
Driftsplanlegging i vannkraftproduksjon – en realopsjonstilnærming

**Hovedoppgave for stud.techn
Gunnar Aronsen
Våren 2002**



Faggruppe for investering, finansiering og økonomistyring
Institutt for industriell økonomi og teknologiledelse
Fakultet for samfunnsvitenskap og teknologiledelse
Norges naturvitenskaplige tekniske universitet

Forord

Denne rapporten er skrevet som hovedoppgave ved Institutt for industriell økonomi og teknologiledelse ved Norges teknisk-naturvitenskaplige universitet i Trondheim våren 2002. Jeg vil gjerne rette en takk til alle som har bidratt til at denne oppgaven, enten det er i form av data eller faglige innspill. Særlig vil jeg takke min veileder Stein-Erik Fleten for mange og gode innspill i arbeidet med denne oppgaven.

Trondheim, 18. Juni 2002

Gunnar Aronsen, stud. tech
aronsen@stud.iot.ntnu.no

Sammendrag

I denne rapporten er det gjort et forsøk på tilpasse en prismodell som skulle ivareta hensynet til korrelasjon mellom totalt tilsig i landet og pris. Som et mål på totalt tilsig har avvik fra middelveien på magasinbefylling i Norge vært benyttet. Dette fordi dataene er tilgjengelig for alle aktører i markedet, og at de er signifikant korrelert med forwardprisen ble en modell forsøkt tilpasset med de som inndata.

Prismodellen som ble forsøkt tilpasset tok utgangspunkt i en kjent enfaktor modell og ble deretter forsøkt modifisert slik at vi fikk en prismodell som tok hensyn til korrelasjon mellom pris og tilsig. Dette lyktes ikke og det er derfor laget en diskret prismodell som kan markedsjusteres. Basisen for denne modellen er Hull & White's [Hull, John C. (1993)] rentemodell og en korreksjon algoritme som for markedsjustering fra Clewlow og Strickland (April 1999). Variansstrukturen til denne modellen viser seg å være fleksibel og kan i prinsipp tilpasse hvilken som helt variansmodell. Fordi den er markedsjustert ivaretaes også hensynet sesongsvingninger. Modellen gir også som utdata som kan benyttes til å lage overgangsmatriser, og prismodellen kan derfor betraktes som en markovmodell. Tilsigsmodellen som er tilpasset bygger på en prismodell fra SEFas [Mo & Green m.flere (2000)], og er også en markovmodell. Modellen er tilpasset uten korreksjoner av metodikken og med mindre justeringer av datasett som kommer fra et konkret kraftverk.

Med de pris og tilsigmodellene på plass ble vannkraftsproducents driftsplanleggingsproblem formulert som ett lineært optimeringsproblem. Som optimering algoritme for å løse problemet ble SDP forsøkt benyttet. Denne algoritmen krever bruk av interpolering og det ble skrevet en algoritme for dette. Problemet ble forsøkt løst i Matlab, og hvilket ikke gikk fordi interpolasjonsalgoritmen ikke kunne benyttes sammen med den lineære solveren i Matlab. Problemet ble også forsøkt løst som ikke lineært og lot seg løse for mindre testsett. Fordi den estimerte løsningstiden for det fulle problemet var ca 90 timer ble dette aldri løst. Hypotesen om at bruken av forwardpriser i driftsplanlegging gir bedre resultater en ved bruk av forventet spotpriser er derfor ikke analysert.

Forord.....	I
Sammendrag	II
Innledning	1
1. Datagrunnlag forward- og futures priser.....	2
1.1. Terminstruktur	2
1.2. Glatting av data	3
1.2.1. Metodikk for glatting.....	4
1.3. Sensurering.....	5
1.4. Datautvalg	5
1.5. Deskriptiv statistikk og kontroll av prisdata	6
1.5.1. Signifikans i korrelasjon	7
2. Datarepresentasjon av tilsig.....	9
2.1. Tilsigsdata.....	9
2.1.1. Informasjon i magasindata	10
2.2. Magasinfyllingsdata.....	10
2.3. Deskriptiv statistikk	11
2.4. Sjekk av fordelingsegenskaper til dataene	13
3. Valg av forwardpris modell	14
3.1. Innledning.....	14
3.2. Metodikk	14
3.3. Teoretisk grunnlag	16
3.4. Overgang fra spotpris til log spotpris modell.....	18
3.5. Valg av første deterministisk funksjon	19
3.6. Estimering av den stokastiske prosessen for enfaktormodellen.....	20
3.7. Bruk av forwardpriser kontra spotpris i tilpassning	22
3.8. Estimering av standard avvik og risikopremie.....	22
3.8.1. Estimering av standard avviket.....	22
3.8.2. Estimering av markedsprisen for risiko.....	23
4. Tilpassning av modeller	24
4.1. Parametrene i basismodellen	24
4.1.1. Alternativ beregning markedspris for risiko	24

4.2.	Hvor bra tilpasset er basis modellen	26
4.2.1.	Antagelsen om normalitet	26
4.2.2.	Tilpassning av full modell.....	27
4.3.	Beregning av korrelasjon mellom forwardpris og magasinnivå	29
4.3.1.	Korrelasjon mellom forwardpriser og tilsig.....	29
4.3.2.	Modeller som er prøvd tilpasset	31
4.3.3.	Hvorfor kan ikke korrelasjon mellom tilsig og pris modelleres her? ..	31
4.4.	Hvordan kan tilsiget inkluderes i andre modeller?	32
4.4.1.	Ved bruk av en tofaktor modell?	32
4.4.2.	Valg av pris modell	34
5.	Generelt om trinomiske trær.....	35
5.1.	Hvorfor bruke dem her.....	35
5.2.	Oppbyggingen av det trinomisk treet	35
5.3.	Beregning overgangssannsynligheter	36
5.4.	Forskjellige modeller for trinomiske treer.....	37
6.	Trinomisk tre for Hull og White modellen	38
6.1.	Betingelse for Hull og White modellen	38
6.2.	Treets form	38
6.3.	Steg 1 i trebyggingen	39
6.3.1.	Overgangssannsynligheter i modellen	41
6.4.	Steg 2 i trebyggingen	42
6.4.1.	Beregning av tilstandspriser.....	42
6.5.	Beregning av korreksjonledd.....	43
6.6.	Tidsavhengig varians	43
6.7.	Implementering av trinomisk tre.....	44
7.	Generelt om driftsplanleggingsproblemet.....	45
7.1.	Innledning.....	45
7.2.	Hva øker kompleksiteten.....	45
7.3.	Formulering av optimeringsproblemet.....	47
7.3.1.	Matematisk formulering av driftsplanleggingsproblemet	47
7.3.2.	Hva inngår i ICF og FCF	48
7.4.	Fastkraftforpliktelsens virkning på driftsplanleggingen	49

8.	Implementering av algoritme	50
8.1.	Hvorfor velg SDP.....	50
8.2.	Framgangs måte ved SDP rekursjon.....	51
8.3.	Seriekorrelasjon i tilstandsvariable.....	52
8.3.1.	Generelt.....	52
8.3.2.	Seriekorrelasjon i pris	53
8.3.3.	Seriekorrelasjon i tilsig	54
9.	Interpolering.....	56
10.	Forenklinger i prismodellen.....	56
11.	Eksplisitt formulering av optimeringsproblemet.....	57
12.	Løsning av optimeringsproblemet	58
13.	Anbefalinger til videre arbeid	59
14.	Oppsummering.....	61
15.	Referanse- og litteraturliste	62
16.	Vedlegg	65
FIGUR 1.2-1:	PRISKURVER FRA TERMINMARKEDET	3
FIGUR 1.5-1:	Z-VERDIER I KORRELASJON	7
FIGUR 1.5-2:	TIDSREKKEPLOT FORWARDPRISER.....	8
FIGUR 2.3-1	AVVIK FRA GJENNOMSNTTLIG MAGASINFYLLING	12
FIGUR 2.4-1	HISTOGRAM, BOKS, TETTHET OG QQ-PLOTT.	13
FIGUR 4.1-1:	MARKEDSPRISEN FOR RISIKO SOM FUNKSJON AV MODNINGSTID	25
FIGUR 4.1-2:	RISIKOPREMIE SOM FUNKSJON AV MODNINGSTID	25
FIGUR 4.2-1:	DIVERSE PLOT VED SJEKK AV NORMALITETSANTAGELSEN.....	26
FIGUR 4.2-2:	OBSERVERT FORWARDPRISER PLOTTET MOT PREDIKERTE	28
FIGUR 4.3-1	KORRELASJON FORWARDPRIS OG TILSIG	29
FIGUR 4.3-2:	Z-VERDIER I TESTEN AV OM KORRELASJON ER SIGNIFIKANT.....	30
FIGUR 4.3-3	KORRELASJONSPLOTT.....	31
FIGUR 5.2-1:	TRINOMISK TRE.....	35
FIGUR 6.2-1:	TRINOMISK TRE I HULL & WHITE MODELLEN.....	39
FIGUR 6.3-1:	FORGRENINGSMØNSTER FRA NODE I HULL & WHITE MODELLEN	40
FIGUR 6.6-1:	OBSERVERT STANDARDAVVIK PLOTTET MOT MODELLERT.....	44
FIGUR 7.3-1:	SAMMENHENG MELLOM ICF OG FCF	49
FIGUR 8.2-1:	DEFINISJON PÅ SYSTEMTILSTANDER.....	51
FIGUR 8.3-1:	OVERGANGER I TRINOMISK TRE	53

TABELL 1.1-1: TIDSPUNKT FOR LEVERING FORWARDER	2
TABELL 1.5-1 DESKRIPTIV STATISTIKK FORWARDPRISER.....	6
TABELL 1.5-2 KORRELASJON MELLOM FORWARDPRISER.....	7
TABELL 2.3-1 DESKRIPTIV STATISTIKK MAGASINDATA	11
TABELL 4.1-1 PARAMETER VERDIER	24

Innledning

Driftsplanlegging i hos vannkraftprodusenter skjer i dag med bruke av forskjellige metoder. Felles for alle er hvordan pris håndteres, og i dag er det vanlig å bruke den forventede spotpris som inndata til disse metodene. Denne prisen er ikke risikjustert og det fører ikke til en optimal løsning av planleggingsproblemet til vannkraftprodusenten. Ved å bruke risikjusterte priser i denne sammenhengen kan man finne en mer optimal løsning og markedsværdien driftskontantstrømmene til vannkraftprodusenten maksimeres [Flethen S.-E (2000)]. At så er tilfelle er ikke en allment godtatt oppfatning, og hensikt med oppgaven er å utvikle et modellrammeverk for å kunne studere ovenfor nevnte påstand.

Forventet spotprisprognoser kommer i dag normalt fra ”bottom-up” modeller som Samkjøringsmodellen og brukes som inndata i driftsplanleggingen av vannkraftverk. I de fleste miljø som jobber med driftsplanlegging av vannkraftverk er bruken av forventet spotpris en etablert praksis i denne sammenheng. Erfaringsdata fra driftsplanlegging gjort med risikjusterte priser, det vil si forwardpriser, er derfor ikke allment tilgjengelig.

Følgen av etablert praksis i bransjen er det kun finnes planleggingsmodeller som bruker forventet spotpris som inndata. For å kunne studere påstanden ovenfor, må det utvikles eller tilpasses modeller for forwardpris, forventet spotpris, tilsig og en optimeringsmodell som bruker pris og tilsigsmodellene som input.

Resultatet fra optimeringen skal danne grunnlaget for studier av påstanden om bruken av forwardpriser er mer optimalt enn forventet spotpris i driftsplanlegging for en vannkraftprodusent.

1. Datagrunnlag forward- og futures priser

1.1. Terminstruktur

Datasettet som vil bli brukt til å tilpasse en modell for forwardprisene er hentet fra Nordpool og dekker tidsrommet fra uke 40 i 1995 og fram til uke 4 i 2002. Grunnlagsdataene er hentet fra forward- og futures priser på kraftbørsen Nordpool. I motsetning til andre børser har terminstrukturen her en annen oppbygging.

Forwarder og futures handles i blokker og gir derfor opphav til en priskurve som ikke er glatt. Futuresprodukter handles innværende år og omfatter ukes og blokkprodukter. Ett år er delt i 13 blokker av 4 uker. Blokk 1 har levering i uke 1-4 og blokk 2 i uke 5-8 og så videre. Blokkene løses opp fredag før første uke i en blokken har levering. Etter at en blokk er løst opp, handles ukene i den blokken som ukesprodukter.

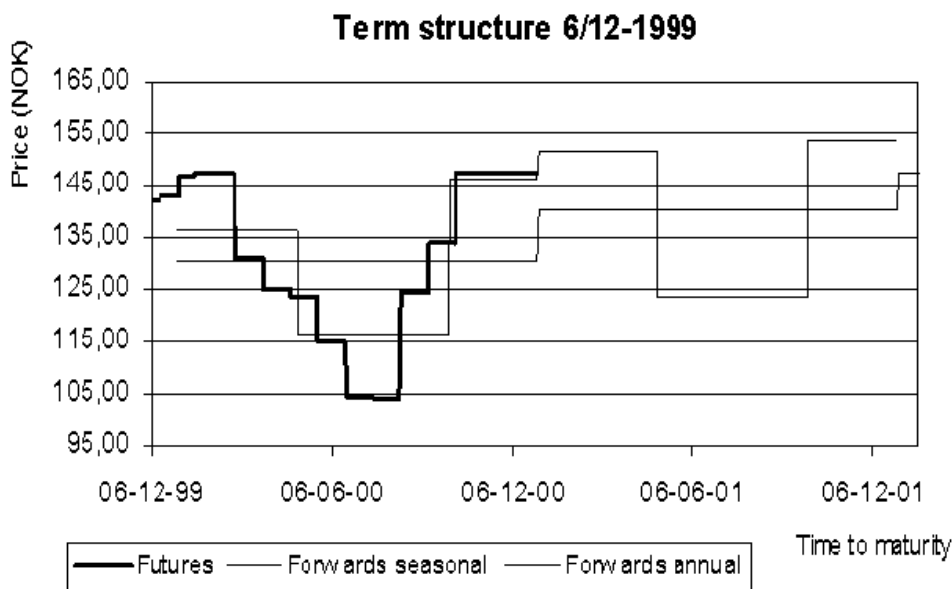
Forwardprodukter handles i blokker som dekker sesonger og år. Sesongene er vinter 1, sommer og vinter 2. Årsprodukter selges i blokker som har levering for inntil fire år fram i tid. Leveringstidsrom for disse produktene er angitt i Tabell 1.1-1.

Sesong	Tid
Vinter 1	1. Januar- 30. April
Sommer	1. Mai- 30. September
Vinter 2	1. Oktober- 31. Desember
År	1. Januar- 31. Desember

Tabell 1.1-1: Tidspunkt for levering forwarder

1.2. Glatting av data

Den beskrevne terminstrukturen gir en priskurve som illustrert i Figur 1.2-1 nedenfor.



Figur 1.2-1: Priskurver fra terminmarkedet

Terminstrukturen fører også til at prisene som kan observeres markedet i enkelte tilfeller bygger på få observasjoner. Særlig gjelder dette for produkter med lang modningstid. Dette er et kjent fenomen også i marked med en høyere likviditet en det nordiske kraftmarkedet [Fleten & Lemming, 2001]. Enkelte handledager kan det også skje at noen terminkontrakter ikke er omsatt, og dermed vil datamaterialet være diskontinuerlig. Særlig gjelder dette kontrakter med lang modningstid. I noen tilfeller vil det være behov for andre modningstidspunkt enn de som kan observeres i markedet til investeringsbeslutninger eller andre formål. I så tilfelle må prisene glettes for at de skal kunne brukes, og metoder for å gjøre dette er vanligvis regresjon eller glattingfunksjoner. Forutsetningen for at disse metodene skal gi gode resultat er at markedet er både modent og likvid. Disse betingelsen er ikke tilstede i kraftmarkedet, og metodene lar seg derfor ikke bruk her [Fleten & Lemming, 2001]. En annen metodikk for å glatte priskurver fra kraftmarkedet er derfor nødvendig.

1.2.1. Metodikk for glatting

Fordi ordinære metoder ikke, som tidligere nevnt, kan brukes til å skape ett glattet datasett, er det utviklet andre metoder for dette formålet. Fleten & Lemming (2001) har laget en helt ny metode for dette formålet som er kalt Term Structure Generation (TSG). Som navnet tilsier er dette en metode for å konstruere prisdata for ikke observerbare modningstidspunkt.

I TSG minimeres en vektet sum av kvadratisk avvik og en glattingsfunksjon ved bruk av optimering slik at en glatt forwardkurve tilpasses. Som inndata til TSG kreves beste pris kjøper er villig til å betale og laveste pris det selges for alle modningstidspunkt som finnes i perioden som det skal lages en forwardkurve. Disse to prisene gir nedre og øvre grenses i restriksjonen som inngår i optimeringsproblemet som løses i TSG. I tillegg inngår også en prisprognose i samme periode fra en eller annen "bottom-up"¹ modell, for ytterligere detaljer se [Fleten & Lemming, 2001].

Ved bruk av metoden kan du for en handledag konstruere et glattet datasett med daglig eller ukentlig oppløsning på forwardpris relativt til dagen eller uken. Her brukes det ukentlig oppløsning og forwardprisen for følgende uker er basert på markeddata for siste handeldag uken før. Konkret innebære dette at om siste handeldag er torsdag i denne uken så vil en ukes forwardprisen gjelde for kraft levert fra og med først kommende mandag.

Å konstruere datasettet på denne måten fører også til at det kan være mer eller mindre enn en uke mellom observasjons tidspunktene. Grunnen er at Nordpool åpner senere og stenger tidligere noen uker. I slike tilfeller kan tiden mellom observasjonene være mindre eller lengre enn en uke. Datasettet inneholder derfor uker med usynkron handel som gir opphav til skjevheter i fordelingen til dataene [Solibakke, 2001b]. Fordi dataene kommer fra en konstruerte forwardkurve og dermed har i boende unøyaktigheter er effekten av dette vanskelig å påvise. Nøyaktigheten til modellen som blir tilpasset vil normalt ikke være god, og slik effekter forutsettes absorbert i den feilmarginen som en slik modell vil ha.

¹ Samkjøringsmodellen er en typisk bottom-up modell som kan generere slik prisprognoser.

1.3. Sensurering

På Nordpool handles det terminkontrakter med modningstidspunkt opptil 3 år. Strukturen som brukes på kontraktene gjør at 3 års modningstid kun oppnåes ved kjøpe i begynnelsen av året. Ved slutten av året er modningstid på 2 år den høyest oppnåelige, og grunnen er at for år tre selges det kun en forward for hele året. Dermed er det i siste uke i 2001 kun 105 uker til en års kontrakt i 2004 har modningstidspunkt. Følgene av dette er at datasettet som blir generert av TSG består av data for 156 uker når forwardpriser genereres i uke 1 og avtar der etter til 104 uker mot slutten av året. For å skape et datasett egnet for statistisk analyse fjernes forwardpriser med modningstid over 104 uker fra datasettet. I tillegg fjernes de 10 første ukene fra datasettet fordi maksimal modningstid er mindre enn 104 uker for disse ukene. Det gjenværende datasette inneholder nå alle genererte forwardprisene med opptil 104 ukers modningstid fra og med uke 50 i 1995 fram til uke 4 i 2002, totalt 319 uker.

1.4. Datautvalg

Det total datasettet består nå av en matrise med dimensjonene 319 x 104. For å lage en modell som beskriver forwardprisen godt nok, er det ikke nødvendig å bruke alle datapunktene for å fange opp hvordan risikopremien er i markedet. Velger derfor å bruke samme datautvalg som Koekebakker & Ollmar (2001), og disse er valgt slik at de reflekterer faktiske handlede kontrakter. Antallet modningstidspunkt som velges er 21, og med verdien T_m i uker fram til modningstidspunkt. Valgte modningstidspunkt er da $T_1, \dots, T_M = [1, 2, 3, 4, 5, 6, 7, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 70, 88, 104]$. Hvor de 7 første representerer ukeskontrakter og de 11 neste månedskontrakter og de siste 3 er sesongkontrakter.

1.5. Deskriptiv statistikk og kontroll av prisdata

Fra de valgte modningstidspunktene er deskriptiv statistikk beregnet for noen få tidspunkt i tabellen nedenfor. En komplett tabell med deskriptiv statistikk for alle tidspunkt finnes i vedlegg 1.

	Uke 1	Uke 7	Uke 12	Uke 24	Uke 52	Uke 70	Uke 104
Minimum	45,3	70,0	69,8	72,2	104,8	104,7	115,6
Første kv.	107,3	109,9	113,3	130,2	137,2	136,8	146,4
Middelverdi	153,3	157,5	159,4	161,7	162,0	161,8	164,2
Median	138,5	149,8	152,7	147,1	157,4	155,8	162,3
Tredje kv.	193,3	189,7	183,7	174,0	180,5	183,7	177,9
Maximum	352,8	354,6	369,2	383,5	260,5	299,9	267,3
Standaravvik	62,3	61,8	61,6	59,4	33,8	37,0	29,1
Skeivhet	0,8	1,1	1,3	1,7	0,7	1,2	0,8
Kurtosis	0,3	1,0	1,7	3,1	0,2	1,9	1,0

Tabell 1.5-1 Deskriptiv statistikk forwardpriser

Hvor skeivhet og kurtosis er definert som følger, øvrige ting antas kjent:

$f(X)$: Fordelingens funksjonen til x .

$$\mu = E[X] = \int_{-\infty}^{\infty} x \cdot f(x) \cdot dx \quad (\text{Likning 1.5-1})$$

$$\sigma^2 = E[(X - \mu)^2] \quad \text{Variansen til en variabel.} \quad (\text{Likning 1.5-2})$$

$$\alpha_3 = \frac{E[(X - \mu)^3]}{\sigma^3} \quad \text{Skeivheten til fordelingen.} \quad (\text{Likning 1.5-3})$$

$$\alpha_4 = \frac{E[(X - \mu)^4]}{\sigma^4} \quad \text{Kurtosis til fordelingen.} \quad (\text{Likning 1.5-4})$$

Verdiene på skeivhet angir i hvilken retning fordelingen er forskjøvet og kurtosis beskriver hvor spiss peaken er i fordelingen er. Alle normalfordelte variable har $\alpha_3 = 0$ og $\alpha_4 = 3$. I Tabell 1.1-1 er skeivhet og kurtosis ikke nær disse verdiene i de fleste tilfellene, noe som indikerer at prisene ikke er normalfordelte. Skeivhetene er positive og innbærer at fordelingen er forskjøvet til høyre. De fleste kurtosisverdiene er mindre enn 3 og innbærer en tykkhalet fordeling og flat fordeling av prisen. En vanlig antakelse er at ved kurtosisen $\alpha_4 = 3 \pm 0.5$ så kan det antas normalfordeling, en antakelse som kun holder for 4 verdier i hele datasettet (se vedlegg 1).

Korrelasjon mellom forwardprisen disse ukene er presentert i tabellen nedenfor.

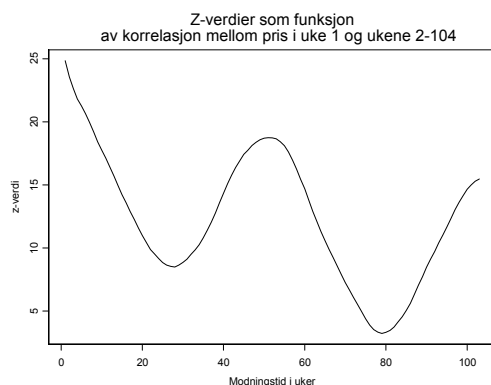
	Uke 1	Uke 7	Uke 12	Uke 24	Uke 52	Uke 70	Uke 104
Uke 1	1,00	0,93	0,87	0,74	0,83	0,60	0,70
Uke 7	0,93	1,00	0,97	0,82	0,79	0,79	0,67
Uke 12	0,87	0,97	1,00	0,89	0,71	0,88	0,58
Uke 24	0,74	0,82	0,89	1,00	0,51	0,86	0,32
Uke 52	0,83	0,79	0,71	0,51	1,00	0,43	0,94
Uke 70	0,60	0,79	0,88	0,86	0,43	1,00	0,34
Uke 104	0,70	0,67	0,58	0,32	0,94	0,34	1,00

Tabell 1.5-2 Korrelasjon mellom forwardpriser

Nedenfor er prisen for forwarder med modningstid på henholdsvis 1 uke, 52 uker og 104 uker plottet i samme diagram. Verdiene fra og med uke 50 i 1995 til og med uke 4 i 2002.

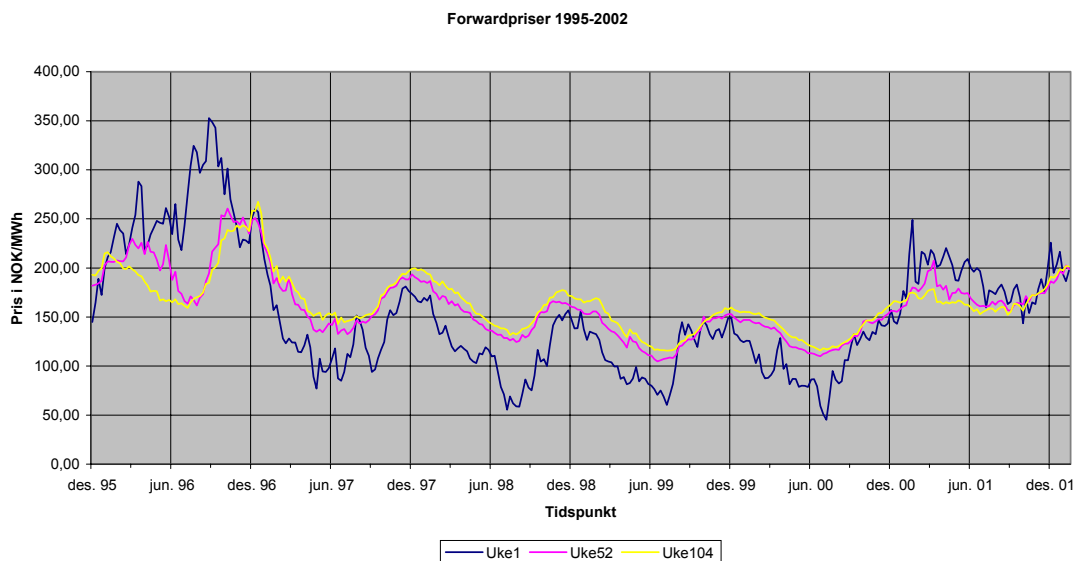
1.5.1. Signifikans i korrelasjon

For å sjekke om korrelasjon er signifikant brukes Kendall test (se vedlegg 5), testen forutsetter ikke at dataene tilhører en bestemt fordeling. Det sjekkes kun om korrelasjon mellom priser i uke 1 og øvrige uker er signifikant. Verdien av teststatistikken $z \sim N(0,1)$ er plottet som en funksjon av modningstid i Figur 1.5-1.



Figur 1.5-1: z-verdier i korrelasjon

Som Figur 1.5-1 viser, er det få lave z-verdier og den laveste verdiene observerte verdien er $z = 3.22$. Det kan derfor konkluderes med at alle priser er signifikant korrelert med prisen i uke 1.



Figur 1.5-2: Tidsrekkeplot forwardpriser

Inngangsverdiene som brukes i TSG modellen er bid- og askpriser fra Nordpool og prisprognoser fra en bottom-up modell. Med disse verdiene tilgjengelig, genereres det en fil for hver handledag/uke av TSG med forwardpriser. I dette tilfellet ble det 329 filer som data skulle hentes fra. Fordi antallet prosesser som dataene går gjennom er mange, er sjansene for feil i datasettet høyt. For å verifiserer at så ikke er tilfelle, er det laget 3 scatterplott² av datasettet, og disse viste ikke noen indikasjoner på at datasettet inneholdt ekstremverdier. En visuell sammenlikning med tilsvarende diagram i [Koekebakker & Ollmar, 2001, side 7] viser at datasettet har likheter som forventet. Ut fra dette samt den deskriptive statistikken³ og scatterplottene, konkluderer jeg derfor med at datasettet ikke inneholder ekstremverdier og er derfor egnet til videre bruk.

² Se vedlegg nr 2

³ Se også vedlegg nr 1

2. Datarepresentasjon av tilsig

2.1. Tilsigsdata

For å kunne lage en modell som fanger opp korrelasjon mellom forwardprisen og tilsiget, bør informasjon om dette inkluderes når i modellen tilpasses. Å finne enkle modeller som gir et enkelt mål på tilsig i form av regn og akkumulert nedbør i form av snø eller grunnvann for hele landet er vanskelig.

I bransjen er det vanlig å betrakte den hydrologiske balansen for hele landet, som tar hensyn til grunnvannstand, magasinnivå og snømagasin. Hvordan denne balansen framkommer, kjenner jeg ikke til, men det er grunn til å tro at antall inputparametere er høyt fordi den dekker hele landet. På lokalt nivå brukes andre modeller, også disse svært kompliserte fordi antall inputparametere er høyt og kalibreringen disse omfattende.

Andre grunner finnes også, og uten å begrunne disse her kan korrelasjon og datamengde nevnes som noen av de viktigste. Innvirkningen av korrelasjon og datamengde er grundig beskrevet i [Gjelsvik, A. (1992)], og taes også opp i [Aronsen & Dybdahl 2001, kap 6].

Et sett av tilgjengelige data som gir en indikasjon på tilsiget er fyllingsgraden⁴ på vannmagasinene i Norge, og disse kan inn hentes fra [SSB 1]. Dataene er gitt som total fyllingsgrad for hele landet og fyllingsgrad for hvert av de 3 delområdene. Statistikken dekker 97,1 % av landets total magasinkapasitet og publiseres hver onsdag. Data som publiseres onsdager er aktuell magasinfylling mandag samme uke. For ytterligere detaljer om område inndeling m.m. se [SSB 1].

⁴ Fyllingsgrad = [Magasinnivå (MWh)]*100 %/[Total magasinkapasitet(MWh)]

2.1.1. Informasjon i magasindata

Informasjon som kan finnes direkte eller indirekte fra magasinfyllingen er blant annet ukentlig forbruk, ukentlig tilsig og energi tilgjengelig for framtidig bruk. Det ukentlige tilsiget av vann til magasinet finnes ved å trekke fra det ukentlige forbruket fra fyllingsmengde (gitt i GWh). Hvor stort tilsig som finnes tilgjengelig i snø ”magasinet” i nedbørsfeltene rundt kraftverkene ikke angitt. Informasjonene om fyllingsgrad er tilgjengelig for alle og inngår trolig som et av mange informasjons-element som påvirker forwardprisen.

I hvilken grad det er sammenheng mellom forwardprisen og magasinfylling kan best beskrives med korrelasjon. Om det finnes en signifikant korrelasjon mellom magasinfylling og forwardpriser kan informasjonen om magasinfylling brukes i en forwardprismodell. En korrekt modellen vil gi prediksjon/simuleringsverdier som er korrelert med magasinfyllingen. Om det er korrelasjon eller ikke mellom magasinfylling og forwardpriser vil bli analysert senere.

2.2. Magasinfyllingsdata

Som tidligere nevnt, er frekvensen i magasindataene publisert med uker, og dette passer meget bra i forhold til forwardprisen som også er generert med TSG med ukesfrekvens. Magasinfyllingsdata er derfor hentet fra uke 50 i 1995 til og med uke 4 i 2002 (319 uker), og dette er samme tidsrom som forwardprisen er generert for. Datasettet er komplett for denne perioden og er derfor ikke manipulert på noen måte. Opprinnelig består datasettet av magasinfyllingsgrad angitt i prosent. Settet består av en kolonne for total fyllingsgrad for landet og 3 kolonner som gir fyllingsgrad i hver del området⁵.

⁵ Andelen av total magasinkapasitet for hvert av del områdene 1-3 er henholdsvis 36, 28 og 36 %.

2.3. Deskriptiv statistikk

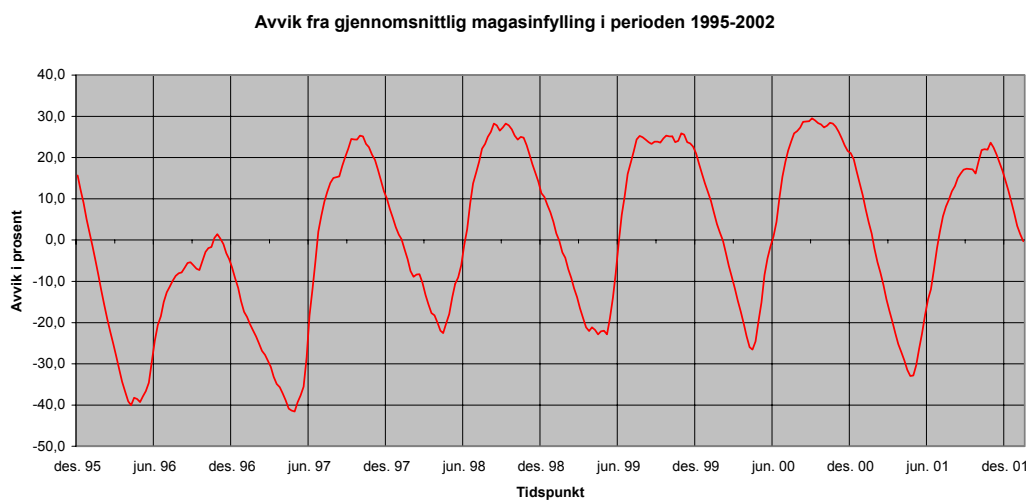
Som nevnt består dataene opprinnelig av 4 kolonner med prosentvis fyllingsgrad på magasin. I tillegg til disse er det også lagt til 4 kolonner med avvik i prosent regnet fra gjennomsnittlig fyllingsgrad for hver opprinnelige kolonne. Gjennomsnittet for hver kolonne er regnet over alle observasjonen i tidsperioden dataene dekker. I tabellen nedenfor er deskriptiv statistikk regnet ut for magasindataene, alle verdier gitt i prosent. Kolonneoverskrift TH er magasinifylling for hele landet, og O1-O3 er for delområdene i landet. Avvik fra gjennomsnittlig fylling i disse områdene er indeksert med ATH-AO3.

	TH	O1	O2	O3	ATH	AO1	AO2	AO3
Minimum	23,50	16,70	27,60	25,40	-41,56	-46,14	-38,43	-41,95
Første kv.	48,95	43,70	52,50	51,90	-16,11	-19,14	-13,53	-15,45
Middelverdi	65,35	62,79	66,43	68,03	0,29	-0,04	0,41	0,67
Median	66,40	63,80	67,50	71,00	1,34	0,96	1,47	3,65
Tredje kv.	84,15	84,85	83,30	85,75	19,09	22,01	17,27	18,40
Maximum	94,60	96,10	94,90	96,50	29,54	33,26	28,87	29,15
Standaravvik	20,17	22,82	18,37	19,23	20,17	22,82	18,37	19,23
Skevhet	-0,28	-0,25	-0,28	-0,40	-0,28	-0,25	-0,28	-0,40
Kurtosis	-1,09	-1,22	-1,03	-0,95	-1,09	-1,22	-1,03	-0,95
LCL	62,43	59,48	63,77	65,24	-2,63	-3,36	-2,26	-2,12
UCL	68,28	66,10	69,10	70,82	3,22	3,27	3,07	3,46

Tabell 2.3-1 Deskriptiv statistikk magasindata

I Tabell 2.3-1 er verdiene som måles utenom LCL og UCL selvforklarende eller beskrevet tidligere i kapittel 1.5. LCL er nedre grense på ett 99 % konfidensintervall for middelveien og UCL det øvre med samme signifikans. Disse verdiene indikerer at middelveien for alle ligger innenfor et smalt område. En naturlig sjekk er om disse verdien kommer fra en normalfordeling eller fra en annen fordeling i tilfelle det skal simuleres forwardpriser. Sjekk av normalitet vil bli utført i neste avsnitt.

Velger å bruke avvik fra gjennomsnittsverdien for hele landet som mål på tilsig. Denne verdien ser vi at har middelvei veldig nær null, selv om den er under påvirkning fra ekstremåret 1996. Nedenfor er en tidsserie plott av verdien.



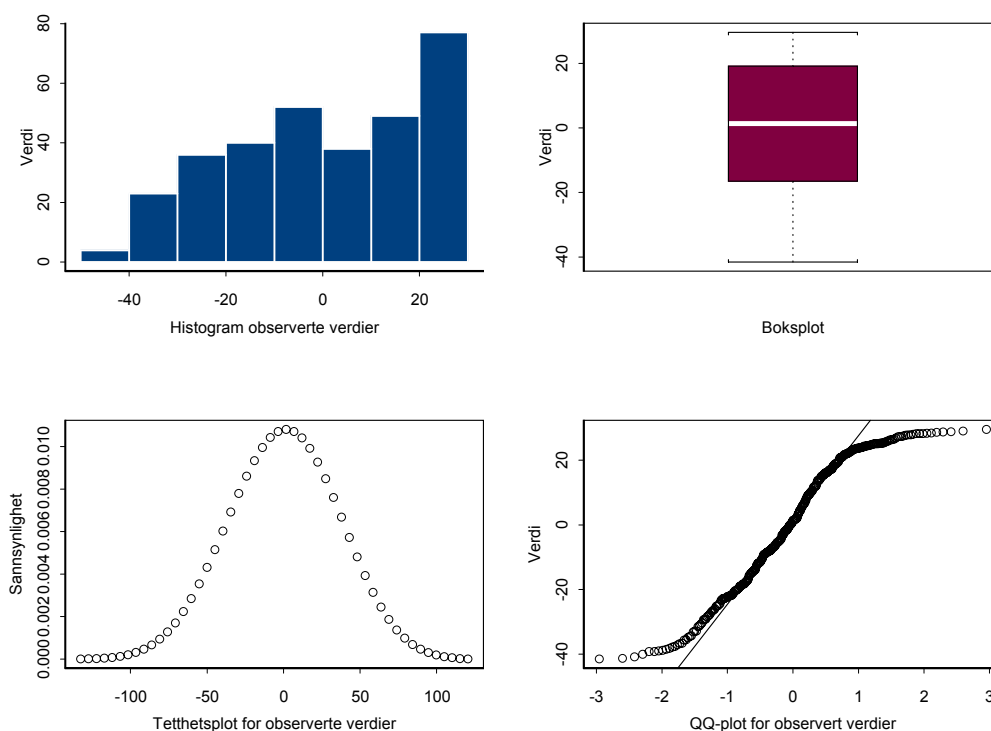
Figur 2.3-1 Avvik fra gjennomsnittlig magasinifylling

Når en ser bort fra året 1996 svinger verdiene omkring null. Betrakter vi kurven for forwardprisene i 1996 har denne også et betydelig avvik fra øvrige år. Noe som tyder på at korrelasjonen mellom tilsig og pris kan ivaretaes ved å bruke magasinifylling som en indikasjon på tilsiget. Om så er tilfelle blir testes i kapittel 4.3.1.

2.4. Sjekk av fordelingsegenskaper til dataene

En første sjekk på om dataene er normalfordelt er å betrakte histogram, boks, tetthet og qq-plott i Figur 2.4-1.

Avvik fra gjennomsnittlig magasinifylling



Figur 2.4-1 Histogram, boks, tetthet og qq-plott.

Tetthetsplottet viser at dataene kanskje er normalfordelt, men øvrige plott viser dette helt åpenbart ikke er tilfelle. Fordi dataene er påvirket av både naturlige fenomen og menneskelig handlinger kan ikke historiske tall materiale som stammer fra før forwardmarkedet brukes til å tilpasse fordelingen. Årsaken er at driftsplanlegging av kraftverkene til en viss grad styrt av markedets forventning til priser og sesong.

I bruke av ordinære fordelinger skaper dette et problem om vi ikke bruker en betinget fordeling. Dette skjer fordi at tiden påvirker sannsynligheten for en positiv eller negativ endring i magasinifyllingsgraden. I vårflommen er det åpenbart at den mest sannsynlige endringen er positiv og en fordeling må ta hensyn til dette.

3. Valg av forwardpris modell

3.1. Innledning

Det er gjort mange studier av egenskapene til forwardpriser i det nordiske kraftmarkedet og omtrent like mange modeller for spot- og forwardpriser er foreslått. Ikke noen av modellene som er foreslått har tatt hensyn til korrelasjonene mellom spot-/forwardpris og tilsig. Derfor er det forsøkt tilpasset flere modeller som tar hensyn til dette.

Utgangspunktet for modellene som er forsøkt tilpasset her er hentet fra [Lucia & Schwartz 2001]. I denne artikkelen er det tilpasset flere modelltyper, blant annet er en- og tofaktormodeller tilpasset. Av hensyn til kompleksiteten i tilpassning er ikke tofaktormodeller betraktet, selv om resultatene fra [Lucia & Schwartz 2001] tyder på at disse kan være bedre.

Enfaktormodellene som er tilpasset har brukt henholdsvis spotpris og log spotpris som utgangspunkt. Funn tyder på at modeller basert på spotpris er omtrent like bra som de som bruker log spotpris. Her velges det å bruke en log spotprismodell for å unngå problemet med negative priser. Årsaken er at en ordinær spotprismodell kan gi negative priser når den brukes på noen av modellene som brukes for å lage diskrete priser.

3.2. Metodikk

For å bli kjent med metoden som er brukt i [Lucia & Schwartz 2001], vil først en modell som er veldig lik modellene i samme artikkel tilpasses. Parametrene i en slik modell bør ikke være så veldig forskjellig, og vil dermed gi en indikasjon på om dette er gjort riktig.

Når første del av tilpassningen er gjennomført, vil det testes flere varianter av den deterministiske funksjonen der hensynet til magasinnivå taes med. At denne sammenhengen taes inn i det deterministiske leddet skyldes at denne informasjonen er tilgjengelig når forwardprisen bestemmes, og dermed pr. definisjon deterministisk på dette tidspunktet.

Magasinnivået, eller avvik som brukes her er selvfølgelig ikke en deterministisk variabel, men den kan behandles slik i tilpassningen av forwardprismodellen, på samme måte spotprisen brukes. Om forwardprismodellen blir bra tilpasset når avviket fra magasinnivå er tatt inn i modellen, så kan det være hensiktsmessig å tilpasse en modell for denne variabelen (avviket fra normal nivå). Forwardprisbaner kan da simuleres ved å trekke utfall fra avviksmodellen og det stokastiske leddet i den opprinnelige forwardpris modellen.

I datagrunnlaget finnes det en periode hvor avviket i magasinnivå og priser var ganske stort, og prisene svingte veldig i dette tidsrommet. En modell som fanger opp korrelasjonen mellom pris og tilsig vil fange opp disse svingningen og avviket fra observert pris og modellert pris vil ikke være stort. Hvordan dette avviket arter seg i en modell som ikke tar hensyn til korrelasjon mellom tilsig og pris vil bli illustrert i den første modellen som tilpasses.

Et mål på hvor godt modellen som tar hensyn til korrelasjon mellom tilsig og pris er tilpasset, er å sammenlikne denne med en som ikke gjør det. Intuitivt vil enn forvente at modeller som ikke ivaretar hensynet til korrelasjon mellom pris og tilsig vil ha større avvik i tilpassningen. Noe som vil skje i de perioder hvor det opptrer ekstrem priser som følge ekstra ordinære tilsigsforhold.

3.3. Teoretisk grunnlag

Prisen kan beskrives slik at den består av ett deterministisk og et stokastisk ledd.

Følgende uttrykk for prisen kan da settes opp:

$$P_t = f(t) + X_t \quad (\text{Likning 3.3-1})$$

$f(t)$: Deterministisk ledd som er predikerbart.

Det antas videre at X_t følger en stokastisk prosess av formen:

$$dX_t = -\kappa \cdot X_t \cdot dt + \sigma \cdot dZ \quad (\text{Likning 3.3-2})$$

$\kappa > 0$, $X(0) = x_0$ og hvor dZ representerer ett inkrement i en standard Brownsk bevegelse Z_t . Derfor følger X_t en stasjonær middelretur prosess, eller en Orstein-Uhlenbeck prosess med 0 som forventningsverdi i det lange løp og med et middelretur rate lik κ .

Ved å omskrive (3.3-1) til $X_t = P_t - f(t)$ kan vi skrive (3.3-1) og (3.3-2) slik:

$$d(P_t - f(t)) = \kappa \cdot (f(t) - P_t) \cdot dt + \sigma \cdot dZ \quad (\text{Likning 3.3-3})$$

Dette uttrykket viser at dersom P_t avviker fra det deterministiske leddet $f(t)$ så trekkes det tilbake med en rate som er proporsjonal med avviket. Den eneste variabelen som har stokastiske egenskaper er da X_t og vi kan derfor kalle X_t for tilstandsvariabel.

Hvor $a(t)$ er den deterministiske funksjonen av t definert ved:

$$a(t) \equiv \frac{1}{\kappa} \cdot \frac{df(t)}{dt} + f(t) \quad (\text{Likning 3.3-5})$$

En eksplisitt løsning av (3.3-2) kan finnes og gir sammen med (3.3-1) denne løsningen:

$$P_t = f(t) + X_0 e^{-\kappa t} + \sigma \cdot \int_0^t e^{-\kappa(s-t)} \cdot dZ(s) \quad (\text{Likning 3.3-6})$$

Ved å sette $X_0 = P_0 - f(0)$ kan vi sette opp disse uttrykkene for betinget varians og forventning under for utsettning om at prisen er normalfordelt.

$$\left. \begin{aligned} E_0(P_t) &\equiv E_0(P_t | X_0) = f(t) + (P_0 - f(0))e^{-\kappa t} \\ \text{Var}_0(P_t) &\equiv \text{Var}_0(P_t | X_0) = \frac{\sigma^2}{2 \cdot \kappa} (1 - e^{-2\kappa t}), \quad \kappa > 0 \end{aligned} \right\} \quad (\text{Likning 3.3-6})$$

Fordi vi ønsker å bruke de risikojusterte prisene i produksjonsplanleggingen må vi risikojustere prisen for tilstandsvariabelen X_t istedenfor den virkelige gitt av (3.3-2).

Ved å ta hensyn til den ikke handlbare naturen til X_t , så vil standard arbitrasjeargumenter med to derivater tillate oss å oppnå den risikonøytrale prosessen for X_t .

$$dX_t = \kappa(\alpha^* - X_t) \cdot dt + \sigma \cdot dZ^* \quad (\text{Likning 3.3-8})$$

$$\alpha^* \equiv \frac{\lambda \cdot \sigma}{\kappa} \quad (\text{Likning 3.3-9})$$

Hvor dZ^* står for et inkrement til Z_t^* , en standard brownsk bevegelse under risikonøytralt sannsynlighetsmål. λ står for pris pr enhet risiko og er knyttet til tilstandsvariabelen X_t . Vi antar λ konstant, men generelt er λ en funksjon av X_t og t .

Ved å følge de samme stegene som tidligere, kan den eksplisitte løsningen av den stokastiske differensiallikningen (3.3-8) finnes ved bruk av Ito's lemma og denne er gitt ved:

$$P_t = f(t) + X_0 e^{-\kappa t} + \alpha^* (1 - e^{-\kappa t}) + \sigma \cdot \int_0^t e^{-\kappa(s-t)} \cdot dZ^*(s) \quad (\text{Likning 3.3-10})$$

Hvor α^* er som definert i (3.3-9). Fra dette har vi at P_t er betinget normal under risikonøytralt mål, med forventningsverdi:

$$E^*(P_t) = f(t) + X_0 e^{-\kappa t} + \alpha^* (1 - e^{-\kappa t}) \quad (\text{Likning 3.3-11})$$

Verdien av et hvert derivat må være lik den forventede verdien, under risikonøytral verdisetting, diskontert med risiko fri rente. Den risiko frie renten antas konstant.

Verdiene av en forward eller futureskontrakt basert på spotprisen med modningstid T må da være:

$$v_0(X_T, T) = e^{-r \cdot T} E_0[P_T - F_0(P_0, T)] \quad (\text{Likning 3.3-12})$$

Hvor $F_0(P_0, T)$ er forwardprisen ved inngåelse av en kontrakt med modningstidspunkt T , og r er den risiko frie renten. Fordi verdien av en slik kontrakt må være lik null når den inngås, oppnår vi følgende løsning på forward og futuresprisen ved å bruke (11) og (1) for $t=0$:

$$F_0(P_0, T) = E_0(P_T) = f(T) + (P_0 - f(0))e^{-\kappa \cdot T} + \alpha^* (1 - e^{-\kappa \cdot T}) \quad (\text{Likning 3.3-13})$$

$$\text{med } \alpha^* = \frac{-\lambda \cdot \sigma}{\kappa}$$

3.4. Overgang fra spotpris til log spotpris modell

En overgang fra en modell basert på spotpris til en modell basert på log spotpris skjer ved å jobbe med den naturlige logaritmen til spotprisen. Det antas da at prosessen som log spotprisen følger er denne.

$$\ln(P_t) = f(t) + Y_t \quad (\text{Likning 3.4-1})$$

Hvor $f(t)$ er som tidligere definert. Den stokastiske prosessen som Y_t følger er gitt ved:

$$dY_t = -\kappa \cdot Y_t \cdot dt + \sigma \cdot dZ \quad (\text{Likning 3.4-2})$$

Ved hjelp av en rekke omskrivninger tilsvarende de brukt for å ulede likning (3.3-4) kan prisprosessen uttrykkes med denne likningen.

$$dP_t = \kappa (b(t) - \ln(P_t)) \cdot P_t \cdot dt + \sigma \cdot P_t \cdot dZ \quad (\text{Likning 3.4-3})$$

Med

$$b(t) \equiv \frac{1}{\kappa} \cdot \left(\frac{\sigma^2}{2} + \frac{df}{dt}(t) \right) + f(t)$$

Ved å anta at Y_t følger prosessen gitt i (3.4-2), forutsettes log spotprisen å være betinget normalfordelt med betinget forventningsverdi og varians gitt ved:

$$\left. \begin{aligned} E_0(\ln(P_t)) &= f(t) + (\ln(P_0) - f(0)) \cdot e^{-\kappa \cdot t} \\ \text{Var}_0(\ln(P_t)) &= \frac{\sigma^2}{2 \cdot \kappa} \cdot (1 - e^{-2\kappa \cdot t}) \quad \kappa > 0 \end{aligned} \right\} (\text{Likning 3.4-4})$$

Dermed har log spotprisen de samme egenskapene som er utledet i forrige avsnitt. Spotprisen P_t har da betinget log normal fordeling, som leder tilfølgende uttrykk for forventet spotpris og varians for denne:

$$E_0(P_t) = \exp\left(f(t) + (\ln(P_t) - f(0)) \cdot e^{-\kappa \cdot t} + \frac{\sigma^2}{2 \cdot \kappa} (1 - e^{-2\kappa \cdot t})\right) \quad (\text{Likning 3.4-5})$$

og

$$Var_0(P_t) = E_0(P_t)^2 \left[\exp\left(\frac{\sigma^2}{2 \cdot \kappa} (1 - e^{-2\kappa \cdot t})\right) - 1 \right] \quad (\text{Likning 3.4-6})$$

Disse finnes ved å bruke standard likninger for forventning og varians til en log normal variabel. Overgangen til risikojusterte priser skjer ved at prosessen for Y_t defineres som i (3.3-8) og ved å bruke (3.4-1) og (3.4-2) kan den risikojusterte log spotprisen uttrykkes slik:

$$\ln(P_t) = f(t) + Y_0 e^{-\kappa \cdot t} + \alpha^* (1 - e^{-\kappa \cdot t}) + \sigma \cdot \int_0^t e^{-\kappa \cdot (t-s)} \cdot dZ^*(s) \quad (\text{Likning 3.4-7})$$

Den risikojusterte forventningsverdien og variansen er da gitt ved:

$$\left. \begin{aligned} E_0^*(\ln(P_t)) &= f(t) + (\ln(P_0) - f(0)) \cdot e^{-\kappa \cdot t} + \alpha^* \cdot (1 - e^{-\kappa \cdot t}) \\ Var_0^*(\ln(P_t)) &= \frac{\sigma^2}{2 \cdot \kappa} \cdot (1 - e^{-2\kappa \cdot t}) \quad \kappa > 0 \end{aligned} \right\} \quad (\text{Likning 3.4-8})$$

Ved å bruke samme argumentasjon som for utledningen av likning (3.3-13) kan følgende uttrykk for forward/futuresprisen settes opp:

$$F_0(P_0, T) = E_0(P_T) = \exp\left[f(T) + (P_0 - f(0))e^{-\kappa \cdot T} + \alpha^* (1 - e^{-\kappa \cdot T}) + \frac{\sigma^2}{4 \cdot \kappa} (1 - e^{-2\kappa \cdot T})\right] \quad (\text{Likning 3.4-9})$$

med $\alpha^* = \frac{-\lambda \cdot \sigma}{\kappa}$: som er risikopremien (Likning 3.4-10)

3.5. Valg av første deterministisk funksjon

Funksjonen $F = f(T)$, som er det deterministiske tidsfunksjonen i forwardprismodellen, er ikke lik de som er brukt i [Lucia & Schwartz 2001], og årsaken til dette er flere. Hovedgrunnen er at det ønskes en enkel modell og derfor droppes dummyvariablene som brukes for å definere sesongledd, se [Lucia & Schwartz 2001 side 18] for flere detaljer om disse. Det kunne også vært brukt en dummyvariabel for det året som har 53 uker(1998), men også dette utelates fordi det kun gjelder for ett år. Følgende deterministisk funksjon velges brukt i likningen som tilpasses først, og denne vil også være utgangspunktet for senere funksjoner:

$$F = f(T) = \alpha + \gamma \cdot \cos\left(\frac{2 \cdot \pi}{52} \cdot (t_a + \tau)\right) \quad (\text{Likning 3.5-1})$$

T = Modningstid for forwarden

t_a = (Dagens dato + Modningstid)

α : Middelveien til sesongfunksjonen.

γ : Amplitudeverdi, utslaget fra middelveiene.

τ : Fase forskyvningen av toppverdi, om denne er lik null oppnåes maks verdien i uke 52.

Som sesongledd er det valgt å bruk en cosinusfunksjon fordi denne har sin maksimale verdi når vinkelen er lik $2 \cdot \pi \cdot n$, $n = 0, 1, 2, 3, \dots, N$. Når $\tau = 0$ betyr dette at sesongfunksjonen har sin maksimale verdi ved $t=52$. Erfaringsmessig oppnåes maksimalpris i perioden januar-februar, og valget av cosinusfunksjonene som sesongledd er derfor mest hensiktsmessig slik at parameteren τ inntar enfornuftige verdi. Merk også modningstidspunktet regnes om til absolutt tid i sesongfunksjonen. Størrelses av dette leddet blir derfor avhengig av når på året forwarden modnes og ikke hvor lenge det er til modningstidspunktet ved inngåelse av kontrakten.

3.6. Estimering av den stokastiske prosessen for enfaktormodellen

I den likning (3.4-9) som er gjengitt neden for er den en rekke parametere som skal bestemmes for at likningen skal kunne brukes. Den enkleste og raskeste metoden er å bruke ulineær minste kvadraters metode for å bestemme disse samtidig. Her brukes statistikkprogrammet S-Plus fordi t-verdier og andre interessante statistiske størrelser beregnes samtidig. Ut fra disse verdiene kan en bestemme hvilke parametere som er signifikante i likningen. Løsningen på den stokastiske differensial likningen (3.4-9) som parameteren skal bestemmes for er gitt ved :

$$F_0(P_0, T) = E_0(P_T) = \exp \left[f(T) + (P_0 - f(0))e^{-\kappa \cdot T} + \alpha^* (1 - e^{-\kappa \cdot T}) + \frac{\sigma^2}{4 \cdot \kappa} (1 - e^{-2\kappa \cdot T}) \right]$$

$$\text{med } \alpha^* = \frac{-\lambda \cdot \sigma}{\kappa}$$

Med sesongleddet fra gitt av (3.5-1):

$$f(T) = \alpha + \gamma \cdot \cos\left(\frac{2 \cdot \pi}{52} \cdot (t_a + \tau)\right)$$

Ved å sette inn for (3.5-1) i (3.4-9) framkommer denne likningen:

$$F_0(P_0, T) = e^{\left(\alpha + \gamma \cdot \cos\left(\frac{2\pi}{52}(t_a + \tau)\right) + (P_0 - f(0))e^{-\kappa \cdot T} + \alpha^* (1 - e^{-\kappa \cdot T}) + \frac{\sigma^2}{4\kappa} (1 - e^{-2\kappa \cdot T}) \right)} \quad (\text{Likning 3.6-1})$$

Merk at $f(0)$ ikke betyr at innverdien i funksjonen er null, men at $f(0)$ er lik verdien på sesongfunksjonen for det tidspunktet som legges til grunn for prisingen av forward eller futuresproduktet. Parametrene i denne likningen kan ikke bestemmes simultant, men finnes stegvis. I første steget løses likningen for prisprosessen uten risikojustering. Likningen som parametrene skal bestemmes for er da denne:

$$F_0(\ln(P_t), T) = f(t) + (\ln(P_0) - f(0)) \quad (\text{Likning 3.6-2})$$

Dette forutsetter at følgende gjelder for prisen.

$$\ln(P)_t = f(t) + Y_t \quad (\text{Likning 3.6-3})$$

$$Y_t = (1 - \kappa)Y_{t-1} + \xi_t \quad (\text{Likning 3.6-4})$$

$$t = 0, 1, 2, \dots, N, \quad \xi_t \sim N(0, \sigma^2) \text{ og } \phi = 1 - \kappa.$$

Med disse betingelsene oppfylt kan følgende sammenheng formuleres:

$$z_t = f(\theta, T) + Y_t \quad (\text{Likning 3.6-5})$$

$$Y_t = \phi Y_{t-1} + \xi_t \quad (\text{Likning 3.6-6})$$

$\theta = [\alpha, \gamma, \tau]$: Vektor av parametere som skal bestemmes.

Ved å omskrive Y_t og sette den inn (3.6-5) får vi likningen (3.6-8) som brukes i regresjonen der verdien på parametrene i første steg finnes etter :

$$Y_{t-1} = \ln(P_{t-1}) - f(t-1) = \phi \cdot z_{t-1} - \phi \cdot f(\theta, (t-1)) \quad (\text{Likning 3.6-7})$$

$$z_t = \phi \cdot z_{t-1} + f(\theta, T) - \phi \cdot f(\theta, (T-1)) + \xi_t \quad (\text{Likning 3.6-8})$$

Eller om vi skriver (3.6-8) helt ut får vi denne likningen hvor parametrene bestemmes ved bruk av ikke lineære minste kvadrates metode:

$$\ln(P_{t_a}) = \phi \cdot \ln(P_{t_a-1}) + \alpha + \gamma \cdot \cos\left(\frac{2 \cdot \pi}{52} \cdot (t_a + \tau)\right) - \phi \cdot \left[\alpha + \gamma \cdot \cos\left(\frac{2 \cdot \pi}{52} \cdot (t_a - 1 + \tau)\right) \right] + \xi_t \quad (3.6-9)$$

3.7. Bruk av forwardpriser kontra spotpris i tilpassning

Likningen i [Lucia & Schwartz 2001] tilpasses ved bruk av en spotprisrekke. I vårt tilfelle har vi flere prisrekker som skal brukes i tilpassningen, og utgangspunkt er dermed ikke det samme som [Lucia & Schwartz 2001]. Som et estimat på spotprisen brukes 1-ukes forwardprisen [Bessembinder m.fl. 1996], og alle parametere utenom risikopremien bestemmes ved bruk av dette datasettet.

Risikopremien bestemmes ved å bruke øvrige forwardpriser i datasettet og det vil si for ukene $j=2, \dots, 7, 12, 16, \dots, 52, 72, 88, 104$ og hvor valget av j nesten er det samme som i Koekebakker & Ollmar (2001) som tidligere nevnt. Forskjellen er at vi ikke bruker første uken til å bestemme gjennomsnittlig risikopremie, fordi dette er brukt til å bestemme øvrige parametere i datasettet.

3.8. Estimering av standard avvik og risikopremie

Som tidligere nevnt vil parametrene som finnes i likning (3.6-9) bestemmes ved bruke av ikke-lineær minst kvadraters metode. Konkret innebærer det at parametrene som inngår i $\theta = [\alpha, \gamma, \tau]$ og ϕ vil bli bestemt med denne metoden. Ved at ϕ er bestemt så er også κ gitt gjennom sammenhengen $\kappa=1-\phi$. Øvrige parametere finnes gjennom beregninger som bruker resultater fra disse beregningen.

3.8.1. Estimering av standard avviket

Som et estimat på standard avviket brukes standardfeilen til regresjonen. Denne verdien er definert som standardavviket til residualet ξ_t . Estimert verdi av standard avvik finnes da ved bruke å bruke standard formelen:

$$\hat{\sigma} = \sqrt{\frac{\sum_{n \in N} (\xi_i - \bar{\xi})^2}{N - R}} \quad (\text{Likning 3.8.1-1})$$

I de fleste statistikkprogram oppgis denne verdien som et av parametrene i outputen fra en regresjonen. Verdien er justert for antall frihetsgrader og justeringsleddet R er i vårt tilfelle lik 4, det samme som antall parametere som er bestemt i regresjonen.

3.8.2. Estimering av markedsprisen for risiko

For å kunne estimere risikopremien over alle observasjoner må vi benytte oss av den tilpassede funksjonen (likning 3.6-1). I denne funksjonen er det kun en verdi som er ukjent (λ , markedsprisen for en enhet risiko) og denne skal finnes ved å bruke observasjoner av forwardprisen som ikke er brukt i tilpassningen av øvrige parametere. Definer da følgende likning.

$$P(\lambda) = e^{\left(\alpha + \gamma \cdot \cos\left(\frac{2\pi}{52} \cdot (t_a + \tau)\right) + (\ln(P_0) - f(0))e^{-\kappa \cdot T} - \frac{\lambda \cdot \sigma}{\kappa} (1 - e^{-\kappa \cdot T}) + \frac{\sigma^2}{4\kappa} (1 - e^{-2\kappa \cdot T}) \right)} \quad (3.8.2-1)$$

Verdi av øvrige parametere inngår som konstanter og verdien av λ finnes ved bruk av ikke lineær minste kvadraters metode når det minimeres over datasettet fra følgende uker ukene $j=2, \dots, 7, 12, 16, \dots, 52, 72, 88, 104$.

4. Tilpassning av modeller

4.1. Parametrene i basismodellen

En tilpassning av likning (24) gir følgende parametere når modellen tilpasses.

Parameter	Verdi	Standard feil	T-verdi
α	4.9763	0.1623	30.67
γ	0.1918	0.06520	2.94
τ	-1.2068	2.7665	-0.44
φ	0.9651	0.0148	65.22

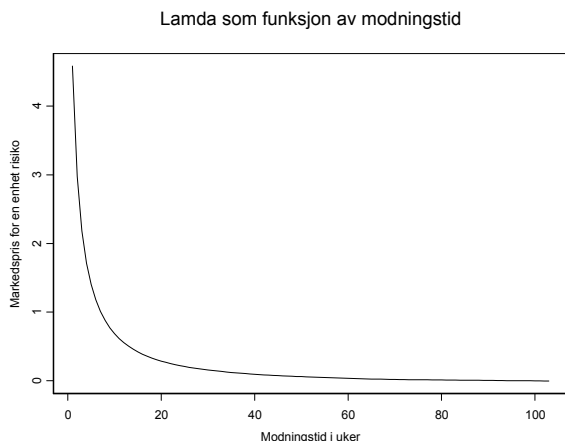
Tabell 4.1-1 Parameter verdier

Sammenlignet med verdiene fra [Lucia & Schwartz 2001], virker disse fornuftig. I tilpassningen som er gjort av Lucia og Schwartz er $[\alpha, \varphi, \tau] = [4.938, 0.986, 0.836]$ i den modellene som det er relevant å sammenligne med. Ser også at alle parametrene utenom τ er signifikante, med relativt høye t-verdier. Antar ut fra dette at tilpassningen er korrekt gjennomført og beregner øvrige verdier som inngår i forwardpris modellen. Standard avviket som inngår i forwardprismodellen beregnes som beskrevet i (3.8.1-1) og er $\sigma = 0.1008 \approx 0.101$. Verdien er i samme størrelses orden som den funnet i [Lucia og Schwartz 2001] hvor $\sigma \approx 0.086$

Markedsprisen for risiko(λ) finnes ved bruk av likning (3.8.2-1) og er et gjennomsnitt over alle modningstidspunkt. Verdien av $\lambda = -0.0207$ og denne er signifikant med en t-verdi lik -20. I motsetning til Lucia & Schwartz (2001) så er verdien her negativ, og dermed er forwardprisen høyere en forventet spotpris. Årsaken er at med $\lambda < 0$, blir risikopremien ($\alpha^* = -\lambda\sigma/\kappa$) positiv. Med en positiv α^* vil forwardprisen ligge over forventet spotpris, og dette er ikke konsistent med markedet.

4.1.1. Alternativ beregning markedspris for risiko

Fordi en negativ verdi på λ er høyst uventet ble det gjort ytterligere analyser av markedsprisen for risiko. I stedetfor å beregne en gjennomsnittlig λ , ble λ verdien for alle modningstidspunkt fra 2 –104 uker ble funnet ved hjelp av (3.8.2-1). Verdiene som ble funnet i denne analysen er framstilt i Figur 4.1-1.

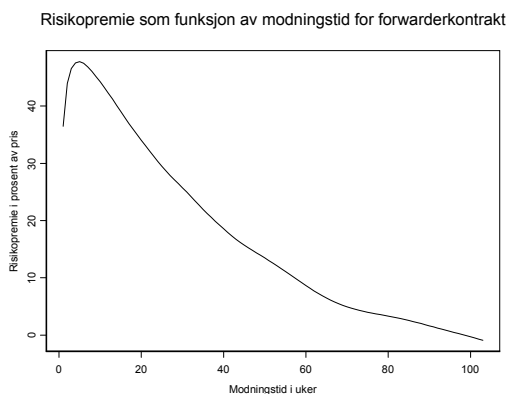


Figur 4.1-1: Markedsprisen for risiko som funksjon av modningstid

Resultatene fra den siste analysen er konsistent med hva som observeres i markedet og viser dermed at bruken av en gjennomsnittlig markedspris for risiko ikke gir noen god modell for forwardprisen. Gitt disse lamdaverdien kan risikopremien regnes ut som en funksjon av modningstiden.

Risikopremien i prosent er gitt ved denne likningen:

$$\text{risikopremie}(T) = \left(1 - e^{-\frac{\lambda\sigma}{\kappa}(1-e^{-\kappa T})}\right) * 100\% \quad (\text{Likning 4.1.1-1})$$



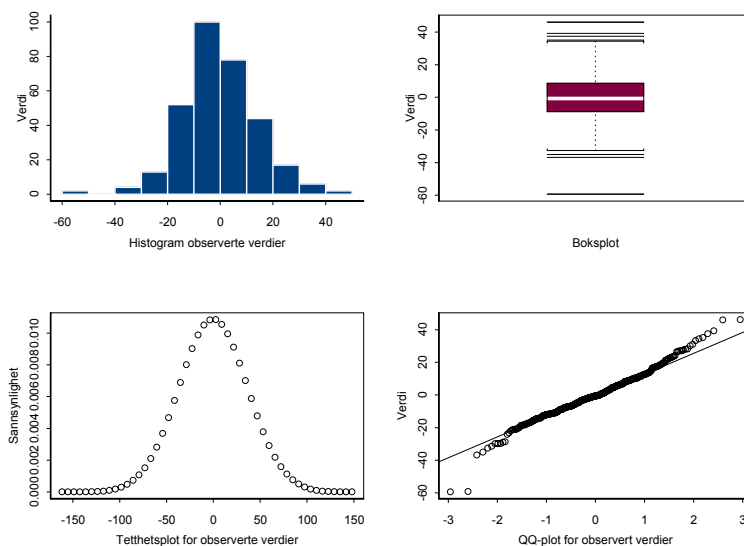
Figur 4.1-2: Risikopremie som funksjon av modningstid

Disse verdiene er noe i overkant med hva som kan observeres i markedet, og skyldes trolig dårlig metodikk for å bestemme lamda. Metoden som brukes her gir rom for å velge en større lamda for å minke det kvadratiske avviket som skal minimeres i algoritmen. Algoritmen korrigerer derfor for dårlig tilpassning av basismodellen ved å velge en høyere lamdaverdi under optimeringen.

4.2. Hvor bra tilpasset er basis modellen

4.2.1. Antagelsen om normalitet

Kravet til normalfordelte residualer ved tilpassningen av (30) (dvs. om $\xi_t \sim N(0, \sigma^2)$) er basisen som modellen bygger på.



Figur 4.2-1: Diverse plot ved sjekk av normalitetsantagelsen

Antagelsen om at residualene er normalfordelt blir ikke styrket ved en studie av plotene ovenfor. Avviket som vi observerer i qq-plottet med at punktene i enden av datasettet ikke ligger på den rette linjen tyder på at antagelsen om normalitet er svak. Et slikt avvik antyder at vi har en tykkhalet fordeling og kan være en indikasjon på at residualene ikke er normalfordelt.

Om så er tilfelle, er bruken av verdier fra modellen høyst tvilsom, og derfor gjøres det en hypotese test om dette holder med en metode fra Johnson & Wichern (1998), s 193. Test statistikk er r_Q og denne er gitt ved følgende likning:

$$r_Q = \frac{\sum_{j=1}^n (x_{(j)} - \bar{x}) \cdot (q_{(j)} - \bar{q})}{\sqrt{\sum_{j=1}^n (x_{(j)} - \bar{x})^2 \cdot \sum_{j=1}^n (q_{(j)} - \bar{q})^2}} \quad (36)$$

Kritisk verdi finnes i tabell 4-2 [Johnson & Wichern (1998)]:

Hvor:

$x_{(j)}$: De ordnede observasjoner av residualene.

$q_{(j)}$: De ordnede observasjoner av standard normal kvantilene.

n : Antall observasjoner i datasettet.

$$\left. \begin{array}{l} n = 318 \\ \alpha = 0.01 \end{array} \right\} \Rightarrow r_{Q(318,0.01)} = 0.9935 \text{ Kritisk verdi.}$$

Setter da opp følgende hypotese:

H_0 : Dataene er normalfordelt.

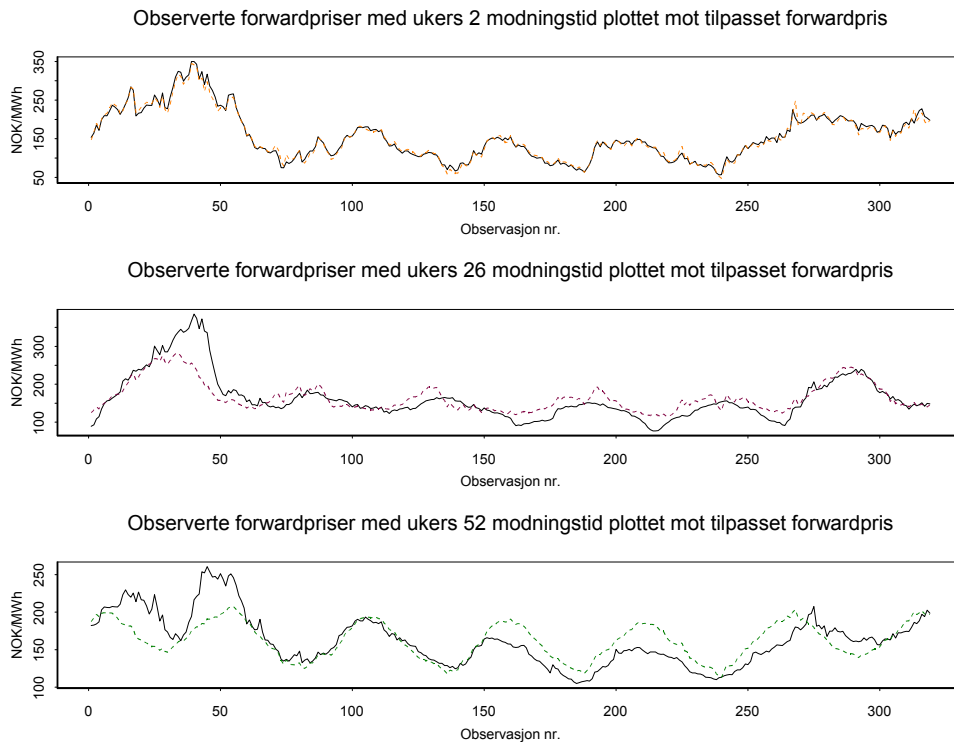
H_1 : Dataene er ikke normalfordelt.

Forkast H_0 når : $r_Q < r_{Q(318,0.01)}$

Beregnet $r_Q = 1.00$ og vi beholder dermed H_0 og forutsetter at residualene er normalfordelt, og implisitt av at prismodellen er lognormal.

4.2.2. Tilpassning av full modell

Ved beregningen av markedsprisen for risiko ble det brukt to framgangsmåter. I den første metoden ble det beregnet en gjennomsnittlig verdi over alle modningstidspunktene . For det andre tilfellet ble det beregnet en markedspris for risiko for hvert eneste modningstidspunkt. Verdien som ble funnet i det første tilfellet gav en negativ markedspris for risiko. En negativ markedspris for risiko innebærer at forwardprisen forventes å være høyere en spotpris ved innløsning, noe som ikke samsvarer hva som er observert i markedet.



Figur 4.2-2: Observert forwardpriser plottet mot predikerte

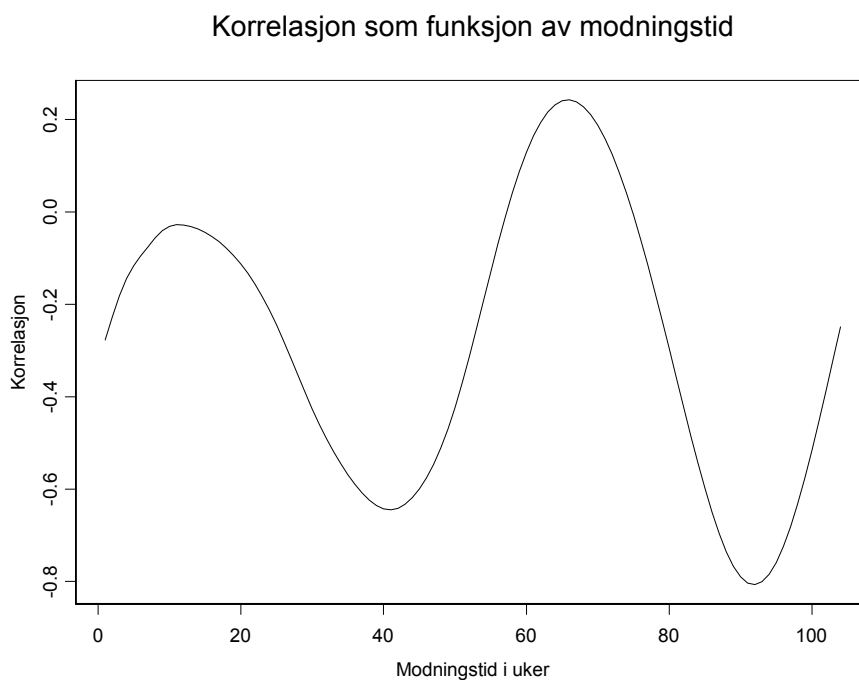
Som det første øverste av de tre plottet viser er tilpassningen meget god når modningstiden er kort. Med økende modningstid ser vi at tilpassningen blir dårligere som ventet, og en del av årsaken kan kanskje tilskrives manglende hensyn til tilsig-/magasinnivå. Særlig sees dette godt i plottet hvor modningstiden er 52 uker, og hvor observasjonene 0-70 er en tørr periode og 150-250 er en våt periode. Effekten av den tørre perioden er en underestimering av forwardprisen, og følgelig en overestimering i den fuktige perioden.

I en modell hvor korrelasjon mellom pris og tilsig taes hensyn til, vil tilpassningen kanskje kunne bli bedre enn her. Fordi usikkerheten i estimer av framtidig nedbør og etterspørsel med mer er økende med tiden, gir dette en øvre grense på hvor godt en slik modell kan tilpasses. Om en bruker et av de vanligste målene for tilpassing, det vil si R^2 , finner vi at denne er lik 0.977 for denne modellen, og denne er regnet ut over alle modningstiden dvs fra 1- 104 uker. Den høye verdien som er funnet på R^2 indikerer at det å tilpasse en modell som tar hensyn til tilsig/magasinnivå med basis i denne modellen kan bli vanskelig.

4.3. Beregning av korrelasjon mellom forwardpris og magasinnivå

4.3.1. *Korrelasjon mellom forwardpriser og tilsig*

Før en starter å velge en modell som tar hensyn til korrelasjon mellom tilsiget og forwardprisene vil det være naturlig å studere denne korrelasjonen. I figuren nedenfor er korrelasjon mellom pris og tilsigs plottet. Med tilsig menes fyllingsgraden i alle norske magasinene⁶.

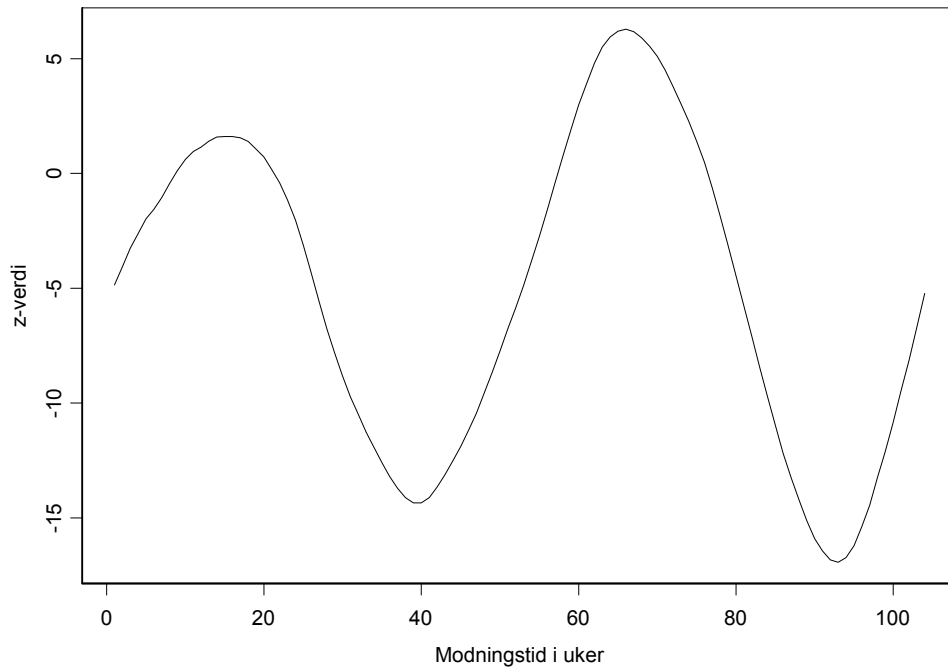


Figur 4.3-1 Korrelasjon forwardpris og tilsig

Korrelasjon mellom forwardprisen og tilsiget følger et mønster som viset i Figur 4.3-1. Når en studerer det komplette datasettet finnes høyest negativ korrelasjon etter 42 uker og neste nåes etter 93 uker. Laveste korrelasjon opptrer etter 12 og 75 uker. Hvorvidt korrelasjon er signifikant sjekkes på samme måte som i kapittel 1.5.1.

⁶ Disse magasinene utgjøre 97,1 % av det samlede volumet i Norge

Teststatistikk z som funksjon av korrelasjon mellom
avvik i magasinnivå og forwardpris



Figur 4.3-2: z-verdier i testen av om korrelasjon er signifikant

I Figur 4.3-2 så ser vi at signifikansnivået på korrelasjon mellom avvik i magasinnivå og forwardpriser svinger voldsomt. Z-verdien ligger i store perioder over eller under nivået der korrelasjon er signifikant. Det er derfor grunn til å tro at avviket fra normal nivået for magasinnivået har en forklaringseffekt på forwardprisene. En slik korrelasjon samsvarer med hva som påvist av Gjolberg & Johnsen (2002).

4.3.2. Modeller som er prøvd tilpasset

En rekke modeller er forsøkt utprøvd for å ta hensyn til korrelasjonen mellom tilsig og forwardprisen. Felles for alle modellene er at tilsiget/magasinnivå er prøvd tatt med som en faktor i den deterministiske delen av modellen. Modellene som er prøvd tilpasset er derfor bare forskjellige variasjoner av den deterministiske funksjonen $f(T)$. Nedenfor er noen av funksjonen som er prøvd tilpasset:

MT : Fyllingsgrad norske kraftverksmagasin.

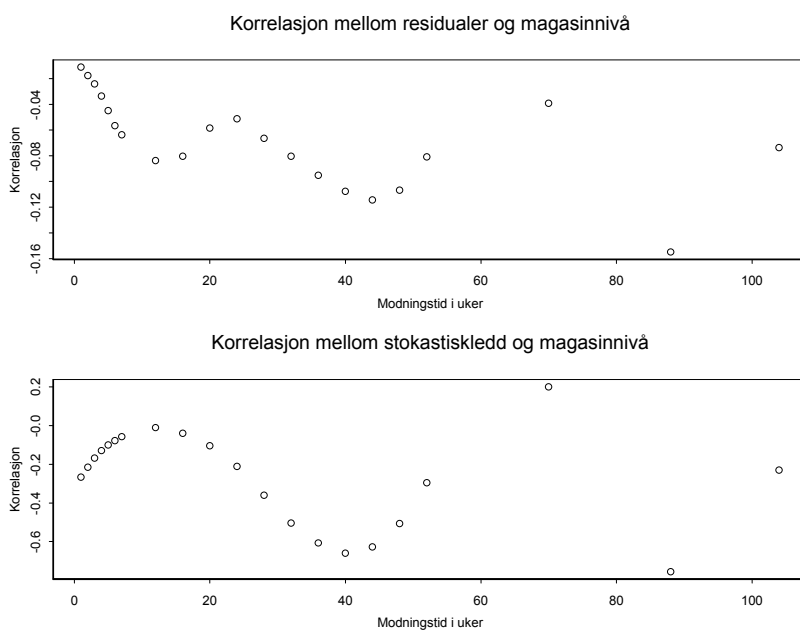
$$1) : f(T) = \alpha \cdot e^{\delta(1) \cdot MT \cdot T} + e^{\delta(1) \cdot MT \cdot T} \cdot \gamma \cdot \cos\left(\frac{2 \cdot \pi}{52}(t_a + \tau)\right)$$

$$2) : f(T) = \alpha \cdot \delta(1) \cdot MT \cdot T + \delta(2) \cdot MT \cdot T \cdot \gamma \cdot \cos\left(\frac{2 \cdot \pi}{52}(t_a + \tau)\right)$$

I tillegg til de to som er vist her, er ca. 20 andre mer eller mindre like modeller forsøkt tilpasset. Ingen av disse hadde en bedre tilpassning enn basismodellen. I de fleste tilfeller har tilpassningen heller blitt dårligere ved at summen av de kvadrerte residualene har økt.

4.3.3. Hvorfor kan ikke korrelasjon mellom tilsig og pris modelleres her?

Årsaken til at sammenhengen mellom tilsig og pris ikke fås med i mine modeller kan skyldes tilpassningsmetoden som brukes. En sjekk av plottene nedenfor forklarer kanskje noe av årsaken.



Figur 4.3-3 Korrelasjonsplott

I det øverst plottet på forrige side ser vi at korrelasjon mellom magasinnivå og residualer fra basismodellen til Lucia & Schwartz (2001) er veldig lav og dette leder til at tilsiget/magasinnivået har liten forklaringsgrad når det forsøkes lagt til i basismodellen.

Samtidig ser vi at korrelasjon mellom det stokastiske leddet i basismodellen til Lucia & Schwartz (2001) og magasinnivået er tydelig, i en korrekt modell ville punktene vært tilfeldig fordelt omkring 0 med lave verdier. Dette tyder på at min modellformulering eller metodikk for løsning er feil. Å bestemme parametrene sekvensielt slik som det er prøvd her er kanskje en metode feil. Alternativt skulle kanskje en mindre del av datasettet vært brukt til å løse basis modellen og resten for å tilpasse leddene som tar hensyn til avvik fra normalnivået i tilsiget.

Modellen er derfor åpenbart ikke helt korrekt, men ifølge [Lucia & Schwartz] gir enfaktormodeller en god nok forklaring på forwardprisen. Velger derfor å bruke noen av parametrene fra basismodellen som input til en prismodell som ikke er korrelert med tilsiget, men som kan markedsjusteres og brukes i optimeringen.

4.4. Hvordan kan tilsiget inkluderes i andre modeller?

4.4.1. Ved bruk av en tofaktor modell?

Et forslag er å tilpasses en tofaktor modell av typen som er [Lucia & Schwartz 2001] her foreslått. Den risikojusterte modellen kan da beskrives slik:

$$P_t = f(t) + X_t + \varepsilon_t \quad (\text{Likning 4.4-1})$$

$$dX_t = -\kappa \cdot X_t \cdot dt + \sigma_X \cdot dZ_X \quad (\text{Likning 4.4-2})$$

$$d\varepsilon_t = \mu \cdot dt + \sigma_\varepsilon \cdot dZ_\varepsilon \quad (\text{Likning 4.4-3})$$

$$dZ_X \cdot dZ_\varepsilon = \rho \cdot dt \quad (\text{Likning 4.4-4})$$

Hvor X_t har de samme egenskapene som i enfaktor modellen. Den andre tilstandsvariabelen følger Geometrisk Brownsk bevegelse og de to Winer-prosessene dZ_X^* og dZ_ε^* er korrelerte gjennom likning (4.4-4).

Den risikojusterte prosessen for disse tilstandsvariablene er :

$$dX_t = -\kappa \cdot (\alpha^* - X_t) \cdot dt + \sigma_X \cdot dZ_X^* \quad (\text{Likning 4.4-5})$$

$$d\varepsilon_t = \mu_\varepsilon^* \cdot dt + \sigma_\varepsilon \cdot dZ_\varepsilon^* \quad (\text{Likning 4.4-6})$$

med

$$\alpha^* \equiv \lambda_X \cdot \frac{\sigma_X}{\kappa} \quad (\text{Likning 4.4-7})$$

$$\mu_\varepsilon^* \equiv \mu_\varepsilon - \lambda_\varepsilon \cdot \sigma_\varepsilon \quad (\text{Likning 4.4-8})$$

hvor λ_X og λ_ε er markedsprisen for risiko forbundet med enhet av hver tilstandsvariabel, verdien antas å være konstant.

$$dZ_X^* \cdot dZ_\varepsilon^* = \rho \cdot dt \quad (\text{Likning 4.4-9})$$

I [Lucia & Schwartz 2001] forutsettes det at ε_t er en faktor som priser langsiktig risiko og dermed at X_t er faktor som priser kortsiktig risiko. Korrelasjon som brukes i likning (4.4-9) representerer korrelasjon mellom kort og langsiktig risiko. I dette tilfellet er det mellom spot og forwardpriser.

I en modell som skal ta hensyn til korrelasjon mellom tilsig og forwardpriser vil framgangsmåten være noe annerledes. I en slik setting vil X_t representere prisen for den generelle risikoen i markedet og ε_t vil da prisen knyttet til tilsiget. Verdien av λ_X vil da være prisen for en enhets endring i X_t og tilsvarende λ_ε for en enhets endring i ε_t (tilsig/magasinnivå).

Korrelasjonen mellom forwardpriser og tilsig vil inngå i likning (4.4-9). Figur 4.3-1 viser at denne korrelasjonen ikke er konstant men er funksjon av modningstiden. For å kunne prisen for forwarder best mulig bør derfor modellen tilpasses for modningstider forskjellige intervaller av modningstiden. Som et konkret eksempel vil dette medføre at du får en likning for å prisen for forwarder med modningstid mindre enn 52 uker, og en for lengre modningstider.

4.4.2. Valg av pris modell

Å tilpasse en slik en tofaktormodell er det ikke er tid å gjøre innfor gjenværende tidsrom av hovedoppgaven. For å finne ut hva som er mest lønnsomt av å bruke av forward eller forventet spotpriser i driftsplanlegging, må det gjennomføres en optimering. Optimeringen kan ikke gjøres uten at det er tilpasset en diskontinuerlig prisprosess og i denne forbindelsen passer det meget bra å bruke log-modellen til Lucia & Schwartz (2001). Hvorfor denne passer bedre enn visse andre modellen vil bli forklart i avsnittet om implementering av den diskrete prisprosessen.

5. Generelt om trinomiske trær

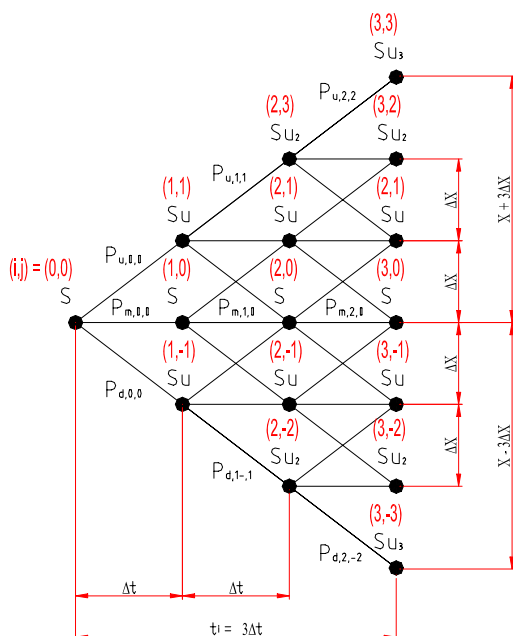
5.1. Hvorfor bruke dem her

For å kunne gjøre en optimering når det er usikkerhet både i pris og tilsig må strukturen på disse dataene tilpasses valget av optimeringsmetoden. I dette tilfellet betyr det at det må regnes ut tilstandspriser og tilhørende overgangssannsynligheter for prisene som skal brukes i optimeringen. Dette gjøres med utgangspunkt i den kontinuerlige enfaktormodellen som er tilpasset, det vil si at parametrene fra denne modellen brukes som inndata i modellen.

Fordi datagrunnlaget for tilpassningen av enfaktormodellen kommer fra en approksimering (ref. Kap. 1.4), er det viktig å ikke ta inn flere unøyaktigheter i beregningene. Ifølge Haug (1997) gir trinomiske tre nøyaktigst tilnærming i når det skal lages diskrete prisutfall av en kontinuerlig prisprosess. Derfor velges det å bruke et trinomisk tre framfor det mer vanlige binomiske treet.

5.2. Oppbyggingen av det trinomisk treet

I ett trinomisk tre forgrener hver node seg på til tre nye noder, og metoden ble introdusert av Boyle (1986). Oppbygging av treet er som vist i figuren nedenfor:



Figur 5.2-1: Trinomisk tre

Hver node i treet indekseres med heltallene i og j , hvor i indekserer den horisontale posisjonen og j den vertikale posisjonen. Horisontal posisjon indikeres med positive heltall $i = 1, \dots, I$. Vertikal plassering indikeres med både positive og negative heltall $j = -J, \dots, 0, \dots, J$, og maksimalt antall steg opp eller ned er alltid like, treet er derfor symmetrisk. Enhver node er derfor unikt identifiserbar ved heltallssettet (i, j) og brukes derfor i utviklingen av trinomiske tre.

I treet er det tre typer overgangssannsynligheter og disse betegnes $p_{u,i,j}$, $p_{m,i,j}$ og $p_{d,i,j}$. Felles for disse er at indeksen (i, j) angir ettstegs overgangssannsynligheten fra denne noden til mulige noder, og disse er $(i+1, j+1)$, $(i+1, j+0)$ og $(i+1, j-1)$. Betydningen av den siste indeksen er dermed slik, u gjelder for bevegelsen opp i treet fra node $(i, j) \rightarrow (i+1, j+1)$, m gjelder for bevegelsen rett fram i treet fra node $(i, j) \rightarrow (i+1, j+0)$ og d gjelder for bevegelsen ned i treet fra node $(i, j) \rightarrow (i+1, j-1)$.

I treet er bevegelser i horisontal retning økning i tid, og tidsinkrementet Δt er like stort i alle steg fra $i = 0, \dots, I$. Bevegelser i vertikal retning er endringer i pris og inkrementet ΔX er like stort for alle steg $j = -J, \dots, -1, 0, 1, \dots, J$. I et tre som spenner over et tidsrommet T og hvor antall tidssteg er lik I , er $\Delta t = T/I$. Tiden som er gått i enhver node i kan da uttrykkes slik $t_i = i \cdot \Delta t$.

5.3. Beregning overgangssannsynligheter

Overgangssannsynlighetene i et trinomisk tre velges slik at forventningsverdien, varians og total sannsynlighet er korrekt. Generelt gjelder at for en modell som er beskrevet i (5.3-1) hvor $d\zeta$ følger geometriske brownske bevegelser, kan overgangssannsynlighetene finnes via de velkjente statistiske likningene (5.3-1)-(5.3-4).

$$dx = \mu \cdot dt + \sigma \cdot d\zeta \quad (\text{Likning 5.3-1})$$

$$E[\Delta x] = p_u \cdot (\Delta x) + p_m \cdot 0 + p_d \cdot (-\Delta x) = \mu \cdot \Delta t \quad (\text{Likning 5.3-2})$$

$$E[(\Delta x)^2] = p_u \cdot (\Delta x)^2 + p_m \cdot 0 + p_d \cdot (-\Delta x)^2 = \sigma^2 \cdot \Delta t + \mu^2 \cdot \Delta t \quad (\text{Likning 5.3-3})$$

$$p_u + p_m + p_d = 1 \quad (\text{Likning 5.3-4})$$

$$\text{Hvor verdien av } E[(\Delta x)^2] = \text{Var}(\Delta x) + (E[\Delta x])^2$$

Ved å løse likningene (5.3-1)-(5.3-4) med hensyn på p_u , p_m og p_d , finnes følgende likninger for overgangssannsynlighetene:

$$\hat{p}_u = \frac{1}{2} \cdot \left(\frac{\sigma^2 \cdot \Delta t + \mu^2 \cdot \Delta t^2}{\Delta X^2} + \frac{\mu \cdot \Delta t}{\Delta X} \right) \quad (\text{Likning 5.3-5})$$

$$\hat{p}_m = 1 - \frac{\sigma^2 \cdot \Delta t + \mu^2 \cdot \Delta t^2}{\Delta X^2} \quad (\text{Likning 5.3-6})$$

$$\hat{p}_d = \frac{1}{2} \cdot \left(\frac{\sigma^2 \cdot \Delta t + \mu^2 \cdot \Delta t^2}{\Delta X^2} - \frac{\mu \cdot \Delta t}{\Delta X} \right) \quad (\text{Likning 5.3-7})$$

5.4. Forskjellige modeller for trinomiske treer

Likningen som er brukt her for å beregne sannsynlighetene (5.3-5)-(5.3-7) i treet tar utgangspunkt i at bevegelsen til ΔX følger likningen (5.3-2). For mange prisprosesser kan dette være en korrekt beskrivelse av ΔX bevegelse, men ikke for alle. Det forutsettes i enfaktormodellen til Lucia og Schwartz (2001) at ΔX følger denne stokastisk prosess beskrevet i (5.3-2). Samme metode for å bygge det trinomiske treet kan derfor velges for å lage diskrete priser.

Det beskrives i Hull (1993) flere metoder som kan benyttes alt ettersom hvilken stokastisk prosess variabelen i modellen antas å følge. Modellen vi har valgt er identisk med Hull og White rentemodell [Hull (1993) Kap. 21.9], og prosedyren for å lage det trinomiske treet til denne modellen brukes derfor her [Hull (1993) Kap. 21.12].

Forskjellen mellom modellene i Hull (1993) og det vanlige trinomiske treet beskrevet her går blant annet på hvordan overgangssannsynligheter beregnes. Størrelsen på treet begrenses også i vertikal retning, dette innebærer at verdiene av $-J$ og J settes lik enn eller annen maksimal verdi. Årsaken til disse endringene vil beskrives grundig i kapitlet om prosedyren for trebyggingen.

6. Trinomisk tre for Hull og White modellen

6.1. Betingelse for Hull og White modellen

Forutsetningen for å kunne bruk det trinomiske treet i Hull & White modellen er at vår prismodell følger den samme prosess variabelen i deres modell. Likningen som beskriver hvilken prosess deres variabel(renten) følger er denne [Hull (1993), likning (21.17)]:

$$\left. \begin{aligned} dr &= \underbrace{(\theta(t) - a \cdot r) \cdot dt + \sigma \cdot dz}_{\Downarrow} \\ dr &= a \cdot \left(\frac{\theta(t)}{a} - r \right) \cdot dt + \sigma \cdot dz \end{aligned} \right\} \quad \text{(Likning 6.1-1)}$$

Prosessen som beskriver vår prisbevegelsen dP_t er gitt ved likning (3.3-4) utledet i kapittel 3.3. Likningen er gjengitt nedenfor:

$$dP_t = \kappa \cdot (a(t) - P_t) \cdot dt + \sigma \cdot dZ \quad \text{(Likning 3.3-4)}$$

Videre forutsettes det at a og σ er konstanter i Hull & White modellen, hvilket de også er i modellen til Lucia & Schwartz (2001). Med parameteren a som en konstant kan $\theta(t)$ omformuleres slik:

$$\theta'(t) = \frac{\theta(t)}{a} \quad \text{(Likning 6.1-2)}$$

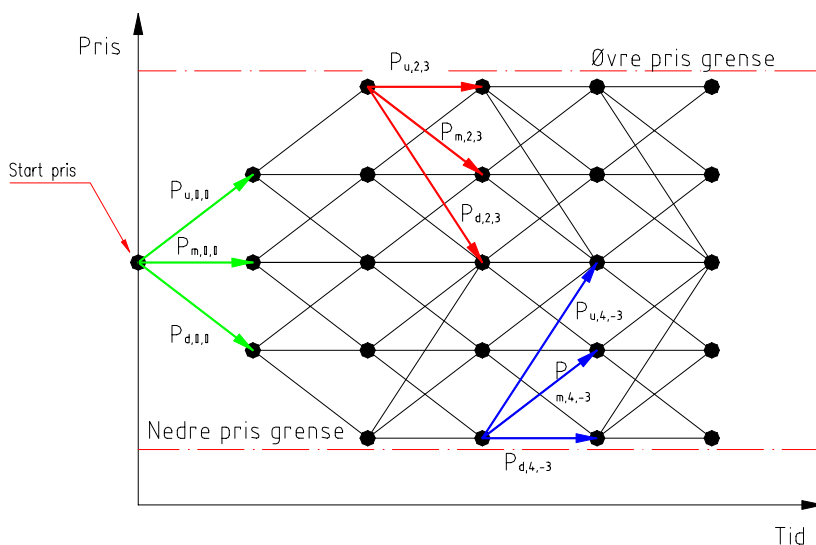
Som leder til at likning (43) kan uttrykkes på denne måten:

$$dr = a \cdot (\theta'(t) - r) \cdot dt + \sigma \cdot dz \quad \text{(Likning 6.1-2)}$$

Dermed ser vi at likning (3.3-4) er identiske med (6.1-2) og at metoden for å lage et trinomiske tre til Hull & White modellen kan benyttes for enfaktormodellen fra Lucia & Schwartz (2001). Metoden kan også benyttes for log spotprismodellen ved å anta at $r = \ln(S_t)$. Denne tilnærmingen gjør også at korreksjonen mot denne sanne terminstrukturen blir enklere, ved at vi kan benytte metodikken fra Clewlow og Strickland (April 1999). Hvordan dette gjøres vil bli grundig forklart i kapittel 6.4, som omhandler steg 2 byggingen av trinomiske tre.

6.2. Treets form

I et ordinært trinomisk tre vil indeksen j øke med en for hver tidssteg som taes. Fordi indeksen j inngår i formlene for overgangssannsynlighet kan dette lede til negativ på disse når j inntar en høy negativ verdi. Hull & White modellen løser problemene med å begrense størrelsen på indeksen j , utseende til treet blir da som i Figur 6.2-1.



Figur 6.2-1: Trinomisk tre i Hull & White modellen

Som Figur 6.2-1 viser, vil forgreningen av treet ikke skje slik den gjør i et ordinært tre der utstrekningen i prisretningen forsetter ettersom tiden går. Her begrenses utstrekningen i prisretningen av en øvre og nedre grense på prisen. Forgreningen av treet foregår derfor på tre måter som de fargede pilene indikerer. For alle noder som ikke ligger langs øvre eller nedre pris grense skjer forgreningen som de grønne pilene viser. Nodene som ligger øverst i treet forgrener seg i henhold til de røde pilene og i de nedre noder forgrenes det i henhold til de blå pilene.

Begrensningen av maksimalt antall utfall som ligger i modellen gjør derfor treet svært egnet for bruk i optimeringen, samtidig som det gir godt framstilling av det diskrete utfallet til prisen.

6.3. Steg 1 i trebyggingen

I følge Hull (1993), er første steg i trebyggingen å konstruere ett tre for X_t som har startverdi lik null og følger en AR(1) prosess hvor $X_t \sim N(0, \sigma^2)$. Dette tilsvarer det stokastiske leddet i prisligningen (3.3-1) og likningen for prosessen er gitt likning (3.3-2).

$$dX_t = -\kappa \cdot X_t \cdot dt + \sigma \cdot dZ \quad (\text{Likning 3.3-2})$$

Videre defineres ΔX som avstand mellom prisene i treet og settes lik:

$$\Delta X_t = \sigma \cdot \sqrt{3 \cdot \Delta t} \quad (\text{Likning 6.3-1})$$

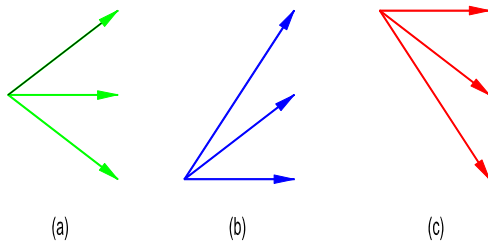
som er et godt valg med hensyn på feil minimering i følge Hull (1993).

Videre defineres node (i,j) slik at :

$$t_i = i \cdot \Delta t \quad (\text{Likning 6.3-2})$$

$$X_j = j \cdot \Delta X \quad (\text{Likning 6.3-3})$$

Som Figur 6.2-1 viser, er det tre alternativer for å regne ut overgangssannsynligheter og forgreningen i disse skjer etter mønsteret vist i figurene under.



Figur 6.3-1: Forgreningens mønster fra node i Hull & White modellen

Å forgrene etter mønsteret vist i Figur 6.3-1 (a) er i de fleste tilfeller god nok. Men når $\kappa > 0$ er det nødvendig å skifte fra Figur 6.3-1 (a) til Figur 6.3-1 (c) for tilstrekkelig store j . Tilsvarende er det nødvendig å skifte fra Figur 6.3-1 (a) til Figur 6.3-1 (b) når j har tilstrekkelig lav nok negativ verdi.

Tidspunktet når det skal skiftes fra en forgrening etter Figur 6.3-1 (a) til forgrening etter metode Figur 6.3-1 (c) bestemmes av verdien j_{max} . Hull & White (1990) viser at sannsynlighetene alltid er positive om verdien av j_{max} defineres slik:

$$j_{max} = \frac{0.184}{\kappa \cdot \Delta t} \quad (\text{Likning 6.3-4})$$

Tilsvarende for skifte fra forgrening etter Figur 6.3-1 (a) til forgrening etter Figur 6.3-1 (b) skjer når verdien j_{min} nåes, hvor j_{min} settes lik $-j_{max}$.

6.3.1. Overgangssannsynligheter i modellen

Gitt forgreningsmønstrene i Figur 6.3-1 så er det tre sett av likninger for overgangssannsynlighet som ettersom hvilket mønster forgreningen skjer etter. Med forgreningen etter Figur 6.3-1 (a) gjelder følgende likninger:

$$\left. \begin{aligned} \hat{p}_u &= \frac{1}{6} + \frac{\kappa^2 \cdot j^2 \cdot \Delta t^2 - \kappa \cdot j \cdot \Delta t}{2} \\ \hat{p}_m &= \frac{2}{3} - \kappa^2 \cdot j^2 \cdot \Delta t^2 \\ \hat{p}_d &= \frac{1}{6} + \frac{\kappa^2 \cdot j^2 \cdot \Delta t^2 + \kappa \cdot j \cdot \Delta t}{2} \end{aligned} \right\} \kappa \quad (\text{Likning 6.3-5})$$

Tilsvarende om forgrening skjer etter mønsteret i Figur 6.3-1 (b) vil disse likningene gjelde for overgangssannsynlighetene gjelde.

$$\left. \begin{aligned} \hat{p}_u &= \frac{1}{6} + \frac{\kappa^2 \cdot j^2 \cdot \Delta t^2 + \kappa \cdot j \cdot \Delta t}{2} \\ \hat{p}_m &= \frac{1}{3} - \kappa^2 \cdot j^2 \cdot \Delta t^2 - 2 \cdot \kappa \cdot j \cdot \Delta t \\ \hat{p}_d &= \frac{7}{6} + \frac{\kappa^2 \cdot j^2 \cdot \Delta t^2 + 3 \cdot \kappa \cdot j \cdot \Delta t}{2} \end{aligned} \right\} (\text{Likning 6.3-5})$$

Sist, om forgreningen skjer et mønster i Figur 6.3-1 (c) vil settet av likninger som gir overgangssannsynligheter være dette.

$$\left. \begin{aligned} \hat{p}_u &= \frac{7}{6} + \frac{\kappa^2 \cdot j^2 \cdot \Delta t^2 - 3 \cdot \kappa \cdot j \cdot \Delta t}{2} \\ \hat{p}_m &= \frac{1}{3} - \kappa^2 \cdot j^2 \cdot \Delta t^2 + 2 \cdot \kappa \cdot j \cdot \Delta t \\ \hat{p}_d &= \frac{1}{6} + \frac{\kappa^2 \cdot j^2 \cdot \Delta t^2 - \kappa \cdot j \cdot \Delta t}{2} \end{aligned} \right\} (\text{Likning 6.3-6})$$

Merk at sannsynligheten kun er en funksjon av j når κ og Δt er konstante. Dette innebærer at nodene med samme verdi på j vil ha samme overgangssannsynlighet. Sannsynlighetene i treet er derfor symmetrisk om noder med $j = 0$.

6.4. Steg 2 i trebyggingen

Prisen som er funnet i steg 1 i byggingen av det trinomiske treet er de stokastiske verdiene i prisligning (1). Verdiene i treet må derfor korrigeres slik at de er konsistente med den opprinnelige terminstrukturen som finnes i markedet. Korreksjonen skjer ved at en regner ut et korreksjonsledd α_m som blir lagt til den stokastiske verdien i nodene.

6.4.1. Beregning av tilstandspriser

For å gjøre det, må det brukes tilstandspriser. Tilstandsprisen $Q_{i,j}$ sier noe om verdien av et derivat i tidspunkt $t = 0$ som gir 1 enhet i avkastning hvis node (i,j) inntreffer og 0 ellers. En annen måte å betrakte denne beregningen på er å si at vi vet at opsjonsprisen er lik 1 og deretter prøve å regne oss tilbake til verdiene i nodene.

Som tidligere nevnt er renten variabelen i Hull & White modellen, og utledningen gjort i steg 2 i treet kan derfor ikke overføres direkte til modellen fra Lucia & Schwartz (2001). Metoden som brukes her er fra Clewlow & Strickland (1999), og denne er lagt for å juster priser opp mot den eksisterende terminstrukturen.

For å kunne benytte denne metoden, defineres $C(0)$ som er prisen i dag på en europeisk opsjon med følgende payoff funksjon $C(S)$ i tidssteg gitt ved:

$$C(0) = \sum_j Q_{i,j} \cdot C(S_{i,j}) \quad (\text{Likning 6.4.1-1})$$

Hvor det summeres over alle noder j i tidspunkt i . Tilstandsprisen finnes ved foroverrettet induksjon og er gitt ved:

$$Q_{(i+1),j} = \sum_{j'} Q_{i,j'} \cdot p_{j',j} \cdot P(i \cdot \Delta t, (i+1) \cdot \Delta t) \quad (\text{Likning 6.4.1-2})$$

Hvor $p_{j',j}$ er sannsynligheten for å gå fra node $(i+1,j') \rightarrow (i+1,j)$ og $P(i \cdot \Delta t, (i+1) \cdot \Delta t)$ står for prisen ved tidspunkt $i \Delta t$ av en rene ned diskontert obligasjon ved tidspunkt $(i+1) \Delta t$. Summasjonen gjøres over alle noder j som i tidssteg i grener til node $((i+1),j)$. $P(i \cdot \Delta t, (i+1) \cdot \Delta t)$ angir diskonterings faktor med risikofri rente fra tidspunkt $i \Delta t$ til $(i+1) \Delta t$. Om vi tar hensyn til vi skal ha en kontinuerlig faktor, kan denne skrives slik:

$$P(i \cdot \Delta t, (i+1) \cdot \Delta t) = e^{-\int_t^{T} r(u) du} = e^{r \Delta t} \quad (\text{Likning 6.4.1-3})$$

6.5. Beregning av korreksjonledd

For å justere det opprinnelige treet til å være konsistent med observerte forwardpriser og den sesongvariasjon som finnes i disse, vil verdien hver node bli korrigeret. Den opprinnelige verdien $x'_{m,j}$ får da ett tillegg α_m , og den nye prisen er da $x_{m,j} = x'_{m,j} + \alpha_m$. For å kunne bruke tilstandspriser for å korrigere for den opprinnelige terminstrukturen brukes en spesial versjon av likning (6.4.1-1).

$$P(0, i \cdot \Delta t) \cdot F(0, i \cdot \Delta t) = \sum_j Q_{i,j} \cdot S_{i,j} \quad (\text{Likning 6.4.2-1})$$

Hvor verdien av $F(0, i\Delta t)$ hentes i terminmarked og hvor verdien av diskonteringsfaktoren $P(0, i\Delta t) = P(0, T_i)$ finnes ved å bruke likning (6.4.1-3) for $i \in 1, \dots, I$. Gitt at $\ln(S_t) = x_{m,j} = x'_{m,j} + \alpha_m$, kan korreksjonsfaktoren finnes av likning (6.4.2-2), likningen er utledet i vedlegg 4.

$$\alpha_i = \ln \left[\frac{P(0, i \cdot \Delta t) \cdot F(0, i \cdot \Delta t)}{\sum_j Q_{i,j} \cdot e^{x'_{i,j}}} \right] \quad (\text{Likning 6.4.2-2})$$

Fordi det er forwardpriser det arbeides med, er verdiene $F(0, i\Delta t)$ allerede diskontert fram til tidssteg i , og diskontering skjer kun ett tidssteg. Diskonteringsfaktoren i (6.4.2-2) blir derfor $P(0, i\Delta t) = e^r$ for alle i .

6.6. Tidsavhengig varians

Bruke av likning (6.3-1) forutsetter at variansen er konstant, hvilket den ikke er. Modellens oppbygging gjør at en hvilken som helst variansmodell kan benyttes, og dermed kan den tidsavhengig implementeres i modellen. Det antas at variansen følger denne modellen:

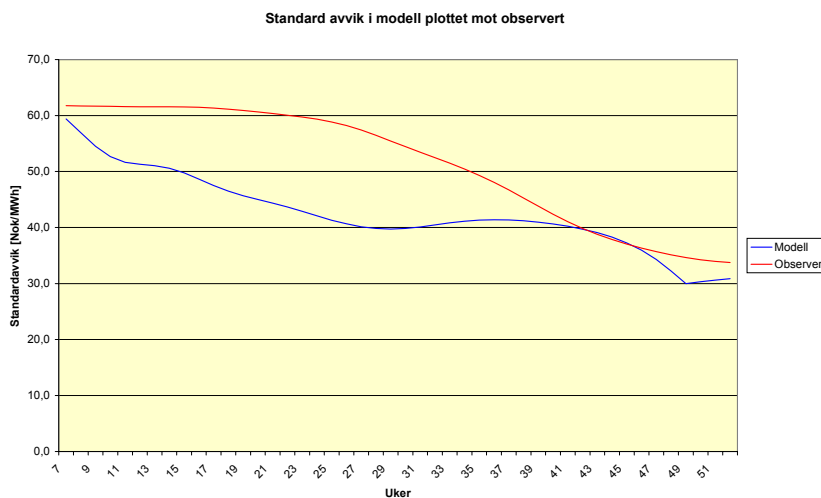
$$\sigma(t) = \begin{cases} \sigma_t = \sigma \cdot (1 - e^{-\kappa \cdot (T_A - t)})^{0.5} & \sigma_t \geq \sigma_A \\ \sigma_A & \sigma_t < \sigma_A \end{cases} \quad (\text{Likning 6.5-1})$$

hvor :

T_A = Tidspunktet standardavvik når den asymptotiske verdien (langtidsvariansen).

σ = Spotvariansen funnet fra basismodell ($\sigma = 0.1$).

σ_A = Asymptotisk standardavvik.



Figur 6.6-1: Observert standardavvik plottet mot modellert

Som Figur 6.6-1 viser, blir det estimerte standardavvik noe lavere enn det faktiske og modellen som er brukt er ikke helt optimal. En annen modell kan lett tilpasses, men fordi denne modellen ble implementert på slutten av prosjektet, er det ikke tid til dette. Dette tyder på at det i denne prismodellen lett kan brukes forskjellige variansmodeller og at disse kan tilpasses uavhengig av øvrige parametere.

6.7. Implementering av trinomisk tre

Det er flere måter å beregne verdien i ett slikt tre, enklest er kanskje å bruke Excel og deretter lage treet for et sett av parametere. I dette tilfellet velges det en annen løsning fordi verdier datamengden som skal trekkes ut fra det trinomiske treet er betydelig og muligheten for å gjøre feil tilsvarende.

Metoden som velges er å skrive et program i Matlab som beregner alle overgangssannsynligheter, tilstandspriser, korreksjonsledd og til slutt prisene basert på ett vilkårlig sett av inputverdier. Jobbene med å programmere et slikt generelt tre er betydelig større enn det å laget et regneark, men bruken av det er mer fleksibel enn av et tre laget for et spesifikt sett av inputverdier. Koden for treet finnes i vedlegg 6 og den finnes også i elektronisk format på CD som er vedlagt rapporten.

7. Generelt om driftsplanleggingsproblemet

7.1. Innledning

For en kraftprodusent innebærer driftsplanlegging at de forventede netto driftsinntektene under visse restriksjoner maksimeres. Oppgaver som inngår i driftsplanleggingen er:

- Prisprognosering
- Å fordele produksjonen av (aktiv) kraft mellom de forskjellige enhetene.

I Norge dekkes ca 99 % av kraftbehovet av vannkraft og planleggingsproblemet blir derfor å bestemme hvor mye skal en produsere i dag, gitt framtidige prisprognoser. Utover dette må også produsenten forholde seg til et usikkert tilsig og planleggingsproblemet til vannkraft produsent har dermed usikkerhet i 2 dimensjoner. Denne stokastikken er en av flere årsaker til at driftsplanlegging innen vannkraft krever forholdsvis kompliserte algoritmer.

7.2. Hva øker kompleksiteten

Om vi sammenlikner vannkraftprodusenten med en termisk kraftprodusent kan en av årsakene til planleggingsproblemet synliggjøres. Den er at alternativkostnaden for vannet ikke er gitt eksogent slik som brenselsprisen er for et termisk kraftverk. Videre vil tappebeslutninger i dag påvirke muligheten til produksjon i fremtiden fordi tilsiget er usikkert. Å vente med produksjon kan betraktes som en realopsjon for produsenten. Denne forskjellen og i tillegg det faktum at både pris og tilsig er seriekorrelert gjør problemet vanskelig.

Å innføre en venteopsjon gjør at vannkraftprodusentens planleggingshorisont nødvendigvis må bli lengre. Særlig gjelder det for kraftverk med flere års magasin, og hvor verdien av å vente er avhengig prisen som betales for framtidig produksjon. Verdien av å vente med produksjon er derfor avhengi av størrelsen på magasinet. I den ene enden av denne skalaen finner vi elvekraftverkene hvor verdien av å vente er null. I et elvekraftverk skjer dette fordi det vannet som ikke brukes renner forbi kraftverket og er dermed tapt for all framtid.

Ved å utvide tidshorizonten må det tas hensyn til at:

- prisen er meget usikker og for å kunne planlegge driften må det foreligge en eller annen prisprognose for planleggingstidsrommet.
- alternativkostnaden for vannet (vannverdi = dualverdien til vannbalansen ref. kapittel 7.3) i et hvert tidspunkt i planleggingsperioden må bestemmes for å verdisette venteopsjonen,
- siden tilsiget ikke er kjent, må det etableres en tilsigsmodell.

De tre punktene ovenfor er knyttet til usikkerheten som gir opsjonsverdien. I tillegg kommer en rekke andre fysiske og reguleringsmessige restriksjoner som påvirker optimeringen i vannkraftprodusentens planleggingsproblem.

De viktigste av disse restriksjonene er:

- minimal og maksimal vannføring
- minimal og maksimal vannstand i magasinet
- at sannsynligheten for flom og tørrkjøring skal minimeres
- at variabel vannhøyde i magasinene medfører at effektproduksjon som funksjon av tappet volum ikke er konvekt. Dette problemet gjelder særlig hvor høydeforskjellen mellom magasin og turbin er liten. Ved et magasin som ligger høyt over turbinene, eksempelvis 800 meter, vil en endring på 20 meter i magasinshøyden utgjøre lite av den totale fallhøyden
- kaskadeeffekt, det vil si at vannet som tappes i ett kraftverk renner ned i magasinet til et annet kraftverk, og som i sin tur renner ned i et nytt magasin og så videre

Alle punktene unntatt variabel vannhøyde, vil inngå som restriksjoner i en optimering. Enkelte av restriksjonene vil i noen optimeringsalgoritmer lede til ikke-lineære restriksjoner og må da utelukkes. Problemet med variabel vannhøyde kan løses med approksimasjon der en lager en representativ middelhøyde i magasinet, eller ved at målfunksjon lages stykkevis lineær. Det er derfor en rekke faktorer som unike for

vannkraftsprodusentens planleggingsproblem og således gjør problemet mer komplekst enn for en termiskkraftprodusent.

7.3. Formulering av optimeringsproblemet

7.3.1. Matematisk formulering av driftsplanleggingsproblemet

Nedenfor er minimeringsproblemet til en vannkraftprodusent formulert.

$$\text{Min } E \left\{ \sum_{t \in T} \sum_{i \in I} \left[\overbrace{c_t(u_{t(i)})}^{ICF} + \overbrace{\alpha_{t+1}(v'_{t+1(i)}, \xi'_{t(i)})}^{FCF} \right] \right\}$$

$$v'_{t+1(i)} = v_{t(i)} - u_{t(i)} - s_{t(i)} + \xi'_{t(i)} + \sum_{j \in J} (u_{t(j)} + s_{t(j)}) \quad \forall i, l, t, j: \text{ Vannbalanse krafverk/magasin } i$$

$$v_{t(i)} \leq \bar{v}_{(i)} \quad \forall i, t \quad : \text{ Maksimalt innhold i magasin } i$$

$$u_{t(i)} \leq \bar{u}_{(i)} \quad \forall i, t \quad : \text{ Maksimal produksjon i kraftverk } i$$

$$u_{t(i)} + s_{t(i)} \leq \bar{V}_{(i)} \quad \forall i, t \quad : \text{ Maksimal vannføring etter kraftverk } i$$

$$u_{t(i)} + s_{t(i)} \geq \underline{V}_{(i)} \quad \forall i, t \quad : \text{ Minimal vannføring etter kraftverk } i$$

Hvor :

$i = 1, \dots, I$ Indeks for vannkraftverk ($I =$ antallet kraftverk/magasin).

$j = 1, \dots, J$ Settet av kraftverk som er oppstrøm for kraftverk i .

$l = 1, \dots, L$ Indeks for tilsigsscenarioer.

$t = 1, \dots, T$ Tidsindeks

$v_{t+1(i)}$ Lagret volum i kraftverk i ved slutten av tidsrom t (beslutningsvariabel).

$v_{t(i)}$ Lagret volum i kraftverk i ved begynnelsen av tidsrom t (kjent).

$\xi'_{t(i)}$ Tilsig til kraftverk i i tidsrom t i scenario l (stokastisk).

$u_{t(i)}$ Vann brukt til produksjon i kraftverk i i tidsrom t (beslutningsvariabel).

$u_{t(j)}$ Vann brukt til produksjon i kraftverk j i tidsrom t (beslutningsvariabel).

$s_{t(i)}$ Overløp i kraftverk i i tidsrom t (beslutningsvariabel).

$s_{t(j)}$ Overløp i kraftverk j i tidsrom t (beslutningsvariabel).

$\bar{u}_{(i)}$ Maksimal produksjon i kraftverk i .

$\bar{v}_{(i)}$ Maksimalt tillatt magasinifilling for magasin i .

$\bar{V}_{(i)}$ Maksimal vannføring etter kraftverk i .

$\underline{V}_{(i)}$ Minimal vannføring etter kraftverk i .

$ICF =$ Funksjon for umiddelbare kostnader.

$FCF =$ Funksjon for framtidige kostnader.

I tillegg til disse restriksjonene kan det være krav om en nedre grense på magasinfyllingen. En slik restriksjon kan være en del av konsesjonskravene eller være gitt av nedre høyde på vanninntaket. Restriksjonen kan skrives slik.

$$u_t(i) = 0 \quad \text{hvis} \quad v'_{t+1}(i) \leq \underline{v}(i) \Rightarrow u_t(i)[v'_{t+1}(i) - \underline{v}(i)] \geq 0 \quad (73)$$

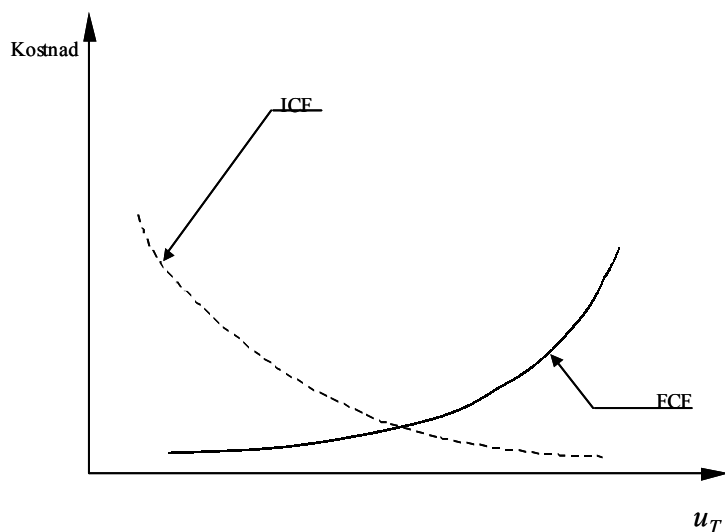
$\underline{v}(i)$ Minimalt tillatte magasinfylling for magasin i

Sammenhengen er ikke-lineær fordi $u_t(i)$ og $v_{t+1}(i)$ er beslutningsvariable og må dermed utelates for å beholde problemet lineært.

7.3.2. Hva inngår i ICF og FCF

Kostnader knyttet til ICF for en ren vannkraftprodusent er salg av kraft til spotmarkedet som igjen avhenger hvordan egen produksjon av vannkraft styres i tidssteg t . ICF er dermed en funksjon som er økende i u_t fordi økt egenproduksjon gir økende kostnader i tidssteg t . I de tilfeller hvor produsenten også har termiske kraftverk vil produksjonskostnadene i tidssteg t inngå i ICF.

Funksjonen FCF representerer alle kostnader som påløper etter tidssteg t ; det vil si fra og med $t+1$ til og med T som er slutten på planleggingshorisonten. FCF vil være økende i u_t fordi det å bruke mye vann i dag leder til at vi har mindre i magasinet i morgen, hvilket innebærer at v_{t+1} blir mindre. Dersom produsenten har termisk kraftproduksjon i dette tidsrommet vil kostnaden for den inngå her. Sammenhengen mellom ICF og FCF ved forskjellige nivåer av produksjonsvolum u_t er vist i Figur 7.3-1.



Figur 7.3-1: Sammenheng mellom ICF og FCF

7.4. Fastkraftforpliktelsens virkning på driftsplanleggingen

Stort sett har alle produsenter avtale om å levere en fast mengde kraft til gitt pris i tidsrommet det planlegges for. En rasjonell produsent vil alltid minimere sin kostnader forbundet med en slik kontrakt. Av dette følger at produsenten kjøper kraft i spotmarkedet for å tilfredstille fastkraftkontrakter når vannverdien er over spotprisen, og produserer selv om vannverdien er lavere enn spotpris. Under forutsetningen av han alltid kan kjøpe nok kraft i spotmarkedet for å tilfredstille sine fastkraftforpliktelser kan denne enkle matematiske formuleringen av minimeringsproblemet settes opp som på neste side:

c_t : Enhetskostnader for overføring og kjøp av strøm i spotmarkedet i tidspunkt t .

ρ : Konverteringsfaktor fra tappevolum til energi i kraftverket.

u_t : Tappet volum fra kraftverket i tidsrom t for å tilfredstille fastkraftkontrakter.

D_t : Fastkraftsforpliktelse for tidsrom t .

π_t : Vannverdi i pr volumenheter i kraftverket i tidsrom t .

e_t : Mengde energi kjøpt i spotmarkedet i tidsrom t .

$$\text{Min}_{u_t, e_t} \sum_{t \in T} ICF_t = \text{Min}_{u_t, e_t} \sum_{t \in T} c_t e_t \quad (74)$$

s.t.

$$\rho \cdot u_t + e_t \geq D_t \Rightarrow \left. \begin{array}{l} \pi_t \geq c_t \Rightarrow e_t = D_t \quad (\text{Kjøper kraft i spotmarkedet}) \\ \pi_t \leq c_t \Rightarrow \rho \cdot u_t = D_t \quad (\text{Produserer kraft selv}) \\ \pi_t = c_t \Rightarrow \rho \cdot u_t + e_t = D_t \quad (\text{Kan kjøpe kraft i spot eller produsere selv}) \end{array} \right\} \Rightarrow (75)$$

Merk at totalt tappet volum alltid kan deles slik: $u_i^{total} = u_i^{salg} + u_i^{kontrakt}$

Når $u_i = u_i^{kontrakt}$ vil restriksjon (75) alltid binde. Årsaken er at u_i velges lik null når vannverdien er høyere en spotpris. Med vannverdi lavere spotpris velges e_i lik null. Når vannverdi er lik spotpris kan all kraft kjøpes i spot eller produseres selv. I alle tilfellene blir $\rho \cdot u_i + e_i = D_i$, og dermed kan det substitueres for e_i i likning (74).

$$(75) \Rightarrow \sum_{j \in J} \rho \cdot u_{j,t} + e_i = D_i \Rightarrow e_i = D_i - \sum_{i \in I} \rho \cdot u_i \quad (76)$$

Setter inn fra uttrykket for e_i fra (76) i målfunksjon.

$$\text{Min}_{u_i; e_i} \sum_{i \in I} c_i \cdot (D_i - \sum_{i \in I} \rho \cdot u_i) \Leftrightarrow \text{Min}_{u_i; e_i} \sum_{i \in I} -c_i \cdot \rho \cdot u_i \quad \text{fordi } (c_i \cdot D_i) = \text{konstant}$$

Vi ser da at fastkraftforpliktelsene blir et konstantledd i minimeringsfunksjonen, og innvirker således ikke inn på løsningen av driftsplanleggingsproblemet. Merk at IFC_t , som er den umiddelbare kostnadsfunksjonen, kun er definert for leddet som dekker kjøp av kraft. I reelle problem vil også andre utgifter inngå, uten at det har betydning for gyldigheten av konklusjonen her.

8. Implementering av algoritme

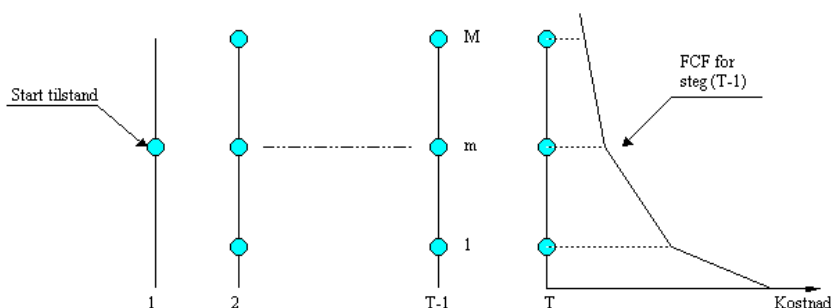
8.1. Hvorfor velg SDP

Det finnes flere optimeringsmetoder for å løse vannkraftprodusentens planleggingsproblem, og den som velges brukt her er SDP (stochastic dynamic programming) [Stage & Larsson (1961)]. Metoden har vært i bruk over 30 år innen produksjonsplanlegging innen vannkraft og er forholdsvis enkel å implementere.

Andre metoder kunne også vært benyttet, og en av disse er stokastisk dual dynamisk programmering (SDDP) [Pereira (1989)]. Andre nyere metoder er Abridged Nested Decomposition (AND) og Reduced Sampling Method (ReSa). ReSa er en kombinasjon av SDDP og AND [Hindsberger og Philpott (2001)]. Felles alle disse metodene er at de er designet for hydroelektriske produksjonssystem med flere magasin hvor regnetiden med SDP vil vært veldig lang. Men i dette tilfellet skal problemet kun løses for ett kraftverk med et magasin, og bruken av SDP er da uproblematisk.

8.2. Framgangs måte ved SDP rekursjon

Metoden er basert på prinsippet til Bellman (1957), og er basert på diskretisering av tilstandsrommet. Løsningen av problemer skjer ved baklengs rekursjon som i vanlig DP(dynamic programming), og forutsetter en oppdeling av tilstandsrommet som nedenfor:



Figur 8.2-1: Definisjon på systemtilstander

For hvert tidssteg t (typisk en uke eller måned) defineres et sett av systemtilstander indeksert med $m=1,2,\dots,M$, for eksempel reservoarnivå på 100%, 90% og så videre til 0%. Videre forutsettes starttilstanden for magasinnivået kjent ved $t=1$. Sett den framtidige kostnadsfunksjonen $FCF_{T+1}=0$, om en ikke har tilgang på langtids analyser fra samkjøringsmodellen eller Vannsimtap. Om slik analyser er tilgjengelig, kan verdien av FCF_{T+1} hentes fra disse. Med disse forutsetningene gitt, start i siste steg T og løs ettsteg planleggingsproblemet for de hydroelektriske produksjon. Ettsteg planleggingsproblemet for hydroelektrisk produksjon er gitt ved:

$$z_t = \text{Min } c_t(u_t) + \alpha_{t+1}(v_{t+1}) \quad (\text{Likning 8.2-1})$$

s.t.

$$v_{t+1} = v_t - u_t - s_t + a_t$$

$$v_{t+1} \leq \bar{v}$$

$$u_{t+1} \leq \bar{u}$$

v_t : Magasin nivå i periode t

u_t : Produksjon i periode t

s_t : Overløp i periode t

a_t : Tilsig i periode t

Ett stegs problemet løses problemet løses for alle tilsigsscenario $k=1,\dots,K$, prisscenario $p=1,\dots,P$ og magasintilstander $m=1,\dots, M$.

Beregn deretter forventet driftskostnad knyttet til magasintilstand m ved å regne forventningsverdien over alle kombinasjoner av pris og tilsigsscenario. Denne verdien vil være første punkt i funksjonen for framtidige forventet kostnad (FCF) i steg $T-1$ når denne beregningen gjøres for magasinnivå M . Gjenta disse beregningen suksessivt for øvrige magasinnivå. Interpoler deretter mellom kostnadene i alle steg og lag $FCF = \alpha_T(v_T)$ for steg $T-1$, som illustrert i Figur 8.2-1.

Prosessen er gjentaes for alle steg $t=T-1, T-2, \dots, 1$, men nå med målfunksjonen minimere summen av de umiddelbare kostnadene (ICF_{t-1}) og de framtidige kostnaden (FCF_t), gitt av den nettopp beregnede FCF'en.

8.3. Seriekorrelasjon i tilstandsvariable

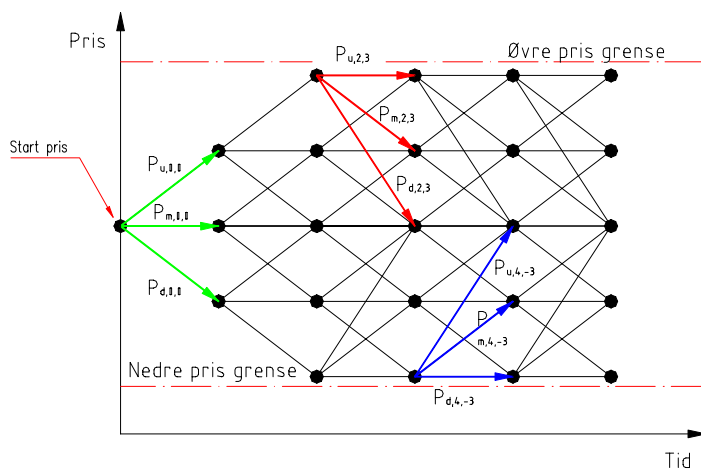
8.3.1. *Generelt*

Både pris og tilsig har seriekorrelasjon. For å oppnå best mulig tilnærming av resultater i optimeringen, taes det hensyn til dette i modellen. Antagelsen som implementeringen av seriekorrelasjon i Samkjøringsmodellen og Vannsimtap bygger på er å anta den kun gjelder i ett tidssteg. Implisitt antaes det dermed at tilsig og pris følger en markovprosess. En vanlig teknikk for å implementere denne seriekorrelasjon blir derfor ved bruk av metoder knyttet til markovegenskapen. Alternativt kan seriekorrelasjon ivaretaes ved bruk av AR(1) prosess, slik som det er gjort i tilsigmodellen som brukes i Plansddp [Gjelsvik (1992)].

Valget av metodene for å ta hensyn til seriekorrelasjonen påvirker løsningen av driftsplanleggingsproblemet og dette aspektet må derfor vurderes i valget av metode. I enkelte tilfeller legger også optimeringsalgoritmen føringer for hvordan dette kan løses og dette skjer blant annet i SDDP-algoritmen [Pereira & Kelman m. flere (1992)]. Kravet i SDDP er at den første deriverte av likningen som beskriver en eller annen stokastisk prosess er en konstant, hvilket i praksis innebærer ettstegs seriekorrelasjon. Valget av SDP som optimeringsalgoritme gir derfor føringer for hvordan seriekorrelasjon kan implementeres, og hvordan dette innvirker på våre tilstandsvariable vil bli diskutert nedenfor.

8.3.2. Seriekorrelasjon i pris

I prismodellen som bygger Hull & White modellen var det nødvendig å beregne alle overgangssannsynligheter for å bestemme tilstandsprisene. Matrisen for overgangssannsynligheter kan konstrueres ved bruk av disse data. I motsetning til overgangsstrukturen i prismodellen som brukes i blant annet Vannsimtap [Mo & Green m. flere (2000)] er det ikke mulig med overgang fra en ekstrem høy pris til en ekstrem lavt pris i løpet av et tidssteg med denne modellen. Årsaken er at hver node i modellen kun har overgang til de tre noder etter mønsteret indikert med rød, grønn og blå piler i figuren under.



Figur 8.3-1: Overganger i trinomisk tre

Følgene av å bruk denne modellen er en overgangssannsynlighetsmatrise gitt tidspunkt t med følgende utseende om vi antar at antall prispunkter $i = 7$ (Figur 8.2-1 har 5):

$$\mathbf{P}_t(i \times i) = \begin{pmatrix}
 p_{1,1} & p_{1,2} & p_{1,3} & 0 & 0 & 0 & 0 \\
 p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 & 0 & 0 \\
 0 & p_{3,2} & p_{3,3} & p_{3,4} & 0 & 0 & 0 \\
 0 & 0 & p_{4,3} & p_{4,4} & p_{4,5} & 0 & 0 \\
 0 & 0 & 0 & p_{5,4} & p_{5,5} & p_{5,6} & 0 \\
 0 & 0 & 0 & 0 & p_{6,5} & p_{6,6} & p_{6,7} \\
 0 & 0 & 0 & 0 & p_{7,5} & p_{7,6} & p_{7,7}
 \end{pmatrix}$$

Legg merk til at selv om ikke alle noder kommuniserer direkte med hverandre, er det en mulighet å bevege seg mellom disse via andre noder (f.eks fra node 1 → node 7). Følgelig er det også mulig å komme til tilbake til noden du startet fra. Det gjelder for alle noder og prisen kan derfor sies å være en markovkjede som er positivt tilbakevendende [Ross (1997)]. Det finnes derfor en sannsynlighet større enn null for å gå fra en ekstrem høy pris til en ekstrem lav pris innen en eller annen tid t , $t \neq \infty$.

I tillegg til diskrete pris og overgangssannsynligheter, gir bruken av prismodellen også et mer realistisk bilde av prisbevegelsene. Årsaken er de nevnte begrensninger i nodekommunikasjon og dermed ikke hopp fra ekstrem lavt til høy pris eller visa versa. Ulempen med modellen er at antall prispunkt er gitt av inputen i Hull & White modellen og kan ikke styres av brukeren. Bruk av denne modellen kan derfor på større system (flere magasin) medføre lange regnetider om antall prispunkter er høyt. I vårt tilfelle er det kun en modell med et magasin som skal brukes og modellen medfører derfor ingen ulemper i dette tilfellet.

8.3.3. Seriekorrelasjon i tilsig

For å kunne teste hypotesen om det er bedre å gjøre driftsplanleggingen med forwardpriser enn med forventet spotpris behøves ett konkret kraftverk med tilhørende data. Via samarbeidspartner er det gitt adgang til et konkret vannkraftverk med nødvendig data for å kunne gjennomføre en driftsplanlegging. Hvilket kraftverk dette er, holdes utenfor oppgaven etter ønske fra samarbeidspartneren. Tilsigsmodellen som brukes er identisk med prismodellen utviklet av Mo & Green med flere (2000) (se vedlegg 9). Modellen er basert på markovegenskapen og ivaretar dermed hensynet til seriekorrelasjon som finnes i tilsiget.

Markovmodellen tilpasses ved opptelling av overganger mellom tilsigsbånd, for ytterlig informasjon se vedlegg 9. For å få en fornuftig struktur på tilsigsdataene i modellen fjernes verdier som ligger under 1 % kvantilen og tilsvarende de som ligger over 99 % kvantilen. Grunnen til at det gjøres er at ekstremverdier i datasettet medfører at tilstandsrommet må deles opp i mange tilsigsbånd. Antall overganger til hvert bånd blir da få, og overgangssannsynligheten blir da også basert på få overganger. Modellen er også laget på en slik måte at sensurering ikke har

konsekvenser for modellen statistiske egenskaper, fordi hensynet til ekstremtilstander blir håndtert i en egen korreksjonsmetodikk [Mo & Green med flere (2000)].

Sensuren av datasettet gjøres ved å regne ut hva verdien er på f.eks 99 % kvantilen for hele datasettet. Verdier som ligger over denne verdien så byttet ut med verdien som tilsvarer 99 % kvantilen. Prosedyren medfører at som ligger over eller under nederste kvantil blir byttet ut i samtlige tilsigsrekker.

9. Interpolering

Ved løsning av ett stegsproblemet i SDP brukes $\alpha_{t+1}(\cdot)$ som funksjonsverdi for framtidige kostnader, det vil si FCF. Verdien av denne finnes med å interpolere i tabellen over funksjonsverdier av $\alpha_t(\cdot)$. Vårt tilfelle med 21 magasinnivå, 7 tilsigsalternativ og 13 priser inneholder denne tabellen $21 \cdot 7 \cdot 13 = 1911$ verdier som brukes for å danne FCF for løsning i tidspunkt t .

Interpolasjonen gjøres ved bruk av en standard Matlab funksjon. På grunn av virkemåten til denne funksjonen må det i tillegg skrives en funksjon som trekker ut nødvendig verdier av datasettet før interpolasjon. Årsaken er at Matlab genererer en matrise på $1911 \cdot 1911 \cdot 1911$ for hver enkelt variabel som inngår i interpolasjon. Dimensjonen på denne matrisen blir dermed større en hva som er tillatt i Matlab. Funksjon trekker derfor ut de 8^7 verdiene med tilhørende variabelverdier som er nødvendig for interpolasjon i tre dimensjoner, og muliggjør dermed bruk av standard funksjonen i Matlab for interpolasjon.

10. Forenklinger i prismodellen

Ved interpolering er det et krav at verdiene som er input til interpolasjon er ikke ligger utenfor verdiene som ligger i tabellen det interpoleres fra. Prismodellens oppbygging er slik at prisen stiger og synker. Maksimal eller minimal pris i periode t kan derfor bli forskjellig fra periode $(t+1)$. I så tilfelle blir løsningen å ekstrapolere, og det krever en annen løsningsmetode, noe som ikke er ønskelig. Problemet omgås ved å sette verdiene i topp og bunnodene konstant. Verdiene i som brukes i toppnoden er derfor lik maksimal observert verdi som finnes i toppnoden over alle perioder det planlegges for. Tilsvarende gjøres for bunnodene. Å settes disse verdien konstant leder til at variansen øker fordi verdien er lik de minimale eller maksimale som observeres over hele tidsperioden. Egenskaper som skeivhet og kurtosis blir også påvirket og fordelingen til prisen blir mer skeiv mot maksimal verdien og noe flatere som følge av lavere kurtosis. Årsaken er prisendringen i toppnoden er størst i absolutt verdi og påvirker fordelingen mer enn de små endringen i bunnoden. Korreksjonen som gjøres av prismodellen har en del uheldige virkninger, men velges for å kunne løse optimeringsproblemet.

⁷ Det behøves 2^n funksjonsverdier med variable for interpolasjon når vi har n variable.

11. Eksplisitt formulering av optimeringsproblemet

Med alle undermodeller til SDP-algoritmen på plass kan optimeringsproblem skrives fult ut, og er gitt slik:

$$\text{Sett FCF : } \alpha_{T+1}(v_{T+1}^m, a_{T+1}^k, p_{T+1}^i) \equiv 0$$

Løs et stegs problemet for alle tidsperioder $t = T, T-1, \dots, 1$.

for $m = 1, \dots, M$

for $k = 1, \dots, K$

for $i = 1, \dots, I$

$$\alpha_t(v_t^m, a_t^k, p_t^i) = \text{Min } E \left\{ -p_t^i \cdot u_t + \sum_{k=1}^K \sum_{j=1}^I p_{kl} \cdot p_{ij} \alpha_{t+1}(v_{t+1}^k, a_{t+1}^l, p_{t+1}^j) \right\} \quad (\text{Likning 11-1})$$

s.t.

$$v_{t+1}^k = v_t^m - u_t - s_t + a_t^k$$

$$s_t - u_t \leq 0$$

$$v_{t+1} \leq \bar{v}$$

$$u_{t+1} \leq \bar{u}$$

nest

nest

Lag en komplett FCF ved å interpoler på $\alpha_t(v_t^m, a_t^k, p_t^i)$ i forrige steg med verdien

$$\{\alpha_t(v_t^m, a_t^k, p_t^i), m = 1, \dots, M, k = 1, \dots, K \text{ og } i = 1, \dots, I\}$$

nest

p_{kl} : Overgangs sannsynlighet fra tilsignnivå k i tidspunkt t til l i periode $t+1$ –

p_{ij} : Overgangs sannsynlighet fra prisnivå i i tidspunkt t til j i periode $t+1$.

Øvrig variable som tidligere definert i likning (8.2-1).

Verdien av M i algoritmen velges slik at et det er mulig å hoppe minst et steg på den diskret magasininnstillingen i ett tidssteg [Gjelsvik, møte 21/5-02]. Det innebærer ved et gjennomsnittlig tilsig på 10, så bør $v^{m+1} - v^m \leq 10$.

12. Løsning av optimeringsproblemet

Algoritmen til det lineære optimeringsproblemet ble forsøkt implementert i Matlab, noe ikke som lot seg gjennomføre på grunn av interpolasjonsprosedyren. Problemet ble derfor omskrevet til et ikke-lineært problem, og dette lot seg løse i kombinasjon med interpoleringsalgoritmen for testcasen som var laget. Løsningstiden pr subproblem i denne algoritmen ble ca 3.2 sekunder og med $1911 \cdot 52 = 99372$ subproblem i det fulle problemet blir estimert løsingstid ca 90 timer.

Den lange løsningstiden antas å henge sammen med at hver iterasjon må hente verdier fra interpolasjonsalgoritmen. Med i størrelses orden 25 iterasjoner pr subproblem blir løsingstiden derfor lang. Det ble også forsøkt å bruke løsningen fra forrige subproblem som startverdi i neste problem og det gav en reduksjon i løsingstid fra opprinnelige 3.7 til 3.2 sekunder. Fordi estimert løsingstid er i størrelses orden 90 timer ble det full optimeringsproblemet løst ikke.

I mangel på tid er det ikke forsøkt med noen annen algoritme, selv om det finnes algoritmer som ikke bruke interpolasjon. En av disse er SDDP, og den kunne vært brukt her. Ved bruk av SDDP vil også problemet med korreksjonen av prismodellen vært unngått. I ettertid kan enn kanskje si at disse problemene burde vært identifiser før valget av algoritme ble gjort. Kanskje er dette korrekt, men før gjennomføringen av oppgaven var kunnskapen om Matlab totalt fraværende. En anbefaling til de som skal løse liknende oppgaver i framtiden må derfor bli slik "Lag kunstige datasett, og bruk disse for å teste algoritmer og program med hensyn på løsingstid i ett tidlig stadium av oppgaven".

13. Anbefalinger til videre arbeid

Hensikten bak alle modeller som er tilpasset eller forsøkt tilpasset er å kunne bruke disse i en optimering som løser et driftsplanleggingsproblem. Ønsket om å løse dette problemet bærer i hypotesen om hva som optimalt å bruke av forwardpriser eller forventet spotpris for å maksimere verdien av kontantstrømmene til en vannkraftprodusent. De anbefalinger som gis her må sees i lys av dette.

Prismodellen som ble forsøkt tilpasset slik at korrelasjon mellom tilsig og pris ble ivarettatt ble ikke god. For de som skulle lykkes å lage en slik modell, kanskje etter det forslaget som er gitt i kapittel 4.4, danner det basis for en diskret prismodell. En slik diskret modell kan være SEFas modell [Mo & Green med flere (2000)] som lager diskrete utfall og beregner overgangssannsynligheter. Modellen har vist seg å ha de ønskede statistiske egenskaper som kreves, og er forholdsvis lett å implementere.

Prismodellen som er tilpasset her har noen egenskaper som kanskje gjør den egnet for videre bruk. Disse egenskapene er hovedsakelig knyttet til god evner til å gjenskape fordelinger, en egenskap knyttet til trinomiske tre [Haug (1997)]. At prisen blir markedsjusterte er den andre. Før videre bruk av modellen skjer, bør de statistiske egenskapene til denne kontrolleres. Modellen viste relativt gode egenskaper for å gjenskape variansstrukturen, og men det bør være mulig å lage en bedre modell for variansen. Særlig med tanke på at dette kan gjøres nesten uavhengig av det trinomiske treet, kun mindre endringer i programkoden er påkrevd. En modell med bedre varians vil også kunne gi en god tilpassning av skeivhet og kurtosis, fordi disse avhenger av variansen.

Tilsigsmodellen bygger som tidligere nevnt på en prismodell fra SEFas [Mo & Green med flere (2000)]. Modellens egenskaper antas derfor å være godt dokumentert. Usikkerhetsmoment i den tilpassede modellen er derfor sensureringen som er gjort og hvor vidt dette påvirker fordelingsegenskapene. Om modellen skal brukes til å verifisere ovenfor nevnte hypotese bør kanskje disse egenskapene sjekkes.

Optimeringsalgoritmen som ble implementert her viste seg å jobbe langsomt, og for de som arbeider med liknende problemer, er og blir anbefalingen å teste algoritmen som skal brukes på mindre testsett. Disse kan gjerne være selvlaget, og uten rot i virkeligheten.

14. Oppsummering

I denne rapporten er det forsøkt å laget en prismodell som skulle ta hensyn til korrelasjon mellom pris og tilsig på landsbasis. Dette lyktes ikke, men det er gitt et forslag om å bruke en tofaktor modell som en mulig løsning på dette.

Videre er det utviklet en prismodell som er basert på ett trinomisk tre. Parametrene som brukes som input til modell er hentet fra en enfaktor modell som ble forsøkt tilpasset. Modellen kan sesongjusteres med bruk av markedspriser og er fleksibel i forholdt til hvilken variansmodell som brukes. Variansmodellen som er implementert viser at varians strukturen fra modellen er ganske bra, men at den også kan forbedres.

Det er også tilpasset en tilsigsmodell, og denne er basert på SEFas prismodell med mindre korreksjoner. Modellen er en velprøvd fra SEFas side, og den antas derfor å ivareta de statistiske kravene som en slik modell skal ha. Tilpassningen av modellen er gjort med et datasett fra ett konkret kraftverk.

Som algoritme for å løse driftsplanleggingsproblemet ble SDP benyttet i sammen med pris og tilsigsmodellene som er tilpasset. Algoritmen ble implementert i Matlab i kombinasjon med en interpolasjonsalgoritme, og ble forsøkt løst som ett lineært problem, noe som mislyktes. Problemet ble deretter skrevet om til ett ikke lineært, som lot seg løse. Den lange løsningstiden for det ikke lineære problemet ble for lang, og planleggingsproblemet er derfor ikke løst. Følgelig er ikke hypotesen om at driftplanlegging med forwardpriser gir bedre økonomiske resultater en med forventet spotpris testet.

15. Referanse- og litteraturliste

Aronsen.G & D.M. Dybdahl. (2001), "Pris og tilsig i driftsplanleggingsproblemet" *Prosjekt i faget SIS 1101 Institutt for industriell økonomi og teknologiledelse, Norges teknisk-naturvitenskaplige universitet (NTNU).*

Bellmann, R. (1957), *"Dynamic programming*, Princeton University Press.Ag

Bessembinder , H & Coughenour, J & Seguin, P & Smoller M, *"Is There a Term Structure of Futures Volatilities? Reevaluating the Samuelson Hypothesis"* , *Journal of Derivatives*, Winter 1996. (Abstrakt i C.F.A. Digest, Sommeren 1997).

Boyle, P. P. (1986), *Option Valuation Using a Three Jump Process*, *International Options Journal* 3 pp 7-12.

Clelow, L og Strickland C. (April 1999), *"Valuing Energy Options in a One Factor Model Fitted to Forward Prices"*.

Fleten, S.-E. (2000), *Portfolio management emphasizing electricity market applications A stochastic programming approach*, Dr.ing avhandling, NTNU, Trondheim

Fleten, S.-E. & Lemming, J. (2001), *Constructing forward price curves in electricity markets*, NTNU, Trondheim

Gjelsvik, A, møte Sintef Energiforskning 21/05-2002.

Gjelsvik, A. (1992), *Stokastisk tilsigsmodell for driftsplanlegging*, Sintef Energiforskning TR A4008, Trondheim

Gjolberg, O & Johnsen, T (2002), *Electricity futures: Inventories and Price and Relationships at Nord Pool*, (Preliminary draft)

Haug, Espen G. (1998), *The Complete Guide To Option Pricing Formulas*, 1. utg, McGraw-Hill, New York, USA

Hindsberger, M. & Philpott, A. B. (2001) *Medium-term Hydro-thermal Scheduling with Stochastic Inflows*, Working paper, 9th International Conference on Stochastic Programming, Berlin, Aug 25-31 2001

Hull, John C. (1993), *Options, Futures and Other Derivative Securities*, 2. utg, Prentice Hall, Englewood Cliffs, USA

Johnson, R. A. & Wichern, D. W. (1998), *Applied Multivariate Statistical Analysis*, fjerde utgave, Prentice Hall, New Jersey

Koekebakker, S. & Ollmar, F. (2001), *Forward Curve Dynamics in the Nordic Electricity Market*, HIA/NHH, Kristiansand/Bergen

Lucia, J. J. & Schwartz, E. S. (Mars 2001), *Electricity Prices and Power Derivatives. Evidence from the Nordic Power Exchange*, Dpto. Economía Financiera y Matamática Universidad de Valencia & The Anderson School at UCLA, Los Angeles USA

Mo, B. & Green, T. & Warland, G. & Haugstad, A. (2000), *Dokumentasjon av ny prismodell til bruk i Vansimtap, Planssddp og Beta*, Sintef Energiforskning Trondheim

Pereira, M.V.F. (1989), *Optimal Stochastic Operations Scheduling of Large Hydroelectrical Systems*, International J. of Electrical Power & Energy Systems 11: s 161-169.

Pereira, M.V.F. & Kelman, R. & Campodonico, N. (1992), *Stochastic Optimization of Complex Hydrothermal System In a Competitive Framwork*, PSRI.

Ross, Sheldon M. (1997), *Introduction to Probability Models, Sixth Edition*, Academic Press, San Diego, California, USA.

SSB 1, Statistisk Sentralbyrå, emne område : Kraft- og vannforsyning, fyllingsgrader
adresse : <http://www.ssb.no/emner/10/08/vannmag/>.

Sollibakke, P. B. (2001b), *A Stochastic Volatility Model Specification with Diagnostics for the Nordic Forward Electric Power Market*, Institutt for bedriftsadministrasjon og økonomi, Høyskolen i Molde

Stage, S. & Larsson Y. (1961), *Incremental cost of water power. AIEE Transactions(Power Apparatus and Systems)*, August: 361.

Nordpools hjemmeside <http://www.nordpool.com>

16. Vedlegg

Vedlegg nr 1 : Deskriptiv statistikk priser	2
Vedlegg nr 2 : Scatterplott pris data	4
Vedlegg nr 3: Script i Splus for import av prisdata	7
Vedlegg nr 4 : Utledning av korreksjon parameteren α_i	8
Vedlegg nr 5: Kendall test.....	9
Kendall's τ : Procedure.....	9
Vedlegg nr 6: Matlab kode prismodell.....	12
Kode for Trinomisk tre	12
Funksjon for å lage overgangsmatrise i prismodell	17
Funksjon for korreksjon av pristre	17
Vedlegg nr 7: Kode implementering av SEFas prismodell for tilsig	18
Funksjon som legger til ekstrem scenarier i overgangssannsynlighetene.....	18
Funksjon som legger til ekstremscenarier verdier i tilsigspunkt.....	18
Funksjon som finner betinget forventingsverdi i hvert tilsigband	19
Funksjon som finner antall observasjoner i hvert tilsigsband.....	20
Funksjon som korrigerer overgangssannsynligheter.....	20
Funksjon som beregner overgangssannsynlighetsmatrisen	22
Funksjon som finne verdien på tilsigspunktene.....	23
Funksjon som fjerner ekstremverdier	24
Vedlegg nr 8: Koder for optimering	25
Funksjon for interpolering av FCF.....	25
Funksjon for optimering	26
Vedlegg nr 8: Diverse S-Plus funksjoner.....	28
Funksjon som finner parameterene i basismodellen.....	28
Funksjon som finner gjennomsnittlig verdi på markedsprisen for risiko	29
Funksjon som finner markedsprisen for risiko som funksjon av modningstid	30
Vedlegg nr 9: Utdrag av dokumentasjon på SEFas prismodell(kopi)	33

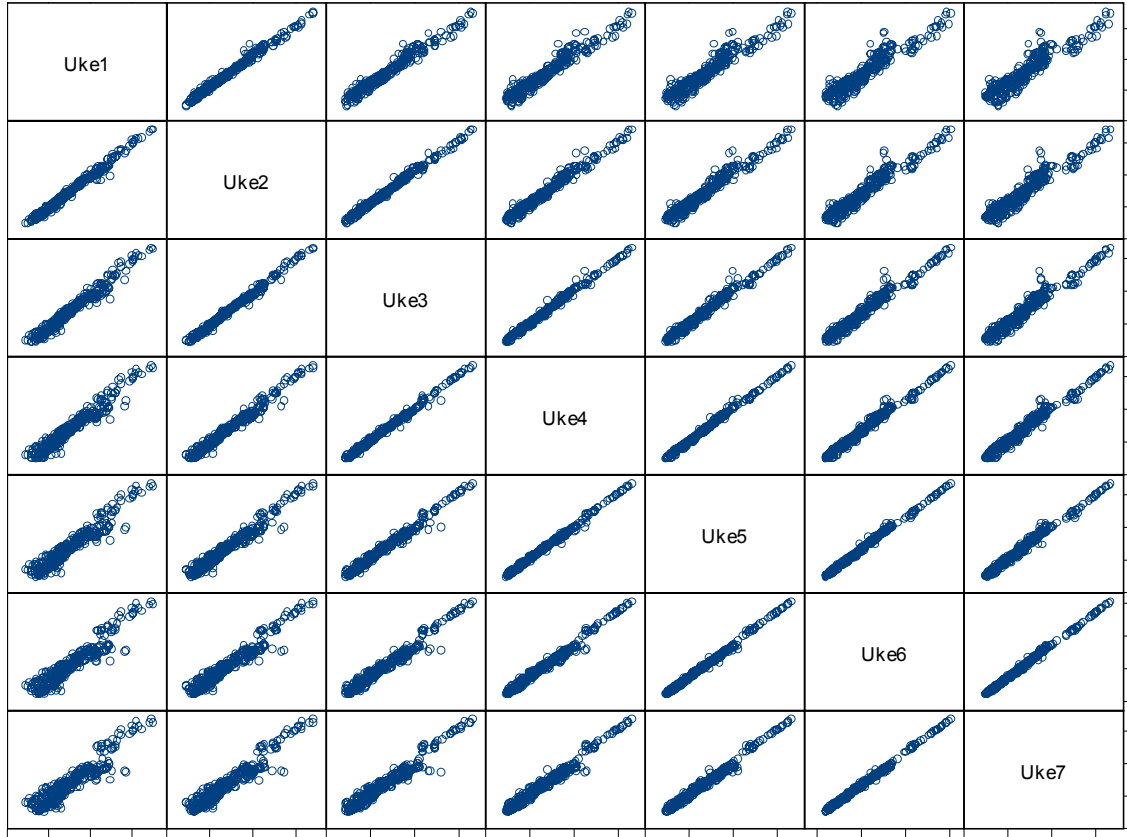
Vedlegg nr 1 : Deskriptiv statistikk priser

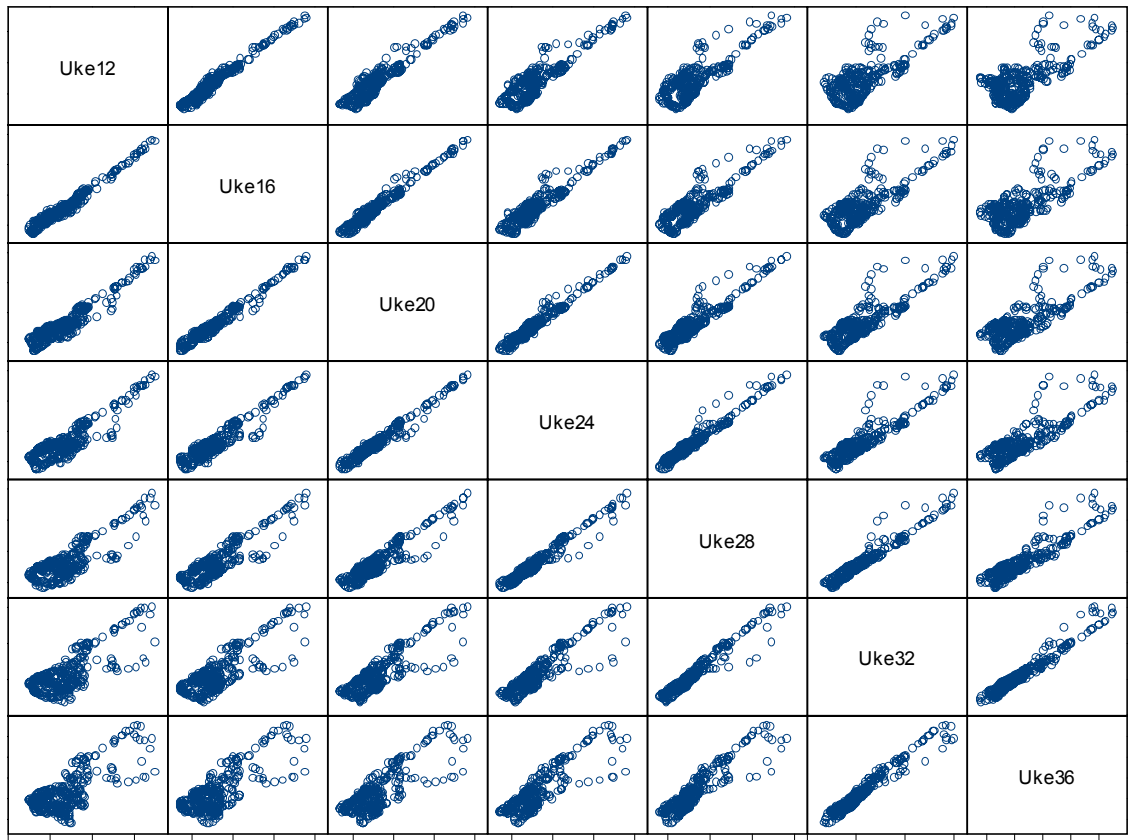
Tabell deskriptiv statistikk forwardpriser Uke 1-7							
	Uke 1	Uke 2	Uke 3	Uke 4	Uke 5	Uke 6	Uke 7
Minimum	45,3	56,3	64,3	66,6	68,2	70,3	70,0
Første kv.	107,3	107,9	107,6	105,6	108,4	109,0	109,9
Middelverdi	153,3	155,0	156,4	157,1	157,3	157,3	157,5
Median	138,5	143,0	146,5	148,7	149,4	149,8	149,8
Tredje kv.	193,3	196,0	195,6	193,7	190,8	189,0	189,7
Maximum	352,8	350,0	351,0	352,0	353,0	353,8	354,6
Standaravvik	62,3	62,4	62,6	62,4	62,1	61,9	61,8
Skevhet	0,8	0,8	0,9	0,9	1,0	1,0	1,1
Kurtosis	0,3	0,3	0,5	0,6	0,8	0,9	1,0

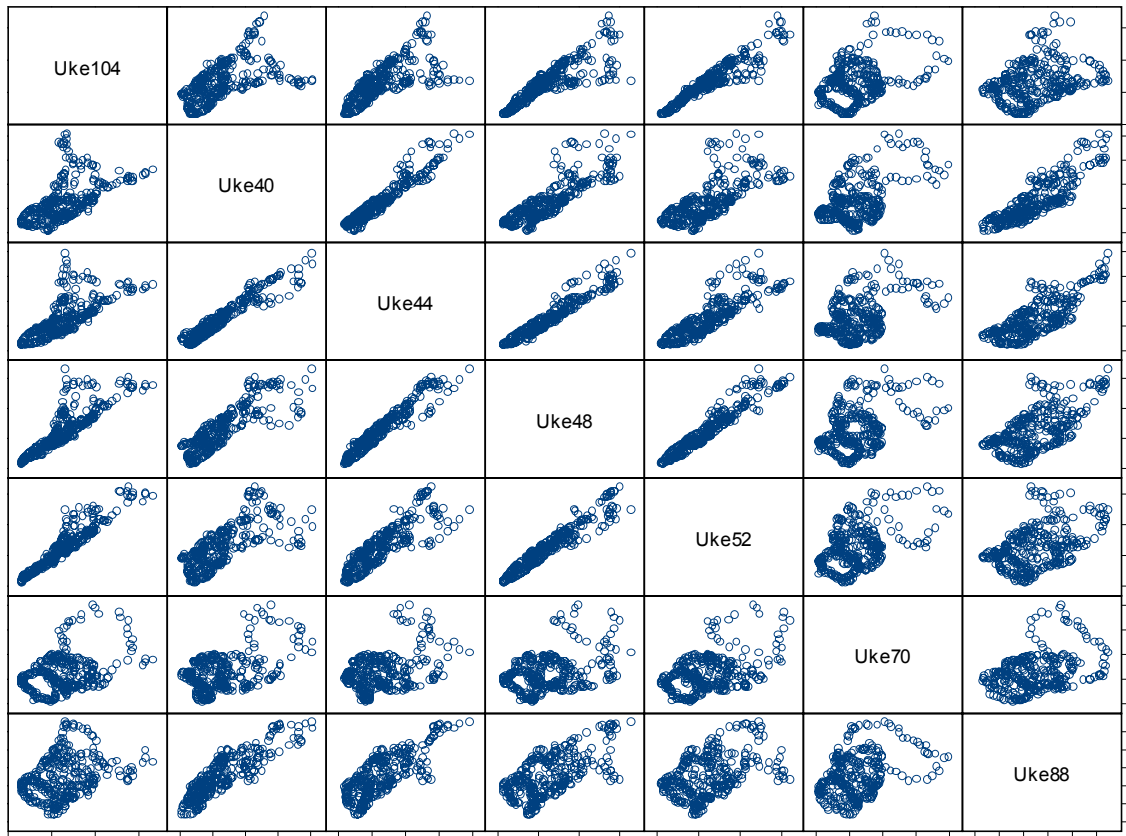
Tabell deskriptiv statistikk forwardpriser Uke 12-36							
	Uke 12	Uke 16	Uke 20	Uke 24	Uke 28	Uke 32	Uke 36
Minimum	69,8	71,9	68,5	72,2	81,9	86,4	88,9
Første kv.	113,3	119,5	123,5	130,2	130,5	128,2	127,9
Middelverdi	159,4	160,7	161,4	161,7	161,8	161,5	162,4
Median	152,7	148,5	149,5	147,1	146,7	149,3	150,6
Tredje kv.	183,7	176,7	173,0	174,0	173,8	176,2	176,7
Maximum	369,2	380,2	384,4	383,5	383,5	350,5	326,4
Standaravvik	61,6	61,5	60,6	59,4	56,5	52,5	48,1
Skevhet	1,3	1,5	1,6	1,7	1,7	1,6	1,4
Kurtosis	1,7	2,3	2,7	3,1	3,1	2,6	1,7

Tabell deskriptiv statistikk forwardpriser Uke 40-104							
	Uke 40	Uke 44	Uke 48	Uke 52	Uke 70	Uke 88	Uke 104
Minimum	102,5	110,1	106,8	104,8	104,7	108,4	115,6
Første kv.	133,6	137,3	138,8	137,2	136,8	140,4	146,4
Middelverdi	163,2	163,5	162,8	162,0	161,8	161,8	164,2
Median	153,1	154,2	154,8	157,4	155,8	157,0	162,3
Tredje kv.	180,1	182,6	182,4	180,5	183,7	182,7	177,9
Maximum	303,4	296,0	264,6	260,5	299,9	237,5	267,3
Standaravvik	42,4	37,9	35,1	33,8	37,0	28,9	29,1
Skevhet	1,2	0,9	0,7	0,7	1,2	0,5	0,8
Kurtosis	0,9	0,4	0,0	0,2	1,9	-0,2	1,0

Vedlegg nr 2 : Scatterplott pris data







Vedlegg nr 3: Script i Splus for import av prisdata

```
# S-Plus funksjon for import av forwardpriser fra prn. filer
fdata <- function(v1,v2,filplassering)
{
  filer          <- files.in.dir(filplassering)
  filnavn1       <- paste("w",v1,".prn",sep="")
  filnavn2       <- paste("w",v2,".prn",sep="")

  for (k in 1:length(filer))
  {
    if(filnavn1 == filer[k]) {startfil <- as.numeric(k)}
    if(filnavn2 == filer[k]) {sluttfil <- as.numeric(k)}
  }

  filliste <- vector(mode="list",length = (sluttfil-startfil + 1))
  teller   <- 1

  while (startfil <= sluttfil)
  {
    adresse <- paste(filplassering,filer[startfil],sep="")
    v4      <- scan(file=adresse,what=character(),sep="\t")
    v5      <- 0
    v6      <- 0

    while(v5 <= length(v4))
    {
      v5 <- v5 + 1
      if(v4[v5] == "Maturity Gen  MPS  Fourier ")
      {v6 <- v5-1}
    }

    v7 <- read.table(adresse, header=T, skip= v6)

    filliste[[teller]] <- v7[,1:2]
    startfil          <- startfil + 1
    teller            <- teller + 1
  }

  list(filliste)
}
```

Vedlegg nr 4 : Utledning av korreksjon parameteren α_i

Fra likning (66) er

$$P(0, i \cdot \Delta t) \cdot F(0, i \cdot \Delta t) = \sum_j Q_{i,j} \cdot S_{i,j}$$

Ved å uttrykke spotprisen i form av $x_{i,j}$ gir det:

$$P(0, i \cdot \Delta t) \cdot F(0, i \cdot \Delta t) = \sum_j Q_{i,j} \cdot e^{x_{i,j}} = \sum_j Q_{i,j} \cdot e^{x'_{i,j} + \alpha_i}$$

$$P(0, i \cdot \Delta t) \cdot F(0, i \cdot \Delta t) = e^{\alpha_i} \sum_j Q_{i,j} \cdot e^{x'_{i,j}}$$

som gir

$$\alpha_m = \ln \left[\frac{P(0, i \cdot \Delta t) \cdot F(0, i \cdot \Delta t)}{\sum_j Q_{i,j} \cdot e^{x'_{i,j}}} \right]$$

Vedlegg nr 5: Kendall test

Kendall's τ : Procedure

1.

Order the pairs of observations by increasing order of X 's, i.e. make sure that

$$X_{(1)} \leq X_{(2)} \leq \dots X_{(n)}$$

2.

Compare each Y with the ones below it, i.e. if Y 's are greater, than they are concordant. If Y 's are smaller, then discordant.

Example 11.3.1 We return to the Age vs. Cholesterol data set, this time given in entirety:

Age (X)	Cholesterol (Y)	Concordant	Discordant
41	274	1	7
45	209	4	3
50	194	5	1
51	270	1	4
54	165	4	0
59	234	2	1
62	281	0	2
68	238	0	1
71	208	0	0
total		17	19

This leads us to suspect that there might not be any tendency (similar numbers of concordant and discordant pairs)

Define the parameter

$$\tau = 2\Pr[(X_i - X_j)(Y_i - Y_j) > 0] - 1$$

Then τ is a parameter which describes the joint distribution, in a similar manner to how the median, mean, and variance describe a univariate distribution.

- $\tau = 0$ implies independence
- $\tau < 1$ implies perfect negative correlation
- $\tau > 1$ implies perfect positive correlation

Just like the sample mean and sample median estimate the (true population) mean and median, we can estimate τ by

$$\hat{\tau} = \frac{K}{n(n-1)/2}$$

where K is defined as the number of concordant pairs, minus the number of discordant pairs. Let K_{data} be the computation of K with the data set of interest.

So, for $H_0 : \tau = 0$ (independence), we can reject H_0 in favor of

- $H_a : \tau > 0$ (positive association), if $K_{\text{data}} \geq k(\alpha, n)$, p-value is $\Pr(K \geq K_{\text{data}})$
- $H_a : \tau < 0$ (negative association), if $K_{\text{data}} \leq -k(\alpha, n)$, p-value is $\Pr(K \leq -K_{\text{data}})$
- $H_a : \tau \neq 0$ (dependence), if $|K_{\text{data}}| \geq k(\alpha/2, n)$, p-value is $2 \times \Pr(K \geq K_{\text{data}})$

where $k(\alpha, n)$ is the upper-tail value of the distribution of K under H_0 , such that $\Pr(K \geq k(\alpha, n)) = \alpha$, and this is given in Table A.21.

The large sample approximation is through

$$Z = \frac{K}{\sqrt{\frac{n(n-1)(2n+5)}{18}}} \sim N(0, 1), \text{ under } H_0.$$

For ties, don't count as a concordant or discordant pair (recall the one-sample tests, with ties for the hypothesized median). This is for both the X

values and the Y values. Also, the large sample approximation needs to be adjusted (see p.187 in the textbook for details).

Example 11.3.2 *So for the data set, $K = -2$. This doesn't seem very large, especially when $k(0.05, 9) = 18$. So we probably will fail to reject the null hypothesis, since there are similar numbers of concordant and discordant pairs. (p -value is 0.54 against $H_1 : \tau > 0$, 0.46 against $H_1 : \tau < 0$, and 0.92 against $H_1 : \tau \neq 0$).*

<http://franz.stat.wisc.edu/~rossini/courses/intro-nonpar/text/text.html>

Vedlegg nr 6: Matlab kode prismodell

Kode for Trinomisk tre

```
function d=trinomisk_tre(startpris,risiko_r_ar,kappa_uke,sigma_uke,totaltid,periodetid)
% Laget for å beregne verdiene i ett trinomisk tre.
% Treet bygger på Hull & White modellen fra 1990.
% En vektor som inneholder den opprinnelige terminstrukturen.
% Startpris er dagens pris i kr/MWh.
% Risiko_r_ar er den risikofrirenten pr år gitt i prosent.
% Kappa_uke er mean-reversion rate pr. uke.
% Sigma_uke er standard avvik pr. uke.
% Totaltid er antall uker treet skal dekke.
% Perioder er antall steg i treet skal innholdet, dvs (perioder +1) = Imax.
% Er totaltid lik 52 uker og antall steg lik 2 blir variansen og mean-reversion regnet
% som verdier gjelder pr halv år.
% Perioder er antallet steg framover i treet.
% Delta T= 1.

%-----
% Skalerer standardavviket, diskonteringsfaktoren og mean-reversion
% slik at de stemmer overens med stegstørrelsen.

global delta_T delta_X kappa % Gjør verdiene globale slik at de kan benyttes i
subfunksjoner.
delta_T = periodetid/totaltid; % Bestemmer tidsintervallet delta T.

%sigma = sigma_uke*(totaltid)^0.5; % Standardavvik for treet's tidsrom.

diskont = exp(-log(1+0.01*risiko_r_ar)*(periodetid/52)); % Diskonteringsrate pr tidsteg.
%-----
% Må ta hensyn til den tidsvarierende variansen.
%
alfa = (sigma_uke^2)/(2*kappa_uke);

sigma_t = zeros(1,totaltid);

for i =1:totaltid
    s_temp= sigma_uke*(1-exp(-2*kappa_uke*(totaltid + 1 -i)))^0.5;

    if(s_temp < 0.05)
        s_temp=0.05;
    end

    sigma_t(1,i) =s_temp;
end

delta_X=zeros(1,totaltid +1);

for i =2:(totaltid +1)
    delta_X(1,i) =sigma_t(1,i-1); % Regner ut den stokastiske stegstørrelsen pr tidsteg.
end

kappa = kappa_uke*periodetid; % Regner ut mean-reversion pr tidssteg.

delta_T = 1;

% Fordi kappa er skalert slik at denne gjelder pr tidssteg kan delta_T utelates dvs. settes lik 1
% i likningene for overgangssannsynlighet. Alternativt kunne kappa vært skalert til å gjelde hele
% for tidene spenner over og delta_T en fraksjon av dette.
%-----
% Finner ut hva som blir breddene på treet.
% Definerer størrelsene Imax og Jmax som bestemmer utseendet på treet.

global Jmax Imax % Gjøre verdiene globale slik at de kan benyttes i funksjoner.

Imax = round(totaltid/periodetid) + 1;

jmax_temp = ceil(0.184/kappa)
Jmax =min(jmax_temp,Imax-1);

Jmin = -Jmax;
%-----
%Lager matrisene som behøves i programmet

P = zeros(f_j(Jmin) + 1,(Imax-1),3); % Matrise for overgangspriser (overgangssannsynligheter)
Q = zeros(f_j(Jmin) + 1,Imax); % Matrise for tilstandspriser (kumulative sannsynligheter)
X = zeros(f_j(Jmin) + 1,Imax); % Matrise for stokastisk priselement
F = zeros(f_j(Jmin) + 1,Imax); % Matrise for forwardprisene
A =zeros(1,Imax); % Midlertidig skal leses inn fra bruker
%-----
% Regner ut verdiene i P-matrisen.

for i=1:(Imax-1)
    for j= max((-i+1),Jmin):min((i-1),Jmax)

        if (j == Jmax)&(Jmax ~= Imax-1)
            temp_u = 1;
        else
            temp_u = 0;
        end
    end
end
```

Hovedoppgave i investering, finansiering og økonomistyring
NTNU Norges Tekniske-Naturvitenskaplige Universitet
Våren 2002

```
if (j == Jmin)&(Jmax ~= Imax-1)
    temp_d = 1;
else
    temp_d = 0;
end

P(f_j(j),i,1:3)=temp_u*p_1(j) + (1-temp_u -temp_d)*p_2(j) +temp_d*p_3(j);

end;
end;
%-----
% Regner ut verdiene i X-matrisen.
for i=1:Imax

    for j=max((-i+1),Jmin):min((i-1),Jmax)
        if(i==1)
            X(f_j(j),i) = j*delta_X(1,1);
        end
        X(f_j(j),i) = j*delta_X(1,i);
    end

end
end
%-----
% Tilstandsprisene regnes ut uten å bli diskontert med den risikofri renten.
% Diskontering gjøres ved å manipulere Q-matrisen.
%-----
% Regner ut tilstandsprisen i hver node.

Q(f_j(0),1) = 1; % Setter verdien på derivatet i startnoden lik 1

i = 2; % Starter å regne ut verdien for nodene med indeks i = 1,
% indekseres med 2 fordi matriser ikke godtar 0,1,2...totaltid som indeks.

while i <= Imax;

% Bruker min og max funksjoner for å begrense (j) når (i) overgår Jmax.

    j = max((-i+1),Jmin);

    while j <= min((i-1),Jmax)
%-----
% Node [max((-i+1),Jmin)] <= j <= [min((i-1),Jmax)] | i < Jmax + 2
% Noder s

% Beregning av tilstandspriser i et begrenset tre ved bruk av vanlig overganger,
% dvs forgrening etter figur 21.7 (a) i Hull (1993). Beregninger som skjer før det
% brukes overgangssannsynligheter beregnet etter figur 21.7(b).

        if(Jmax ~= Imax-1)&(i < Jmax + 2)
            q1_1 = zeros(1,3);
            for k = -1:1
% Linje 1 : Bidrag fra node ovenfor, samme nivå og ovenfor.

                q1_1(1,k+2) = Q(f_j(j + k),i-1)*P(f_j(j + k),i-1,2 + k);

            end

            Q(f_j(j),i) = sum(q1_1);

        end
%-----
% Node j = (Jmin|Jmax > 1)
% Nederste node i et begrenset tre.

        if (j == Jmin)&(Jmax ~= Imax-1)&(i >= Jmax+2)&(Jmax > 1)

% Tar IKKE hensyn til at når Jmin = -1 kan noden nåes fra Jmax - 2 = Jmin.
% Dette gjøres lengre nede i algoritmen.

            q2_1 = zeros(1,3);
            for k=0:1
% Linje 1: Bidrag fra node på samme nivå og ovenfor.

                q2_1(1,k+1) = Q(f_j(j + k),i-1)*P(f_j(j + k),i-1,3);

            end

            Q(f_j(j),i) = sum(q2_1);

        end
%-----
% Node j = (Jmin + 1|Jmax > 1)
% Nest nederste noden i et begrenset tre.

        if (j == Jmin + 1)&(Jmax ~= Imax-1)&(i >= Jmax+2)&(Jmax > 1)

% Tar ikke hensyn til at dette kan være midt noden når Jmax==1.
% Det gjøres lengre nede i algoritmen.

            q3_1 = zeros(1,3);

            for k=0:1
% Linje 1: Bidrag fra node på sammen nivå og noden ovenfor.
% Linje 2: Bidrag fra noden nedenfor, blir lagt til når k==1.

                q3_1(1,k+1) = Q(f_j(j + k),i-1)*P(f_j(j + k),i-1,(2 + k))...
                    + Q(f_j(j - 1),i-1)*P(f_j(j - 1),i-1,2)*k;

            end

            Q(f_j(j),i) = sum(q3_1);

        end

    end

end
```

Hovedoppgave i investering, finansiering og økonomistyring
NTNU Norges Tekniske-Naturvitenskaplige Universitet
Våren 2002

```
-----
% Node j = (Jmin + 2|Jmax > 2)
% Tredje nederste node.

    if (j == Jmin + 2) & (Jmax ~= Imax-1) & (i >= Jmax+2) & (Jmax > 2)

% Tar hensyn ikke hensyn til at noden kan nå til midt noden i de tilfeller
% når Jmax==2. Det taes heller ikke hensyn tilfellet Jmax==1, da vil denne noden
% kommunisere med øvre rand node. Dette gjøres lengere ned i algoritmen.

        q4_1 = zeros(1,3);

        for k=-1:1
% Linje 1: Bidrag fra node nedenfor, på samme nivå og ovenfor.
% Linje 2: Bidrag fra noden 2 nedenfor, dvs rand noden.
            q4_1(1,k+2) = Q(f_j(j+k), i-1)*P(f_j(j+k), i-1, 2+k)...
                +Q(f_j(j-2), i-1)*P(f_j(j-2), i-1, 1)*(1+k)*(1-k);

        end
        Q(f_j(j), i) = sum(q4_1);
    end

-----
% Node [max((-i+1), Jmin)] <= j <= [min((i-1), Jmax)] | [abs(j)~=Jmax|abs(j)~=Jmax-1|abs(j)~=Jmax-2]
% Dette nodene som ikke kommuniserer med kant nodene.

% Beregning av tilstandspriser i et begrenset tre ved bruk av vanlig overganger,
% dvs forgrening etter figur 21.7 (a) i Hull (1993).

    if (Jmax ~= Imax-1) & (i >= Jmax+2) & (Q(f_j(j), i) == 0) & (abs(j)~=Jmax) & (abs(j)~=Jmax-1) & (abs(j)~=Jmax-2)

        q5_1 = zeros(1,3);

        for k = -1:1
% Linje 1 : Bidrag fra node ovenfor, samme nivå og ovenfor. Beregnes ikke
% om vi er i de tre øverste eller nederste nodene i ett begrenset tre.

            q5_1(1,k+2) = Q(f_j(j+k), i-1)*P(f_j(j+k), i-1, 2+k);

        end

        Q(f_j(j), i) = sum(q5_1);
    end

-----
% Node j = Jmax - 2

% Tredje øverste node når treet begrenses. Denne noden kan også være nederst noden
% i treet når Jmax==1.

    if (j == Jmax - 2) & (Jmax ~= Imax-1) & (i >= Jmax+2)

% Tar hensyn til at denne noden kan være nedre node i treet når Jmax ==1.
% Med Jmax==2 vil dette være midtre node, og denne beregnes også her.

        q6_1 = zeros(1,3);

        if (Jmax > 2) % Innbærer ikke noe spesielt.
            for k=-1:1
% Linje 1: Bidrag fra node nedenfor, samme nivå og ovenfor som vanlig.
% Linje 2: Bidrag fra noden to ovenfor blir lagt til når k==0.
                q6_1(1,k+2) = Q(f_j(j+k), i-1)*P(f_j(j+k), i-1, 2+k)...
                    +Q(f_j(j+2), i-1)*P(f_j(j+2), i-1, 3)*(1+k)*(1-k);
            end
        end

        if (Jmax == 2) % Innbærer at denne noden er midtre node treet
            for k=-1:1
% Linje 1: Bidrag fra node nedenfor, samme nivå og ovenfor som vanlig.
% Linje 2: Bidrag fra noden to ovenfor blir lagt til når k==0.
% Linje 2: Bidrag fra noden to nedenfor blir lagt til når k==0.
                q6_1(1,k+2) = Q(f_j(j+k), i-1)*P(f_j(j+k), i-1, 2+k)...
                    +Q(f_j(j+2), i-1)*P(f_j(j+2), i-1, 3)*(1+k)*(1-k)...
                    +Q(f_j(j-2), i-1)*P(f_j(j-2), i-1, 1)*(1+k)*(1-k);
            end
        end

        if (Jmax == 1) % Innbærer at denne noden er nedre node i treet
            for k=0:1
% Linje 1: Bidrag fra nedre og midt node.
% Linje 2: Bidrag fra øvre rand node, blir lagt til når k==1.
                q6_1(1,k+1) = Q(f_j(j+k), i-1)*P(f_j(j+k), i-1, 3)...
                    +Q(f_j(j+2), i-1)*P(f_j(j+2), i-1, 3)*k;
            end
        end

        Q(f_j(j), i) = sum(q6_1);
    end

-----
% Node j = Jmax - 1
% Nest øverste node når treet begrenses.

    if (j == Jmax - 1) & (Jmax ~= Imax-1) & (i >= Jmax+2)

% Tar hensyn til at når Jmax = 1 kan noden nåes fra Jmax - 2 = Jmin.
% Dette innebære at denne noden er mitt i treet i de tilfeller når Jmax==1.

        q7_1 = zeros(1,3);

        if (Jmax == 1) % Innbære at denne noden er midt i treet.

            for k=-1:1
```

Hovedoppgave i investering, finansiering og økonomistyring
NTNU Norges Tekniske-Naturvitenskapelige Universitet
Våren 2002

```
% Linje 1: Bidrag fra noden på ovenfor, sammen nivå og nedenfor.
    q7_1(1,k+2) = Q(f_j(j+k),i-1)*P(f_j(j+k),i-1,2);
    end
    end
    if (Jmax > 1) % Innebære ikke noe spesielt.
        for k=-1:0
% Linje 1: Bidrag fra noden på sammen nivå og noden nedenfor.
% Linje 2: Bidrag fra øvre rand node.
            q7_1(1,k+2) = Q(f_j(j+k),i-1)*P(f_j(j+k),i-1,2+k)...
                +Q(f_j(j+1),i-1)*P(f_j(j+1),i-1,2)*(-k);
            end
        end
        Q(f_j(j),i)=sum(q7_1);
    end
-----
% Node j = Jmax
% Øverste node når treet begrenses.

    if (j == Jmax)&(Jmax ~= Imax-1)&(i >= Jmax+2)

% Tar hensyn til at når Jmax = 1 kan noden nåes fra Jmax - 2 = Jmin..
        q8_1 = zeros(1,3);
        if (Jmax==1)
            for k= -1:0
% Linje 1 : Bidrag fra øvre rand node og nest øverst node ved tidspunkt.
% Linje 2 : Bidrag fra node med indeks Jmin, gjelder når Jmax=1.
                q8_1(1,k+2) = Q(f_j(j+k),i-1)*P(f_j(j+k),i-1,1)...
                    + Q(f_j(j-2),i-1)*P(f_j(j-2),i-1,1)*(-k);
            end
        end
        if (Jmax > 1)
            for k= -1:0
% Linje 1 : Bidrag fra øvre rand node og nest øverst node.
                q8_1(1,k+2) = Q(f_j(j+k),i-1)*P(f_j(j+k),i-1,1);
            end
        end
        end
        Q(f_j(j),i)=sum(q8_1);
    end
-----
% Beregning av tilstandspriser i vanlig tre ved bruk av vanlig overganger,
% dvs forgrening etter figur 21.7 (a) i Hull (1993).

    if(Jmax == Imax-1)
        q9_1 = zeros(1,3);
        for k = -1:1
            q9_1(1,k+2)= Q(f_j(j+k),i-1)*P((f_j(j+k)),i-1,2+k);
        end
        Q(f_j(j),i)= sum(q9_1);
    end

    j = j + 1;
    end

    i = i + 1;
end
-----
% Diskontering av tilstandsprisen som er beregnet i Q-matrisen.
% Dette gjøres med å multiplisere hver kolonne med diskonteringsfaktoren opphøyde i
% antall tidssteg. (Diskonteringsfaktor < 1).
for k =1:(Imax-1)
    Q(:,k+1)= Q(:,k+1).*diskont^k;
end
-----
%Beregner verdien av korreksjonsleddene

Am = zeros(1,Imax); % Korrigerede verdier.
Am(1,1) = log(startpris(1,1));
Qsum = sum(Q.*exp(X));
for i=2:Imax
    temp_2 = startpris(1,i-1)*diskont;
    Am(1,i)=log(temp_2/Qsum(1,i));
end

temp=zeros(f_j(Jmin)+1,Imax);

for i =1:Imax
    for j = max(-i+1,Jmin):min((i-1),Jmax)
        temp(f_j(j),i) = exp(X(f_j(j),i) + Am(1,i));
    end
end
end
```

Hovedoppgave i investering, finansiering og økonomistyring
NTNU Norges Tekniske-Naturvitenskaplige Universitet
Våren 2002

```
F = F +temp;
F
d{1,1}=P;
d{1,2}=F;

%-----
% Intern funksjon for å håndtere indekser. Denne må lages fordi Matlab ikke kan bruke
% negative verdier i matrise indekseringen indeksering.

function sub_1 =f_j(j)

global Jmax;

temp = Jmax + 2 - j;

sub_1= temp;
%-----

% Intern subfunksjon for å regne overgangssannsynlighet om forgrening skjer
% i henhold til figur 21.7-(c) Hull (1993).
% Denne brukes når vi er i nodene (i,Jmax|jmax_temp < totaltid), dvs at ve beveger oss langs den
% øvre kanten i treet.

function sub_2 =p_1(j);

global kappa delta_T

% Sannsynlighet for å gå rett fram, dvs. fra node (i,j) til (i+1,j)
pu = 7/6 +((kappa*j*delta_T)^2 -3*kappa*j*delta_T)/2;

% Sannsynlighet for å gå ett steg ned, dvs. fra node (i,j) til (i+1,j-1)
pm = -1/3 - (kappa*j*delta_T)^2 + 2*kappa*j*delta_T;

% Sannsynlighet for å gå to steg ned, dvs. fra node (i,j) til (i+1,j-2)
pd = 1- pu -pm;

sub_2 =[ pu pm pd];
%-----

% Intern subfunksjon for å regne overgangssannsynlighet om forgrening skjer
% i henhold til figur 21.7-(a) Hull (1993). Denne brukes når vi er i nodene (i,j|Jmin < j < Jmax)

function sub_3 =p_2(j);

global kappa delta_T

% Sannsynlighet for å gå opp, dvs. fra node (i,j) til (i+1,j+1)
pu = 1/6 +((kappa*j*delta_T)^2 -kappa*j*delta_T)/2;

% Sannsynlighet for å gå rett fram, dvs. fra node (i,j) til (i+1,j)
pm = 2/3 - (kappa*j*delta_T)^2 ;

% Sannsynlighet for å gå rett fram, dvs. fra node (i,j) til (i+1,j-2)
pd = 1- pu -pm;;

sub_3 =[ pu pm pd];
%-----

% Intern subfunksjon for å regne overgangssannsynlighet om forgrening skjer
% i henhold til figur 21.7-(b) Hull (1993). Denne brukes når vi er i nodene (i,Jmin|jmax_temp < totaltid)

function sub_4 =p_3(j);

global kappa delta_T

% Sannsynlighet for å gå opp, dvs. fra node (i,j) til (i+1,j+2)
pu = 1/6 +((kappa*j*delta_T)^2 + kappa*j*delta_T)/2;

% Sannsynlighet for å gå rett fram, dvs. fra node (i,j) til (i+1,j+1)
pm = -1/3 - (kappa*j*delta_T)^2 -2*kappa*j*delta_T ;

% Sannsynlighet for å gå rett fram, dvs. fra node (i,j) til (i+1,j)
pd = 1- pu -pm;

sub_4 =[ pu pm pd];
%-----
```


Funksjon for å lage overgangsmatrise i prismodell

Inndata er overgangssannsynlighets matrisen fra det trinomiske treet.

```
function ut=Overgangm(inndata)
% Korreksjon av pristre slik at dette kan brukes i en interpolasjons algoritme.
% Korreksjon av pristre slik at dette kan brukes i en interpolasjons algoritme.

RC      = size(inndata);

n_row   = RC(1,1);
n_tid   = RC(1,2);

Dinn    =(n_row-3)/2 + 1;
% Kutter vekk øvre og nedre rad i matrisen.

inndata = inndata(2:(n_row-1),:,:);
n_row   = n_row-2;
% Stokker om dataene slik at de havner på vanlig form brukt innen markovmodeller.

Pt_pris = zeros(n_row,n_row,n_tid);

for t =1:n_tid
    for j=1:n_row
        for i=1:3

            k= -i;
            if(i==2)
                k=0;
            end
            if(i==3)
                k=1;
            end

            if(j==1)
                k=i;
                Pt_pris(j,(j + k - 1) ,t) = inndata(j,t,i);
            end
            if(j==n_row)
                k= -1;
                Pt_pris(j,(j + (k - 3) ) ,t) = inndata(j,t,i);
            end

            if(j~=1)&(j~=n_row)
                Pt_pris(j,(j + k),t) = inndata(j,t,i);
            end

        end
    end
end

ut =Pt_pris;
```

Funksjon for korreksjon av pristre

Inndata er pristreet fra trinomiske prismodellen.

```
function ut=Korreksjon tre(inndata)
% Korreksjon av pristre slik at dette kan brukes i en interpolasjons algoritme.
% Korreksjon av pristre slik at dette kan brukes i en interpolasjons algoritme.

RC      = size(inndata);

n_row   = RC(1,1);
n_col   = RC(1,2);

Dinn    =(n_row-3)/2 + 1;
% Kutter vekk øvre og nedre rad i matrisen.

inndata = inndata(2:(n_row-1),:,:);
% Finner verdiene som blir øvre og nedre pris i treet.

max_ovre = max(inndata(1,:));

min_nedre= min(inndata(end,Dinn:end));
% Erstatter verdiene i øvre og nedre rad i treet.

inndata(1,Dinn:end)   = max_ovre;
inndata(end,Dinn:end) = min_nedre;

ut = inndata;
```

Vedlegg nr 7: Kode implementering av SEFas prismodell for tilsig

Funksjon som legger til ekstrem scenarier i overgangssannsynlighetene

```
function ut = E_Overgang_Pt(inndata)
% Funksjon legger til ekstrem scenarier overgangssannsynlighet.

m_dim = size(inndata);
n_row = m_dim(1,1);
n_col = m_dim(1,2);
n_tid = m_dim(1,3);
side_m = zeros(n_row,1,n_tid);
midt = [side_m inndata side_m];
side_tb = zeros(1,2,52);
topp = [inndata(1, :, :) side_tb];
bunn = [side_tb inndata(n_row, :, :)];
ut = abs([topp; midt; bunn]);
```

Funksjon som legger til ekstremscenarier verdier i tilsigspunkt

```
function ut = E_tilsig(inndata,ovre,nedre)
% Funksjon legger til ekstrem scenarier

m_dim = size(inndata);
n_row = m_dim(1,1);
n_col = m_dim(1,2);
ovre_vek = repmat(ovre,1,n_col);
nedre_vek = repmat(nedre,1,n_col);

% Avhensyn til interpolasjon må verdiene være stigende oppover i kolonnen.
% Vet at i inndatamatriksen i enkelte tilfeller har null i nederste
% rad. Denne verdien sette lik 1.2*nedre. Dette er ikke statistisk korrekt,
% men en praktisk løsning på problemet.

for i =1:n_col
    if(inndata(n_row,i)<nedre)
        inndata(n_row,i)=nedre*1.2
    end
end

ut = [ovre_vek;inndata;nedre_vek];
```

Funksjon som finner betinget forventingsverdi i hvert tilsigband

```
function ut = F_Everdi(inndata,n)

% Inndata forutsettes å være en matrise med like mange rader som det finnes tilsigsobservasjoner
% for, dvs. 72 år her. Antall kolonner er avhengig om det er oppløsningen på datasettet, her
% er det ukens oppløsning og dermed 52 kolonner.
% n      : Antall prispunkter
%-----
% Definerer forskjellige verdier

n_tp    = n;                % Antall tilsigspunkter i markovmodellen
n_row   = length(inndata(:,1)); % Antall rader (år) med tilsigsdata.
n_col   = length(inndata(1,:)); % Antall kolonner med tilsigsdata, her 52 når det er ukessoppløsning.
%-----
% Finner områdene størrelsen på tilsigsbandene

inndata =log(inndata);
maxverd = repmat(max(max(inndata)),1,n_col);
minverd = repmat(min(min(inndata)),1,n_col);
omrade_mat =zeros(n_tp+1,n_col);

    for i =1:n_col
        omrade_mat(:,i)=rot90(linspace(minverd(1,i),maxverd(1,i),n_tp+1)',2);
    end

inndata =exp(inndata);
omrade_mat=exp(omrade_mat);
%-----
% Finner ut hvilke verdier som det skal regnes gjennomsnitt i periode (t+1) og
% samtidig har tilhørt tilsigspunkt (i) i periode t.

tell_mat1 = zeros(n_row,n_tp,n_col);

for t =1:n_col
    for i = 1:n_tp

        T_1 = t-1;

        if (t==1)
            T_1 = 52;
        end

        u_bound_t_1 = repmat(omrade_mat(i,T_1),n_row,1);
        l_bound_t_1 = repmat(omrade_mat(i+1,T_1),n_row,1);

        temp = (inndata(:,T_1) >= l_bound_t_1) & (u_bound_t_1 >= inndata(:,T_1));

        tell_mat1(:,i,t)=temp;

    end
end

%-----
% Endrer dataformat til en matrise med dimensjon (tilsigspunkt*tidsoppløsning)=(n_tp X n_col).

temp_antall = sum(tell_mat1);
antall_tband = zeros(n_tp,n_col);

for t =1:n_col
    antall_tband(:,t) =temp_antall(1,:,t)';
end

%-----
% Setter inn 1 for verdier = 0, for å unngå deling på null.
for t= 1:n_col
    for n = 1:n_tp
        if (antall_tband(n,t)==0)
            antall_tband(n,t)=1;
        end
    end
end

%-----
% Regner ut middelverdien i hvert tilsigband.
sum_tband =zeros(n_tp,n_col);

for t =1:n_col
    for n =1:n_tp

        sum_tband(n,t) = sum(inndata(:,t).*tell_mat1(:,n,t));

    end
end
E_verdi = sum_tband./antall_tband;
%-----
ut=E_verdi;
```

Funksjon som finner antall observasjoner i hvert tilsigsband

```
function ut = F_Nverdi(inndata,n)

% Inndata forutsettes å være en matrise med like mange rader som det finnes tilsigsobservasjoner
% for, dvs. 72 år her. Antall kolonner er avhengig hvilken oppløsningen datasettet har, her
% er det ukens oppløsning og dermed 52 kolonner.
%-----
% Definerer forskjellige verdier

n_tp = n; % Antall tilsigspunkter i markovmodellen

n_row = length(inndata(:,1)); % Antall rader (år) med tilsigsdata.

n_col = length(inndata(1,:)); % Antall kolonner med tilsigsdata, her 52 når det er ukensoppløsning.
%-----
% Finner områdene størrelsen på tilsigsbandene.

inndata =log(inndata);

maxverd = repmat(max(max(inndata)),1,n_col);

minverd = repmat(min(min(inndata)),1,n_col);

omrade_mat =zeros(n_tp+1,n_col);

for i =1:n_col

    omrade_mat(:,i)=rot90(linspace(minverd(1,i),maxverd(1,i),n_tp+1)',2);

end

inndata =exp(inndata);

omrade_mat=exp(omrade_mat);

%-----
% Opptelling av antall observasjoner i hvert tilsigsband.

tell_mat1 = zeros(n_row,n_tp,n_col);

for t =1:n_col
    for i = 1:n_tp

        u_bound = repmat(omrade_mat(i,t),n_row,1);
        l_bound = repmat(omrade_mat(i+1,t),n_row,1);

        temp = (inndata(:,t) >l_bound)&(u_bound >= inndata(:,t));

    tell_mat1(:,i,t)=temp;
    end
end
%-----
% Endrer dataformat til en matrise med dimensjon (tilsigspunkt*tidsoppløsning)=(n_tp X n_col).

temp_antall = sum(tell_mat1);
antall_tband = zeros(n_tp,n_col);

for t =1:n_col
    antall_tband(:,t) =temp_antall(1,:,t)';
end
%-----
ut=antall_tband;
```

Funksjon som korrigerer overgangssannsynligheter

```
function d=Korreksjon_Pt(Tilsigp_i,Nverdi_i,Pt_i,Everdi_i,lamda_i)
% Script skrevet for å korrigere overgangssannsynligheten som er funnet med opptelling.
% MERR! endingen " i" er bare brukt for å unngå advarsler fra Matlab pga. av globale variable.
%-----
% Tilsigp: Matrise med dim (Antall tilsigspunkt X Oppløsning) =(5 X 52)
% Nverdi: Matrise som inneholder antall tilsigspunkt innenfor hvert tilsigsband.
% Pt: Ukorrigererte markov overgangssannsynligheter, dim =(5 x 5 x 52).
% Everdi : Betinget middelerdi i tilsigsband (j) gitt overgang (i).
% Lamda: Vektfaktor, her brukes 0.01(samme som i prismodell).
%-----
% Merk at programmet er skrevet slik at det kan brukes for et vilkårlig antall prispunkt og
% perioder. Dimensjonene som er angitt på matrisen gjelder derfor kun i denne hovedoppgaven.
%
% Merk også at programmet forutsetter at fmincon bruker medium-scale algoritmen.
%-----
global Tilsigp Nverdi Pt Everdi lamda n_row n_col tid

Tilsigp = Tilsigp_i;
Nverdi = Nverdi_i;
Pt = Pt_i;
Everdi = Everdi_i;
lamda = lamda_i;

RC = size(Tilsigp); % Leser inn størrelsen på matrisen, gitt ved (rekker x kolonner).

n_row = RC(1); % Definerer variabelen for antall rekker(og tilsigspunkt), variabelen er global.

n_col = RC(2); % Definerer variabelen for antall kolonner, variabelen er global.

tid = 1; % Definerer en global uke teller.
%-----
% Starter while sløyfen og korrigerer overgangssannsynlighetene i alle 52 ukene
```

Hovedoppgave i investering, finansiering og økonomistyring
NTNU Norges Tekniske-Naturvitenskaplige Universitet
Våren 2002

```
while(tid <=52)
%*****
% Definerer forskjellige variable som inngår i restriksjoner.

Ni = Nverdi(:,tid);

if(tid~=52)
    Nj = Nverdi(:,tid +1);
else
    Nj = Nverdi(:,1);
end

%*****
%Restriksjoner i minimeringsfunksjonen.

lb = repmat(0,n_row^2,1); % Nedre grense for overgangssannsynligheten.
ub = repmat(1,n_row^2,1); % Øvre grense for overgangssannsynligheten.
Aeq = zeros(2*n_row,n_row^2); % Definerer matrisen som brukes for likheter
T = tid; % Setter tiden lik en annen variabel for å ikke
% endre den globale verdien.

% Leser inn restriksjon(2) i gitt i dokumentasjon av prismodell.
for i =1:n_row
    if(i==1)
        for j=1:n_row;
            Aeq(i,j)=1;
        end
    else
        for j=((i-1)*n_row +1):i*n_row;
            Aeq(i,j)=1;
        end
    end
end

% Leser inn restriksjon (3) gitt i dokumentasjon av prismodell.
i_mat =zeros(n_row,n_row); % Indeks matrise

for i=0:(n_row-1)
    k=(1+i):n_row:((n_row-1)*n_row + 1 + i);
    i_mat(i+1,:)= k;
end

for i = (n_row+1):2*n_row
    for j=1:n_row
        Aeq(i,i_mat(i-n_row,j))= Ni(j,1);
    end
end
% Leser inn høyre siden på restriksjon (3) gitt i dokumentasjon av prismodell.

beq =zeros(2*n_row,1);

beq(1:n_row,1)=1; % Summen av overgangssannsynlighetene skal være lik 1, fra rest. (2).

beq((n_row+1):2*n_row,1) = Nj(:,1);% Høyre siden i restriksjon (3).
%*****
%Lager startverdien til funksjonen.

x_0 =zeros(1,n_row^2);

for i =1:5
    x_0(1,((i-1)*n_row +1):n_row*i)= Pt(i,:,tid) ;
end

default = optimset('fmincon');
options = optimset(default, 'MaxFunEvals',100*n_row^2, 'MaxIter',400, 'TolFun',1.0000e-6, 'TolX',1.0000e-6, ...
    'TolCon',1e-2, 'LargeScale','off');%,'Display','iter','Diagnostics','on');

[x] = fmincon(@f_1,x_0,[],[],Aeq,beq,lb,ub,[],options);

if(tid==1)
X = zeros(n_row,n_row,tid);
end

for i=1:n_row
    for j=1:n_row
        if(i==1)
            k=j;
        else
            k = (i-1)*5 + j;
        end
        X(i,j,tid) = x(k);
    end
end

d = X;

%*****
% Teller for antall runder i while sløyfa og end ordre for sløyfen.
tid = tid + 1;
end

%*****
%Intern subfunksjon som brukes i minimeringen
```

```
function sub_1 =f_1(x)

% Må definere variablene som globale for at de skal kunne brukes inn i funksjonen.

global Tilsigp Pt Everdi lamda n_row tid

imax = n_row;
jmax = imax;

T = tid;
T_1 = tid + 1;

if (T==52)
    T_1=1;
end

Tp = Tilsigp(1:imax,T);
Ei = Everdi(1:imax,T_1);
Pij = Pt(1:imax,1:jmax,T);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Regner ut første delsummen i optimerings funksjonen.

pij_Pj =zeros(imax,jmax);

for i=1:imax
    for j=1:jmax
        if (i==1)
            k=j;
        else
            k = (i-1)*5 + j;
        end
        pij_Pj(i,j) = x(k)*Tp(j,1);
    end
end
pij_Pj = sum(pij_Pj)';

S1_i = zeros(imax,1);

for i =1:imax
    S1_i(i,1) = (pij_Pj(i,1)-Ei(i,1))^2;
end

S1 = sum(S1_i);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Regner ut andre delsummen i optimerings funksjonen.

delta_pij =zeros(imax,jmax);

for i=1:imax
    for j=1:jmax
        if (i==1)
            k=j;
        else
            k = (i-1)*5 + j;
        end
        delta_pij(i,j) = (x(k)-Pij(i,j))^2;
    end
end

S2 = lamda*sum(sum(delta_pij));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sub_1 = S1 +S2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Funksjon som beregner overgangsannsynlighetsmatrisen

```
function ut = Overgang_Pt(inndata,n)

% Inndata forutsettes å være en matrise med like mange rader som det finnes tilsigsobservasjoner
% for, dvs. 72 år her. Antall kolonner er avhengig om det er oppløsningen på datasettet, her
% er det ukas oppløsning og dermed 52 kolonner.
% n : Antall prispunkter
%-----
% Definerer forskjellige verdier

n_tp = n; % Antall tilsigspunkter i markovmodellen

n_row = length(inndata(:,1)); % Antall rader (år) med tilsigsdata.

n_col = length(inndata(1,:)); % Antall kolonner med tilsigsdata, her 52 når det er ukasoppløsning.
%-----
% Finner områdene størrelsen på tilsigsbandene

inndata = log(inndata);

maxverd=repmat(max(max(inndata)),1,n_col);
minverd=repmat(min(min(inndata)),1,n_col);
omrade_mat =zeros(n_tp+1,n_col);

for i =1:n_col
    omrade_mat(:,i)=rot90(linspace(minverd(1,i),maxverd(1,i),n_tp+1)',2);
end
```

```
end
%-----
% Finner overgangssannsynligheten for hver uke basert på opptelling.

tell_mat1 = zeros(n_tp,n_tp,n_col);

    for k = 1:n_tp
        for i = 1:n_tp
            for t =1:n_col

% For å kunne regne overgangssannsynligheter for uke 1 må data fra uke 52 brukes,
% for å kunne gjøre det må tidindekseringen manipuleres litt.
T_1 = t-1;

if(t==1)
    T_1 = 52;
end

        u_bound_t_1 = repmat(omrade_mat(k,T_1),n_row,1);
        l_bound_t_1 = repmat(omrade_mat(k+1,T_1),n_row,1);

        u_bound_t = repmat(omrade_mat(i,t),n_row,1);
        l_bound_t = repmat(omrade_mat(i+1,t),n_row,1);

        temp = (inndata(:,t) >=l_bound_t)&(u_bound_t >= inndata(:,t))&...
                (inndata(:,T_1) >= l_bound_t_1)&(u_bound_t_1 >= inndata(:,T_1));

tell_mat1(k,i,t) = sum(temp);
            end
        end
    end
%-----
% Regner ut overgangssannsynligheten med å dele hvert element i raden på antallet overganger totalt
% til denne raden.
for i =1:52
temp = sum(tell_mat1(:, :,i)');
% Må lage en sløyfe som hindrer at det blir delt på null.
for j =1:n_tp
if(temp(j,1)==0)
temp(j,1)=1;
end
end

tell_mat1_temp(:, :,i) = tell_mat1(:, :,i)./repmat(temp,1,n_tp);
end

Pt =tell_mat1_temp;
%-----
ut=Pt;
```

Funksjon som finne verdien på tilsigspunktene

```
function ut = Tilsigspunkt(inndata,n)

% Inndata forutsettes å være en matrise med like mange rader som det finnes tilsigsobservasjoner
% for, dvs. 72 år her. Antall kolonner er avhengig hvilken oppløsningen datasettet har, her
% er det ukens oppløsning og dermed 52 kolonner.
%-----
% Definerer forskjellige verdier

n_tp = n; % Antall tilsigspunkter i markovmodellen

n_row = length(inndata(:,1)); % Antall rader (år) med tilsigsdata.

n_col = length(inndata(1,:)); % Antall kolonner med tilsigsdata, her 52 når det er ukessoppløsning.
%-----
% Finner områdene størrelsen på tilsigbandene.

inndata =log(inndata);

maxverd = repmat(max(max(inndata)),1,n_col);
minverd = repmat(min(min(inndata)),1,n_col);

omrade_mat =zeros(n_tp+1,n_col);

for i =1:n_col

    omrade_mat(:,i)=rot90(linspace(minverd(1,i),maxverd(1,i),n_tp+1)',2);
end

inndata =exp(inndata);

omrade_mat=exp(omrade_mat);

%-----
% BEREGNING AV MIDDELVERDIEN I HVERT PRISBAND

tell_mat1 = zeros(n_row,n_tp,n_col);

for t =1:n_col
    for i = 1:n_tp

        u_bound = repmat(omrade_mat(i,t),n_row,1);
        l_bound = repmat(omrade_mat(i+1,t),n_row,1);
```

```
temp = (inndata(:,t) > l_bound) & (u_bound >= inndata(:,t));

tell_matl(:,i,t)=temp;
end
end
%-----
% Endrer dataformat til en matrise med dimensjon (tilsigs punkt*tidsoppløsning)=(n_tp X n_col).

temp_antall = sum(tell_matl);
antall_tband = zeros(n_tp,n_col);

for t = 1:n_col
    antall_tband(:,t) = temp_antall(1,:,t)';
end
%-----
% Setter inn 1 for verdier = 0, for å unngå deling på null.
for t = 1:n_col
    for n = 1:n_tp
        if(antall_tband(n,t)==0)
            antall_tband(n,t)=1;
        end
    end
end
end
%-----
% Regner ut middeler verdien i hvert tilsigband.
sum_tband = zeros(n_tp,n_col);

for t = 1:n_col
    for n = 1:n_tp

        sum_tband(n,t) = sum(inndata(:,t).*tell_matl(:,n,t));

    end
end
mean_tband = sum_tband./antall_tband;
%-----
ut=mean_tband;
```

Funksjon som fjerner ekstremverdier

```
function ut = Trimfunk(inndata)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funksjon som trimmer vekk verdiene som ligger over 99% kvantilen og
% under 1% kvantilen.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

imax = length(inndata(:,1));
jmax = length(inndata(1,:));
Antall= imax*jmax;

min_ver =min(min(inndata)); % Finner minste verdi i matrisen.
max_ver =max(max(inndata)); % Finner minste verdi i matrisen.

% Setter start verdien på min_pr, denne vil alltid være lik (1/Antall)*100%.
min_pr = (-sum(sum( inndata > min_ver) ) + Antall)*(100/Antall);
max_pr = sum(sum( inndata > max_ver))* (100/Antall);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Kjøren sløyfer for å finne hvilken grenseverdiene.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while(min_pr < 1)
min_ver = min_ver + 0.01;
min_pr = (-sum(sum( inndata > min_ver) ) + Antall)*(100/Antall);
end

while(max_pr < 1)
max_ver = max_ver - 0.01;
max_pr = sum(sum( inndata > max_ver))* (100/Antall);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bytter alle verdier som er større eller mindre enn grenseverdien.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:imax
    for j=1:jmax

        if(inndata(i,j) < min_ver)
            inndata(i,j) = min_ver;
        end

        if(inndata(i,j) > max_ver)
            inndata(i,j) = min_ver;
        end

    end
end
end
```



```
ut = inndata;
```

Vedlegg nr 8: Koder for optimering

Funksjon for interpolering av FCF

```
function ut = intermat3D(alfa,x1,y1,z1)
% =====
% x1 : Magasin nivå uke T, tilstand i m
% y1 : Tilsig i uke T, scenario k
% z1 : Pris i uke T, scenario i
% =====

n_row =length(alfa(:,1));          % Finner antall rader i matrisen som mates inn.

% =====
% Justerer verdien slik at de aldri er like maks eller min verdien som er lest inn i alfa.

if(x1==max(alfa(:,2))) dxu = -eps; else dxu = 0; end
if(y1==max(alfa(:,3))) dyu = -eps; else dyu = 0; end
if(z1==max(alfa(:,4))) dzu = -eps; else dzu = 0; end

if(x1==min(alfa(:,2))) dxl = eps; else dxl = 0; end
if(y1==min(alfa(:,3))) dyl = eps; else dyl = 0; end
if(z1==min(alfa(:,4))) dzl = eps; else dzl = 0; end

% if(y1==max(alfa(:,3))) y1 =y1-10; end
% if(z1==max(alfa(:,4))) z1 =z1-10; end
%
% if(x1==min(alfa(:,2))) x1 =x1+1; end
% if(y1==min(alfa(:,3))) y1 =y1+1; end
% if(z1==min(alfa(:,4))) z1 =z1+1; end

% =====
xyz_inn = [x1 y1 z1];              % Lager en vektor av verdien som det skal interpoleres på.

xyz_inn_mat = repmat(xyz_inn,n_row,1); % Lager en matrise som inneholder dataene som det skal interpoleres på.

dxyz = alfa(:,2:end)-xyz_inn_mat;  % Lager en deltamatrikse for å finne hvilke to vektor som skal trekkes ut.

% =====
% Disse vektoren velges slik at x1,y1 og z1 ligger "mellom" disse.

% =====
U_num = 0;                          % Definerer radnummeret som ligger ovenfor inndataen og som skal brukes i interpolasjon.
rdist_U= 1e10;                       % Setter den romlige distansen lik et tilfeldig høyt tall.

% Regner avstanden den romlige avstanden fra all sett med {[x y z] >=[0 0 0] } til [x1 y1 z1].
% Avstanden er den minst observert settes hyp_num = rad nummeret.

for i = 1:n_row
    if (dxyz(i,1) > dxu) & (dxyz(i,2) > dyu) & (dxyz(i,3) > dzu)
        rdist_tempU = (dxyz(i,1)^2 + dxyz(i,2)^2 + dxyz(i,3)^2)^0.5;
        if (rdist_tempU <= rdist_U)
            U_num=i;
            rdist_U=rdist_tempU;
        end
    end
end

% =====
L_num = 0;                          % Definerer radnummeret som ligger nedenfor inndataen og som skal brukes i interpolasjon.
rdist_L= 1e10;                       % Setter den romlige distansen lik et tilfeldig høyt tall.

% Regner avstanden den romlige avstanden fra all sett med {[x y z] >=[0 0 0] } til [x1 y1 z1].
% Avstanden er den minst observert settes hyp_num = rad nummeret.

for i = 1:n_row
    if (dxyz(i,1) < dxl) & (dxyz(i,2) < dyl) & (dxyz(i,3) < dzl)
        rdist_tempL = (dxyz(i,1)^2 + dxyz(i,2)^2 + dxyz(i,3)^2)^0.5;
        if (rdist_tempL <= rdist_L)
            L_num=i;
            rdist_L=rdist_tempL;
        end
    end
end

% =====
% Trekker ut grense verdien som skal brukes i interpolasjon.

x = [alfa(L_num,2); alfa(U_num,2)];
y = [alfa(L_num,3); alfa(U_num,3)];
z = [alfa(L_num,4); alfa(U_num,4)];

[X Y Z] =ndgrid(x,y,z);

% =====
% Må plukke ut funksjonverdien for alle ordnede permutasjoner av x,y,z settet som er funnet.
xyz_temp = zeros(8,3);

x_t      = [1;1;1;2;2;2;2;2];
```

```
y_t = [1;1;2;2;1;1;2;2];
z_t = [1;2;1;2;1;2;1;2];

for i=1:8
    xyz_temp(i,:) = [x(x_t(i,1),1) y(y_t(i,1),1) z(z_t(i,1),1)];
end

f_xyz = [zeros(8,1) xyz_temp];

for i = 1:n_row
    for j = 1:8
        if (f_xyz(j,2) == alfa(i,2)) & (f_xyz(j,3) == alfa(i,3)) & (f_xyz(j,4) == alfa(i,4))
            f_xyz(j,1) = alfa(i,1);
        end
    end
end

% Lager matrisen med funksjonsverdier som tilhører X,Y,Z
F = zeros(2,2,2);

for I = 1:8
    for i = 1:2
        for j = 1:2
            for k = 1:2
                if (f_xyz(I,2) == X(i,j,k)) & (f_xyz(I,3) == Y(i,j,k)) & (f_xyz(I,4) == Z(i,j,k))
                    F(i,j,k) = f_xyz(I,1);
                end
            end
        end
    end
end

ut = interp(X,Y,Z,F,x1,y1,z1,'spline');
```

Funksjon for optimering

```
function ut = SDP2(Pris_i,Opris_i,Tilsig_i,Otilsig_i,m)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program for løsning av driftsplanleggingsproblemet til en vannkraftprodusent.
% Inndata er følgende:
% Pris      : Pris fra trinomisk tre.
% Opris     : Overgangssann. i det trinomiske treet.
% Tilsig    : Tilsig fra markovmodell (Bygger på Sintefs prismodell).
% Otilsig   : Overgangssannsynligheter fra tilsigsmodellen.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Setter inn en del konstanter, blant annet magasinnivå ved starten.
% Bruker enheten km^3 som mål på fylling og tilsig.

global Mstart EekV Vmax Vmin Umax Umin

Mstart = 41300;          % Magasinfylling (km^3) uke 1 1999.
EekV   = 223e-3;       % MWh/km^3: Energi ekvivalent for kraftverket.
Vmax   = 76500;       % Øvre grense for magasinfylling (km^3).
Vmin   = 0;           % Nedre grense (km^3), dvs at vannspeilet har minste tillatte kotehøyde.
Umax   = 10577;       % Maksimalt produksjonsvolum (km^3/uke) pr uke.
Umin   = 0;           % Minste tillatte produksjonsvolum (km^3) pr. uke.
Smin   = 0;           % Setter minste tillatte overløp.

global Pris Opris Tilsig Otilsig

Pris   = Pris_i;       % Gjør prisen global slik denne kan brukes i minimerings funksjonen.
Opris  = Opris_i;     % Gjør også denne global
Tilsig = Tilsig_i*1000; % Omgjør tilsiget slik at enheten er den samme som på øvrige verdier.
Otilsig = Otilsig_i;  % Gjør også denne global

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Regner ut en del konstanter som er gitt av inndataene.

P_RC = size(Pris);
T_RC = size(Tilsig);

global imax jmax tmax kmax lmax mmax

tmax = P_RC(1,2)-1;    % Maksimalt antall tidsskritt.
imax = P_RC(1,1);     % Antall prispunkter.
jmax = imax;          % Antall mulige overganger fra hvert prispunkt (scenarier).
kmax = T_RC(1,1);     % Antall til tilsigspunkt.
lmax = kmax;          % Antall mulige overganger fra hvert tilsigspunkt (scenarier).
mmax = m + 1;         % Antallet diskre utfall for magasinet.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Hovedoppgave i investering, finansiering og økonomistyring
NTNU Norges Tekniske-Naturvitenskaplige Universitet
Våren 2002

```
% Definerer indeks variable som er globale. Disse skal kunne brukes i sub-funksjoner.
global T I K M
T = 1;
I = 1;
K = 1;
M = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definerer diverse matriser.

global Utfalls_mat Inter_mat

Utfallsrom = rot90(linspace(Vmin,Vmax,mmax)',2);

Utfalls_mat = repmat(Utfallsrom,1,52); % Matrise for som gir magasinets utfallsrom.

Inter_mat = zeros(mmax*imax*kmax,7); % Matrise som skal innholde data for interpolasjon.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Må runde starttilstanden av til nærmeste definerte node i utfallsrommet.

m_s_s = 1;
m_dist = 1e9;
for i = 1:mmax
    temp = abs(Utfallsrom(i,1)-Mstart);
    if(temp <= m_dist)
        m_s_s = i;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Leser inn koeffisientene i restriksjonene som ikke er tidsavhengig.

lb = zeros(3,1);
ub = zeros(3,1);
Aeq = zeros(1,3);
A = zeros(1,3);
b = zeros(1,1);

lb(:,1) = [Umin;Smin;Vmin]; % Setter lower bound på variablene.
ub(:,1) = [Umax;2*Umax;Vmax]; % Setter upper bound på variablene.
A(1,:) = [-2 1 0]; % Setter inn konstanten som gjelder for ulikhets restriksjonen.
Aeq(1,:) = [1 1 1]; % Setter inn konstanten som gjelder for likhets restriksjonen.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for t = 51:-1:50
    T=t;
    int teller = 1; % Teller som skal plassere data riktig i matrisen som det skal interpoleres i.
    global Inn_inter
    if(T>=tmax)
        Inn_inter =Inter_mat(:,1:4,T+1);
    end

for m = mmax:-1:1
    M=m;
    for i = imax:-1:1
        I=i;
        for k =kmax:-1:1
            K=k;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Leser inn variabelverdiene inn i interpolasjon matrisen.

Inter_mat(int teller,2,T) = Utfalls_mat(M,T);
Inter_mat(int teller,3,T) = Tilsig(K,T);
Inter_mat(int teller,4,T) = Pris(I,T);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lager restriksjonene de tilstandsavhengige restriksjonene.

beq = zeros(1,1);
beq(1,1) = Utfalls_mat(M,T) + Tilsig(K,T);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funksjon basert på lineær metoder
%
% x0 = [5000 0 Mstart];
% options = optimset('linprog');
% options = optimset(options,'LargeScale','off');
%
% [opt_val,alfa] = linprog(@f_1,A,b,Aeq,beq,lb,ub,x0,options);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funksjon basert på ikke-lineær metoder
if(int teller==1)
x0 = [5000 0 Mstart];
end
options = optimset('fmincon');

options = optimset(options,'MaxIter',40,'TolFun',10,'TolX',10,...
'TolCon',10,'LargeScale','off','Display','iter');%,'Diagnostics','on');

tic
[opt_val,alfa] = fmincon(@f_2,x0,A,b,Aeq,beq,lb,ub,[],options);
tid_iter =toc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Inter_mat(int_teller,1,T) = -(-alfa)^0.5;
Inter_mat(int_teller,5,T) = opt_val(1);
Inter_mat(int_teller,6,T) = opt_val(2);
Inter_mat(int_teller,7,T) = opt_val(3);
x0 =[opt_val(1) opt_val(2) opt_val(3)]

int_teller = int_teller + 1;

end % End funksjon for K teller
end % End funksjon for I teller
end % End funksjon for M teller
T
end % End funksjon for T teller

ut =Inter_mat;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sub funksjon når linprog brukes
function Sub_2 =f_2(x)

global T I K jmax lmax tmax Eekv Pris Opris Otilsig Tilsig Inn_inter

ICF = -Eekv*Pris(I,T)*x(1) +x(3) -x(3);
SCF = 0*x(2);

temp = zeros(jmax,lmax);

if(T~=tmax)
    for J = 1:jmax
        for L = 1:lmax

            temp(J,L) = Opris(I,J,T)*Otilsig(K,L,T+1)*intermat3D(Inn_inter,x(3),Tilsig(L,T+1),Pris(J,T+1));

        end
    end
end

Sub_2 = -(ICF + SCF + sum(sum(temp)))^2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Vedlegg nr 8: Diverse S-Plus funksjoner

Funksjon som finner parameterene i basismodellen

```
# Script som er laget bruke nls()-funksjonen i S plus på en kolonne
# og verdier funnet her brukes som startverdier i optimeringen i MATLAB.
# Inndata er av klasse data.frame
# Tidsvektor er en vektor med numeriske elementer.

# Scriptet brukes til å tilpasse den noe modifiserte modellen fra Lucia og Schwartz.
# Det taes ikke hensyn til tilsig.

funksjon5 <- function(inndata,tid,magasin) # Les inn y-verdier og tidspunktene som funksjonen skal tilpasses
for.
{
    deltat <- c(1,seq(2,7,1),seq(12,52,4),70,88,104) #
    Verdier som legges til tiden t, slik at sesongfunksjon får rette verdien.
    antallkol <- length(inndata)
    # Antallet kolonner i pris matrisen.
    antallrad <- length(inndata[,1]) - 1
    # Antall rader i matrisene som skal lages.
    yt <- inndata[2:(antallrad + 1),] #
    Leser inn y(t) verdien i den midlertidige matrisen.
    yt1 <- inndata[1:antallrad,]
    # Leser inn y(t-1) verdien i likningen i den midlertidige matrisen.

    tid <-
matrix(rep(tid[2:(antallrad+1),1],antallkol),antallrad,antallkol)

    for(i in 1:antallkol)
    {
        tid[,i] <- tid[,i] + deltat[i]
    }

    antallelem <- antallrad*antallkol

    Yt <- numeric(antallelem)
    Yt1 <- numeric(antallelem)
    Tid <- numeric(antallelem)

for(i in 1:antallkol)
{
    if(i==1) { ovre <-1 }
    else{ovre <- antallrad*(i-1) + 1}

    nedre <- antallrad*i
```

```
Yt[ovre:nedre]      <- yt[1:antallrad,i]
Yt1[ovre:nedre]    <- yt1[1:antallrad,i]
Tid[ovre:nedre]    <- tid[1:antallrad,i]
}

dnls <- data.frame(Yt=Yt,Yt1=Yt1,Tid=Tid) #
Lager data framen som skal brukes i nls-fit.
funksjonen param(dnls,"alfa") <- 167 # Definer startverdien på alfa i
funksjonen param(dnls,"gamma") <- 26 # Definer startverdien på gamma i
funksjonen param(dnls,"tau") <- -1.8 # Definer startverdien på tau i
funksjonen param(dnls,"fi") <- 0.984 # Definer startverdien på fi i funksjonen

# Formelen som tilpasses, dvs. modell nr 1 i Lucia og Schwartz

nlspara <- nls.control(maxiter = 200, tolerance=.001, minscale=.001, trace = T)
verdier <- nls(Yt~fi*Yt1 + alfa + gamma*cos(2*pi*(Tid+tau)/52)+
              -fi*(alfa + gamma*cos(2*pi*(Tid - 1
+tau)/52)), dnls, control=nlspara)

alfa <- verdier$parameter[1]
gamma <- verdier$parameter[3]
tau <- verdier$parameter[4]

residualer <- matrix(verdier$residuals,antallrad,antallkol)
stok <- yt - (alfa + gamma*cos(2*pi*(Tid+tau)/52))

korrl <- cor(residualer,magasin[2:319,])
korr2 <- cor(stok,magasin[2:319,])

par(mfrow = c(2, 1))
plot(deltat,korr1[,1],xlab="Modningstid i uker", ylab="Korrelasjon",main="Korrelasjon
mellom residualer og magasinnivå",type="p")
plot(deltat,korr2[,1],xlab="Modningstid i uker", ylab="Korrelasjon",main="Korrelasjon
mellom stokastiskledd og magasinnivå",type="p")

return(verdier) # Scriptet returnerer parameterene fra minste kvadraters metode.
}
```

Funksjon som finner gjennomsnittlig verdi på markedsprisen for risiko

```
# Skript for å finne gjennomsnittlig risikopremie ved Kosebakk og Ollmar sitt utvalg.
risikopremiel <- function(inndata,tid,knum) # Les inn y-verdier og tidspunktene som funksjonen skal tilpasses
for.
{
  deltat <- c(1,seq(2,7,1),seq(12,52,4),70,88,104) #
  Verdier som legges til tiden t, slik at sesongfunksjon får rette verdien.
  antallkol <- length(inndata)
  # Antallet kolonner i pris matrisen.
  antallrad <- length(inndata[,1])
  # Antall rader i matrisene som skal lages.
  yt <- inndata[1:(antallrad + 1 ),] #
  Leser inn y(t) verdien i den midlertidige matrisen.

  alfa <- 4.9763 # Definerer
  verdien av alfa i funksjonen
  gamma <- 0.1918 # Definerer
  verdien av gamma i funksjonen
  kappa <- 0.035 # Definerer
  verdien av kappa i funksjonen
  sigma <- 0.1008 # Definerer verdien
  av sigma i funksjon

  yo <- log(inndata[,1])-(alfa +gamma*cos(2*pi*tid/52))
  tid <- matrix(rep(tid[1:antallrad,1],antallkol),antallrad,antallkol)

  for(i in 1:antallkol)
  {
    tid[,i] <- tid[,i] + deltat[i]
  }

  dtid <- 0*tid

  for(i in 1:antallkol)
  {
    dtid[,i] <- dtid[,i] + deltat[i]
  }

  antallelem <- antallrad*antallkol

  Yt <- numeric(antallelem)
  Yo <- numeric(antallelem)
}
```

```

        Tid          <-      numeric(antallelem)
        Dtid         <-      numeric(antallelem)

for(i in 1:antallkol)
{
    if(i==1) {      ovre <-1 }
    else{ovre <-  antallrad*(i-1) + 1}

    nedre <- antallrad*i

    Yt[ovre:nedre] <- yt[1:antallrad,i]
    Yo[ovre:nedre] <- yo[1:antallrad]
    Tid[ovre:nedre] <- tid[1:antallrad,i]
    Dtid[ovre:nedre] <- dtid[1:antallrad,i]
}

# Kapper av vektoren de 319 første elementene, de skal ikke brukes i tilpassningen.

Yt          <-      Yt[(antallrad +1):antallelem]
Yo          <-      Yo[(antallrad +1):antallelem]
Tid         <-      Tid[(antallrad +1):antallelem]
Dtid        <-      Dtid[(antallrad +1):antallelem]

    dnls <-
data.frame(Yt=Yt,Yo=Yo,Tid=Tid,Dtid=Dtid,alfa=alfa,gamma=gamma,sigma=sigma,kappa=kappa) # Lager data frammen som
skal brukes i nls-fit.
    param(dnls,"lamda") <-      0.1      # Definer startverdien på lamda i
funksjonen

# Formelen som tilpasses, dvs. modell nr 2 i Lucia og Schwartz

nlspara <-      nls.control(maxiter = 200, tolerance=.001, minscale=.001, trace = T)

verdier <- nls(Yt~exp(alfa +gamma*cos(2*pi*Tid/52))+Yo*exp(-kappa*Dtid)+
- (1/kappa)*lamda*sigma*(1-exp(-
kappa*Dtid))+
+ (1/(4*kappa))*(sigma^2)*(1-exp(-
2*kappa*Dtid))), dnls, control=nlspara)
    RMSE <- (sum(verdier$residuals^2)/length(verdier$residuals))^0.5
    SSE <- sum(verdier$residuals^2)
    SST <- sum(Yt^2)
    R2 <- 1 -SSE/SST

    lamda <- verdier$parameter

    yttest <- inndata[,knum]
    tidp <- tid[,knum]
    dtidp <- dtid[,knum]

    yfitted <- exp(alfa +gamma*cos(2*pi*tidp/52))+yo*exp(-kappa*dtidp)+
- (1/kappa)*lamda*sigma*(1-exp(-
kappa*dtidp))+
+ (1/(4*kappa))*(sigma^2)*(1-exp(-
2*kappa*dtidp)))

    hovedtekst <- paste("Observert forwardpris i uke ",deltat[knum]," plottet mot tilpasset
pris",sep="")
    par(mfrow = c(1, 1))
    tsplot(yttest,yfitted,xlab="Observasjon nr. 1 = uke 50 1995- nr.318= uke 4 2002",
ylab="NOK/MWh",main=hovedtekst)

    return(verdier,R2,RMSE) # Skriptet returnerer parameterene fra minste kvadraters metode.
}

```

Funksjon som finner markedsprisen for risiko som funksjon av modningstid

```

# Skript laget for å finne risikopremien som funksjon av modningstiden.
# Risikopremien for en gitt uke brukes deretter til å sjekke tilpassningen for denne uken.
# Regner ut risikopremien for hver uke.
risikopremie4 <- function(inndata,tid,knum1,knum2,knum3) # Les inn y-verdier og tidspunktene som funksjonen
skal tilpasses for.

{
    deltat <- c(seq(1,104,1))
    # Verdier som legges til tiden t, slik at sesongfunksjon får rette verdien.

    antallkol <- length(inndata)
    # Antallet kolonner i pris matrisen.
    antallrad <- length(inndata[,1])
    # Antall rader i matrisene som skal lages.

```

Hovedoppgave i investering, finansiering og økonomistyring
 NTNU Norges Tekniske-Naturvitenskaplige Universitet
 Våren 2002

```

yt                <- inndata[1:antallrad,]
# Leser inn y(t) verdien i den midlertidige matrisen.

alfa              <- 4.9763                # Definerer
verdien av alfa i funksjonen
gamma            <- 0.1918                # Definerer
verdien av gamma i funksjonen
kappa           <- 0.035                 # Definerer
verdien av kappa i funksjonen
sigma          <- 0.1008                # Definerer verdien
av sigma i funksjon

x                <- log(inndata[,1])-(alfa +gamma*cos(2*pi*(tid+1)/52))
tid             <- matrix(rep(tid[1:antallrad,1],antallkol),antallrad,antallkol)

for(i in 1:antallkol)
{
    tid[,i] <- tid[,i] + deltat[i]
}

dtid <- 0*tid
for(i in 1:antallkol)
{
    dtid[,i] <- dtid[,i] + deltat[i]
}

lamdavektor     <- numeric(antallkol)
SSEvektor       <- numeric(antallkol)

for (k in 1:antallkol)
{
    Yt           <- inndata[,k]
    Tid          <- tid[,k]
    Dtid        <- dtid[,k]
    X           <- matrix(cbind(x))

# Lager data framen som skal brukes i nls-fit.

dnls <- data.frame(Yt=Yt,X=X,Tid=Tid,Dtid=Dtid,alfa=alfa,gamma=gamma,sigma=sigma,kappa=kappa)
param(dnls,"lamda") <- 0.001 # Definer startverdien på lamda i funksjonen

# Formelen som tilpasses, dvs. modell nr 2 i Lucia og Schwartz

nlspara <- nls.control(maxiter = 200, tolerance=.001, minscale=.001, trace = F)

verdier <- nls(Yt~exp(alfa +gamma*cos(2*pi*Tid/52))+X*exp(-kappa*Dtid)+
- (1/kappa)*lamda*sigma*(1-exp(-
kappa*Dtid))+
+ (1/(4*kappa))*(sigma^2)*(1-exp(-
2*kappa*Dtid))), dnls, control=nlspara)

if (k == 1) {verdivektor <- list(verdier)}
else{verdivektor <-c(verdivektor,list(verdier))}

if(k==1){fittedm <- verdier$fitted.values}
fittedm <- as.matrix(cbind(fittedm,verdier$fitted.values))

lamdavektor[k] <- verdier$parameter
SSEvektor[k] <- sum(verdier$residuals^2)
}

SSE <- sum(SSEvektor)

ytp1 <- inndata[,knum1]
ytp2 <- inndata[,knum2]
ytp3 <- inndata[,knum3]

ytfp1 <- fittedm[,knum1]
ytfp2 <- fittedm[,knum2]
ytfp3 <- fittedm[,knum3]

hovedtekst1 <-paste("Observerte forwardpris i uke ",deltat[knum1]," plottet mot tilpasset
forwardpris",sep="")
hovedtekst2 <-paste("Observerte forwardpris i uke ",deltat[knum2]," plottet mot tilpasset
forwardpris",sep="")
hovedtekst3 <-paste("Observerte forwardpris i uke ",deltat[knum3]," plottet mot tilpasset
forwardpris",sep="")

par(mfrow = c(3, 1))
tsp1plot(ytp1,ytfp1,xlab="Observasjon nr.",ylab="NOK/MWh",main=hovedtekst1)
tsp2plot(ytp2,ytfp2,xlab="Observasjon nr.",ylab="NOK/MWh",main=hovedtekst2)
tsp3plot(ytp3,ytfp3,xlab="Observasjon nr.",ylab="NOK/MWh",main=hovedtekst3)

R2 <- 1-SSE/sum(sum(inndata^2))

```

```
return(verdivektor,lamdavektor,R2) # Scriptet returnerer parameterene fra minste kvadraters  
metode.  
}
```


Vedlegg nr 9: Utdrag av dokumentasjon på SEFas prismodell(kopi)