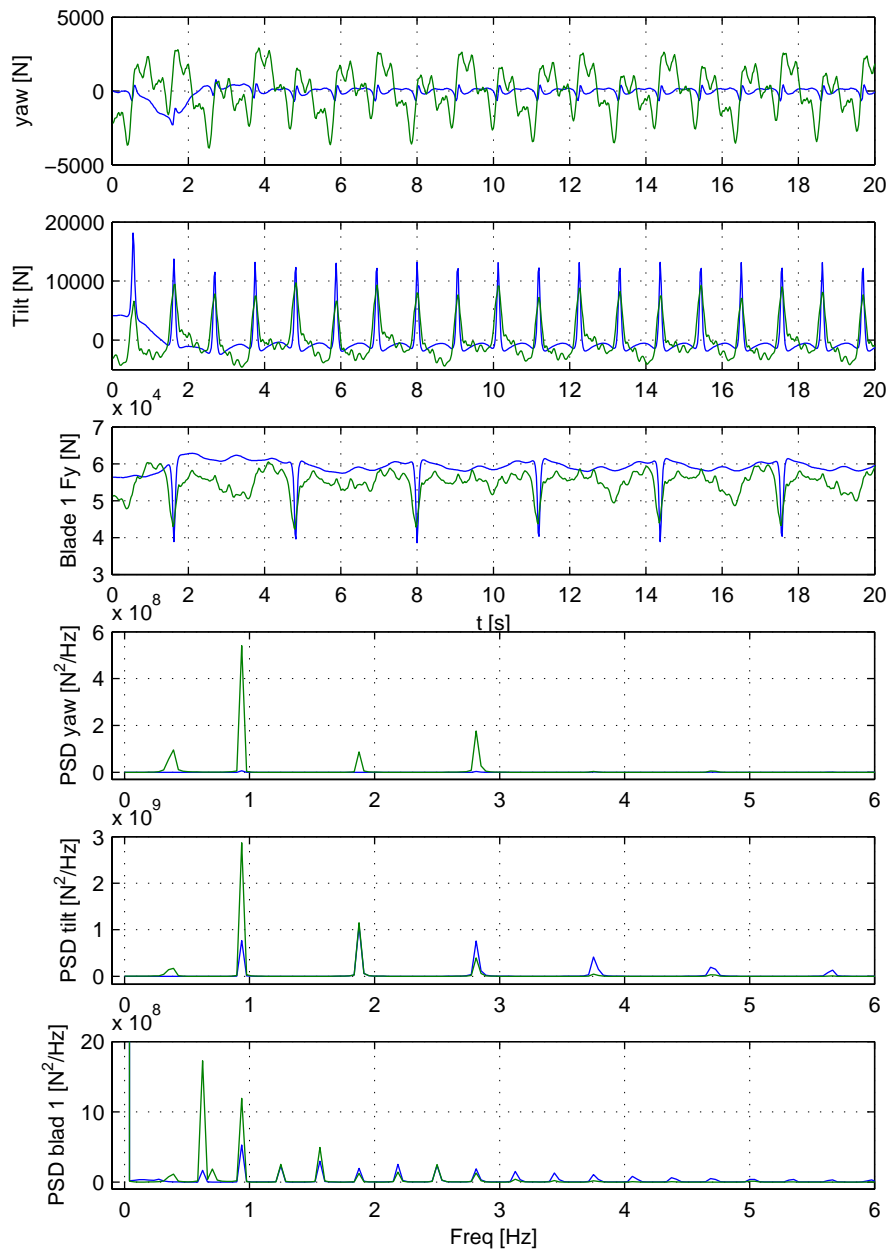




10.1.3 Sammenligning mellem simulink og HAWC

Af sammenligningen ses det, at HAWC simuleringen giver større belastninger end simulink simulationen, hvilket er at forvente. Selvom HAWC er betydelig mere kompleks, kan der stadig ses en god overensstemmelse mellem de to modeller og indvirkningen af den individuelle regulering.





Konklusion

Konklusionen er opdelt i tre dele. Først ses der på, hvilke resultater der er opnået med kollektiv pitch, derefter ses på resultaterne fra den individuelle regulering og sluttelig ses på, hvilke områder der kunne være interessante at gå mere i dybden med for fremtidige undersøgelser.

11.1 Kollektiv regulering

Der er lavet en model over vindmøllen, som består af en designmodel og en lidt mere kompleks simulationsmodel. Ud fra designmodellen er der designet 4 forskellige reguleringer: en PI og tre LQI regulatorer. Yderligere er der lavet Bumpless transfer mellem de forskellige operationsområder for vindmøllen for alle regulatorerne. Ved test af reguleringen er simulationsmodellen samt PI regulatoren verificeret op mod Risø's vindmøllemodel HAWC. De resterende regulatorer er ikke implementeret i denne model. Sammenligningen mellem reguleringen i simulationsmodellen fra dette projekt og HAWC viste, at der var god overensstemmelse. HAWC indeholder mere dynamik, hvilket bevirker, at man skal være forsigtig med at designe regulatorerne for hårdt, når den simple designmodel anvendes, da dette kan give anledning til ustabilitet, hvis designet overføres til HAWC.

PI reguleringen gav rimelige resultater for regulering af omdrejninger og effekt, se kapitel 5. Den gav dog ikke direkte mulighed for at reducere belastningen på drivakslen. LQI reguleringen gav betydeligt forbedrede resultater ved regulering af effekten. For LQI II og III er der mulighed for direkte at designe regulatoren med henblik på at reducere belastningen på drivakslen. Designes reguleringen efter dette kriterie sker det på bekostning af en mere varierende effekt. LQI III gav ikke en forbedret regulering over LQI II som forventet. De bedste resultater er således opnået ved at anvende en LQI II regulering, der har dynamikken i drivakslen med i designet. Dette design er dog ikke blevet verificeret i HAWC.

11.2 Individuel regulering

Til den individuelle regulering er der ligeledes lavet en design- og simulationsmodel. Selve den mekaniske struktur i modellen er den samme som den kollektive model, men den aerodynamiske del er udbygget, så vindmøllen kan påvirkes med et 3D vindfelt, og de enkelte blade kan betragtes hver for sig. Den individuelle regulering består af to PI- og to P regulatorer, der arbejder på bladkræfterne transformeret til yaw- og tiltakserne. Resultatet af den individuelle regulering kan ses i kapitel 10 og viser, at der er en klar reduktion i kræfterne i yaw- og tiltakserne. Der er opnået en kraftig reduktion af 0p kræfterne i begge akser, og 1p kraften på bladene er også reduceret betydeligt. Yderligere er der ved indførsel af P reguleringen opnået en beskedent reduktion i 3p kræfterne.

Modellen og den individuelle regulering er igen verificeret op mod HAWC, hvilket er gjort ved at betragte indvirkningen af et deterministisk vindfelt på de to modeller. Denne sammenligning kan ses i kapitel 10 og gav en god overensstemmelse mellem de to. Simuleringer i HAWC har betydelig større kraftpåvirkninger specielt i yaw-aksen. Der er ikke blevet lavet en stokastisk model af 3D vindfeltet, og der er derfor ikke noget grundlag for at sammenligne modellerne på dette område.

Det er at forvente, at de ikke kan være i total overensstemmelse mellem de to modeller, da HAWC er betydelig mere kompleks. Der er på trods af dette vist en god overensstemmelse mellem de to modeller ved indførsel af den individuelle regulering, hvilket viser, at reguleringen godt kan designes på den simple model og derefter implementeres på den mere komplekse. Her er det igen vigtigt at designet ikke bliver presset for hårdt, da dette gør reguleringen mindre robust over for modelændringer, som så kan give problemer, når reguleringen overføres til HAWC.

11.3 Perspektivering

Der er flere aspekter af projektet, der kunne være interessante at gå videre med for at se, hvilke forbedringer i reguleringen der kan opnås. Her listes og gennemgås de i prioriteret rækkefølge.

Kollektiv pitch:

- Inkludere tårndynamik i designmodel.
- Yderligere undersøgelse af design med aktuatorer.
- Robusthedsanalyse.
- HAWC implementering af LQI reguleringerne.

Individuel pitch:

- Inkludere tårndynamik i designmodel.
- Inkludere deterministisk vindmodel i designmodellen.
- Udvide vindmodellen med stokastisk variation.
- Medtage stokastisk vindmodel i designmodellen.
- Undersøge andre individuelle reguleringsmetoder.
- Designe LQ- eller LQI regulering.

For den kollektive regulering kan designmodellen udbygges, så den har tårndynamikken med i designet. Det blev set, at netop tårnsvingningen gav anledning til problemer, når der blev implementeret en regulator i HAWC, der var designet for hårdt. Da tårnsvingningen fra simulink modellen ikke stemmer helt overens med HAWC simuleringerne, kunne der med fordel laves en dybere undersøgelse af tårndynamikken, inden denne medtages i designmodellen.

Implementeringen af PI regulatoren i HAWC viste at en regulator optimeret på simulink modellen, gav anledning til ustabilitet. Det vil derfor være på sin plads at lave en robusthedsanalyse af designet. Der kan laves en mere dybdegående undersøgelse af, hvorfor reguleringen ikke blev forbedret af at medtage aktuatorer i designmodellen.

LQI design forventes at kunne forbedre reguleringen i HAWC, som det var tilfældet i simulink.

Da tårnsvingningen, som nævnt, har direkte indvirkning på kræfterne i flapvis retning, ville modellen for tårnet også med fordel kunne inkluderes i designmodellen for den individuelle pitch-regulering.

Det individuelle design bygger på en antagelse af, at der er et homogent vindfelt. Der vil muligvis kunne opnås en bedre regulering, hvis den kendte deterministiske model for vinden medtages i designet. Dette ville kræve, at reguleringen er afhængig af azimuth-vinklen. Regulering ud fra Colman transformationen vil i dette tilfælde nok ikke være den bedste løsning, da der ikke længere vil eksistere de symmetrier, der gør transformationen så simpel. Der kunne i stedet anvendes

den nævnte metode med at regulere de tre blade, hvor middelkraften mellem dem anvendes som reference. Giver dette positive resultater, kunne modellen udbygges, så designet også inkluderer den stokastiske variation.

Den individuelle regulering i dette projekt er lavet som PI og P regulering. Der kunne med fordel undersøges mere komplekse reguleringer som LQ og LQI regulering.

Litteraturliste

- [von Kármán 48] von Kármán T, Progress in Statistical Theory of Turbulence, *Proceedings of the National Academy of Science Vol. 34*, 1948.
- [Coleman 57] R.P. Coleman, A.M Feingold, Theory of Self-Excited Mechanical Oscillations of Helicopter Rotors with Hinged Blads, *NASA TN 3844*, NASA, Februar 1957.
- [Davenport 61] A.G. Davenport, The spectrum of horizontal gustiness near ground in high winds, *Q.J.R. Meteorol. soc. Vol 87:194 - 211*, 1961.
- [Kaimal 72] Kaimal, J.C, J.C Wyngaard, Y. Izumi, and O.R. Cote, Spectral Characteristics of Surface-Layer Turbulence, *Quarterly Journal of the Royal Meteorological Society Vol. 98*, 1972.
- [Risø 80] Peter S. Andersen, Ulrik Krabbe, Per Lundsager og Helge Petersen, Basismatriale for beregning af propelvindmøller *Risø-M-2153*, 1980.
- [Højstrup 82] Højstrup Jørgen, Velocity Spectra in the Unstable Planetary Layer, *Journal of the Atmospheric Science Vol. 39, No. 10*, 1982.
- [Knudsen 83] Torben Knudsen, Regulering af vindmøller, *IMSOR DTU*, 1983.

- [Veers 84] P.S. Veers, Modeling Stochastic Wind Loads on Vertical Axis Wind Turbines, *Technical Report SAND83-1909 Sandia National Laboratories, Albuquerque, NM* September 1984.
- [Winkelaar 92] D. Winkelaar, SWIFT Program for Three-Dimensional Wind Simulation Part 1: Model Description and Program Verification, *ECN-R-92-013*, December 1992.
- [Antti 94] Antti K. Piironen, Analysis of Volume Imaging Lidar Signals, *University of Wisconsin*, 1994.
- [Østergaard 94] Per Østergaard, Pitch-regulering af en Vindmølle, *IMM DTU*, 1994.
- [Manwell 02] J.F. Manwell, J.G. McGowan and A.L. Rogers, Wind Energy Explained, *ISBN 0-470-84615-7*, 2002.
- [Sørensen 02] P. Sørensen, A.D. Hansen, P. André, C. Rosas, Wind models for simulation of power fluctuations from wind farms, *Journal of Wind Engineering and Industrial Aerodynamics 90 1381-1402*, 2002.
- [Rosas 03] Pedro Rosas, Dynamic Influences of Wind Power on the Power System, *PhD thesis, Ørsted DTU*, 2003.
- [Reg2 04] Noter til undervisning i faget 31340, *Ørsted DTU*, 2004.
- [Hansen 05] Martin O.L Hansen, Aerodynamics of Wind Turbines, *ISBN 1-902916-06-9 DTU*, 2000.
- [Ahlström 05] Anders Ahlström, Aeroelastic Simulation of Wind Turbine Dynamics, *Doctoral Thesis from Royal Institute of Technology Department of Mechanics SE-100 44 Stockholm*, April 2005.
- [Risø 05] Morten H. Hansen, Anca Hansen, Torben J. Larsen, Stig Øye, Poul Sørensen and Peter Fuglsang, Control Design for a Pitch-Regulated, Variable-speed Wind Turbine, *ISBN 87-550-3409-8*, januar 2005.
- [Marshall 05] Marshall L. Buhl Jr., A New Empirical Relationship between Thrust Coefficient and Induction Factor for the Turbulent Windmill State, *NREL*, August 2005.
- [Engelen 05] K. Thomasen, H. Markou, T. Buhl (Risø), N.K. Poulsen (DTU), T.G. van Engelen, D. Winkelaar (ECN), B. Marrant (TUD), V. Politis (CRES), V. Riziotis & S. Voutsinas (NTUA), Morphological Study of Aerolastic Control Concepts for Wind Turbine, December 2005.
- [Kjølstad 06] Niels Kjølstad Poulsen, Discrete Time Control of Continuous Time Systems, *IMM DTU*, Januar 2006.

Symbolliste

P		$[W]$	Leveret elektrisk effekt.
P_{nom}	= 2e6	$[W]$	Nominel elektrisk effekt.
P_{ref}		$[W]$	Reference elektrisk effekt.
P_{err}		$[W]$	Fejlsignal på elektrisk effekt.
P_r		$[W]$	Effekten i vinden der rammer rotorarealet.
ω_g		$[rad/s]$	Omdrejningstal ved generator.
ω_{gnom}	= 167.6	$[rad/s]$	Nominelt omdrejningstal ved generator.
ω_{gmin}	= 50.27	$[rad/s]$	Minimum omdrejningstal ved generator.
ω_{gref}		$[rad/s]$	Reference omdrejningstal ved generator.
ω_{gerr}		$[rad/s]$	Fejlsignal på omdrejningstal ved generator.
ω_r		$[rad/s]$	Omdrejningstal ved rotor.
ω_{rnom}	= 1.97	$[rad/s]$	Nominelt omdrejningstal ved rotor.
ω_{rmin}	= 0.59	$[rad/s]$	Minimum omdrejningstal ved rotor.
λ		$[rad^{-1}]$	Tip speed ratio.
λ_{nom}		$[rad^{-1}]$	Nominelt tip speed ratio.
N_g	= 1:85		Gearudveksling.
H	= 80	$[m]$	Højde af tårn.
R	= 40	$[m]$	Længde af et blad.
J_r	= 8.7e6	$[kgm^2]$	Inertimomet i rotoren.
J_g	= 150	$[kgm^2]$	Inertimomet i generatoren.
J_{tot}	= 9.78e6	$[kgm^2]$	Total inertimomet.

r_{top}	=	1.2	[m]	Tårnradius top.
r_{rod}	=	2.15	[m]	Tårnradius bund.
Dtr	=	3.486	[m]	Afstand fra rotor- til tårncentrum.
m_t	=	200	[T]	Ækvivalent masse til tårnmodellering.
f_s	=	0.55	[Hz]	Resonansfrekvens for drivaksel.
K_s	=	1.039e8	[N/m]	Fjederkonstant i drivaksel.
D_s	=	1.039e6	[s^{-1}]	Dæmpningsfaktor i drivaksel.
f_t	=	0.3	[Hz]	Resonansfrekvens for tårn.
K_t	=	6.948e5	[N/m]	Fjederkonstant i tårn.
D_t	=	2.316e4	[s^{-1}]	Dæmpningsfaktor i tårn.
ω_n	=	8.88	[Hz]	Udæmpet egenfrekvens for pitch-aktuator.
ζ	=	0.9		Dæmpningsforhold for pitch-aktuator.
τ_{sv}	=	0.1	[s]	Tidskonstant i generator.
ρ	=	1.2	[kgm^{-3}]	Massefylde af luft.
α			[$grader$]	Angrebsvinkel.
v			[m/s]	Vindhastighed i navhøjde.
v_r			[m/s]	Ækvivalent vindhastighed over rotor areal.
v_m			[m/s]	Middel vindhastighed i navhøjde.
v_{rel}			[m/s]	Relativ vindhastighed.
T_r			[Nm]	Rotormomentet.
T_g			[Nm]	Generatormomentet.
T_{gref}			[Nm]	Reference generatormomentet.
θ			[$grader$]	Kollektiv pitch-vinkel.
θ_i			[$grader$]	Pitch-vinkel for blad i .
θ_{ref}			[$grader$]	Kollektiv reference pitch-vinkel.
θ_0			[$grader$]	Bladets smig.
$\dot{\theta}_{max}$	=	10	[deg/s]	Maks hastighed for pitch-ændring.
$\Delta\phi$			[$grader$]	Drivakslens vrid.
c_p				Rotorens effektkoefficient.
c_{pnom}				Rotorens nominelle effektkoefficient.
c_t				Rotorens kraftkoefficient.

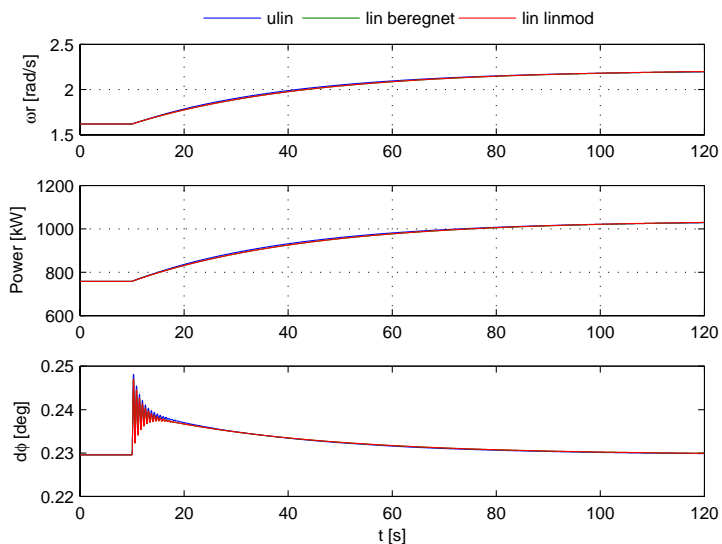
y_t		[m]	Tårnposition i y-aksen.
c_l			Lift koefficient.
c_d			Drag koefficient.
L		[N]	Lift.
D		[N]	Drag.
F_t		[N]	Kraft i y-aksen.
F_x		[N]	Kraft på et blad i x-aksen.
F_y		[N]	Kraft på et blad i y-aksen.
Q		[Nm]	Drejningsmoment på drivakslen.
B	=	3	Bladantal.
N	=	10	Antal bladelement inddelinger.
F			Tip loss faktor.
ϕ		[<i>grader</i>]	Azimut-vinkel.
z_0		[mm]	Ruhedslængde.
c		[m]	Bladkorde.
a			Aksial induktionsfaktor.
a'			Tangentiel induktionsfaktor.

Testplot

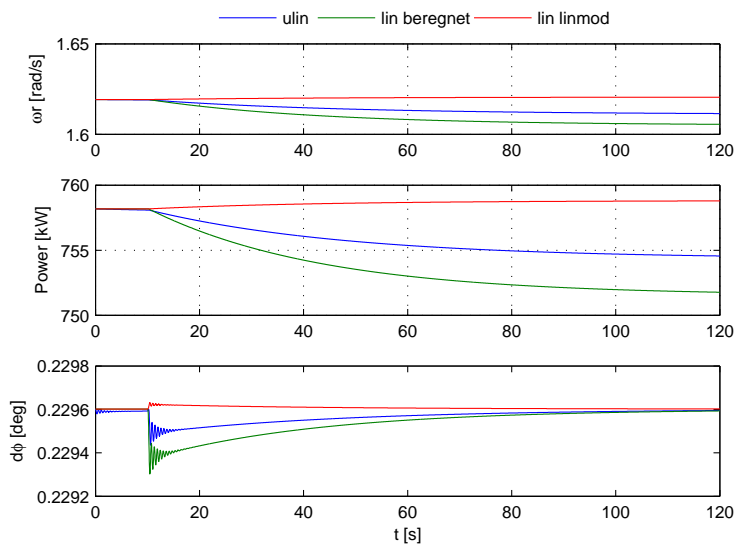
Liste over testplot:

- 1 Verifikation af lineær vindmøllemodel.
- 2 Bodeplot PI & P individuel.
- 3 Kollektiv test.

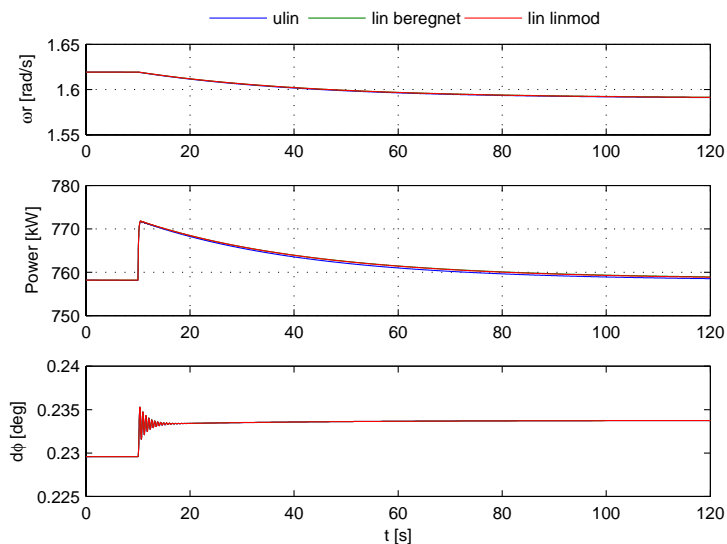
C.1 Verifikation af lineær vindmøllemodel



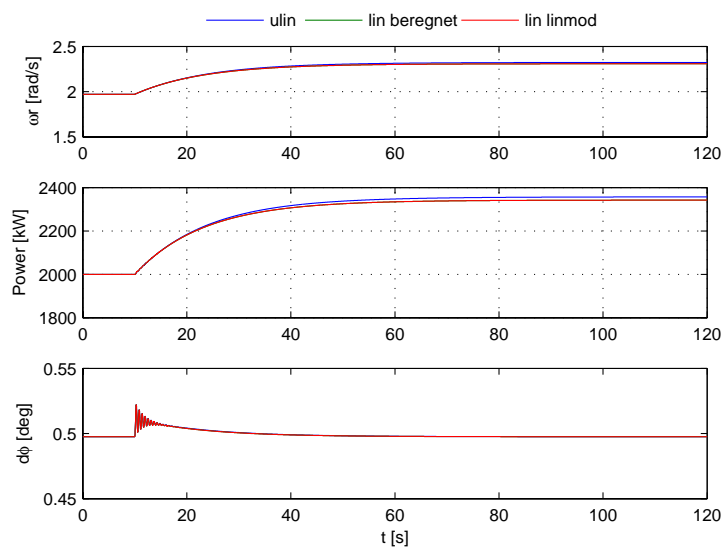
Figur C.1: Verifikation af lineær model ved 8 m/s. Trin på vind med amplituden 1 m/s til tiden 10 sek



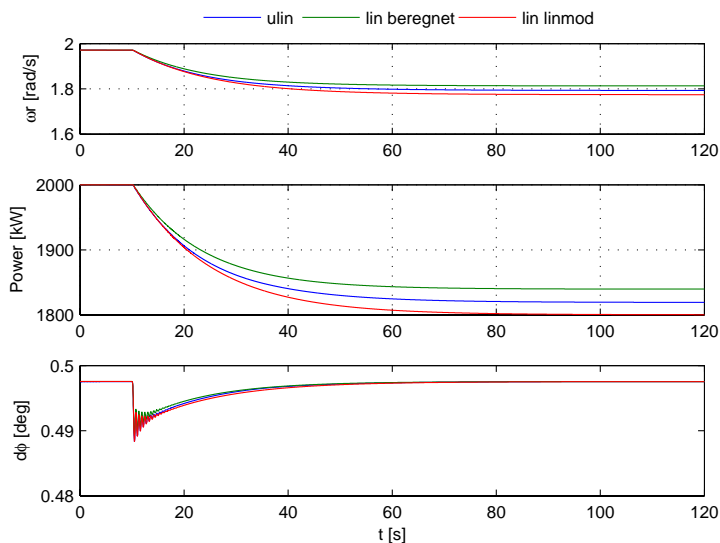
Figur C.2: Verifikation af lineær model ved 8 m/s. Trin på pitch med amplituden 1 grad til tiden 10 sek



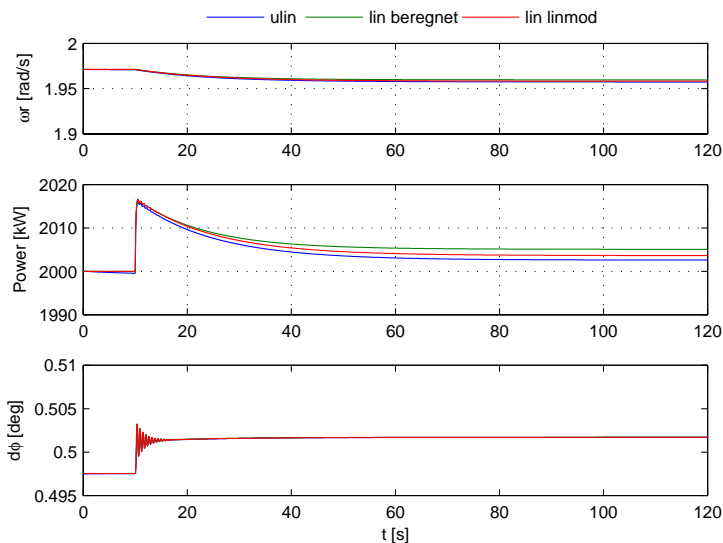
Figur C.3: Verifikation af lineær model ved 8 m/s. Trin på modmoment med amplitude 100 N til tiden 10 sek



Figur C.4: Verifikation af lineær model ved 14 m/s. Trin på vind med amplituden 1 m/s til tiden 10 sek

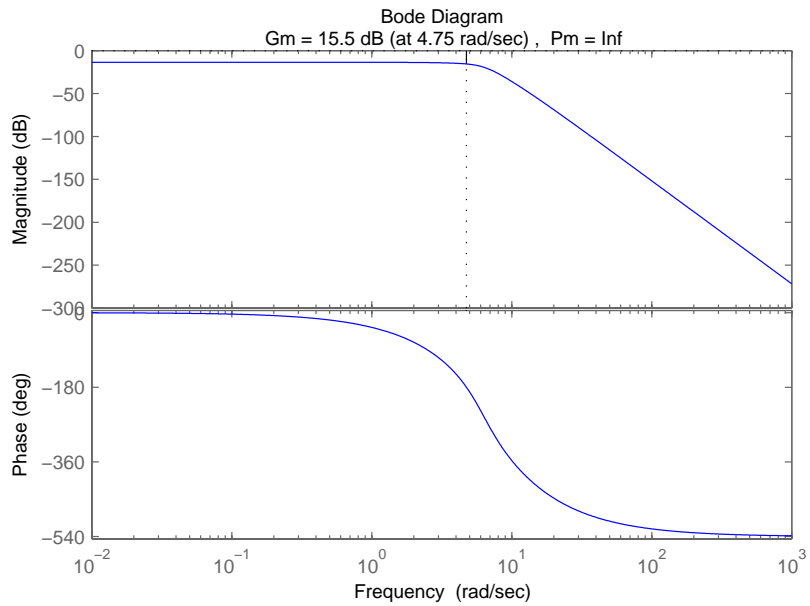


Figur C.5: Verifikation af lineær model ved 14 m/s. Trin på pitch med amplituden 1 grad til tiden 10 sek

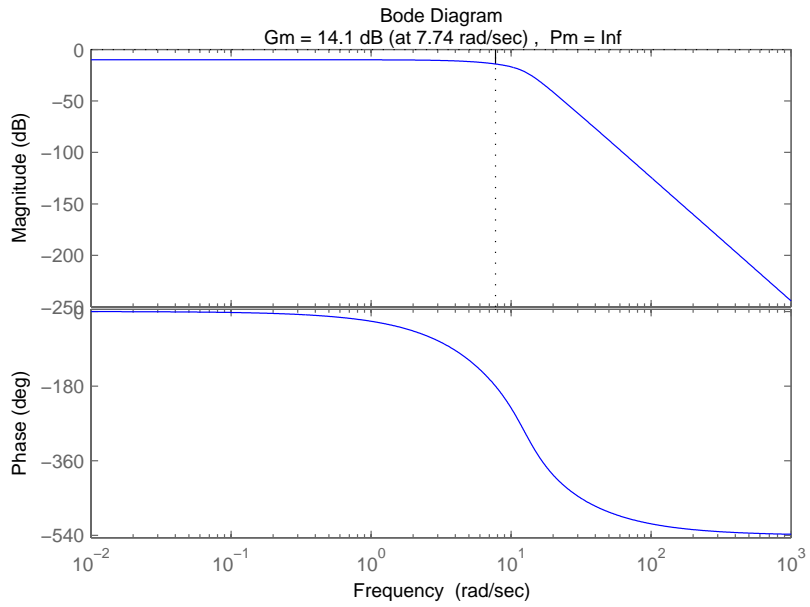


Figur C.6: Verifikation af lineær model ved 14 m/s. Trin på modmoment med amplitude 100 N til tiden 10 sek

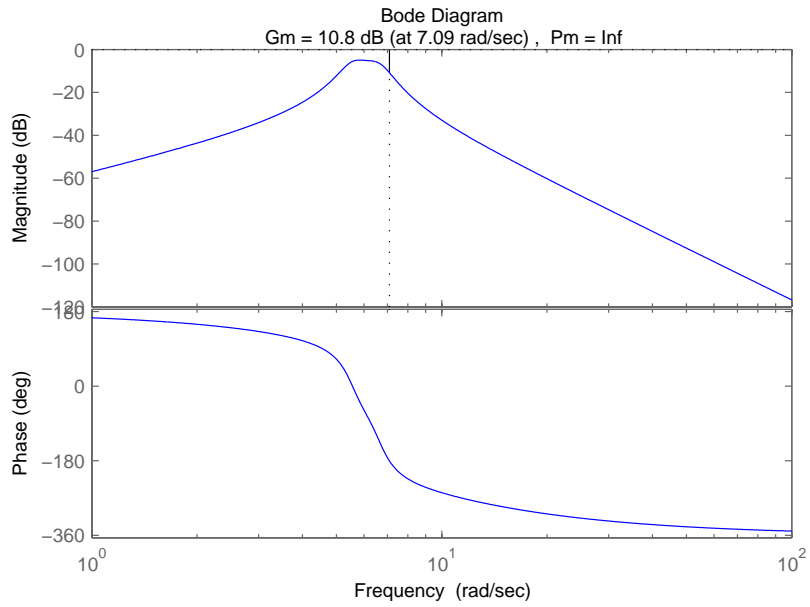
C.2 Bodeplot PI & P individuel



Figur C.7: Bodeplot af PI regulering. Tiltakse



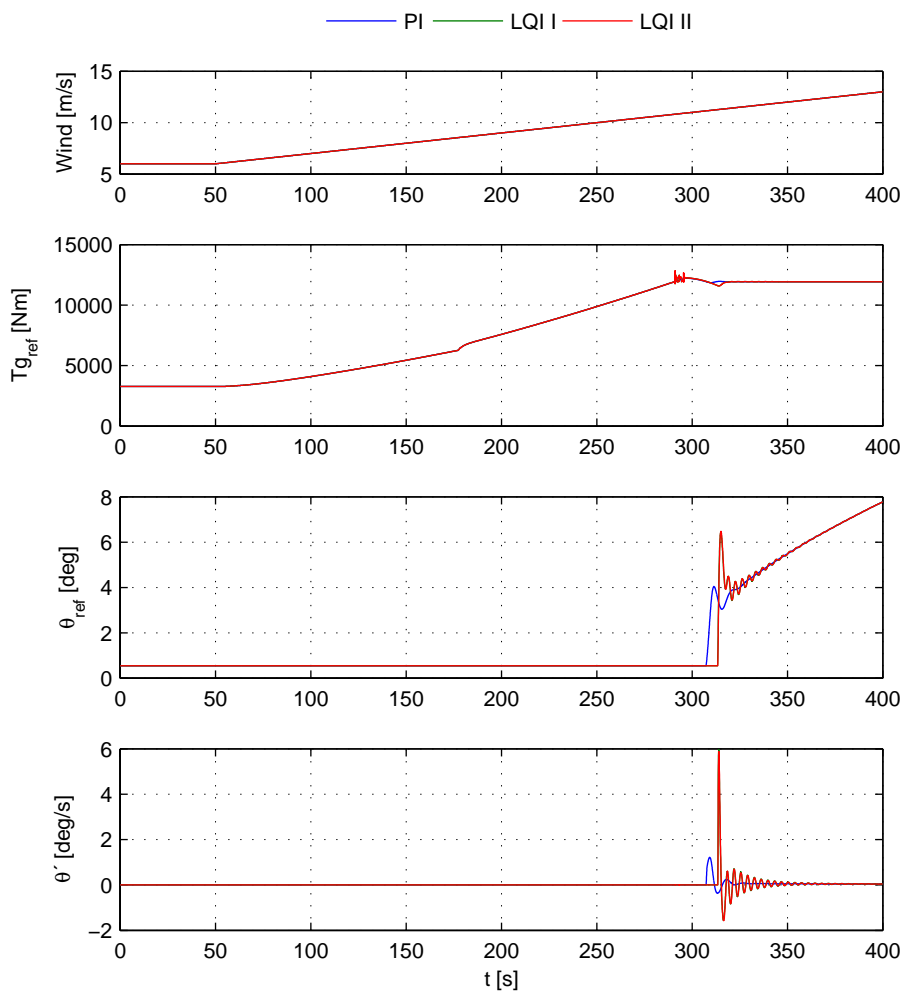
Figur C.8: Bodeplot af PI regulering. Yaw-akse

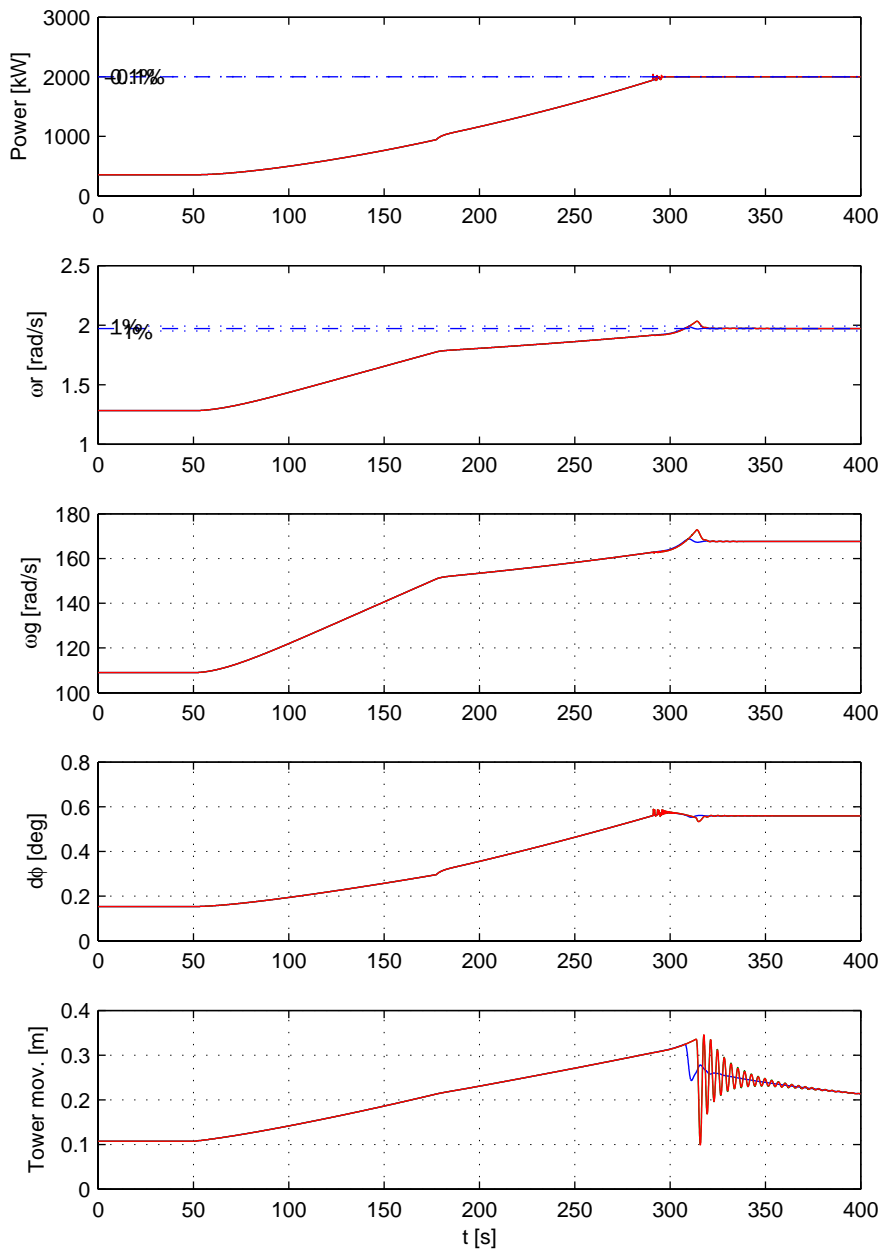


Figur C.9: Bodeplot af P regulering. 3p Yaw- & tiltakse

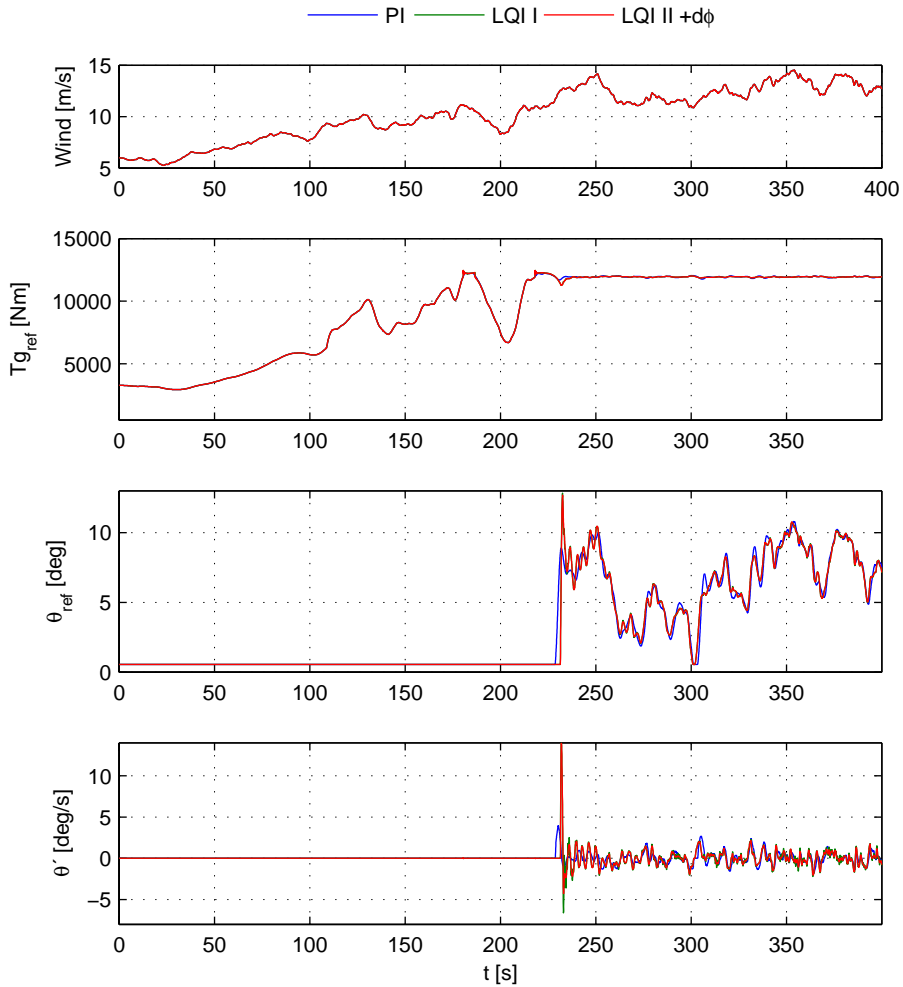
C.3 Kollektiv test

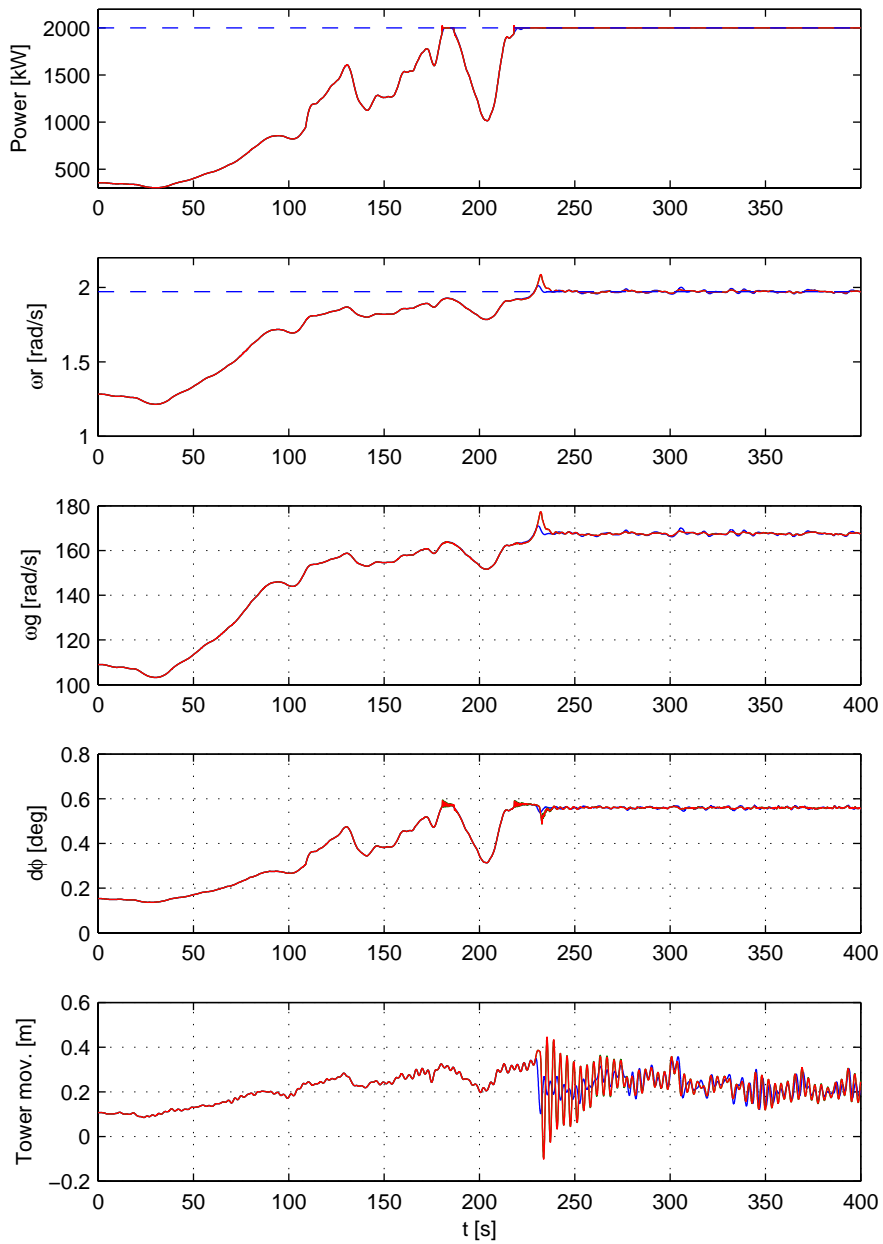
Test med vind stigende som rampe. Bemærk regulatorskiftet til tiden 300 s. I tiden indtil ca. 180 s effektorimeres ($p\omega$ området) herefter stiger effekten hurtigere og hastigheden langsommere (highområdet.).





Testen er foretaget med stokastisk vind stigende som en rampe. Turbulens intensiteten er 13.55%. Det vigtigste at bemærke i denne test er det store dyk i effekten omkring 200 s. Det er værd at undersøge om $p\omega$ kurven kan ændres og afhjælpe dette problem.





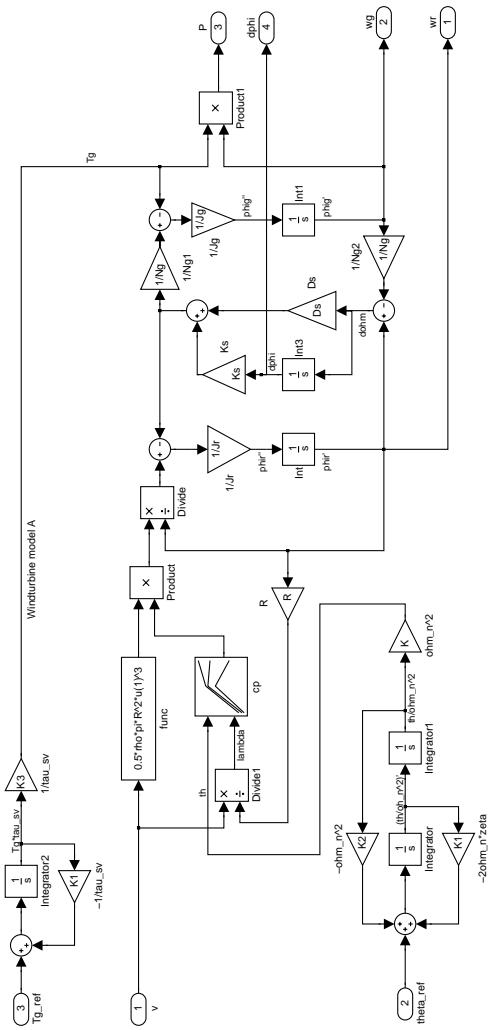
BILAG D

Simulink diagrammer

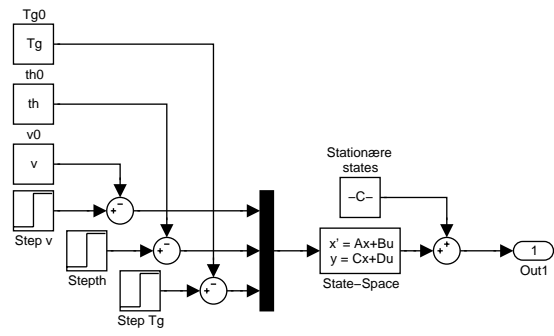
D.1 Liste over simulink modeller:

- 1 Komplet designmodel.
- 2 Komplet lineær designmodel.
- 3 LQI I designmodel.
- 4 LQI II designmodel.
- 5 LQI III designmodel.
- 6 PI implementering.
- 7 Individuel implementering.
- 8 LQI I implementering.
- 9 LQI II implementering.
- 10 LQI III implementering.

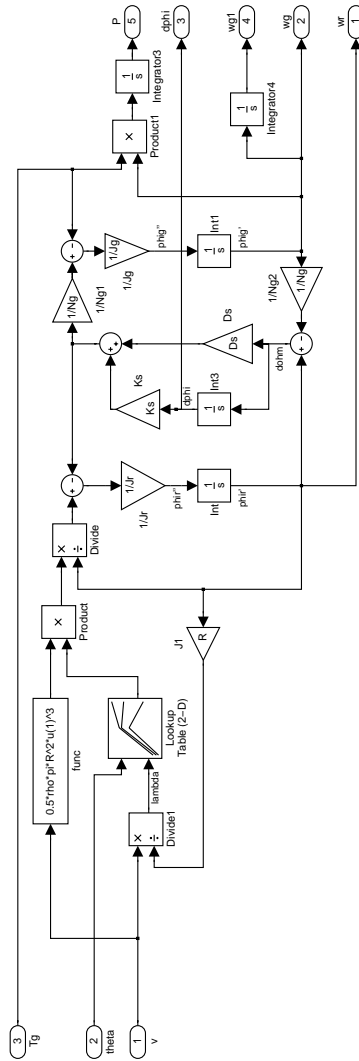
D.1.1 Komplet designmodel



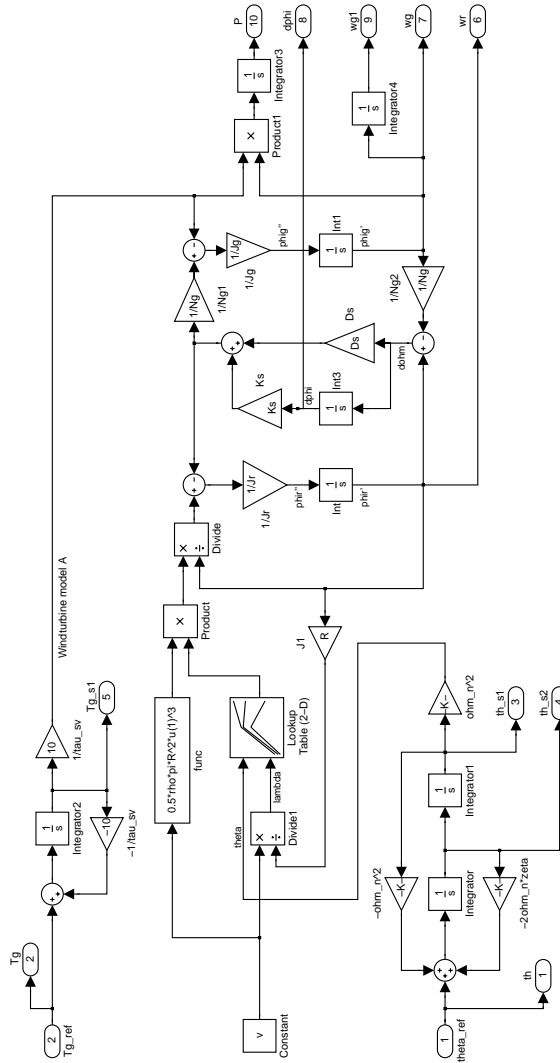
D.1.2 Komplet lineær designmodel



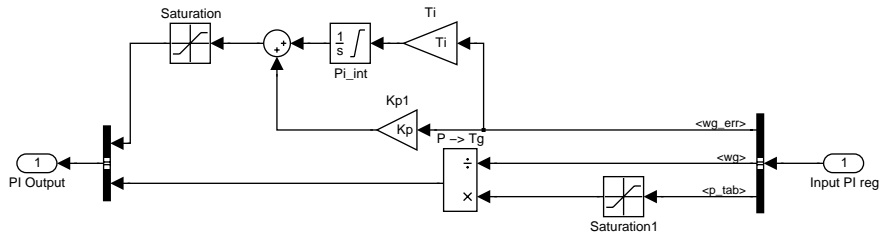
D.1.4 LQI II designmodel



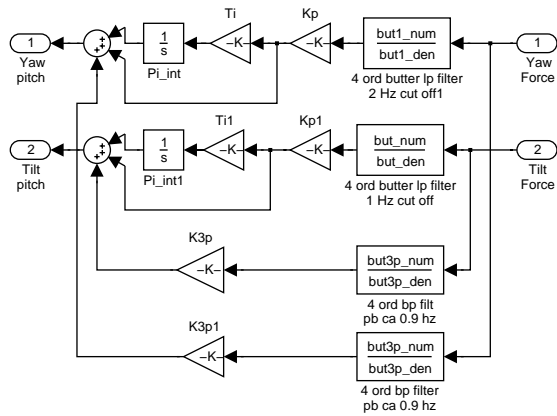
D.1.5 LQI III designmodel



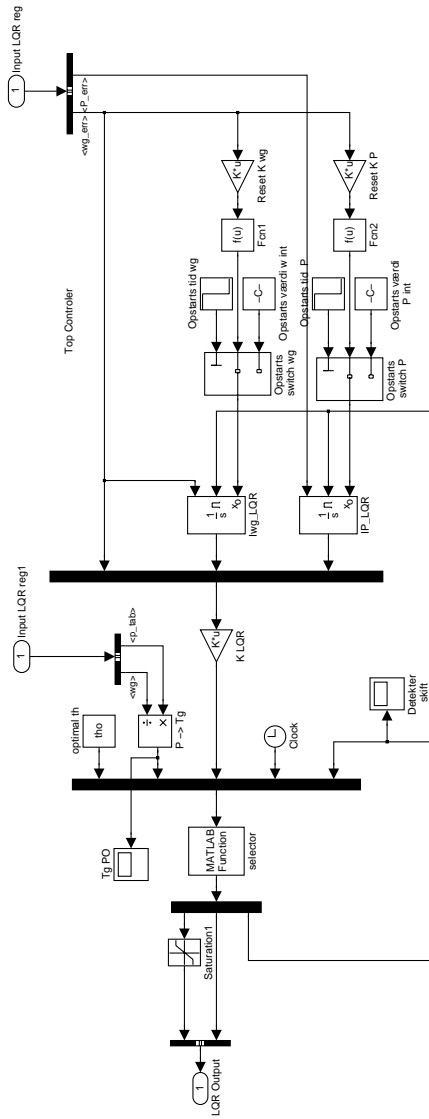
D.1.6 PI implementering



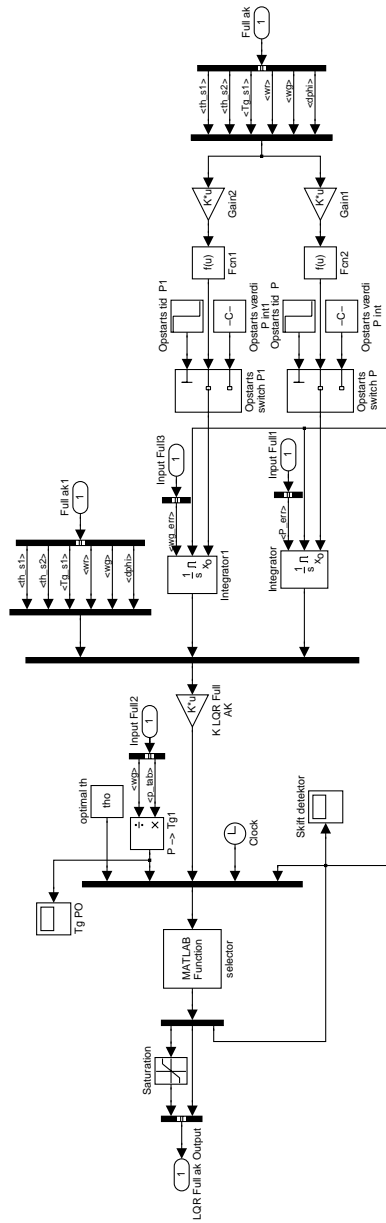
D.1.7 Individuel implementering



D.1.8 LQI I implementering



D.1.10 LQI III implementering



MatLab kode

E.1 Liste over anvendte MatLab filer

- 1 RUN_WT_BEM_IP.m.
- 2 syspar.m
- 3 Regulator_full.m
- 4 Regulator_PI.m
- 5 vind_model.m
- 6 speclin.m
- 7 Aktuator_sys.m
- 8 filters.m
- 9 BEM.m
- 10 lin_BEM.m
- 11 DVM.m
- 12 coleman.m
- 13 invColeman.m
- 14 PO_tabel.m
- 15 selector_full.m
- 16 init_figdata.m
- 17 sim2figdata.m
- 18 HAWC2figdata.m
- 19 stpkt0.m

20 fopt.m
 21 deriv.m
 22 Prsol.m

E.1.1 RUN_WT_BEM_IP.m

```

global tho maxcp wrnom Pnom lamo rho R Ng cptable Fail
global Tgnom reg_Top reg_po wr th
global old_mode_LQR old_mode_Full old_mode_Full_ak
global T1_LQR T2_LQR T1_Full T2_Full T1_Full_ak T2_Full_ak
global shear_mdl shadow_mdl shear_lin shear_z0 H Tr_r Tr_t Dtr
global int_top_wg_start int_top_P_start
global output_sim t_sim figdata
global azim_start

clc;
load wt_2M_BEM.mat           % CP/CT: wt_v29 / wt_2M / wt_2M_BEM_fine
HAWC_data                   % Bladstruktur og aerodynamik fra HAWC
syspar;                      % Generelle data om møllen, tårn ...
filters;                     % Filtre til individuel pitch
fopt(0);                     % Find ss værdier mm. udfra CP data
Aktuator_sys;               % Hent aktuator dynamik
para_wind=wind_model(0.135);% Beregn koef. til Højstrupfilter
torque_dyn = 1;              % 0 = ideel / 1 = Aktuator (ALTID 1)
Fail = [];                   % Opret BEM fejl rapportering.
T1_LQR = inf;                % initialisering af var. til skift
T2_LQR = 0;                  % initialisering af var. til skift
T1_Full = inf;               % initialisering af var. til skift
T2_Full = 0;                 % initialisering af var. til skift
T1_Full_ak = inf;           % initialisering af var. til skift
T2_Full_ak = 0;             % initialisering af var. til skift
[wg_po,P_po] = P0_tabel(.35, .9, 0); %wg-P tabel
azim_start = 0;              % Hvis der ikke køres HAWC data ind.

%***** kontrol af plot *****
name = 'LQI II +d\phi';      % navn i legend på plots
numb = 1;                    % placering i struktur
plot_sel = [1];              % placeringer der skal plottes
scal = 1;                    % Tilføjer procent / promille liner

init_figdata                 % Initialisere struktur til figur data
%figdata = hawc2figdata(figdata,'HAWC',3); % indlæs data fra HAWC

%***** MODEL DYNAMIK *****
gear_dyn = 1;                % 0 = stiv aksel / 1 = elastisk aksel
tower_dyn = 1;               % 0 = stift tårn / 1 = Tårn dynamik
pitch_dyn = 1;               % 0 = ideel aktuator / 1 = aktuator dynamik
blade_mdl = 0;               % 0 = CP/CT kurve baseret / 1 = BEM blad beregning.
reg_loop = 3;                % 0 = openloop / 1 = PI / 2 = LQR / 3 = LQR Full / 4 = AK.
vind_mdl = 1;                % 0 = punktvind / 1 = stokastisk punktvindvind 2D
shadow_mdl = 1;              % 0 = ingen tårnskygge / 1 = tårnskygge model.
shear_mdl = 0;               % 0 = ingen shear / 1 = lineær / 2 = log
shear_lin = 0.1;             % 0.2 = 20% forskel mellem 60 og 140 meter.
shear_z0 = 0.05;            % .0002= stille vand / .008 = lavt græs / 0.05 = kornmark [m]
indi_reg = 0;                 % 0 = ingen individuel regulering / 1 = individuel regulering
indi_reg3p = 0;              % 0 = ingen extra regulering ved 3p / 1 = 3p regulering

%***** Opsætning af input *****
v = 6;                        % Vindhastighed
sim_res = 0.025;              % Opløsning i simuleringen

```

```

sim_time = 40;          % Tid simuleringen kører

%SLOPE INPUT
slope_sim = 0.0;      % Rampe hældning
slope_t = 0;         % tid inden rampe starte
%TRAPPE INPUT
sim_step = [0 1];    % Angiver step sekvens eks. [1 2 -4 ....]
%BEGRÆNS INPUT
input_min = 0;       % min
input_max = 25;      % max

%***** Generer input *****
t_trappe = sim_time/length(sim_step);    %Længde på trin i sek.
t_vec = round([0 : t_trappe : sim_time]./sim_res).*sim_res;
sim_input_t = []; sim_input_v = [];      % opret tomme vektorer

for i = (2:(length(t_vec)-1))
    sim_input_t = [sim_input_t t_vec(i)-sim_res t_vec(i)];
    sim_input_v = [sim_input_v sim_step(i-1) sim_step(i-1)];
end
sim_input_t = [0 sim_input_t sim_time-sim_res];
sim_input_v = [sim_input_v sim_step(end) sim_step(end)];

%***** INT reset værdier *****
Regulator_PI;          % Beregn Regulator parametre
Regulator_LQR;        % Beregn Regulator parametre
Regulator_Full;       % Beregn Regulator parametre
Regulator_Full_ak;    % Beregn Regulator parametre
[wr,th,Tg,lam,cp,P]=stpkt0(v);

%***** INT opstarts værdier *****
[wr,th,Tg,lam,cp,P]=stpkt0(v);          %Beregning af stationære punkter
%TARN
ct0=interp2(cptable.lam',cptable.th',cptable.ct,lam,th); %beregning af kraftten på ...
F0=0.5*rho*pi*R^2*v^2*ct0;             % tårnet ved en given vind
int_tower=F0/Kt;                        %init værdi for int i tårn
%TORQUE Aktuator
Tg_start = Tg/C.torque;
th_start = [th/wn^2 0];

%OPSTARTS MODE
if (P == Pnom)
    old_mode_LQR = 0;    % 1 = po 0 = top sæt start betingelse korrekt
    old_mode_Full = 0;
    old_mode_Full_ak = 0;
else
    old_mode_LQR = 1;
    old_mode_Full = 1;
    old_mode_Full_ak = 1;
end

%***** SIMULERING *****
tic
[t_sim x_reg output_sim] = sim('WT_BEM_IP.mdl',[0 sim_time-sim_res],[,],...
                               [sim_input_t' sim_input_v']);
toc                               %Udskriver simulations tid

% %til frekvens analyse.
% freq1 = fft(output_sim([1000:length(output_sim(:,16))],16),4096);
% PSD_tilt = freq1.*conj(freq1)./1024;
% freq1 = fft(output_sim([1000:length(output_sim(:,17))],17),4096);
% PSD_yaw = freq1.*conj(freq1)./1024;
% freq1 = fft(output_sim([1000:length(output_sim(:,13))],13),4096);
% PSD_blade_tilt = freq1.*conj(freq1)./1024;
% freq1 = fft(output_sim([1000:length(output_sim(:,18))],18),4096);

```

```

% PSD_blade_rot = freq1.*conj(freq1)./4096;
% f = (1/sim_res)*(0:256)./4096;

%***** Plot *****
if (Fail ~= 0) %BEM Fejlrapport
    disp(' ')
    disp(' Konvergerede ikke!');
    disp(' Tid Radius alfa a a^')
    disp(' ')
    disp(Fail)
    disp(' Tid Radius alfa a a^')
elseif (blade_mdl == 1)
    disp('Alle BEM beregninger konvergerede :-)')
end

P_plot = Pnom/1000; w_plot = wrnom; Tg_plot = Tgnom;

figdata = sim2figdata(figdata,name,numb);

figure(1)
subplot(3,1,1,'align')
plotfig(figdata,'wr',plot_sel,'t')
if(scal==1)
    grid off; hold on
    plot([0 figdata.x(end)],[w_plot w_plot], '--');
    plot([0.05*figdata.x(end) figdata.x(end)],[w_plot*0.99 w_plot*.99], ':');...
    text(0,w_plot*.99,' -1%', 'fontsize',8)
    plot([0.05*figdata.x(end) figdata.x(end)],[w_plot*1.01 w_plot*1.01], ':');...
    text(0,w_plot*1.01,' 1%', 'fontsize',8)
    hold off;
end
subplot(3,1,2,'align')
plotfig(figdata,'p',plot_sel)
if(scal==1)
    grid off; hold on;
    plot([0 figdata.x(end)],[P_plot P_plot], '--');
    plot([0.05*figdata.x(end) figdata.x(end)],[P_plot*0.999 P_plot*.999], ':');...
    text(0,P_plot*.999,' -0.1%', 'fontsize',8)
    plot([0.05*figdata.x(end) figdata.x(end)],[P_plot*1.001 P_plot*1.001], ':');...
    text(0,P_plot*1.001,' 0.1%', 'fontsize',8)
    hold off;
end
subplot(3,1,3,'align');
plotfig(figdata,'v',plot_sel,'b')
grid on;

figure(2)
subplot(3,1,1,'align')
plotfig(figdata,'thref',plot_sel,'t')
grid on
subplot(3,1,2,'align')
plotfig(figdata,'thdt',plot_sel)
grid on
if(scal==1)
    grid on; hold on;
    plot([0.05*figdata.x(end) figdata.x(end)],[Pitch_r_lim Pitch_r_lim], ':');...
    text(0,Pitch_r_lim,' limit', 'fontsize',8)
    plot([0.05*figdata.x(end) figdata.x(end)],[ -Pitch_r_lim -Pitch_r_lim], ':');...
    text(0,-Pitch_r_lim,' limit', 'fontsize',8)
    hold off;
end
subplot(3,1,3,'align')
plotfig(figdata,'tgrep',plot_sel,'b')
grid on

```

E.1.2 syspar.m

```

% -----
% Test mølle - Data
% -----

rho=1.25;           % Air density
R=80/2;            % Length of the blade
H=80;              % Hub heighth.
Tr_t = 1.2;        % Tårn radius top
Tr_r = 2.15;       % Tårn radius bund (root)
Dtr = 3.486;       % Afstand fra rotor centrum til tårn centrum.
Ng=85;             % gear ration
Jr=8.7e6;          % Rotor inertti fra Ref.
Jg=150;            % Generator inertti fra Ref
Jt=Jr+Ng^2*Jg;    % totale inertti

fs=0.55;           % Fixed-free resonance frequency drive shaft
Ks=(fs*2*pi)^2*Jr; % Spring constant of drive shaft
Ds=Ks/100;         % Damping

mt=250e3;          % Mass of tower. Groft estimat oprindeligt 200 ton men 250 passer bedre
ft=0.3;            % Tower resonas frequency, data fra r-1500 rapport
Kt= mt*(2*pi*ft)^2; % Spring constant of tower
Dt= Kt/30;         % Damping of tower ESTIMAT.

Pnom=2e6;          % Nominel Effekt
rpm_nom=1600;      % Nominel rpm, Generator side
wgnom=rpm_nom/60*2*pi; % Nominle omdrejningshastighed generator rad/s
wrnom=wgnom/Ng;    % Nominel rotor omdrejnings hastighed rad/s
alfa=0.3;          % wr_low=alfa*wrnom Minimum omdrejnings hastighed generator
%alfa=0;           % Full range generator
Tgnom = Pnom/wrnom/85; % Nominelt modmoment
Pitch_r_lim = 10;  % max pitch vel. deg/s

```

E.1.3 Regulator_full.m

```

%Top linarisering-----
[wr,th,Tg,lam,cp,P]=stpkt0(12); %Beregning af stationære punkter
wg = wr*Ng;
dphi = (Tg/Ks)*Ng;
inp=[th Tg];
[A2 B2 C2 D2] = linmod('Lin_LQR_Full',[wr wg dphi 0 0],inp);

%***** LQR REG *****
%LQI II
R2 = diag([.15/th^2 .1/Tg^2 10/wr^2 0/wg^2 0/dphi^2 100/wg^2 3000/P^2]);
%LQI II +d_dpi
%R2 = diag([.15/th^2 .1/Tg^2 10/wr^2 0.00001/wg^2 10/dphi^2 100/wg^2 3000/P^2]);
%LQI II +d_dpi -effekt
%R2 = diag([.15/th^2 .1/Tg^2 10/wr^2 0.00001/wg^2 10/dphi^2 100/wg^2 1000/P^2]);

Q = C2'*R2*C2;
N = C2'*R2*D2;
R2 = D2'*R2*D2;
reg_Full = lqr(A2,B2,Q,R2,N);
reg_Full = -reg_Full;

%Find udtryk til beregning af start værdier.
[wr,th,Tg,lam,cp,P]=stpkt0(v);

```

```

wg = wr*Ng;
syms wrs wgs dphis PIs wgIs th_sym Tg_sym
inp_vec = [wrs wgs dphis wgIs PIs];
eq1 = sym(inp_vec*reg_Full(1,:)'-th_sym);
eq2 = sym(inp_vec*reg_Full(2,:)'-Tg_sym);
int_start_val = solve(eq1,eq2,PIs,wgIs); %PIs og wgIs er states i regulatoren

%Find start værdier.
wrs = wr;
dphis = (Tg/Ks)*Ng;
if(gear_dyn==0), dphis = 0; end
wgs = wg;
th_sym = th;
Tg_sym = Tg;
P_start = subs(int_start_val.PIs);
wg_start = subs(int_start_val.wgIs);

%Resetværdier
syms wrs wgs dphis PIs wgIs th_sym Tg_sym
th_sym = tho;
Tg_sym = Tgnom;
P_reset = subs(int_start_val.PIs);
wg_reset = subs(int_start_val.wgIs);
k_int_P0 = subs(P_reset,{wrs, wgs, dphis},{0,0,0});
k_int_P(1) = (subs(P_reset,{wrs, wgs, dphis},{1,0,0}))-k_int_P0;
k_int_P(2) = (subs(P_reset,{wrs, wgs, dphis},{0,1,0}))-k_int_P0;
k_int_P(3) = (subs(P_reset,{wrs, wgs, dphis},{0,0,1}))-k_int_P0;

k_int_wg0 = subs(wg_reset,{wrs, wgs, dphis},{0,0,0});
k_int_wg(1) = (subs(wg_reset,{wrs, wgs, dphis},{1,0,0}))-k_int_wg0;
k_int_wg(2) = (subs(wg_reset,{wrs, wgs, dphis},{0,1,0}))-k_int_wg0;
k_int_wg(3) = (subs(wg_reset,{wrs, wgs, dphis},{0,0,1}))-k_int_wg0;

```

E.1.4 Regulator_PI.m

```

%***** PI REG *****
K3p=0.0001;

Kp = 1; %1; %til SIM 1.5;
Ti = 0.8; %2; %til SIM 1.25
%Ti = 0.3; %med indsat lp filter på w

%Kp_indi = 0.0003; %uden filter
%Ti_indi = 0.001;
Kp_indi = 0.000013; %0.8 hz cut off butter
Ti_indi = 0.000047;
%Kp_indi = 0.00002; %2 hz cut off butter
%Ti_indi = 0.0001;
%Kp_indi = 0.000025; %4 hz cut off butter
%Ti_indi = 0.00015;
%Kp_indi = -0.00005; %8 hz cut off butter
%Ti_indi = -0.000238;
%Kp_indi = 0.00007; %16 hz cut off butter
%Ti_indi = 0.00035;

%Kp_indi = 0.0003; %4 hz cut off ellip
%Ti_indi = 3.7500e-004;
%Kp_indi=0.00013;
%Ti_indi=6.5000e-004;

```


E.1.5 vind_model.m

```

function [para_wind] = wind_model(tau)

%*****%
% filename: spec_par.m %
% Using linear model H(s) to %
% approximate effective wind speed %
% by comparing spectrum Se(f) and %
% spectrum for H(s) %
% Se(f): spectral density function of %
% effective wind speed %
% k %
% H(s) = ----- %
% (1+p1*s)(1+p2*s) %
%*****%
% Codes is made by Henrik K.J. %
% Modified by Allan J. Larsen %

global omega specdata
global H R

h=H;
hi=1000;
r=R;
%vm=[17]';
vm=[1:1:25]';

k1=h/(1+15*h/hi);
k2=sqrt(pi)*r;

omega=logspace(-3,1,100);
f=omega/(2*pi);

k=zeros(length(vm),1);
p1=zeros(length(vm),1);
p2=zeros(length(vm),1);
var=zeros(length(vm),1);

th0=[7;0.7];
for i=1:length(vm)
    Sp=(22*tau^2*vm(i)*k1)/((1+33*k1/vm(i)*f)^(5/3));
    F=((1+8/3*k2/vm(i)*f)*(1+4*k2/vm(i)*f));
    Se=Sp./F;

    k(i)=22*tau^2*vm(i)*k1;
    kv(i)=sqrt(k(i));
    specdata=20*log10(Se/k(i));

    opt=optimset('fminsearch');
    opt.Display='notify';
    th=fminsearch('spec_lin',th0,opt);
    th0=th;
    p1(i)=th(1);
    p2(i)=th(2);

% if vm(i)<40,
% PHIE=20*log10(Se);
% Sv=(kv(i)^2)/((1+p1(i)^2*omega.^2)*(1+p2(i)^2*omega.^2));
% PHIV=20*log10(Sv);
%
% semilogx(omega,PHIE,'r',omega,PHIV,'--'); grid;
% title(['Power spectrum (vm=',num2str(vm(i)), ' m/s)'])
% xlabel('Frequency [rad/sec]'),ylabel('Gain [db]')

```

```
% pause(.1),
% end
end
para_wind = [vm kv' p1 p2];
```

E.1.6 speclin.m

```
function err=spec_lin(p)
%*****%
% filename: spec_lin.m %
% Linear function H(s) %
% 1 %
% H(s) = ----- %
% (1+p2*s)(1+p2*s) %
% p: parameters [p1 p2]' %
% err: sum of the squared residuals %
%*****%

global omega
global specdata

Sv=1./((1+p(1)^2*omega.^2).*(1+p(2)^2*omega.^2));
q=20*log10(Sv)-specdata;
err=sum(q.^2);
end
```

E.1.7 Aktuator_sys.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Pitch aktuator (fra model v29const.m) %%%%%%%%%%%
wn=8.88;
xsin=0.9;
num_pitch=[wn^2];
den_pitch=[1 2*xsin*wn wn^2];
A_pitch=[0 1; -wn^2 -2*xsin*wn];
B_pitch=[0; 1];
C_pitch= [wn^2 0];
D_pitch=[0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Torque aktuator for generator 1. ord. med en tidskonstant på 0.1 %%%%%%%%%%%
A_torque=-10; %rettet fra 10
B_torque=1;
C_torque=10; % rettet fra 10
D_torque=0;
tau_sv = 0.1;
```

E.1.8 filters.m

```
%Filtre til Yaw og tilt samt digital implementering i HAWC
%implementeres som anden ordens kaskade koblede filtre.
clc

%Analogt butterworth LP filter bruges i simulink
```

```

% 1 Hz. LP filter
[but_num,but_den]=butter(4,1*(2*pi),'s');
figure(8);
bode(but_num,but_den);
hold on

%digitalt til implementering i HAWC
[a,b]=butter(4,1/20);          %0 < 1 Hz < 20 Hz -> 0<1/20<1
[LP_sos]= tf2sos(a,b) ;      %fra overførings funktion til 2. ordens kaskade koblede
g=1;
flt1 = tf(LP_sos(1,[1:3]),[LP_sos(1,[4:6])],0.025); % test om de to filtre passer.
flt2 = tf(LP_sos(2,[1:3]),[LP_sos(2,[4:6])],0.025);
flt = (g*flt1*flt2);
bode(flt);
hold off
format long;
disp('LP 1.0 Hz.')
disp('første filter')
disp(LP_sos(1,:))
disp('andet filter')
disp(LP_sos(2,:))
format short;

%Analogt butterworth LP filter bruges i simulink
[but1_num,but1_den]=butter(4,2*2*pi,'s');
figure(9);
bode(but1_num,but1_den);
hold on

%digitalt til implementering i HAWC
% 2Hz. LP filter
[a,b]=butter(4,2/20);          %0 < 2 Hz < 20 Hz -> 0<2/20<1
[LP_sos]= tf2sos(a,b) ;      %fra overførings funktion til 2. ordens kaskade koblede
g=1;
flt1 = tf(LP_sos(1,[1:3]),[LP_sos(1,[4:6])],0.025); % test om de to filtre passer.
flt2 = tf(LP_sos(2,[1:3]),[LP_sos(2,[4:6])],0.025);
flt = (g*flt1*flt2);
bode(flt);
hold off;
format long;
disp('LP 2.0 Hz.')
disp('første filter')
disp(LP_sos(1,:))
disp('andet filter')
disp(LP_sos(2,:))
format short;

%Analogt butterworth BP filter bruges i simulink
[but3p_num,but3p_den]=butter(2,[0.85*(2*pi) 1.1*(2*pi)],'s'); % [0.9 0.98]figure(10);
bode(but3p_num,but3p_den);
hold on

%digitalt til implementering i HAWC
[a,b]=butter(2,[0.85 1.1]/20);          %0 < 0.8 Hz < 20 Hz -> 0<0.8/20<1
[BP_sos]= tf2sos(a,b);          %fra overførings funktion til 2. ordens kaskade koblede
g=1;
flt1 = tf(BP_sos(1,[1:3]),[BP_sos(1,[4:6])],0.025); % test om de to filtre passer.
flt2 = tf(BP_sos(2,[1:3]),[BP_sos(2,[4:6])],0.025);
flt = (g*flt1*flt2);
bode(flt);
hold off

format long;
disp('BP 0.8 Hz.')
disp('første filter')

```

```

disp(BP_sos(1,:))
disp('andet filter')
disp(BP_sos(2,:))
format short;

```

E.1.9 BEM.m

```

function [out]=BEM(input)
% Blade Element Momentum.
%
% Beregner drejningsmoment og kraft i vindretning på tårn.
% input 1: vind (10 punkter)
%         r = [3.14 7.02 10.90 14.78 18.66 22.54 26.42 30.30 34.18 38.06]
% input 2: pitch vinkel th
% input 3: rotor omdrejnings hastighed wr
% input 4: blad vinkel i forhold til azimut.
% input 5: simulations tid, som bruges til fejlrapportering.
% output 1: Torque for blad
% output 2: thrust i vind retningnen.

global Fail blade

dQ=[];data=[];Cp=[];per=[];it_done=0;Ct = 0;
alfa_old=[0 0];
rho = 1.225; %Luft densitet
R = 40; %Rotor radius
Rh = 1.2; %Radius Hub part
tol = 0.01; %Tolerance for itteration af a og a'.
sek = 10; %Sekmenter bladet indeles i
dr = (R-Rh)/sek; %Bladelement længde
r = [Rh+(dr/2):dr:R-(dr/2)];
sigma = 3*blade.cord./(2*pi*r);
a_old = [0.1 0.1];
am_old = [0.01 0.01];
%shear_fak = 5;
%th = tho;
%wr = 2;
%v = repmat(input(1),1,10);
v = input(1).*DVM(input(4),r);
th = input(2);
wr = input(3);
time = input(5);

for i = 1:sek;
stop = 0;
it = 1;
a = a_old(1)+(a_old(1)-a_old(2));
a_ = am_old(1)+(am_old(1)-am_old(2));
while(stop~=1)
a = min(a,0.9999999);
a = max(a,0.0000001);
phirad = atan( (v(i)*(1-a)) / ((wr*r(i))*(1+a_)) );
phideg = atan( (v(i)*(1-a)) / ((wr*r(i))*(1+a_)) )*(360/(2*pi));
alfa = phideg-(blade.tw(i)+th);
F = (2/pi)*(acos(exp(-(((3/2)*(1-r(i)/R)) / ((r(i)/R)*sin(phirad))))));

Cl = blade.cl(round((alfa*100)+18001));
Cd = blade.cd(round((alfa*100)+18001));
Cn = Cl*cos(phirad)+Cd*sin(phirad);
a1 = 1/( ((4*F*(sin(phirad)^2))/(sigma(i)*Cn))+1 );
a_1 = 1/((4*F*cos(phirad)/(sigma(i)*Cl))-1);

```

```

if (Cn<=0)
    a1 = 0.0000001;
end
if (a1 > 0.4)
    Ct = (8/9) + (4*F-(40/9))*a + ((50/9)-4*F)*a^2; %fra Note incl F
    a1 = 1/2/(-50*(sin(phirad))^2+36*(sin(phirad))^2*F+9*sigma(i)*Cn)*(-40*(...
        sin(phirad))^2+36*(sin(phirad))^2*F+18*sigma(i)*Cn-6*(-48*(sin(phirad))...
        ^4*F+18*(sin(phirad))^2*sigma(i)*Cn+36*(sin(phirad))^4*F^2)^(1/2));
end
if( (a^2*(1+tol) > a1^2 & a^2*(1-tol) < a1^2 & ...
    a_^2*(1+tol) > a_1^2 & a_^2*(1-tol) < a_1^2 )
    data(i,:) = [r(i) alfa it F Ct a a_ phidegl];
    stop = 1;
    it=1;
else
    a = a1;
    a_ = a_1;
    stop = 0;
    if (it == 100)
        Fail = [Fail ;time r(i) alfa a a_];
        data(i,:) = [r(i) alfa it F Ct a a_ Cn];
        %disp('Konvergere ikke bruger estimat fra 2 tidligere')
        stop = 1;
    end
    it=it+1;
end
end
alfa_old = [alfa alfa_old(1)];
a_old = [a a_old(1)];
am_old = [a_ am_old(1)];
if(i == 1)
    a_old(2) = a;
    am_old(2) = a_;
end
vrel(i) = v(i)*(1-a)/(sin(phirad));
dQ(i) = (F*0.5*rho*vrel(i)^2*(Cl*sin(phirad)-Cd*cos(phirad))*blade.cord(i)*r(i)*dr);
dF(i) = (F*0.5*rho*vrel(i)^2*(Cl*cos(phirad)+Cd*sin(phirad))*blade.cord(i)*dr);
end
torque = sum(dQ);
thrust = sum(dF);
out = [torque, thrust];

```

E.1.10 lin_BEM.m

```

function [out out1]=lin_BEM(input,sel)
% Blade Element Momentum beregning af vindmølle blad.
% Beregner drejningsmoment, kraft i vindretning på tårn.
% input er vind (10 punkter), pitch vinkel th, rotor omdrejnings
% hastighed wr.
% sel er input er angiver hvilken af v=1 th=2 wr=3 Tg=4 der skal laves diff med hensyn til
% vinden der bruges til diff er en midlet vind over hele vingen

% Vinden skal opgives i 10 punkter fordelt over et 40 meter blad:
% r = [3.14 7.02 10.90 14.78 18.66 22.54 26.42 30.30 34.18 38.06]

global Fail blade;

HAWC_data
syspar;

dQ=[];data=[];Cp=[];per=[];it_done=0;Ct = 0;Fail = [];

```

```

alfa_old=[0 0];
rho      = 1.225;           %Luft densitet
R        = 40;             %Rotor radius
Rh       = 1.2;           %Radius Hub part
tol      = 0.01;          %Tollerance for itteration af a og a'.
sek      = 10;            %Sekmenter bladet indeles i
dr       = (R-Rh)/sek;
r        = [Rh+(dr/2):dr:R-(dr/2)];
sigma    = 3*blade.cord./(2*pi*r);
a_old    = [0.1 0.1];
am_old   = [0.01 0.01];

v0 = repmat(input(1),1,10);
th0 = input(2);
wr0 = input(3);

if(sel==1)
    v=v0-0.1;th=th0;wr=wr0;
end
if(sel==2)
    v=v0;th=th0-0.1;wr=wr0;
end
if(sel==3)
    v=v0;th=th0;wr=wr0-0.01;
end
if(sel==4)
    v=v0;th=th0;wr=wr0;
end

torque_vec=[];
thrust_vec=[];
mid_v_vec=[];
th_vec=[];
wr_vec=[];

for k=1:5

    for i = 1:sek;
        stop = 0;
        it = 1;
        a = a_old(1)+(a_old(1)-a_old(2));
        a_ = am_old(1)+(am_old(1)-am_old(2));
        while(stop~=1)
            a = min(a,0.9999999);
            a = max(a,0.0000001);
            phirad = atan( (v(i)*(1-a)) / ((wr*r(i))*(1+a_)) );
            phideg = atan( (v(i)*(1-a)) / ((wr*r(i))*(1+a_)) )*(360/(2*pi));
            alfa = phideg-(blade.tw(i)+th);
            F = (2/pi)*(acos(exp( -(((3/2)*(1-(r(i)/R))) / ((r(i)/R)*sin(phirad))))));

            Cl = blade.cl(round((alfa*100)+18001));
            Cd = blade.cd(round((alfa*100)+18001));
            Cn = Cl*cos(phirad)+Cd*sin(phirad);
            a1 = 1/( ((4*F*(sin(phirad)^2))/(sigma(i)*Cn))+1 );
            a_1 = 1/((4*F*cos(phirad)/(sigma(i)*Cl))-1);
            if (Cn<=0)
                a1 = 0.0000001;
            end
            if (a1 > 0.4)
                Ct = (8/9) + (4*F-(40/9))*a + ((50/9)-4*F)*a^2;           %fra Note incl F
                a1 = 1/2/(-50*(sin(phirad))^2+36*(sin(phirad))^2*F+9*sigma(i)*Cn)*(-40*(...
                    sin(phirad))^2+36*(sin(phirad))^2*F+18*sigma(i)*Cn-6*(-48*(sin(phirad))...
                    ^4*F+18*(sin(phirad))^2*sigma(i)*Cn+36*(sin(phirad))^4*F^2)^(1/2));
            end
        end
    end
end

```

```

if( (a^2*(1+tol) > a1^2 & a^2*(1-tol) < a1^2 & ...
    a_~2*(1+tol) > a_1^2 & a_~2*(1-tol) < a_1^2) )
    data(i,:) = [r(i) alfa it F Ct a a_ phideg];
    stop = 1;
    it=1;
else
    a = a1;
    a_ = a_1;
    stop = 0;
    if (it == 50)
        Fail(i,:) = [th r(i)];
        [r(i) alfa it F Ct; a a_ 0 a1 a_1];
        data(i,:) = [r(i) alfa it F Ct a a_ Cn];
        disp('Konvergere ikke bruger estimat fra 2 tidligere')
        stop = 1;
    end
    it=it+1;
end
end
alfa_old = [alfa alfa_old(1)];
a_old = [a a_old(1)];
am_old = [a_ am_old(1)];
if(i == 1)
    a_old(2) = a;
    am_old(2) = a_;
end
vrel(i)=v(i)*(1-a)/(sin(phirad));
dQ(i) = (F*0.5*rho*vrel(i)^2*(Cl*sin(phirad)-Cd*cos(phirad))*blade.cord(i)*r(i)*dr);
dF(i) = (F*0.5*rho*vrel(i)^2*(Cl*cos(phirad)+Cd*sin(phirad))*blade.cord(i)*dr);
end
torque = sum(dQ);
thrust = sum(dF);
torque_vec = [torque_vec torque];
thrust_vec = [thrust_vec thrust];

if(sel==1)
    mid_v_vec = [mid_v_vec sum(v)/length(v)];
    v=v+0.05;
end
if(sel==2)
    th_vec = [th_vec th];
    th=th+0.05;
end
if(sel==3)
    wr_vec = [wr_vec wr];
    wr=wr+0.005;
end
end
if(sel==1)
    dtorquedv = deriv(torque_vec,mid_v_vec,sum(v0)/length(v0));
    dthrustdv = deriv(thrust_vec,mid_v_vec,sum(v0)/length(v0));
    out =dtorquedv/Jt;
    out1=dthrustdv;
end
if(sel==2)
    dtorquedth = deriv(torque_vec,th_vec,th0);
    dthrustdth = deriv(thrust_vec,th_vec,th0);
    out =dtorquedth/Jt;
    out1=dthrustdth;
end
if(sel==3)
    dtorquedwr = deriv(torque_vec,wr_vec,wr0);
    out =dtorquedwr/Jt;
    out1=0;
end
end

```

```

if(sel==4)
    dtorquedTg = Ng/Jt;
    out =dtorquedTg;
    out1=0;
end

```

E.1.11 DVM.m

```

function [output] = DVM(a,r)
% Generere lineær shaer på vingen udfra blades vinkel fra azimuth.
% input a:      blad vinkel i grader.
% input r:      radier hvor shear skal beregnes.
% output shear_g: Gain som skal ganges på vindfeltet for at generere shear effekt.
%
global shear_mdl shadow_mdl shear_lin shear_z0 H Dtr Tr_r Tr_t
H_c = 5;          %cone højde

az = a/(180/pi);
z = cos(az);
x = sin(az);
sampz = r.*z;
sampx = r.*x;
dts =sqrt((sin(az).*r).^2+Dtr^2);
beta = acos(Dtr./dts);
if(shear_mdl == 1 )
    shear_g = 1+sampz*(shear_lin/80);
elseif(shear_mdl == 2)
    shear_g = ((log((sampz+H)/shear_z0)) ./ (log(H/shear_z0)));
else
    shear_g = ones(1,length(r));
end

if(shadow_mdl == 1)
    for i = 1:length(r)
        if (sampz(i) > -H_c)
            radius(i) = max((-1.2/H_c)*sampz(i)+0,0);    %Simuleret med cone på toppen
        else
            %beregnet afstand mellem blad og tårn når tårn er konisk
            radius(i) = ((Tr_t-Tr_r)/H)*sampz(i) + Tr_t;
        end
    end
    output = shear_g .* (1+radius.^2.*( (sampx.^2-Dtr^2)./(sampx.^2+Dtr^2).^2));
else
    output = shear_g;
end

```

E.1.12 coleman.m

```

function [out]=Coleman(inp)

asi = inp(4)*2*pi/360;
asi1 = (inp(4)+120)*2*pi/360;
asi2 = (inp(4)+240)*2*pi/360;
shift = 0*2*pi/360;

out1 = 1*inp(1) + cos(asi+shift)*inp(2) + sin(asi+shift)*inp(3);
out2 = 1*inp(1) + cos(asi1+shift)*inp(2) + sin(asi1+shift)*inp(3);

```



```
out3 = 1*inp(1) + cos(asi2+shift)*inp(2) + sin(asi2+shift)*inp(3);

out=[out1 out2 out3];
```

E.1.13 invColeman.m

```
function [out]=invColeman(inp)

asi = inp(4)*2*pi/360;
asi1 = (inp(4)+120)*2*pi/360;
asi2 = (inp(4)+240)*2*pi/360;

out1 = 1/3*inp(1) + 1/3*inp(2) + 1/3*inp(3);
out2 = 2/3*cos(asi)*inp(1) + 2/3*cos(asi1)*inp(2) + 2/3*cos(asi2)*inp(3);
out3 = 2/3*sin(asi)*inp(1) + 2/3*sin(asi1)*inp(2) + 2/3*sin(asi2)*inp(3);

out=[out1 out2 out3];
```

E.1.14 PO_tabel.m

```
function [wg_po, p_po] = po_tabel(low, high, plt)
% M file for PO reg.
% [wr,p] = po_tabel(low,high,plot)
% input low indicates were po curve is cut to zero
% input high were it it put to Pnom (2MW)
% high and low are in percent (eks. .35 .95)
global maxcp lamo rho R Pnom wrnom

if nargin<=2, plt = 0; end
%high = 0.95;
high2= 0.99;
wr_vec = [low:0.005:high].*wrnom;
P_vec = (0.5*rho*pi*(R^2)*maxcp*((lamo*R.*wr_vec).^3));
wr_vec = [0.3*wrnom wr_vec wrnom*high2];

wg_po = [wr_vec 2.3].*85; % omregnig til wg i stedet for wr
p_po = [0 P_vec Pnom*1.1 Pnom*1.1];

if (plt == 1)
    %sammenlign.
    figure(4)
    plot(wg_po,p_po,'LineWidth',2)
    % hold on
    % plot(wr.*60*85/(2*pi),power./1000,'r')
    xlabel('rpm [rad(sec.)')
    ylabel('power [W]')
    hold off
end
```

E.1.15 selector_full.m

```
function [output]=selector_Full(input)
```

```

global tho Tgnom T1_Full T2_Full old_mode_Full
%*****
% input 1: th po
% input 2: Tg po
% input 3: th top
% input 4: Tg top
% input 5: time
% input 6: change OK
% output 1: th control signal
% output 2: Tg control signal
% output 3: Styre signal til Top integrator

%SKIFTE BETINGELSE
if (input(2) <= Tgnom)
    reg_mode = 1;           %PO
else
    reg_mode = 0;         %TOP
end

%HYSTERESE
time = input(5);
if(old_mode_Full~= reg_mode)           % Skift detekteret
    if (time > T2_Full & time > T1_Full & (input(3) < tho*1.1))           % Skift tilladt
        new_mode = reg_mode;
        T1_Full = inf;
        T2_Full = time+1.0;           %Tid 2 er l s efter skift (5 sek)
    elseif (time > T2_Full & T1_Full == inf)           %Skift IKKE tilladt start T1_Full opt lling
        new_mode = old_mode_Full;
        T1_Full = time+0.0;           %Tid 1 er l s inden skift sker (1 sek)
    else
        new_mode = old_mode_Full;
    end
else
    %Ikke sket et skift
    new_mode = old_mode_Full;
    T1_Full = inf;
end

%SET OUTPUT
old_mode_Full = new_mode;
if(new_mode == 1)
    output(3) = 1;           %PO
else
    output(3) = 0;           %TOP
end

if (input(6)==0)
    output([1 2]) = input([3 4]); %TOP
else
    output([1 2]) = input([1 2]); %PO
end

```

E.1.16 init_figdata.m

```

% function [figdata] = init_figdata(figdata)

figdata=[];

figdata.ax = 't [s]';
figdata.wr.ax = '\omegar [rad/s]';
figdata.wg.ax = '\omegag [rad/s]';
figdata.p.ax = 'Power [kW]';

```

```

figdata.ty.ax = 'Tower mov. [m]';
figdata.tydt.ax = 'Tower acc. [m/s]';
figdata.dp.ax = 'd\phi [deg]';
figdata.vp.ax = 'Wind point [m/s]';
figdata.v.ax = 'Wind [m/s]';
figdata.th1.ax = '\theta_{ref} 1 [deg]';
figdata.th2.ax = '\theta_{ref} 2 [deg]';
figdata.th3.ax = '\theta_{ref} 3 [deg]';
figdata.th.ax = '\theta [deg]';
figdata.thref.ax = '\theta_{ref} [deg]';
figdata.thdt.ax = '\theta' [deg/s]';
figdata.tg.ax = 'Tg [Nm]';
figdata.tgref.ax = 'Tg_{ref} [Nm]';
figdata.tilt.ax = 'Tilt [N]';
figdata.yaw.ax = 'yaw [N]';
figdata.total.ax = 'Coleman mom [N]';
figdata.bf1.ax = 'Blade 1 Fy [N]';
figdata.bf2.ax = 'Blade 2 Fy [N]';
figdata.bf3.ax = 'Blade 3 Fy [N]';
figdata.bm1.ax = 'Blade 1 Qy [Nm]';
figdata.bm2.ax = 'Blade 2 Qy [Nm]';
figdata.bm3.ax = 'Blade 3 Qy [Nm]';

```

E.1.17 sim2figdata.m

```

function [figdata]=sim2figdata(figdata,set_name, set_num)
global output_sim t_sim
% set_name = 'LQR II'
% set_num = 2;

figdata.x = t_sim;
figdata.legend(set_num) = {set_name};

figdata.wr.y(:,set_num) = output_sim(:,1);
figdata.wg.y(:,set_num) = output_sim(:,2);
figdata.p.y(:,set_num) = output_sim(:,3)/1000;
figdata.ty.y(:,set_num) = output_sim(:,4);
figdata.tydt.y(:,set_num) = output_sim(:,5);
figdata.dp.y(:,set_num) = output_sim(:,6)*(180/pi);
figdata.vp.y(:,set_num) = output_sim(:,7);
figdata.v.y(:,set_num) = output_sim(:,8);
figdata.th1.y(:,set_num) = output_sim(:,9);
figdata.th2.y(:,set_num) = output_sim(:,10);
figdata.th3.y(:,set_num) = output_sim(:,11);
figdata.thref.y(:,set_num)= output_sim(:,12);
figdata.th.y(:,set_num) = output_sim(:,25);
figdata.tgref.y(:,set_num)= output_sim(:,13);
figdata.tg.y(:,set_num) = output_sim(:,24);
figdata.thdt.y(:,set_num) = output_sim(:,14);
figdata.tilt.y(:,set_num) = output_sim(:,15);
figdata.yaw.y(:,set_num) = output_sim(:,16);
figdata.bf1.y(:,set_num) = output_sim(:,17);
figdata.bf2.y(:,set_num) = output_sim(:,18);
figdata.bf3.y(:,set_num) = output_sim(:,19);
figdata.bm1.y(:,set_num) = output_sim(:,20);
figdata.bm2.y(:,set_num) = output_sim(:,21);
figdata.bm3.y(:,set_num) = output_sim(:,22);
figdata.total.y(:,set_num)= output_sim(:,23);

```

E.1.18 HAWC2figdata.m

```

function [figdata]=hawc2figdata(figdata,set_name, set_num);
global azim_start
% set_name = 'HAWC PI 2'
% set_num = 1;

data_sim = load ('c:\hawc\res\ri_pitch.dat');
data_reg = load ('c:\hawc\datafil.dat');
s1 = dir ('c:\hawc\res\ri_pitch.dat');
s2 = dir ('c:\hawc\datafil.dat');
sek1 = (str2num(s1.date(end-1))*10)+str2num(s1.date(end));
sek2 = (str2num(s2.date(end-1))*10)+str2num(s2.date(end));
if (abs(sek1-sek2) >= 10); disp('          FEJL I DATAFILER TIDER!!');end;
n_reg = length(data_reg);
n_sim = length(data_sim);
data_reg = data_reg([(n_reg-(n_sim-1)):end],:); %Smid starten væk da der er indsving i...
                                                %hawc der ikke beregner startværdier
azim_start = data_sim(1,2)+180;                %bruges så simulink og HAWC er i fase

figdata.x = data_sim(:,1);
figdata.legend(set_name) = {set_name};
figdata.wr.y(:,set_num) = data_sim(:,3);
figdata.p.y(:,set_num) = data_sim(:,5);
figdata.v.y(:,set_num) = data_sim(:,6);
figdata.th.y(:,set_num) = -data_sim(:,4);
figdata.th1.y(:,set_num) = -data_sim(:,7);
figdata.th2.y(:,set_num) = -data_sim(:,8);
figdata.th3.y(:,set_num) = -data_sim(:,9);
figdata.dp.y(:,set_num) = data_sim(:,19)*(180/pi);
figdata.ty.y(:,set_num) = data_sim(:,20);
figdata.bf1.y(:,set_num) = data_sim(:,10).*1000;
figdata.bf2.y(:,set_num) = data_sim(:,12).*1000;
figdata.bf3.y(:,set_num) = data_sim(:,14).*1000;
figdata.bm1.y(:,set_num) = data_sim(:,11).*1000;
figdata.bm2.y(:,set_num) = data_sim(:,13).*1000;
figdata.bm3.y(:,set_num) = data_sim(:,15).*1000;

figdata.thref.y(:,set_num) = -(data_reg(:,12)+data_reg(:,13)+data_reg(:,14))./3
figdata.tgref.y(:,set_num) = data_reg(:,10);
figdata.tg.y(:,set_num) = data_reg(:,11);
figdata.total.y(:,set_num)= data_reg(:,5);
figdata.tilt.y(:,set_num)= data_reg(:,6);
figdata.yaw.y(:,set_num) = data_reg(:,7);

%***** Beregning af afledte og manglende data (thdt og wg) *****
thdt = []; dphidt = [];
dphidt = [figdata.dp.y([2:end],set_num) ; figdata.dp.y(end,set_num)];
dphidt = (dphidt -[figdata.dp.y(1,set_num) ; figdata.dp.y([1:end-1],set_num)])./0.025;
figdata.wg.y(:,set_num) = (figdata.wr.y(:,set_num)-((dphidt.*(pi/180))).*85;

thdt = [figdata.th.y([2:end],set_num) ; figdata.th.y(end,set_num)];
figdata.thdt.y(:,set_num) = (thdt -[figdata.th.y(1,set_num) ;...
                                figdata.th.y([1:end-1],set_num)])./0.025;

```

E.1.19 stpkt0.m

```

function [wr,th,Tg,lam,cp,P,A,B]=stpkt0(v,th0)
% Usage: [wr,th,Tg,lam,cp,P]=stpkt0(v,th0)
%        [wr,th,Tg,lam,cp,P,A,B]=stpkt0(v,th0)

```

```

global R wrnom Pnom Ng rho alfa Jt
global maxcp lamo tho lamtab thtab cptab
global cptable
global int_mtd

if nargin<2, th0=max(5,v); end
% if isempty(th0), th0=max(5,v); end
% if isempty(R|Jt), syspar; end;
% if isempty(cptable), load cpdata; end;
th0=max(th0,3);

wr=v/R/lamo;
% ----- low and stop end -----
if wr<wrnom*alfa,
wr=wrnom*alfa;
lam=v/wr/R;
if lam<min(lamtab),
%disp('stop mode')
th=NaN; cp=NaN; P=0; wr=0; Tg=NaN;
else
th=interp1(lamtab,thtab,lam);%,int_mtd);
cp=interp1(lamtab,cptab,lam);%,int_mtd);
P=0.5*rho*pi*R^2*v^3*cp;
Tg=P/wr/Ng;
end
if P<0,% stop end
P=0; cp=NaN; Tg=NaN; wr=0; lam=NaN; th=NaN;
end
% ----- high and top end -----
elseif wr>wrnom,
wr=wrnom;
lam=v/wr/R;
th=interp1(lamtab,thtab,lam);%,int_mtd);
cp=interp1(lamtab,cptab,lam);%,int_mtd);
P=0.5*rho*pi*R^2*v^3*cp;
Tg=P/wr/Ng;
if P>Pnom,% top end
P=Pnom;
[th,cp]=Prsol(P,v,wr,th0);
Tg=P/wr/Ng;
end
% ----- mid end -----
else
lam=lamo;
th=tho;
cp=maxcp;
P=0.5*rho*pi*R^2*v^3*cp;
Tg=P/wr/Ng;
end

% -----
if nargin>6,
if wr>0,
dcpdlam=interp2(cptable.lam,cptable.th,cptable.dcpdlam,lam,th);%,int_mtd);
dcpdth=interp2(cptable.lam,cptable.th,cptable.dcpdth,lam,th);%,int_mtd)

A=(-P/wr^2+(0.5*rho*pi*R^2*v^3*dcpdlam*(-v/R/wr^2))/wr)/Jt;

B=[ (0.5*rho*pi*R^2*v^3*dcpdth/wr)/Jt; % th
-Ng/Jt; % Tg
(0.5*rho*pi*R^2*v^3*cp+0.5*rho*pi*R^2*v^3*dcpdlam/wr/R)/Jt/wr]; % v
B=B(:)';
else

```

```

    A=NaN; B=[NaN NaN NaN];
end
end

return
if wr>0,
    fprintf(' Check : %8.4f \n',dwt0b([wr;th;Tg;v],1));
end;

```

E.1.20 fopt.m

```

function [maxcp,lamo,tho,lam,mtho]= fopt(iplt)
% Usage: [maxcp,lamo,tho,lam,mtho]= fopt(iplt)
%
% Finds:
%
% -the optimal cp (maxcp) for the lam and th (lamo and tho)
% -the optimal th (mtho) as function of lam
%
% fopt(0) No plots
% fopt(1) The lambda iterations - for debugging only
% fopt(2) Optimal theta vs lambda (1) and Opt Cp vs lambda (2)
% fopt(3) 3D plot of Cp vs lambda and th + Opt. th vs lambda
% fopt(4) Opt. theta vs lambda + the optimal (lam,th)
% fopt(5) P as function of lambda and theta

global cptable
global maxcp lamo tho lamtab thtab cptab
% +Global optimum+ +Local optimum table+
global int_mtd

if nargin<1,
    iplt=0;
end

% load cpdata.mat

lam=cptable.lam;
th=cptable.th;
cp=cptable.cp;

%[cp,lam,th]=udtynd(cp,lam,th,[],0.6,-15,30);
% 0.4 -10 15
% -----
% rough estimate of optimum of Cp wrt. th for given lambda
% (lmaxcp, ltho, lam)

[lmaxcp,th_idx]=max(cp); % depend on lamda
ltho=th(th_idx); % depend on lamda

[tmaxcp,lam_jdx]=max(lmaxcp); % Which lambda gives the best cp
ttho=th(th_idx(lam_jdx));
tlamo=lam(lam_jdx);

maxcp=tmaxcp;tho=ttho;lamo=tlamo;

% -----
% Local search for maxcp as function of lambda

lowl=NaN; lowl=-1;

```

```

nofp=2;
mtho=zeros(length(lam),1); mmaxcp=zeros(length(lam),1);
for i=1:length(lam),
    nr=max(1,th_idx(i)-nofp):min(length(th),th_idx(i)+nofp); nr=nr(:);
    x=th(nr);
    y=cp(nr,i);
    X=[ones(length(nr),1) x x.^2];
    parm=inv(X'*X)*X'*y;
    dparm=[2*parm(3) parm(2)];
    mtho(i)=roots(dparm);
    mmaxcp(i)=[1 mtho(i) mtho(i)^2]*parm;
    if iplt==1,
        den=0.05;
        xl=th(min(nr)):den:th(max(nr)); xl=xl(:);
        X=[ones(length(xl),1) xl xl.^2];
        yh=X*parm;

        plot(th,max(lowl,cp(:,i)),'-'); grid;
        hold on;
        plot(ltho(i),max(lowl,lmaxcp(i)),'+');
        plot(mtho(i),max(lowl,mmaxcp(i)),'*');
        plot(x,y,'r+');
        plot(xl,yh,'k-');
        hold off;
        axis([-10 15 lowl 0.5]);
        pause(0.1)
        %pause
    end
end

lamtab=lam; % For the sake of the globals
thtab=mtho;
cptab=mmaxcp;

% -----
if iplt==2,
    [dumcp,plam,dumth,lamdx]=udtynd(cp,lam,th,[],0.5,[],[]);
    pmtho=mtho(lamdx);
    plot(plam,pmtho); grid;
    xlabel('\lambda'); ylabel('\theta')
    title('Optimal \theta vs. \lambda')
    pause;
    pmmaxcp=mmaxcp(lamdx);
    plot(plam,pmmaxcp); grid;
    xlabel('\lambda'); ylabel('C_p')
    title('Optimal C_p vs. \lambda')
end
% -----
% Search for global maxcp
jdx=lam_jdx-3:lam_jdx+3;
x=lam(jdx);
y=mmaxcp(jdx);
X=[ones(length(jdx),1) x x.^2];
phat=pinv(X'*X)*X'*y;
dphat=[ 2*phat(3) phat(2)];
lamo=roots(dphat);
tho=interp1(lam,mtho,lamo,'cubic');
maxcp=interp2(lam,th,cp,lamo,tho,'cubic');

% -----
if iplt==3,
    %[pcp,plam,pth,lamdx,thdx]=udtynd(cp,lam,th,[],0.5,-15,30);
    [pcp,plam,pth,lamdx,thdx]=udtynd(cp,lam,th,[],0.4,-15,20);
    meshc(plam,pth,max(0,pcp));
    xlabel('\lambda=v/\omega/R'); ylabel('\theta'); zlabel('c_p');

```

```

%view(-37.5,30)
view(-40,45)

hold on
plot3(tlamo,ttho,tmaxcp,'+');
plot3(lamo,tho,maxcp,'*');
plot3(lamo,tho,0,'*');
plot3(plam,mtho(lamd),max(0,mmaxcp(lamd)), 'b-');
plot3(plam,mtho(lamd),zeros(1,length(plam)), 'b-');
hold off
elseif iplt==4,
[pcp,plam,pth,lamd,thdx]=udtynd(cp,lam,th,[],0.5,-15,30);
plotyy(plam,mtho(lamd),plam,mmaxcp(lamd)); grid;
%axis([0 0.4 0 20]);
hold on
plot(lamo,tho,'*');
%plot(lam,10*mmaxcp)
hold off
xlabel('\lambda=v/\omega/R'); ylabel('\theta');
title('Optimal \theta (and obtained C_p) vs. \lambda');
elseif iplt==5,
lam=cptable.lam; th=cptable.th; cp=cptable.cp;
[pcp,plam,pth,lamd,thdx]=udtynd(cp,lam,th,[],0.6,-15,30); % p for plot
syspar
P=[];
for i=1:length(plam),
P=[P 0.5*rho*pi*R^2*plam(i)^3*pcp(:,i)];
end
meshc(plam,pth,P);
hold on
%plot3(lamtab',thtab,0.5*rho*pi*R^2*lamtab.^3*cptab,'b-');
plot3(lamtab(lamd),thtab(lamd),0.5*rho*pi*R^2*lamtab(lamd).^3.*cptab(lamd),'b-');
hold off
xlabel('\lambda=v/\omega/R'); ylabel('\theta'); zlabel('P_r/(\omega_rR)^3');
title('Scaled produced power')
% view(-90,0)
% view(0,0)
elseif iplt==6,
[pcp,plam,pth,lamd,thdx]=udtynd(cp,lam,th,[],0.6,-15,30); % p for plot
syspar
plot(lamtab(lamd),0.5*rho*pi*R^2*lamtab(lamd).^3.*cptab(lamd),'b-');
xlabel('\lambda=v/\omega/R'); ylabel('P_r/(\omega_rR)^3');
grid
end

```

E.1.21 deriv.m

```

function drv=deriv(y,x,x0)
% Usage: drv=deriv(y,x,x0)

% Fits a second order polynomial through the data
% and determine the first derivative in x0.

Y=y(:); x=x(:); x0=x0(:);
n=length(y);
X=[ones(n,1) x x.^2];
th=inv(X'*X)*X'*Y;
dth=[th(2); 2*th(3)];
m=length(x0);
X=[ones(m,1) x0];
drv=X*dth;

```


E.1.22 Prsol.m

```

function [th,cp]=Prsol(P,v,w,th0)
% Usage: [th,cp]=Prsol(P,v,w)
%
% Finds the th which gives Pr=P for a given v and wr.
% th0 is not used any longer

global cptable
global R rho
global maxcp lamo tho lamtab thtab cptab
% +Global optimum+ +Local optimal cp(lam) th(lam)+
global int_mtd

if isempty(cptable), load ../../cp/anca/cpdata; end;
if isempty(R), syspar; end
if isempty(maxcp), fopt(0); end;

lam=v/R/w;
lmaxcp=interp1(lamtab,cptab,lam);
lPmax=0.5*rho*pi*R^2*v^3*lmaxcp;
if P>lPmax,
    disp('Infeasible solution 1');
else
    cps=interp1(cptable.lam,cptable.cp',lam,'spline');
    lthmax=interp1(lamtab,thtab,lam); % which th gives max cp
    idx=find(cptable.th>=lthmax);
    ths=cptable.th(idx); % Which th's are in the pitch area
    cps=cps(idx); % which cp's are in the pitch area
    Prs=0.5*rho*pi*R^2*v^3*cps;
    [mp,jdx]=min(abs(Prs-P));
    % Here we are
    nop=3;
    if jdx==1,
        if P<0.5*rho*pi*R^2*v^3*cps(1),
            disp('Too small cp table')
        else
            x=ths(1:2*nop+1);
            Y=Prs(1:2*nop+1);
        end
    elseif jdx==length(cps),
        x=ths(end-2*nop-1:end);
        Y=Prs(end-2*nop-1:end);
    else
        x=ths(jdx-nop:jdx+nop);
        Y=Prs(jdx-nop:jdx+nop);
    end
    X=[x.^2 x ones(1+2*nop,1)];
    parm=inv(X'*X)*X'*Y;
    th=max(roots(parm'-[0 0 1]*P));
    cp=P/(0.5*rho*pi*R^2*v^3);
end

%-----
%-----
function [th,cp]=Prsol_1(P,v,w,th0)
% Usage: [th,cp]=Prsol(P,v,w,th0)
%
% Finds the th which gives Pr=P for a given v and wr.

global cptable
global R rho

if isempty(cptable), load ../../cp/anca/cpdata; end;

```

```
if isempty(R), syspar; end

if nargin<4,
    th0=max(5,v);
elseif isempty(th0),
    th0=max(5,v);
end;

th=th0;
opt=optimset('fsolve');;
opt.Display='off';
opt.Display='iter';
th=fsolve(@powere,th0,opt,v,w,P);
[perr,cp]=powere(th,v,w,P);

%-----
function [perr,cp]=powere(th,v,w,P)
%-----
%-----
global cptable
global R rho
global int_mtd

lambda=v/R/w; cp=[];
if lambda>min(cptable.lam),
    if lambda<max(cptable.lam),
        cp=interp2(cptable.lam,cptable.th,cptable.cp,lambda,th,int_mtd);
        perr=0.5*rho*pi*R^2*v^3*cp-P;
    else
        perr=-P;
        disp('High lambda');
        pause
    end
else
    perr=-P;
    disp('Low lambda');
    pause
end
%-----
```

BILAG F

HAWC

F.1 HAWC releterede filer

Bilag F indeholder HAWC script filen. De vigtigste datafiler fra HAWC samt de 3 PASCAL filer der udgør regulatoren.

Liste over filer der reletere til HAWC:

- 1 i_pitch.htc.
- 2 HAWC_ae.001
- 3 HAWC_pc.001
- 4 HAWC_gb.001
- 5 HAWC_ge.001
- 6 i_pitch.pas
- 7 i_procedures.pas
- 8 i_global.pas

F.1.1 i_pitch.htc

HAWC scriptfilen *.HTC (HAWC Time simulation Command file) indeholder opsætningen om hvordan simuleringe skal forløbe. Scriptfilen indeholder således

de samme kommandoer som kan gives til HWAC kommandoprompten direkte. Den viste scriptfil simulerer i 20 sek. med 100 sek. indsving. Vinden er turbulent med en faktor på 13.55%, middelvinden i 80 meter er 14 m/s., der er logaritmisk shear regnet ud fra en ruhedslængde på 0.05 mm. Regulatoren i_pitch.dll kaldes for hver 0.025 sek.

```

HEAD HAWC input for "Aeroelastisk int. styring" /*
      data baseret diverse møller /*
      80m tårnhøjde. /*
      test af individuel pitch 03 2006 ;
DEFINE_STRING CASE_IDENTIFICATION i_pitch ;
DEFINE_STRING STRUCTURE_FILE_EXT 001 ;
DEFINE_STRING AERODYN_FILE_EXT 001 ;
DEFINE_STRING PROFILE_FILE_EXT 001 ;
DEFINE_STRING GENERATOR_EFFICIENCY_FILE_EXT 001 ;
DEFINE_STRING GEAR_BOX_EFFICIENCY_FILE_EXT 001 ;
DEFINE_STRING EXTERNAL_REGULATION_NAME i_pitch.dll ;
DEFINE_STRING TURB_FILE_PATH .\turb\ ;
DEFINE_STRING MANN_TURB_FILE_EXT 1 ;
PARAMETERS TURBINE 3 2 4 2 ;
PARAMETERS GEAR_BOX 85.0 2 1 ;
PARAMETERS GENERATOR 150.0 9.99E3 2000.0 0.75 1500.0 1 ;
PARAMETERS TILT 0.00 ;
PARAMETERS HAM_DYNAMIC_STALL 1.50 6.00 0.75 0.14 *
                                0.53 0.30 0.70 0.95 1 12 ;
;
PARAMETERS EXTERNAL_REGULATION 0.025 ; Ts
EXT_CONTR_SENS GENERATOR_SPEED 1;
EXT_CONTR_SENS PITCH_BEARING_ANGLE 1 1 1 *
                                    2 3 4 ;
EXT_CONTR_SENS AZIMUTH 180 5;
EXT_CONTR_SENS BLADE_FORCE_MOMENT 1 1 0 1 0 0 0 0 *
                                    0 6 0 0 0 0 ;
EXT_CONTR_SENS BLADE_FORCE_MOMENT 2 1 0 1 0 0 0 0 *
                                    0 7 0 0 0 0 ;
EXT_CONTR_SENS BLADE_FORCE_MOMENT 3 1 0 1 0 0 0 0 *
                                    0 8 0 0 0 0 ;
EXT_CONTR_SENS ELECTRICAL_POWER 9;
EXT_CONTR_SENS SIMULATED_TIME 15 ;
;
PARAMETERS PITCHABLE_BLADE_PART 2 3 0.0 0.0 0.0 ;
ROTOR BASIC_LAYOUT 3 2 0.00 0.00 0.00 *
                    0.00 0.00 0.00 *
                    0.00 0.00 0.00 0.0 ;
;
DAMPING TOWER 0.0 0.0 0.0 3.06E-3 3.06E-3 1.45E-3 ;
DAMPING SHAFT 0.0 0.0 0.0 1.00E-3 1.00E-3 3.20E-3 ;
DAMPING BLADE 0.0 0.0 0.0 1.39E-3 9.00E-4 1.39E-3 ;
DEFINITION AEROCAL_INDUCTION 32 1.0 1.0 1.0 0.1 1.0 *
                              4.0 4.0 4.0 0.4 4.0 *
                              2.0 2.0 2.0 2.0 1.0 ;
; DEFINITION AEROCAL_MODIFICATIONS 1 1 15 ;
;
DEFINITION WIND_FIELD 1.25 14.00 80.00 0.00 0.00 0.1355 0.050 1 ;
DEFINITION TURBULENCE_MANN 80.0 32 4096 80 1 1.0 0.8 1.0 0.5 1 2;
DEFINITION WIND_SIZE_PULSE 105 250 1.0769;
; DEFINITION WIND_RAMP 50 70 1 1.6667;
; DEFINITION TOWER_AERODYNAMIC_LOAD 0.6;
DEFINITION TOWER_SHADOW 1.200 2.150 1.5 1;
; Blade nodes
NODE BLADE 1 1 0.000 0.000 0.000 ;
NODE BLADE 1 2 0.000 0.000 2.000 ;
NODE BLADE 1 3 0.000 0.000 7.000 ;

```

```

NODE BLADE 1 4 0.000 0.000 12.000 ;
NODE BLADE 1 5 0.000 0.000 20.000 ;
NODE BLADE 1 6 0.000 0.000 25.000 ;
NODE BLADE 1 7 0.000 0.000 28.000 ;
NODE BLADE 1 8 0.000 0.000 31.000 ;
NODE BLADE 1 9 0.000 0.000 34.000 ;
NODE BLADE 1 10 0.000 0.000 36.000 ;
NODE BLADE 1 11 0.000 0.000 38.000 ;
NODE BLADE 1 12 0.000 0.000 40.000 ;
NODE BLADE 2 1 0.000 0.000 0.000 ;
NODE BLADE 2 2 0.000 0.000 2.000 ;
NODE BLADE 2 3 0.000 0.000 7.000 ;
NODE BLADE 2 4 0.000 0.000 12.000 ;
NODE BLADE 2 5 0.000 0.000 20.000 ;
NODE BLADE 2 6 0.000 0.000 25.000 ;
NODE BLADE 2 7 0.000 0.000 28.000 ;
NODE BLADE 2 8 0.000 0.000 31.000 ;
NODE BLADE 2 9 0.000 0.000 34.000 ;
NODE BLADE 2 10 0.000 0.000 36.000 ;
NODE BLADE 2 11 0.000 0.000 38.000 ;
NODE BLADE 2 12 0.000 0.000 40.000 ;
NODE BLADE 3 1 0.000 0.000 0.000 ;
NODE BLADE 3 2 0.000 0.000 2.000 ;
NODE BLADE 3 3 0.000 0.000 7.000 ;
NODE BLADE 3 4 0.000 0.000 12.000 ;
NODE BLADE 3 5 0.000 0.000 20.000 ;
NODE BLADE 3 6 0.000 0.000 25.000 ;
NODE BLADE 3 7 0.000 0.000 28.000 ;
NODE BLADE 3 8 0.000 0.000 31.000 ;
NODE BLADE 3 9 0.000 0.000 34.000 ;
NODE BLADE 3 10 0.000 0.000 36.000 ;
NODE BLADE 3 11 0.000 0.000 38.000 ;
NODE BLADE 3 12 0.000 0.000 40.000 ;
; Nacelle nodes
NODE NACELLE 1 0.0 0.000 0.000 ; Tower top
NODE NACELLE 2 0.0 -0.600 0.000 ; Gearbox
NODE NACELLE 3 0.0 -2.186 0.000 ; Main shaft flange
NODE NACELLE 4 0.0 -3.486 0.000 ; Rotor centre
; Tower nodes
NODE TOWER 1 0.000 0.000 0.000 ;
NODE TOWER 2 0.000 0.000 -25.000 ;
NODE TOWER 3 0.000 0.000 -50.000 ;
NODE TOWER 4 0.000 0.000 -80.000 ;
;
; 1 3 2 m0 Ix Iy(axel) Iz
DEFINITION CONCENTRATED_MASS 1 3 2 75.3E+3 409.4E+3 20.0E+3 367.7E+3 ;
TYPES BLADE_AERODYN_3 1 1 1 ;
TYPES BLADE_STRUCTURE_3 1 1 1 ;
TYPES SHAFT_STRUCTURE 1 ;
TYPES TOWER_STRUCTURE 1 ;
TYPES USE_CALCULATED_BEAMS ;
INITIAL AZIMUTH 0.000 1.2 0.000 ;
;
WRITE BEAM_DATA_TO_RDA_DIRECTORY ;
;
RESULTS BLADE_CO_SYS_FORCE 2;
RESULTS BLADE_CO_SYS_DEFORM 1;
RESULTS FORCE_MOMENT_SIGN 4;
;
RESULTS TIME;
RESULTS AZIMUTH 0.0;
RESULTS ROTOR_SPEED;
RESULTS ELECTRICAL_POWER;
RESULTS WIND_SPEED_AT_HUB 0 1 0 0 0 1;
; RESULTS WIND_SPEED_AT_ROTOR_DISK 0 1 0 0 0 40 0 5; Bruges til at kontrollere shear..

```

```

RESULTS PITCH_BEARING_ANGLE 1 1 1 ;
RESULTS PITCH_ANGLE;
RESULTS BLADE_FORCE_MOMENT 1 1 0 1 0 0 1 0; Total Force on Blade 1 (F_y M_y)
RESULTS BLADE_FORCE_MOMENT 2 1 0 1 0 0 1 0; Total Force on Blade 2 (F_y M_y)
RESULTS BLADE_FORCE_MOMENT 3 1 0 1 0 0 1 0; Total Force on Blade 3 (F_y M_y)
RESULTS BLADE_DEFORMATION 1 12 0 1 0 0 0 0; Balde bend on Blade 1
RESULTS BLADE_DEFORMATION 2 12 0 1 0 0 0 0; Balde bend on Blade 2
RESULTS BLADE_DEFORMATION 3 12 0 1 0 0 0 0; Balde bend on Blade 3
;
; RESULTS SHAFT_FORCE_MOMENT 4 0 1 0 1 1 1 ; Rotor centre
RESULTS SHAFT_DEFORMATION 4 0 0 0 0 1 0 ; Rotor centre
RESULTS TOWER_DEFORMATION 4 0 1 0 0 0 0 ; Tower top
;
DEFINITION RESPONSE_CALC_LIMITS 1.0E-3 1.0E-3 0.0 5.0E-1 1 10 ;
RESULTS RESPONSE_TIME_OFFSET 100.0;
WRITE RESPONSE 20.0 0.025 1 1 0 1 2 1 5 1 ;
STOP ;

```

F.1.2 HAWC_ae.001

Datafilen HAWC_ae.001 indeholder data til beregning af aerodynamiske belastninger. Her hentes blad korde, tykkelse og drejning.

```

1 40 m rotorradius
1 14 ch th twist x_ae y_ae PC set
0.000 0.001 100.000 0.000 0.000 0.000 1
1.200 2.450 100.000 5.100 0.000 0.000 1
5.200 2.650 80.000 6.700 0.000 0.000 1
9.200 3.000 52.000 9.400 0.000 0.000 1
13.200 3.200 33.000 9.400 0.000 0.000 1
17.200 2.800 26.000 5.400 0.000 0.000 1
21.200 2.400 22.200 3.500 0.000 0.000 1
25.200 2.000 20.000 2.300 0.000 0.000 1
29.200 1.800 19.000 1.300 0.000 0.000 1
33.200 1.550 18.500 0.500 0.000 0.000 1
37.200 1.180 18.000 0.000 0.000 0.000 1
39.200 0.620 16.300 1.000 0.000 0.000 1
39.600 0.400 15.700 2.400 0.000 0.000 1
40.000 0.010 15.000 6.000 0.000 0.000 1

```

F.1.3 HAWC_pc.001

Datafilen HAWC_pc.001 indeholder information om den aktuelle bladprofil dvs. lift og drag koefficienter for angrebsvinkler mellem $\pm 180^\circ$.

```

1 Converted format
7 NACA 634xx
1 35 15.00000 NACA 63415, Modified to Re = 6 million
-180.00000 0.00000 0.10000 0.00000
-170.00000 0.80000 0.24000 0.40000
-150.00000 0.80000 0.58000 0.32500
-120.00000 0.49000 1.06000 0.32500
-90.00000 -0.11400 1.30000 0.32500

```

-60.00000	-0.75500	1.06000	0.32500
-30.00000	-1.20000	0.45000	0.22000
-20.00000	-1.20000	0.20000	0.05000
-15.00000	-1.10000	0.10000	0.00000
-10.00000	-0.82000	0.01200	0.00000
-6.00000	-0.36500	0.00900	-0.03750
-2.00000	0.08570	0.00600	-0.06600
2.00000	0.54610	0.00600	-0.07800
4.00000	0.77630	0.00650	-0.08400
6.00000	0.99500	0.00960	-0.09000
8.00000	1.17000	0.01320	-0.09000
10.00000	1.31400	0.01690	-0.09000
12.00000	1.44500	0.02350	-0.09000
14.00000	1.45500	0.04700	-0.09000
16.00000	1.44830	0.08200	-0.09000
18.00000	1.36660	0.12424	-0.09670
20.00000	1.32500	0.17700	-0.10330
25.00000	1.25906	0.32423	-0.12000
30.00000	1.19000	0.47536	-0.14000
40.00000	1.06000	0.69000	-0.18000
50.00000	0.93000	0.90000	-0.22000
60.00000	0.75500	1.06000	-0.26000
70.00000	0.55500	1.18000	-0.29250
80.00000	0.34000	1.26000	-0.32500
90.00000	0.11400	1.30000	-0.32500
100.00000	-0.11000	1.27000	-0.32500
120.00000	-0.49000	1.06000	-0.32500
150.00000	-0.80000	0.58000	-0.32500
170.00000	-0.80000	0.24000	-0.40000
180.00000	0.00000	0.10000	0.00000

F.1.4 HAWC_gb.001

Datafilen HAWC_gb.001 indeholder gear effektivitets data.

```
1
1 2
150.0 150.0
2000.0 2000.0
```

F.1.5 HAWC_ge.001

Datafilen HAWC_ge.001 indeholder generator effektivitets data.

```
2
1 2
150.0 150.0
750.0 750.0
2 2
150.0 141.0
2000.0 1881.0
```

F.2 PASCAL DLL implementering

Regulatoren, som HAWC bruger lægges i en DLL fil. Denne er programmeret i PASCAL og består af 3 filer. `i_pitch.pas` som indeholder selve regulator designet og håndtere kaldene fra HAWC. Filen `i_procedures.pas` indeholder underrutiner og `i_global.pas` prototypeerklæringer og konstanter. Disse kompiles til `i_pitch.dll` som HAWC kalder under simulationen.

F.2.1 `i_pitch.pas`

```

Library i_pitch;
uses
  i_procedures in 'i_procedures.pas',
  i_global in 'i_global.pas',
  math;
var
  outfil : textfile;

Procedure regulator_init;
begin
  with regulatorvar do
  begin
    omega_ref := 1.97119539048771; // Nominet omdrejningstal
    deltat    := 0.025 ;           // 40 Hz sampling frequency
    gearratio := 85.0;             // gear ratio
    Pmax      := 2000*1000;        // P_e_max [W]
    Tgnom     := Pmax/(omega_ref*gearratio);

    Kp_pit    := 0.00003*degrad;   //ny implementering som sim
    Ti_pit    := (Kp_pit/0.2);     //ny implementering som sim
    Kp_piy    := 0.000045*degrad;  //ny implementering som sim
    Ti_piy    := (Kp_piy/0.18);    //ny implementering som sim

    P_pit     := -0.00005*degrad;
    int_p1    := 0;
    int_p2    := 0;
  end;

  with butter_tilt do
  begin
    butter_a[1,1]:= 0.00003123897692; //Butterworth filter
    butter_a[1,2]:=0.00006247795566; //sample freq. 40Hz
    butter_a[1,3]:=0.00003123897715; //4.oder LP filter cutoff 1Hz tilt
    butter_b[1,1]:= -1.72593339503704;
    butter_b[1,2]:=0.74744737190787;
    butter_a[2,1]:=1.00000000000000;
    butter_a[2,2]:=1.999999994169253;
    butter_a[2,3]:=0.99999999260419;
    butter_b[2,1]:= -1.86380049207514;
    butter_b[2,2]:=0.88703299965261;
  end;

  with butter_yaw do
  begin
    butter_a[1,1]:= 0.00003123897692; //Butterworth filter
    butter_a[1,2]:= 0.00006247795566; //sample freq. 40Hz
    butter_a[1,3]:=0.00041659920749; //4.oder LP filter cutoff 2Hz yaw
  end;

```



```

butter_b[1,1]:=-1.47967421693119;
butter_b[1,2]:=0.55582154328249;
butter_a[2,1]:=1.00000000000000;
butter_a[2,2]:=1.99999994169253;
butter_a[2,3]:=0.9999999260419;
butter_b[2,1]:=-1.70096433194353;
butter_b[2,2]:=0.78849973981530;
end;

with buttervar4 do
begin
  butter_a[1,1]:= 0.00037506961630; // Butterworth filter 4
  butter_a[1,2]:= 0.00075013923260; // samlpe freq. 40 Hz.
  butter_a[1,3]:= 0.00037506961630; // 4.order BP filter, center 0.95 Hz
  butter_b[1,1]:=-1.94292925859177;
  butter_b[1,2]:= 0.97016695255396;
  butter_a[2,1]:= 1.00000000000000;
  butter_a[2,2]:=-2.00000000000000;
  butter_a[2,3]:= 1.00000000000000;
  butter_b[2,1]:=-1.95613002690003;
  butter_b[2,2]:= 0.97506716111284;
end;
buttervar5:=buttervar4;

with PI_com_var do
begin
  Kpro :=1*degrad;
  Kint :=0.77*degrad;
  //Kpro := 1.5*6.1*degrad*30/(pi*85);
  //Kint := 1.6*2*2.14*degrad*30/(pi*85);

  outmax:= 25*degrad;
  outmin:= 0*degrad;
end;

with LQRvar do
begin
  LQR_gain[1,1] := 1.48248806201612;
  LQR_gain[1,2] := 0.83055896829454;
  LQR_gain[1,3] := 0.00000648960286;
  LQR_gain[2,1] := -31.09081573851427;
  LQR_gain[2,2] := 42.40076563540902;
  LQR_gain[2,3] := -0.19018102406458;
  th_outmax := 25;
  th_outmin := 0;
  Tg_outmax := 15000;
  Tg_outmin := 0;
end;

firstordervar1.tau := 0.2;
firstordervar2.tau := 0.2;
firstordervar3.tau := 0.2;
firstordervar_gen.tau := 0.1;

assign(outfil,'datafil.dat');
rewrite(outfil);
close(outfil);
assign(outfil,'test.dat');
rewrite(outfil);
close(outfil);

end;

Procedure regulation(var inputvektor : array_15;

```

```

                                var outputvektor: array_4);stdcall;

var
  Pelekref   : single;
  omega_err  : single;
  omegag_err: single;
  ctrl       : array_2;
  azimuth    : array_3;
  thetaref   : array_3;
  omega      : single;
  omegag     : single;
  pitang     : array_3;
  pitcole    : array_3;
  Mblade     : array_3;
  Mcole      : array_3;
  Mcole_pi   : array_2;
  Mcole_p    : array_2;
  pitang_com : single;
  i          : integer;
  power_mes  : single;
  power_err  : single;

begin
  with regulatorvar do          // **REGULATOR ROUTINE**
    begin
      inc(stepno);
      if stepno=1 then
        begin
          regulator_init;
          end;
          omegag := inputvektor[1];
          pitang[1] := inputvektor[2]*(-1);
          pitang[2] := inputvektor[3]*(-1);
          pitang[3] := inputvektor[4]*(-1);
          azimuth[1] := inputvektor[5];
          azimuth[2] := inputvektor[5]+(120*degrad);
          azimuth[3] := inputvektor[5]+(240*degrad);
          Mblade[1] := inputvektor[6];
          Mblade[2] := inputvektor[7];
          Mblade[3] := inputvektor[8];
          power_mes := inputvektor[9];

// Collective pitch reg
          pitang_com := (pitang[1]+pitang[2]+pitang[3])/3;
          Pelekref := GetPref(omegag/gearratio,pitang_com,omega_ref,stepno);
          Pelekref := Pelekref/omegag;
          omegag_err := omegag-(omega_ref*gearratio);
          power_err := power_mes-Pmax;
          pitcole[1] := PIreg(stepno,deltat,PI_com_var,omegag_err);
          ctrl := LQR(stepno,deltat,omegag_err,power_err,LQRvar);

// LQR reg
          if (stepno >= (1400000)) then
            begin
              pitcole[1] := ctrl[1]*degrad;
              Pelekref := ctrl[2];
            end;

// Colemann transformatin
          Mcole[1] := ((1/3)*Mblade[1]) +((1/3)*Mblade[2]) +((1/3)*Mblade[3]);
          Mcole[2] := ((2/3)*cos(azimuth[1])*Mblade[1])+((2/3)*cos(azimuth[2])*Mblade[2])+...
                    ((2/3)*cos(azimuth[3])*Mblade[3]);
          Mcole[3] := ((2/3)*sin(azimuth[1])*Mblade[1])+((2/3)*sin(azimuth[2])*Mblade[2])+...
                    ((2/3)*sin(azimuth[3])*Mblade[3]);
          Mcole_pi[1] := butterworth(stepno,butter_tilt,Mcole[2]);
          Mcole_pi[2] := butterworth(stepno,butter_yaw,Mcole[3]);

```

```

Mcole_p[1] := butterworth(stepno,buttervar4,Mcole[2]);
Mcole_p[2] := butterworth(stepno,buttervar5,Mcole[3]);

// Individuel Pitch reg.
if (stepno>(1200)) then
begin
  pitcole[2] := (Kp_pit*Mcole_pi[1])+int_p1;          //Tilt PI reg
  int_p1     := (Mcole_pi[1]*Ti_pit*deltat)+int_p1;
  if (int_p1)>5 then int_p1:=5; if (int_p1)<-5 then int_p1:=-5;
  pitcole[2] := pitcole[2]+P_pit*Mcole_p[1];

  pitcole[3] := (Kp_piy*Mcole_pi[2])+int_p2;          //Yaw PI reg
  int_p2     := (Mcole_pi[2]*Ti_piy*deltat)+int_p2;
  if (int_p2)>5 then int_p1:=5; if (int_p2)<-5 then int_p1:=-5;
  pitcole[3] := pitcole[3]+P_pit*Mcole_p[2];

  thetaref[1]:=pitcole[1]+(pitcole[2]*cos(azimuth[1]))+(pitcole[3]*sin(azimuth[1]));
  thetaref[2]:=pitcole[1]+(pitcole[2]*cos(azimuth[2]))+(pitcole[3]*sin(azimuth[2]));
  thetaref[3]:=pitcole[1]+(pitcole[2]*cos(azimuth[3]))+(pitcole[3]*sin(azimuth[3]));
end
else
begin
  for i:=1 to 3 do thetaref[i]:=pitcole[1];
end;

// Skriv til output fil
assign(outfil,'datafil.dat');
append(outfil);
write(outfil,' ',omegag:1:5);
write(outfil,' ',0);
write(outfil,' ',Mcole_p[1]:6:2);
write(outfil,' ',Mcole_p[2]:6:2);
write(outfil,' ',Mcole[1]:6:2);
write(outfil,' ',Mcole[2]:6:2);          //udskriv Tilt
write(outfil,' ',Mcole[3]:6:2);          //udskriv Yaw
write(outfil,' ',power_err:6:1);
write(outfil,' ',power_mes:6:1);
close(outfil);

  outputvektor[1]:=Pelekref;
  outputvektor[2]:=thetaref[1]*(-1);
  outputvektor[3]:=thetaref[2]*(-1);
  outputvektor[4]:=thetaref[3]*(-1);
end;
end;

Procedure generator(var deltat,powref,powlos,omegagen,mgenera : single);stdcall;
begin
  with generatorvar do
  begin
    mgenera:=firstorderfilter(deltat,stepno,firstordervar_gen,powref);

    assign(outfil,'datafil.dat');
    append(outfil);
    write(outfil,' ',powref:5:1);          //udskriv Tg_ref
    write(outfil,' ',mgenera:5:1);        //udskriv Tg
    close(outfil);
  end;
end;

Procedure pitchservo(var deltat : single;
                    var thetaref : array_3;
                    var theta : array_3);stdcall;
begin
  with pitchservovar do

```

```

begin
  inc(stepno);
  theta[1] :=firstorderfilter(deltat,stepno,firstordervar1,thetaref[1]);
  theta[2] :=firstorderfilter(deltat,stepno,firstordervar2,thetaref[2]);
  theta[3] :=firstorderfilter(deltat,stepno,firstordervar3,thetaref[3]);

  assign(outfil,'datafil.dat');
  append(outfil);
  write(outfil,' ',theta[1]);           //udskriv th1
  write(outfil,' ',theta[2]);         //udskriv th2
  writeln(outfil,' ',theta[3]);       //udskriv th3
  close(outfil);

end;
end;

exports
regulation,generator,pitchservo;

begin
end.

```

F.2.2 i_procedures.pas

```

unit i_procedures;
interface
  uses i_global,math;

var
  OUTFIL: TEXTFILE;

function firstorderfilter(deltat:single;
                          stepno:integer;
                          var firstordervar:Tfirstordervar;
                          x:single):single;

Function GetPref(ogagen:single;
                 pit_com:single;
                 oga_ref:single;
                 stepno:integer):single;

function butterworth(stepno:integer;
                     var buttervar:Tbuttervar;
                     input:single):single;

function PIreg(stepno:integer;
               deltat :single;
               var PIvar:TPIvar;
               error : single):single;

function LQR(stepno : integer;
             deltat : single;
             wg_err,p_err : single;
             var LQRvar : TLQRvar):array_2;

implementation

Procedure Readwp();
begin
  with wpdatavar do
    //P-Omega tabel

```

```

begin
  wpdata[1,1]:= 0;      wpdata[1,2]:= 0;
  wpdata[2,1]:= 0.6000; wpdata[2,2]:= 0;
  wpdata[3,1]:= 0.7750; wpdata[3,2]:= 78700;
  wpdata[4,1]:= 0.8750; wpdata[4,2]:= 113300;
  wpdata[5,1]:= 0.9750; wpdata[5,2]:= 156700;
  wpdata[6,1]:= 1.0750; wpdata[6,2]:= 210100;
  wpdata[7,1]:= 1.1750; wpdata[7,2]:= 274300;
  wpdata[8,1]:= 1.2750; wpdata[8,2]:= 350500;
  wpdata[9,1]:= 1.3750; wpdata[9,2]:= 439600;
  wpdata[10,1]:=1.4750; wpdata[10,2]:= 542600;
  wpdata[11,1]:=1.5750; wpdata[11,2]:= 660600;
  wpdata[12,1]:=1.6750; wpdata[12,2]:= 794600;
  wpdata[13,1]:=1.7750; wpdata[13,2]:= 950600;
  wpdata[14,1]:=1.9510; wpdata[14,2]:=2000000;
  wpdata[15,1]:=2.2000; wpdata[15,2]:=2000000;
end;
end;

Function interpolate(x,x0,x1,f0,f1:single):single;
begin
  if x0=x1 then interpolate:=f0
  else interpolate:=(x-x1)/(x0-x1)*f0+(x-x0)/(x1-x0)*f1;
end;

Function GetPref(ogagen:single;
                 pit_com:single;
                 oga_ref:single;
                 stepno:integer):single;

const
  pit_limit = 2.0*degrad;
var
  x,x0,x1,f0,f1,Pref : single;
  i : integer;
begin
  if stepno=1 then Readwp();

  if pit_com>pit_limit then ogagen:=oga_ref;
  with wpdatavar do
  begin
    i:=0;
    repeat
      inc(i);
    until wpdata[i,1]>ogagen;
    x:=ogagen;
    x0:=wpdata[i-1,1];
    x1:=wpdata[i,1];
    f0:=wpdata[i-1,2];
    f1:=wpdata[i,2];
    Pref:=interpolate(x,x0,x1,f0,f1);
    if pit_com<pit_limit ...
      then Pref:=interpolate(pit_com,0.0,pit_limit,Pref,regulatorvar.Pmax);
  end;
  GetPref:=Pref;
end;

function firstorderfilter(deltat:single;
                          stepno:integer;
                          var firstordervar:Tfirstordervar;
                          x:single):single;

var
  y :single;
begin
  y :single;          // første orders filter
  with firstordervar do
  begin
    // 1
    // -----
    // tau*s+1
  end;
end;

```

```

    if stepno=1 then
    begin
        x1:=x;
        y1:=x;
    end;
    y:=(deltat*(x+x1-y1)+2*tau*y1)/(deltat+2*tau);
    x1:=x;
    y1:=y;
end;
firstorderfilter:=y;
end;

function butterworth(stepno:integer;           //Anden ordens filtre
                    var buttervar:Tbuttervar; //2 kaskadekoblede
                    input:single):single;

var
    i :integer;
    temp :single;
begin
    with buttervar do
    begin
        IF (STEPNO=1) THEN
            for i:=1 to 3 do
            begin
                inp[i]:=input;
                tem[i]:=input;
                oup[i]:=input;
            end;

            inp[3]:=inp[2];
            inp[2]:=inp[1];
            inp[1]:=input;
            TEMP:=butter_a[1,1]*inp[1]+butter_a[1,2]*inp[2]+butter_a[1,3]*inp[3]
                -butter_b[1,1]*tem[1]-butter_b[1,2]*tem[2];

            tem[3]:=tem[2];
            tem[2]:=tem[1];
            tem[1]:=temp;
            TEMP:=butter_a[2,1]*tem[1]+butter_a[2,2]*tem[2]+butter_a[2,3]*tem[3]
                -butter_b[2,1]*oup[1]-butter_b[2,2]*oup[2];

            oup[3]:=oup[2];
            oup[2]:=oup[1];
            oup[1]:=temp;

            butterworth:=oup[1];
        end;
    end;
end;

function LQR(stepno : integer;
            deltat : single;
            wg_err,p_err : single;
            var LQRvar : TLQRvar):array_2;

var
    th,Tg : single;
begin
    with LQRvar do
    begin
        if (stepno = (140000)) then
        begin
            int_wg := 6.704012;          int_P := -26818.950099;
        end;

        int_wg := int_wg+deltat*wg_err;
        int_p := int_p + deltat*p_err;
    end;
end;

```

```

th := wg_err*LQR_gain[1,1]+int_wg*LQR_gain[1,2]+int_p*LQR_gain[1,3];
Tg := wg_err*LQR_gain[2,1]+int_wg*LQR_gain[2,2]+int_p*LQR_gain[2,3];

if (th<th_outmin) then th:=th_outmin
else if (th>th_outmax) then th:=th_outmax;

if (Tg<Tg_outmin) then Tg:=Tg_outmin
else if (Tg>Tg_outmax) then Tg:=Tg_outmax;

LQR[1] := th; //omregn til styresignal i [rad]
LQR[2] := Tg; //omregn til styresignal i power [W]

assign(outfil,'test.dat');
append(outfil);
write( outfil,' ',int_p:7:5);
writeln( outfil,' ',int_wg:7:5);

close(outfil)
end;
end;

function PReg(stepno:integer;
deltat :single;
var PIvar:TPIvar;
error : single):single;
var
outres:single;

begin
with PIvar do
begin
if stepno=1 then
begin
outset1:=0;
error1:=0;
end;
outset := outset1+error*Kint*deltat;
if (outset<outmin) then outset:=outmin
else if (outset>outmax) then outset:=outmax;
outpro:=Kpro*error;
outset1:=outset;
error1:=error;
outres:=outset+outpro;
if (outres<outmin) then outres:=outmin
else if (outres>outmax) then outres:=outmax;
PIreg:=outres;
end;
end;
end.

```

F.2.3 i_global.pas

```

unit i_global;

interface
const
maxwpset = 10;
maxwplines = 100;
type
array_2 = array[1..2] of single;

```

```

array_3 = array[1..3] of single;
array_4 = array[1..4] of single;
array_15 = array[1..15] of single;

Tregulator = Record
  stepno      : integer;
  deltat      : single;
  wpname      : string;
  omega_ref   : single;
  gearratio   : single;
  Pmax        : single;
  Tgnom       : single;
  Kp_pit      : single;
  Ti_pit      : single;
  Kp_piy      : single;
  Ti_piy      : single;
  P_pit       : single;
  int_p1      : single;
  int_p2      : single;
end;

Tgenerator = Record
  stepno      : integer;
end;

Tpitchservo = Record
  stepno      : integer;
end;

Twpdata = Record
  wpdata      : array[1..maxwplines,1..2] of single;
  lines       : integer;
end;

Tglobal = Record
  theta       : array_3;
end;

Tbuttervar=Record
  butter_a    : array[1..3,1..3] of single;
  butter_b    : array[1..2,1..2] of single;
  inp,tem,oup: array[1..3] of single;
end;

TLQRvar=Record
  LQR_gain    : array[1..3,1..3] of single;
  int_wg,int_p,int_wg1 : single;
  th_outmin,th_outmax : single;
  Tg_outmin,Tg_outmax : single;
  wg_err1,Itemp,KI    : single;
  int_pp1,pp_err1,Ipptemp : single;
end;

TPIvar=Record
  Kpro,Kdif,Kint      : single;
  outmin,outmax       : single;
  error1              : single;
  outset1             : single;
  outset,outpro,outdif : single;    // burde være lokale, men pga. udskrift ...
end;

Tfirstordervar=Record
  tau : single;
  x1,y1: single;
end;

```



```
const
  pi      = 3.14159265359;
  degrad  = 0.0174532925;
  raddeg  = 57.2957795131;
var
  regulatorvar : Tregulator;
  generatorvar : Tgenerator;
  pitchservovar : Tpitchservo;
  wpdatavar    : Twpdata;
  globalvar    : Tglobal;
  datafil      : textfile;
  PI_com_var   : TPIvar;
  butter_tilt  : Tbuttervar;
  butter_yaw   : Tbuttervar;
  buttervar4   : Tbuttervar;
  buttervar5   : Tbuttervar;
  LQRvar       : TLQRvar;
  firstordervar1: Tfirstordervar;
  firstordervar2: Tfirstordervar;
  firstordervar3: Tfirstordervar;
  firstordervar_gen : Tfirstordervar;
implementation
end.
```

