

# Vurdering av operasjonell tilgjengelighet av prosessanlegg for CO<sub>2</sub>-fangst

**Kristian Engdal**

Master i energi og miljø  
Oppgaven levert: Juni 2009  
Hovedveileder: Olav Bolland, EPT



# Oppgavetekst

CO<sub>2</sub> er identifisert som den største bidragsyteren til global oppvarming. Det største potensialet for reduksjon av CO<sub>2</sub>-utslipp til atmosfæren er gjennom fangst og lagring av CO<sub>2</sub> fra kraftproduserende industri. Utfordringen for denne industrien er at CO<sub>2</sub>-fangst-prosessen gjør kraftproduksjonen mindre lønnsom. Det koster relativt mye energi å skille ut CO<sub>2</sub> fra slike prosesser. Dette er motivasjonen for at det foregår mye forskning og utvikling av ny teknologi for kostnadseffektive CO<sub>2</sub>-fangst prosesser. Tekniske utfordringer er både oppskalering av kjent teknologi, utvikling av ny teknologi, samt optimalisering av prosessen. I alle tilfeller fører dette til stor teknisk og økonomisk risiko ved å ta teknologien i bruk på store prosessanlegg. Ved svært store investeringer, ønsker man å kunne vurdere og håndtere risikoen før man tar i bruk teknologien.

Med utgangspunkt i en prosjektoppgave utført høsten 2008 skal det gjennomføres en evaluering av tilgjengelighet for et absorpsjonsanlegg for separasjon av CO<sub>2</sub> fra eksosgass.

I evaluering av tilgjengelighet vil det eksistere usikkerhet i kvantifiserte størrelser. For å aggregere og kvantifisere denne usikkerheten, kan det antas at tilgjengeligheten i de ulike delprosessene er stokastisk uavhengige og fordelt etter et prinsipp, som for eksempel 'gjennomsnittlig-tid-til-feil'.

Data for tilgjengelighet for absorpsjonsprosesser av den type det er aktuelt med her, finnes i liten grad. Oppgaven er derfor mest fokusert på metode, og tallresultater vektlegges i den grad data for tilgjengelighet på enhetene i prosessen finnes.

Det skal etableres en metodikk for beregning av tilgjengelighet for et absorpsjonsanlegg for separasjon av CO<sub>2</sub> fra eksosgass, basert på en beskrivelse av tilgjengelighet med utgangspunkt i eksisterende data. Metodikken skal benyttes til å beregne tilgjengelighet med basis i data i litteratur og egne estimater.

Oppgaven bearbeides ut fra følgende punkter

1. Litteraturstudie for å finne informasjon om tilgjengelighet på absorpsjonsprosesser generelt, for absorpsjonsprosesser for separasjon av CO<sub>2</sub>, og for enhetsoperasjonene som inngår i et slikt anlegg.
2. Utarbeide metodikk for å beskrive tilgjengelighet ved hjelp av egnede fordelingsfunksjoner for feilhendelser, samt hvordan regne kombinasjoner av flere feilhendelser i et integrert anlegg, slik at hele anleggets tilgjengelighet fremkommer.
3. Data for tilgjengelighet skal skaffes tilveie så langt det er praktisk mulig, og ellers skal estimater benyttes.
4. Metodikken skal demonstreres i form av et utvalg av regneeksempler.



## Forord

Denne oppgaven er resultatet av mitt arbeid med masteroppgave ved Norges teknisk-naturvitenskapelige universitet, fakultetet for ingeniørvitenskap og teknologi, institutt for energi- og prosesssteknikk, våren 2009. Oppgaven er delvis en fortsettelse av en prosjektoppgave skrevet høsten 2008 om CO<sub>2</sub>-fangst og tilgjengelighetsstudier.

Arbeidet med denne masteroppgaven har både vært en utfordring og en god erfaring å ha med seg i overgangen fra studietilværelsen til arbeidslivet. En av de viktigste erfaringene jeg har fra denne prosessen er å jobbe selvstendig med et tema som er lite utviklet fra før. Det har vist seg at oppgaven ble mye vanskeligere enn antatt i og med at jeg har måttet sette meg inn i det meste av informasjon uten videre kjennskap til emnene på forhånd. Når det i tillegg har vært lite informasjon å hente på temaet har det til tider vært veldig krevende. Det har uansett vært en spennende opplevelse som er godt å ha med seg videre med tanke på fremtidige utfordringer i arbeidslivet. I tillegg har jeg tilegnet meg mye kunnskap innenfor fagområder som jeg ikke var kjent med på forhånd.

I tillegg til min veileder ved NTNU, professor Olav Bolland, vil jeg takke min medveileder, fra prosjektoppgaven høsten 2008, Tore Myhrvold hos Det Norske Veritas AS, for gode råd på veien.

Trondheim, juni 2009

Kristian Engdal



## Sammendrag

Denne oppgaven viser en metodikk for beregning av operasjonell tilgjengelighet av et amineranlegg for separasjon av CO<sub>2</sub> fra eksosgass. Det fins foreløpig ikke anlegg som separerer CO<sub>2</sub> fra eksosgass i en storskala sammenheng slik at det naturlig nok er veldig lite informasjon å oppdrive om tilgjengelighet til slike systemer. Det er derfor gjort flere antakelser og forenklinger med tanke på tilnærming til feilårsaker og feildata. Det er valgt å bruke en metode utarbeidet av Thangamani G. et al. [1] som grunnlag for oppbygningen av metodikken i denne oppgaven. Årsaken til dette er at man får et veldig oversiktlig og godt bilde av hvordan systemet er bygd opp med feiltrestrukturen som er benyttet i denne metoden. Monte Carlo simulering er videre benyttet til å simulere og beregne ”Tid-til-feil” og operasjonell tilgjengelighet. Man har også gjort et forsøk på å sammenlikne den empiriske fordelingen til de simulerte resultatene med spesifikke statistiske fordelinger. Resultatene viser at en normal-, gamma- eller weibullfordeling passer godt for de simulerte resultatene. Det er derimot ikke grunnlag for å trekke noen konklusjoner ut i fra de resultatene som er vist i denne oppgaven fordi de fleste feildataene er hentet fra annet prosessutstyr enn det som er i et CO<sub>2</sub>-rensaneanlegg.

Det er ikke utviklet en god modell for simulering og beregning av reparasjonsrater, som nok vil være den største svakheten til metodikken. Det har dessverre vært vanskelig å finne gode løsninger for hvordan reparasjonsratene kunne implementeres med det informasjonsgrunnlaget og datatilgjengeligheten som fins. Det har derfor blitt lagt lite vekt på utarbeidelsen av modell for reparasjonsratene. Man må likevel understreke at det er viktig å kunne modellere gode og realistiske reparasjonsrater med tanke på å få riktige anslag for dødtiden til anlegget og dermed også tilgjengeligheten.

På grunn av at det er benyttet mye feildata for utstyr som ikke er tilknyttet et CO<sub>2</sub>-rensaneanlegg er det vanskelig å gi noen spesifikke konklusjoner for tilgjengeligheten til et slikt anlegg. Man kan på et generelt grunnlag se at redundans av utstyrskomponenter har en stor effekt på reduksjonen av korrektiv vedlikeholdstid, sammenliknet med et seriekoblet system hvor man har redusert feilraten til enkelte feilhendelser betraktelig. Slik som anlegget er satt opp i denne oppgaven vil ikke den korrektive vedlikeholdstiden ha mye å si for dødtiden til anlegget. Dette er fordi man har implementert en forebyggende vedlikeholdsfase hvert andre år som varer i to måneder og dermed er en dominerende faktor for dødtiden.

Arbeidet gjort i denne oppgaven viser at RAM-analyser av CO<sub>2</sub>-renseseprosesser er et område som foreløpig er lite utforsket. Det har vist seg at utarbeidelsen av en praktisk metode for tilgjengelighetsberegninger i CO<sub>2</sub>-renseseprosesser er veldig vanskelig å få til med det materialet som fins i dag. Metodikken utarbeidet i denne oppgaven er langt fra ferdigutviklet, men vil fungere som et fundament og godt utgangspunkt for videre arbeid innenfor området.

# Innholdsfortegnelse

<b>FORORD</b> .....	<b>i</b>
<b>SAMMENDRAG</b> .....	<b>iii</b>
<b>INNHOLDSFORTEGNELSE</b> .....	<b>iv</b>
<b>FIGURISTE</b> .....	<b>vi</b>
<b>TABELLISTE</b> .....	<b>vii</b>
<b>SYMBOLLISTE</b> .....	<b>viii</b>
<b>FORKORTELSER</b> .....	<b>x</b>
<b>1 INNLEDNING</b> .....	<b>1</b>
1.1 Bakgrunn for oppgaven .....	1
1.2 Tolkning av oppgaven .....	2
1.3 Kildehenvisninger.....	3
<b>2 TILGJENGELIGHET</b> .....	<b>5</b>
2.1 Statistisk og matematisk grunnlag .....	5
2.2 Retningslinjer for tilgjengelighetsstudier i oljeindustrien .....	6
2.2.1 <i>Definisjoner av uttrykk benyttet i RAM-studier</i> .....	7
2.2.2 <i>Regularitetsanalyse</i> .....	9
2.2.3 <i>Datagrunnlag for tilgjengelighetsstudier</i> .....	11
2.3 Analyseverktøy .....	12
2.3.1 <i>FMEA/FMECA</i> .....	12
2.3.2 <i>Feiltreanalyse</i> .....	13
<b>3 PROSESSMODELL</b> .....	<b>15</b>
3.1 CO <sub>2</sub> -rensaneanlegg for rensing av eksosgass .....	15
3.2 Feilårsaker i aminrensaneanlegget .....	18
3.3 Datagrunnlag .....	20
<b>4 BAKGRUNNSTEORI TIL METODIKKEN</b> .....	<b>23</b>
4.1 Fordelingsfunksjoner .....	23
4.1.1 <i>Poissonprosess</i> .....	24
4.1.2 <i>Normalfordeling</i> .....	25
4.1.3 <i>Eksponentialfordeling</i> .....	26
4.1.4 <i>Gammafordeling</i> .....	28
4.1.5 <i>Weibullfordeling</i> .....	29
4.1.6 <i>Lognormalfordeling</i> .....	30
4.2 Systemanalyse .....	30



4.2.1	<i>Reparerbare systemer</i> .....	31
4.2.2	<i>Monte Carlo simulering</i> .....	33
<b>5</b>	<b>METODIKKEN FOR BESKRIVELSE AV TILGJENGELIGHET</b> .....	<b>35</b>
5.1	Hovedstruktur .....	35
5.2	Ulike oppsett av systemet .....	37
5.3	Total simuleringstid og antall simuleringer .....	37
5.4	Beregning av operasjonell tilgjengelighet .....	38
5.5	Beskrivelse av simuleringprosessen .....	39
<b>6</b>	<b>RESULTATER OG DISKUSJON</b> .....	<b>43</b>
6.1	Hovedoppsett .....	43
6.1.1	<i>Feilhendelsene med lavest og høyest feilrate</i> .....	44
6.1.2	<i>Feiltrærne og systemet</i> .....	48
6.2	Redundans av alle pumper i aminkretsen .....	51
6.2.1	<i>Feiltrærne og systemet</i> .....	52
6.3	Operasjonell tilgjengelighet .....	54
6.4	Seriestruktur med feilhendelser som har redusert feilrate .....	55
6.4.1	<i>Reduksjon av feilraten til feilhendelse 1</i> .....	55
6.4.2	<i>Reduksjon av feilraten til feilhendelsen med høyest feilrate i hvert feiltre</i> .....	56
6.4.3	<i>Reduksjon av feilraten til de tre feilhendelsene med høyest feilrate</i> .....	56
<b>7</b>	<b>KONKLUSJON OG VIDERE ARBEID</b> .....	<b>59</b>
7.1	Konklusjon .....	59
7.2	Videre arbeid .....	60
<b>8</b>	<b>REFERANSER</b> .....	<b>63</b>
	<b>VEDLEGG</b> .....	<b>65</b>
	A: Tilpasning av simulerte verdier til spesifikke fordelinger i MATLAB .....	65
	B: MOCUS (Metode for å oppnå “cut sets”) .....	66
	C: Forenklet FMECA av et aminrenseanlegg .....	69
	D: MATLAB-kode .....	72
D.1:	<i>Hovedoppsett</i> .....	72
D.2:	<i>Oppsett med redundans av alle aminpumper</i> .....	80
	E: Feiltreoppbygning .....	89
E.1:	<i>Hovedoppsett</i> .....	89
E.2:	<i>Oppsett med redundans av alle aminpumper</i> .....	92
	F: Avvik mellom simuleringene .....	95

## Figurliste

<i>Figur 2.1: Illustrasjon som viser sammenhenger mellom enkelte regularitetsbegrep (oversatt fra engelsk fra NORSOK Z-016 [7]).....</i>	<i>9</i>
<i>Figur 3.1: Flytdiagram for aminrenseprosessen.....</i>	<i>15</i>
<i>Figur 4.1: "Badekar-kurve" for feilraten .....</i>	<i>24</i>
<i>Figur 5.1: Flytskjema for simuleringsprosessen. ....</i>	<i>42</i>
<i>Figur 6.1: Empirisk CDF for feilhendelse 9 sammenliknet med empiriske CDF for flere fordelingsfunksjoner.....</i>	<i>46</i>
<i>Figur 6.2: IFR/DFR-test for feilhendelse 9.....</i>	<i>46</i>
<i>Figur 6.3: Empirisk CDF for feilhendelse 12 sammenliknet med empiriske CDF for flere fordelingsfunksjoner.....</i>	<i>47</i>
<i>Figur 6.4: IFR/DFR-test for feilhendelse 12.....</i>	<i>48</i>
<i>Figur 6.5: Empirisk CDF for systemet sammenliknet med empiriske CDF for flere fordelingsfunksjoner.....</i>	<i>49</i>
<i>Figur 6.6: IFR/DFR-test for systemet. ....</i>	<i>50</i>
<i>Figur 6.7: Empirisk CDF for systemet sammenliknet med empiriske CDF for flere fordelingsfunksjoner.....</i>	<i>53</i>
<i>Figur 6.8: IFR/DFR-test for systemet. ....</i>	<i>53</i>
<i>Figur B.1: Eksempel på feiltre .....</i>	<i>66</i>
<i>Figur C.1: Forenklet FMECA av aminrenseanlegg del 1 .....</i>	<i>69</i>
<i>Figur C.2: Forenklet FMECA av aminrenseanlegg del 2.....</i>	<i>70</i>
<i>Figur C.3: Forenklet FMECA av aminrenseanlegg del 3.....</i>	<i>71</i>
<i>Figur E.1: Feiltre 1 i hovedoppsett .....</i>	<i>89</i>
<i>Figur E.2: Feiltre 2 i hovedoppsett .....</i>	<i>90</i>
<i>Figur E.3: Feiltre 3 i hovedoppsett.....</i>	<i>91</i>
<i>Figur E.4: Feiltre 1 i oppsett med redundans av aminpumper .....</i>	<i>92</i>
<i>Figur E.5: Feiltre 2 i oppsett med redundans av aminpumper .....</i>	<i>93</i>
<i>Figur E.6: Feiltre 3 i oppsett med redundans av aminpumper .....</i>	<i>94</i>

## Tabelliste

<i>Tabell 3.1: Oversikt av de ulike reaksjonskomponentene i aminreanseanlegget.</i> .....	17
<i>Tabell 3.2: Nøkkeltall for ulike feilhendelser utarbeidet fra OREDA.</i> .....	21
<i>Tabell 5.1: Andelen feilhendelser som har MTTF lengre enn den totale simuleringstiden ved ulike simuleringsperiode og ulikt antall simuleringer.</i> .....	38
<i>Tabell 6.1: MTTF og MTTR for enkelte feilhendelser, feiltrær og totalsystem.</i> .....	43
<i>Tabell 6.2: Maksimum-, minimum-, middel- og medianverdi i tillegg til standardavvik for feilhendelse 9 og 12 og systemet basert på en empirisk CDF av de simulerte verdiene.</i> .....	44
<i>Tabell 6.3: Teoretisk verdi for MTTF til feilhendelse 9 sammenliknet med simulert verdi.</i> .....	44
<i>Tabell 6.4: Teoretisk verdi for MTTF til feilhendelse 12 sammenliknet med simulert verdi.</i> .....	45
<i>Tabell 6.5: MTBF og MTTR for de ulike uønskede topphendelsene i systemet.</i> .....	48
<i>Tabell 6.6: MTTF og MTTR for feiltrær og totalsystemet i oppsettet med redundans i aminpumper.</i> .	51
<i>Tabell 6.7: MTBF og MTTR for de ulike uønskede topphendelsene i systemet.</i> .....	52
<i>Tabell 6.8: Systemtilgjengelighet og total korrektiv vedlikeholdstid sammenliknet med maksimal teoretisk verdi uten korrektivt vedlikehold.</i> .....	54
<i>Tabell 6.9: Seriestruktur med forbedret feilrate for feilhendelse 1.</i> .....	55
<i>Tabell 6.10: Seriestruktur med forbedret feilrate for feilhendelse 1,9 og 22.</i> .....	56
<i>Tabell 6.11: Seriestruktur med forbedret feilrate for feilhendelse 9,10 og 11.</i> .....	57
<i>Tabell F1: Avvik mellom de ulike simuleringsrundene ved <math>p = 1000</math> simuleringer.</i> .....	95
<i>Tabell F2: Avvik mellom de ulike simuleringsrundene ved <math>p = 10000</math> simuleringer.</i> .....	95

## Symbolliste

Symbol	Forklaring	Benevning
$t$	Tid	[s]
$T$	Tid til feil inntreffer	[s]
MTTF	Gjennomsnittlig tid til feil	[s]
MTTR	Gjennomsnittlig tid til reparasjon	[s]
MTBF	Gjennomsnittlig til mellom feil	[s]
$\Delta t$	Endring i tid	[s]
$F(t)$	Sannsynlighetsfunksjon	[1]
$R(t)$	Pålitelighetsfunksjon	[1]
$P(t)$	Sannsynlighet	[1]
$f(t)$	Sannsynlighetstetthetsfunksjon	[s <sup>-1</sup> ]
$z(t)$	Feilrate	[s <sup>-1</sup> ]
$X_i$	Binær strukturvariabel	[1]
$\lambda$	Feilintensitet	[s <sup>-1</sup> ]
$v$	Middelverdi	[s]
$\tau^2$	varians	[s <sup>2</sup> ]
$N(v, \tau^2)$	Normalfordeling	[1]
$\Phi(\bullet)$	Standardnormalfordeling	[1]
$\phi(\bullet)$	Normalfordelt sannsynlighetstetthetsfunksjon	[s <sup>-1</sup> ]
$Y$	Skiftparameter	[s]
$\alpha$	fleksibilitetsparameter	[1]
$\Gamma(\alpha)$	Gammafunksjonen	[1]
$\Phi(\mathbf{X}(t))$	Strukturfunksjonen	[1]
$E(t)$	Forventningsverdi	[variabel]

$A(t)$	Tilgjengelighet	[1]
$\mu$	Reparasjonsrate	$[s^{-1}]$
$\omega$	Raten av forekomster av feilhendelser	$[s^{-1}]$
$T_t$	Total simuleringstid	[s]
$T_s$	Forebyggende reparasjonstid	[s]
$T_d$	Total driftstid	[s]
$T_0$	Korrektiv vedlikeholdstid	[s]
$T_r$	Total vedlikeholdstid	[s]
$cr$	Andel av feilhendelser som har MTTF høyere enn totaltiden	[1]
$p$	Antall simuleringer	[1]
$q$	Sannsynlighet for feil i et tidsskritt	[1]

## Forkortelser

Gjennomsnittlig tid til feil	-	MTTF
Gjennomsnittlig tid til reparasjon	-	MTTR
Gjennomsnittlig tid mellom feil	-	MTBF
Pålitelighet, tilgjengelighet og vedlikehold	-	RAM
Stigende feilrate	-	IFR
Avtakende feilrate	-	DFR
Kumulativ fordelingsfunksjon	-	CDF
Sannsynlighetstetthetsfunksjon	-	PDF
Feilmodus og –effekt analyse	-	FMEA
Feilmodus, –effekt og kritikalitets-analyse	-	FMECA
Offshore reliability data	-	OREDA
Metode for å oppnå ”cut sets”	-	MOCUS
Raten av forekomster av feilhendelser	-	ROCOF
Feilhendelse	-	FH
Feiltre	-	FT

# 1 Innledning

## 1.1 Bakgrunn for oppgaven

I dag er fangst og lagring av CO<sub>2</sub> et mye omtalt tema i verdenssamfunnet. Mange peker på at utslipp av CO<sub>2</sub> til atmosfæren har hovedskylden for de store klimaforandringene som verden står overfor. Det er derfor lagt ned store ressurser for å komme opp med gode løsninger for fangst og lagring av CO<sub>2</sub> fra industrielle prosesser. Den største kilden for fangst av CO<sub>2</sub> er kraftproduserende industri. Det er derimot et betydelig effektivitetstap knyttet til rensing av eksosgass fra kraftproduserende industri. Energikostnadene i forbindelse med utskilling av CO<sub>2</sub> fra industrielle prosesser er relativt høye i tillegg til de tekniske utfordringene som må overvinnnes. Det er derfor en stor økonomisk og teknisk risiko ved å ta i bruk CO<sub>2</sub>-renseteknologi på store prosessanlegg. På grunn av at det er knyttet veldig store investeringer til denne teknologien ønsker man å vurdere og håndtere risikoen før den tas i bruk. Det finnes i dag mange ulike metoder og verktøy for evaluering av risiko og pålitelighet til en rekke industriprosesser. CO<sub>2</sub>-rensing av eksosgass er derimot en teknologi som fortsatt ikke er tatt i bruk i en storskala sammenheng og dermed også et nokså uprøvd område innenfor RAM-fagfeltet.

Det er et stort behov for å få på plass et godt fundament for evaluering av tilgjengeligheten til et anlegg for CO<sub>2</sub>-rensing av eksosgass. Beregning av tilgjengeligheten er viktig i seg selv, men det er minst like viktig å identifisere hvilke feilhendelser som har størst innvirkning på systemtilgjengeligheten. En systematisk tilnærming til problemet er derfor nødvendig for å kunne si noe om årsaker og sammenhenger i anlegget.

Denne oppgaven gir først en oversikt av hvilke faktorer en må ta hensyn til ved utarbeidelsen av en systemanalyse. Deretter er det utarbeidet en metodikk for beregning av operasjonell tilgjengelighet til et absorpsjonsanlegg for rensing av eksosgass. Metodikken tar i bruk en feiltrestruktur for å sette de ulike feilhendelsene i sammenheng mens en Monte Carlo simulering er benyttet til å simulere "levetider" for de forskjellige feilhendelsene. Metodikken er delvis basert på en liknende metode beskrevet i et eksempel på en simuleringsprosess for et pulverkatalytisk krakkingsanlegg, av Thangamani G. et al. [1], hvor målet var å beregne tilgjengeligheten til anlegget ved hjelp av eksponentielt fordelte levetider og reparasjonstider. Metoden beskrevet av Thangamani G. et al. ble blant annet valgt på grunn av den oversiktlige struktureringen av de forskjellige feilhendelsene inndelt i ulike feiltrær.

Beregningen av de ulike systemvariablene er basert på simulerte verdier for "gjennomsnittlig-tid-til-feil" for hver feilhendelse. Det er også gjort et forsøk på å si noe om hvordan simuleringsresultatene er fordelt for eventuelt å kunne dra nytte av statistiske modeller som et sammenlikningsgrunnlag.

Utgangspunktet for et relevant datagrunnlag til denne oppgaven var nokså begrenset ved at det ikke finnes storskala separasjonsanlegg for rensing av eksosgass. En mulig kilde for

relevant data er StatoilHydro som har erfaring med CO<sub>2</sub>-separasjon fra naturgasstrømmen på blant annet Sleipnerfeltet. Det har dessverre ikke vært mulig å få tak i relevant data fra StatoilHydro etter at det ble gjort en forespørsel til bedriften [2]. Et alternativ til å få tak i relevant data hadde vært å utføre en prosesssimulering for en CO<sub>2</sub>-renseprosess i et simuleringsprogram som for eksempel Aspen HYSYS [3] eller ProII [4]. Dette er derimot en veldig komplisert og tidkrevende prosess som ikke var et realistisk alternativ innenfor rammene til denne oppgaven. De dataene som er benyttet i denne oppgaven er derfor basert på data fra OREDA [5]. Dette er erfaringsdata om pålitelighet som er hentet fra ulike offshore olje- og gassinstallasjoner. Mesteparten av komponentene som er i et aminrenseanlegg er ikke representert i OREDA. Det har derfor blitt foretatt en rekke forenklinger og tilpasninger i utvelgelsen av data til simuleringsprosessen. Resultatene man har fått fra simuleringsprosessen gir derfor ikke nødvendigvis et riktig bilde på tilgjengeligheten til et CO<sub>2</sub>-renseanlegg. Dataene som er benyttet i denne oppgaven er brukt som et verktøy for å få testet metodikken. Man kan derfor ikke trekke konkrete slutninger om hva usikkerheten i datagrunnlaget vil ha og si for resultatene når man ikke opererer med ”ekte” data.

## 1.2 Tolkning av oppgaven

I oppgaveteksten under avsnittet om mål for oppgaven står det skrevet at man skal utføre et litteraturstudie for å finne informasjon om tilgjengelighet for absorpsjonsprosesser generelt og for absorpsjonsprosesser for separasjon av CO<sub>2</sub>. I denne oppgaven er det i første omgang valgt å se på begrepet ”tilgjengelighet” for å gi en innføring i hva det innebærer. Deretter er det valgt å gå nærmere inn på de ulike retningslinjene for tilgjengelighetsstudier i oljeindustrien. Det har ikke vært mulig å oppdrive egne retningslinjer for tilgjengelighetsanalyser av CO<sub>2</sub>-renseprosesser slik at man har benyttet det samme utgangspunktet som er gitt for generell oljeindustri. Informasjon om tilgjengelighet for de ulike enhetsoperasjonene som inngår i et rensesanlegg har vært vanskelig å oppdrive. Som et resultat av dette er et vanlig aminrenseanlegg presentert så godt det lar seg gjøre med den informasjonen som har vært tilgjengelig.

Det er ikke et eget diskusjonskapittel i denne oppgaven. Det er i stedet valgt å diskutere ulike valg og resultater underveis i rapporten. Dette er gjort som et forsøk på å få en mer oversiktlig rapport ved at man får diskutert ulike aspekter og resultater så fort de dukker opp.

Rapporten starter ved at man i kapittel 2 beskriver begrepet ”tilgjengelighet” og ulike retningslinjer og verktøy knyttet til dette. I kapittel 3 er en prosessmodell for et separasjonsanlegg for rensing av eksosgass presentert med identifisering av feilårsaker og feildata. Kapittel 4 gir en gjennomgående presentasjon av teori knyttet til metodikken. Det er blant annet tatt med et eget delkapittel om ulike fordelingsfunksjoner for å beskrive forskjellige fordelinger av en feilhendelse eller et system. Dette benyttes i resultatene av simuleringene for å se om man kan se en sammenheng mellom ulike teoretiske fordelingsfunksjoner og simulerte verdier. I kapittel 4.2 blir generelle sammenhenger i



forbindelse med systemanalyse og simulering presentert i tillegg til likninger som benyttes i beregningsdelen av metodikken. Kapittel 5 gir en beskrivelse av alle fasene til den utviklede metodikken og går også inn på den spesifikke beregningsmodellen som er valgt for beregning av operasjonell tilgjengelighet. Resultatene blir presentert og diskutert i kapittel 6. Tilslutt blir konklusjonen og forslag til videre arbeid lagt fram i kapittel 7. Referanser til ulike kilder er plassert i kapittel 8 og vedlegget består av underkapitlene A til F, der blant annet feiltrærne og simuleringskoden ligger.

### **1.3 Kildehenvisninger**

Enkelte kilder er regelmessig benyttet for å presentere flere deler av teorigrunnlaget. Det er derfor valgt å presentere disse kildene i innledningen til kapitlene hvor de er mye brukt, i stedet for å referere til dem i hvert avsnitt. Andre kilder som er mindre benyttet er referert til direkte i teksten med nummer og i enkelte tilfeller også navn på hovedforfatter. Referanser til større oppslagsverk har kildehenvisning med spesifikke sidetall, mens det ikke er referert til spesifikke sidetall i artikler og mindre omfattende kilder.



## 2 Tilgjengelighet

### 2.1 Statistisk og matematisk grunnlag

I pålitelighetsstudier av tekniske systemer er man avhengig av å se på ulike modeller av systemet. Det er viktig at modellene er såpass oversiktlige og enkle at de kan håndteres av matematiske og statistiske metoder. Man må også kunne anta at modellen er mest mulig realistisk slik at resultatene man oppnår kommer til praktisk nytte etterpå. Resultatene er altså ikke mer ”korrekte” enn modellen er realistisk. Det statistiske grunnlaget i dette kapittelet er basert på teori hentet fra Rausand, M. et al. ([6], s.18-25).

Påliteligheten til en gjenstand er ofte gitt av en sannsynlighetsmodell som modellerer livsløpet til gjenstanden. Man setter da opp en pålitelighetsfunksjon  $R(t)$ . Avhengig av hvilke forutsetninger som ligger til grunn vil man kunne bruke ulike matematiske modeller for å finne sannsynligheten for at enheten ”overlever” i tidsintervallet  $(0, t]$ .

Pålitelighetsfunksjonen er avhengig av tiden det tar til en feil inntreffer. *Tid til feil inntreffer*  $T$  er kontinuerlig fordelt av *sannsynlighetstetthetsfunksjonen (PDF)*  $f(t)$  og gitt av fordelingsfunksjonen:

$$F(t) = P(T \leq t) = \int_0^t f(u) du \quad \text{for } t > 0 \quad (2.1)$$

Her er  $F(t)$  sannsynligheten for at enheten svikter innenfor tidsintervallet  $(0, t]$  og betegnes ofte som den *akkumulerte tetthetsfunksjonen (CDF)*.

Pålitelighetsfunksjonen er gitt som:

$$R(t) = 1 - F(t) = P(T > t) \quad \text{for } t > 0 \quad (2.2)$$

Videre kan man sette opp sannsynligheten for at en enhet vil svikte i tidsintervallet  $(t, t + \Delta t]$  når man vet at den fungerer ved tiden  $t$ :

$$P(t < T \leq t + \Delta t | T > t) = \frac{P(t < T \leq t + \Delta t)}{P(T > t)} = \frac{F(t + \Delta t) - F(t)}{R(t)} \quad (2.3)$$

Ved å dele likning (2.3) på  $\Delta t$  og samtidig la  $\Delta t \rightarrow 0$  vil man få *feilraten*  $z(t)$  til enheten som er en annen viktig målefunksjon for påliteligheten til en enhet:

$$z(t) = \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} \frac{1}{R(t)} = \frac{f(t)}{R(t)} \quad (2.4)$$

Når  $\Delta t$  er liten og antallet identiske enheter i drift ved  $t = 0$  er veldig høyt, vil  $z(t) \cdot \Delta t$  grovt sett representere det relative forholdet av enheter som er i drift ved tiden  $t$ , men som svikter i tidsintervallet  $(t, t + \Delta t]$ .

Siden sannsynlighetstettheten er definert som  $f(t) = \frac{d}{dt}F(t)$  vil likning (2.4) kunne skrives som:

$$z(t) = -\frac{R'(t)}{R(t)} = -\frac{d}{dt} \ln R(t) \quad (2.5)$$

Ved å integrere likning (2.5) med hensyn på tiden står man igjen med følgende uttrykk for pålitelighetsfunksjonen:

$$R(t) = e^{-\int_0^t z(u)du} \quad (2.6)$$

Dette viser at pålitelighetsfunksjonen  $R(t)$  så vel som fordelingsfunksjonen  $F(t)$  entydig er gitt av feilraten  $z(t)$ . Sannsynlighetstettheten kan ved hjelp av likning (2.4) og (2.6) skrives som:

$$f(t) = z(t)e^{-\int_0^t z(u)du} \quad \text{for } t > 0 \quad (2.7)$$

Dette er et statistisk og matematisk grunnlag som må ligge til grunn når man skal utføre tilgjengelighetsanalyser. Videre i dette kapittelet blir det sett nærmere på hvilke retningslinjer en må forholde seg til ved gjennomføring av en analyse av *pålitelighet, tilgjengelighet og vedlikehold (RAM)* innenfor prosessindustrien.

## 2.2 Retningslinjer for tilgjengelighetsstudier i oljeindustrien

Når man skal utføre omfattende analyser av for eksempel et prosessanlegg er det viktig at man følger de retningslinjene som er satt for at resultatene skal bli korrekte. Det fins derfor en rekke standarder som har som mål å presentere disse retningslinjene på en best mulig måte. I denne oppgaven blir det sett nærmere på NORSOK standard Z-016 [7] som dekker analyser av pålitelighet og vedlikehold av komponenter, systemer og operasjoner assosiert med oljeleting og –boring, utvinning, prosessering og transport av oljeressurser. Denne standarden, i tillegg til den nyutviklede internasjonale standarden ISO 20815 [8], vil gi det beste grunnlaget for RAM-studier for et CO<sub>2</sub>-renseanlegg av standardene som fins i dag.

NORSOK Z-016 presenterer en rekke krav til planlegging, utførelse og bruk av pålitelighetsteknologi. I denne kapittelet vil hovedfokuset ligge på punktene:

- Planlegging og implementering av pålitelighetsteknologi.
- Anvendelse av pålitelighets- og vedlikeholdsdata.

I resten av kapittel 2.2 vil NORSOK standard Z-016 [7] benyttes som teorigrunnlag der andre kilder ikke er angitt.

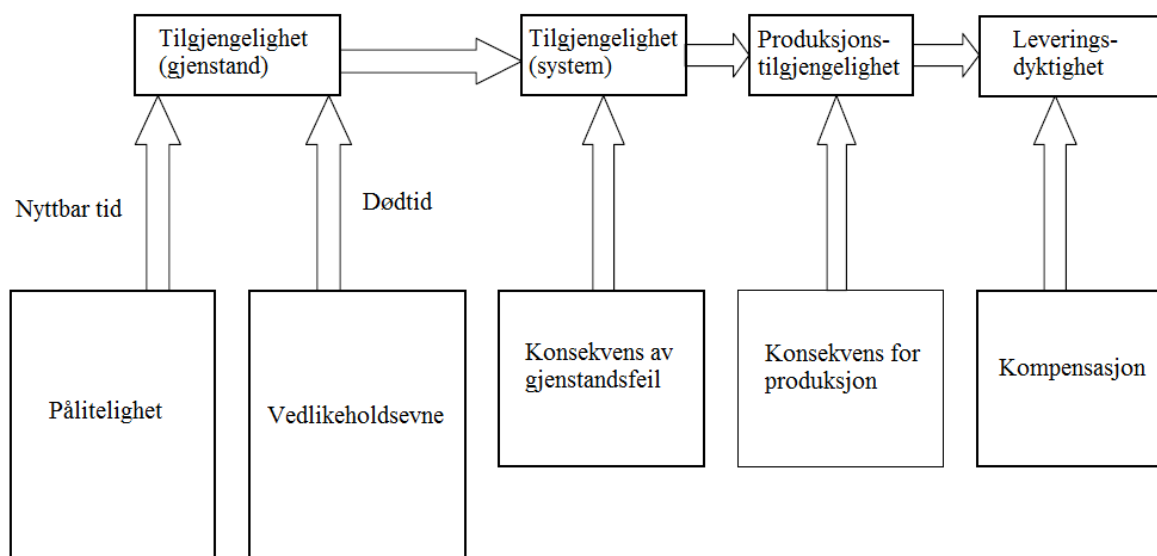
### 2.2.1 Definisjoner av uttrykk benyttet i RAM-studier

For å kunne forstå hva som menes med de ulike uttrykkene i RAM-studiene vil det i første omgang bli satt opp en liste med definisjoner på viktige uttrykk som vil bli brukt senere:

<i>Aktiv reparasjonstid</i>	Den delen av dødtiden hvor reparasjonen utføres på en gjenstand utenom logistiske forsinkelser, forberedelse til reparasjon og forberedelse til produksjon.
<i>Beregnet levetid</i>	Planlagt brukstid for totalsystemet.
<i>Driftstid</i>	Tidsintervallet hvor gjenstanden utfører sin påkrevde funksjon.
<i>Dødtid</i>	Tidsintervallet hvor en gjenstand er nede som følge av en feil eller en udugelighet til å utføre en påkrevd funksjon.
<i>Feil</i>	Avslutning på egenskapen en gjenstand har til å utføre en påkrevd funksjon.
<i>Feilmekanisme</i>	Fysisk-, kjemisk- eller annen prosess som fører eller har ført til feil.
<i>Feilrate</i>	Antall feil relativt til samsvarende operasjonstid eller kalendertid.
<i>Feilårsak</i>	Effekten som fører til at en feil observeres på gjenstanden som har feilet.
<i>Fellesfeil</i>	Feil i forskjellige gjenstander som har samme grunnleggende årsak, men hvor feilene ikke er avhengige av hverandre.
<i>Forebyggende vedlikehold</i>	Vedlikehold utført ved forutbestemte intervaller eller i henhold til foreskrevne kriterier som har som mål å redusere sannsynligheten for feil eller forringelsen av funksjonaliteten til gjenstanden.
<i>Gjennomsnittlig tid mellom feil (MTBF)</i>	$MTBF = MTTF + MDT$ . Hvis dødtiden er lik reparasjonstiden; $MTBF = MTTF + MTTR$ .
<i>Gjennomsnittlig tid til feil (MTTF)</i>	En beregnet tid til feil som kan både være lengre eller kortere enn beregnet levetid til systemet.
<i>Gjennomsnittlig tid til reparasjon (MTTR)</i>	En beregnet tid av den aktive reparasjonstiden.
<i>Hviletid</i>	Den delen av nyttbar tid hvor gjenstanden ikke er i funksjon.

<i>Korrektivt vedlikehold</i>	Vedlikehold som er utført etter en feiloppdagelse og har som mål å sette en gjenstand i stand til å utføre en påkrevd funksjon.
<i>Leveringsdyktighet</i>	Raten av leveringer i forhold til planlagte leveringer over en gitt tidsperiode når effekten av kompensasjonselementer som bytte fra andre produsenter og nedstrøms bufferlager er inkludert.
<i>Nyttbar tid</i>	Tidsintervallet hvor en gjenstand er i nyttbar tilstand som er beskrevet av at den kan utføre en påkrevd funksjon gitt at de eksterne ressursene er anskaffet.
<i>Produksjonstilgjengelighet</i>	Raten av produksjon i forhold til planlagt produksjon over en gitt tidsperiode.
<i>Pålitelighet</i>	Egenskapen en gjenstand har til å utføre en påkrevd funksjon under gitte forhold for et gitt tidsintervall.
<i>Regularitet</i>	Et uttrykk brukt til å beskrive hvordan et system er i stand til å møte ytelseskrav.
<i>Tilgjengelighet</i>	Egenskapen en gjenstand har til å utføre en påkrevd funksjon under gitte forhold ved et gitt tidspunkt eller et gitt tidsintervall, forutsatt at påkrevde eksterne vedlikeholdsressurser er anskaffet. Denne egenskapen er uttrykt som andelen av tid gjenstanden er i sin funksjonelle tilstand.
<i>Usikkerhet</i>	Mangel av kunnskap om en ukjent størrelse. Usikkerheten er uttrykt ved sannsynligheter.
<i>Vedlikeholdsevne</i>	Egenskapen en gjenstand har til å gå tilbake til tilstanden hvor den kan utføre en påkrevd funksjon når vedlikehold gjennomføres under gitte forhold i tillegg til at man bruker uttrykte prosedyrer og ressurser.

## Regularitet



Figur 2.1: Illustrasjon som viser sammenhenger mellom enkelte regularitetsbegrep (oversatt fra engelsk fra NORSOK Z-016 [7])

### 2.2.2 Regularitetsanalyse

Regularitetsanalysen skal gi et grunnlag for beslutninger som er viktige for valg av løsninger og mål for å oppnå en optimal økonomi med de gitte restriksjonene. Selv om økonomi er noe av det viktigste som ligger til grunn for en industriell prosess vil det ikke i denne oppgaven være noen diskusjon rundt økonomien i et anlegg. Hovedfokuset i denne oppgaven vil ligge på metodikken som ligger til grunn for analysen. Med det sagt er det viktig å merke seg at regularitetsanalyser skal være konsekvent i tillegg til at antakelser og pålitelighetsdata skal kunne spores opp.

Det kreves nøye planlegging før en kan utføre en grundig regularitetsanalyse. For det første skal målsettingen med analysen klart defineres på forhånd. Typiske målsettinger kan være å:

- Identifisere operasjonelle faktorer eller utstyr som er kritisk til regularitet.
- Forutsi produksjonstilgjengelighet, leveringsdyktighet, tilgjengelighet og pålitelighet.
- Anskaffe informasjon til andre aktiviteter som risikoanalyse eller vedlikeholds- og reservedelsplanlegging.
- Bekrefte regularitetsmålsettinger eller -krav.

I denne oppgaven vil det fokuseres på å finne tilgjengeligheten for et system og samtidig se i hvor stor grad enkelte feilhendelser eller redundans av komponenter har betydning for systemtilgjengeligheten.

Vanligvis skal man sette ned en gruppe med flere personer som kan være med på utførelsen av analysen. Det er ofte lønnsomt å ha med personer som er eksperter innenfor det området man skal undersøke. Regularitetsanalyse er en tverrfaglig aktivitet som krever nært samarbeid med andre relevante fagfelt.

Systemet som skal analyseres må defineres med de nødvendige grensene til omgivelsene slik at man vet hva som skal være med eller ikke i analysen. Det er ofte flere grupper av systemer i for eksempel en fullstendig gassproduksjonskjede. Hvis man kun skal se på CO<sub>2</sub>-rensprosessen må man vite hvor man skal sette grensene for systemet.

Man må definere de ulike operasjonsmodusene som skal være med i analysen. Relevante operasjonsmoduser kan være oppstart, normal drift, drift med delvis last og innskrenking av driften. Samtidig er det nødvendig å definere operasjonsfasen eller tidsperioden for analysen.

Ytelsesmålene man ønsker å forutsi skal defineres i starten. I tillegg må man ta stilling til om regularitetseffekten fra revisjonsstans eller katastrofehendelser skal inkluderes. Med katastrofehendelser menes alvorlige og ualminnelige hendelser som fører til langvarig stans av produksjonen.

Når man skal utføre en grundig regularitetsanalyse er det viktig å gå gjennom all relevant teknisk dokumentasjon. Man kan dele dokumentasjonen inn i to typer:

<i>Systembeskrivelse</i>	Dokumentasjon som beskriver alle tekniske og operasjonelle aspekter som påvirker resultatet av regularitetsanalysen i tillegg til at det er nødvendig for å identifisere systemet som skal analyseres.
<i>Pålitelighetsdata</i>	Det er viktig å kunne dokumentere pålitelighetsdata som er benyttet. Det er mulig å benytte seg av ekspertbedømmelser, men erfaringsbaserte estimater skal benyttes hvis de er tilgjengelige.

Utvikling av modellen for systemet inkluderer følgende aktiviteter:

- Funksjonell nedbrytning av systemet
- Evaluering av konsekvensene for feil og vedlikehold for de ulike undersystemene.
- Evaluering av hendelser som er med i modellen inkludert fellesfeil.
- Modellutvikling og dokumentasjon.

Avhengig av målsettingen for regularitetsanalysen er det viktig å se på ulike ytelsesmål. Produksjonstilgjengelighet og leveringsdyktighet er de mest brukte ytelsesmålene. I denne oppgaven derimot begrenses regularitetsanalysen til en *tilgjengelighetsanalyse* av systemer.



Det kan ofte være til stor nytte å utføre sensitivitetsanalyser for å evaluere effekten alternative antakelser, variasjon i feil- eller reparasjonsdata eller ulike systemkonfigurasjoner har på resultatene. I tillegg til ytelsesmålene er det viktig å kartlegge hvilke kritiske elementer som fins i systemet for å kunne utbedre påliteligheten til systemet ytterligere.

Usikkerheten i verdiene til de forutsatte ytelsesmålene skal diskuteres og om mulig kvantifiseres. Kvantifiseringen kan ta form av en usikkerhetsfordeling basert på forventningsverdien til ytelsesmålet eller en oversikt av spredningen av fordelingen som for eksempel standardavvik. Hovedfaktorene som fører til variabilitet og dermed usikkerhet i ytelsesmålene skal identifiseres og diskuteres. Faktorer som bidrar til usikkerhet rundt hvordan systemytelsen modelleres skal også identifiseres og diskuteres.

### **2.2.3 Datagrunnlag for tilgjengelighetsstudier**

Systematisk innsamling og behandling av erfaringer fra den daglige driften er regnet for å være en investering og et middel til forbedring av produksjons- og sikkerhetsutstyr. Hensikten med å etablere og opprettholde databaser med pålitelighetsdata er å kunne gi tilbakemeldinger som er med på å støtte opp under produktdesign, forbedringer i nåværende utstyr, utviklingen av vedlikeholdsprogram, identifiseringen av viktige faktorer til produksjonsutilgjengelighet og ikke minst forbedringen av antakelsene brukt til beslutningsstøtte.

For å kunne forutsi tid til feil eller tid til reparasjon for en gjenstand må man bestemme seg for en sannsynlighetsmodell. Modelltypen avhenger av hensikten med analysen. En eksponentiell levetidsfordeling (kapittel 4.1.3) kan for eksempel være hensiktsmessig. Hvis man skal reflektere over en tendens vil det være nødvendig å bruke en modell som tar hensyn til tidsavhengige feilrater. Etableringen av en feiltids- eller reparasjonstidsmodell skal basere seg på innsamlet pålitelighetsdata.

Etableringen av nøyaktig og relevant pålitelighetsdata krever at man utfører en datakvalifiseringsprosess. Denne prosessen innebærer at man forsøker å få tak i originale datakilder samtidig som man tolker alle tilgjengelige estimeringsmetoder og statistikk som er nødvendig for bruk i analysen. Utvelgelse av data skal blant annet basere seg på følgende prinsipper:

- Data skal stamme fra samme type utstyr.
- Data skal om mulig stamme fra identiske utstyrmodeller.
- Data skal stamme fra perioder med stabil drift som betyr at man helst ikke bør bruke data som er innhentet fra for eksempel oppstarten til et anlegg som kjennetegnes av mange startproblemer.
- Data skal om mulig stamme fra utstyr som har blitt utsatt for sammenlignbare drifts- og vedlikeholdsforhold.
- Mengden av feilhendelser brukt til å estimere eller forutsi pålitelighetsparametere bør være stor nok til å forhindre skjevfordeling av resultatene.

Data fra hendelsesdatabaser som for eksempel OREDA-databasen [5] møter kravene som stilles ovenfor. I tilfelle det er mangel på data er det nødvendig med kvalitativ ingeniørbedømmelse, og sensitivitetsanalyser av inndata bør gjennomføres.

## 2.3 Analyseverktøy

Når man skal utføre en regularitetsanalyse er det hensiktsmessig å ta i bruk enkelte analyseverktøy som hjelper til med å skaffe en tilstrekkelig oversikt av systemsammenhengene. To av de mest sentrale verktøyene er *feilmodus og -effekt analyse (FMEA)* og *feiltreanalyse*. Det er i hovedsak feiltreet som vil bli benyttet i forbindelse med metodikken i denne oppgaven, men det er som regel alltid greit å gjennomføre en FMEA for å lage et grunnlag for feiltreanalysen. Ved å utføre en FMEA vil man kunne vite mer om hva som forårsaker en tophendelse i et feiltre. Nedenfor vil de to verktøyene bli forklart nærmere ved hjelp av Rausand, M. et al ([6], s.73-93)

### 2.3.1 FMEA/FMECA

En av de første systematiske teknikkene for feilanalyse var FMEA – ”Failure Mode and Effects Analysis”. Den ble oppfunnet i 1950-årene for å studere feil i militærsystemer. FMEA er som regel det første ledd i en studie av systemets pålitelighet. Hovedfunksjonen til FMEA er å forhindre at produkt- og prosessproblemer oppstår. Metoden går ut på å gå gjennom så mange komponenter og undersystem som mulig for å identifisere feilmoduser, årsaker og effekter av slike feil. FMEA blir gjennomgått både i design- og produksjonsfasen. På denne måten kan man oppdage feil på et tidligst mulig tidspunkt slik at enkle og rimelige utbedringer kan gjennomføres [9].

Hvis kritikaliteter eller prioriteringer er tilknyttet feilmodusene vil en FMEA gå over til en FMECA – ”Failure Mode, Effects and Criticality Analysis”.

FMEA eller FMECA lister opp potensielle feil og identifiserer virkningen av effekten av disse feilene. På denne måten vil det dannes et grunnlag for en kvantitativ pålitelighets- og tilgjengelighetsanalyse.

Man kan utføre en FMEA på to ulike tilnæringsmåter

1. Topp-bunn tilnærming – Her starter man på systemnivå og jobber seg ned til komponentnivå.
2. Bunn-topp tilnærming – Her starter man på komponentnivå og jobber seg oppover.

De fleste FMEA og FMECA benytter seg av bunn-topp tilnærmingen, men topp-bunn

tilnærmingen vil i mange tilfeller være mer effektiv. Man deler inn i flere undersystem og sjekker hver av dem for feilmoduser før man går videre ned til komponentnivå og analyserer disse. På denne måten kan man gå bort fra et undersystem hvis det viser seg ikke å inneholde noen feilmoduser. Dette vil spare både tid og innsats i form av bemanning.

FMEA er en forholdsvis enkel analyse å gjennomføre som ikke behøver overoppsyn av spesielt kompetent personell. Men det er viktig at de som skal gjennomføre analysen er inneforstått med hvordan systemet virker og hvilke restriksjoner som forhindrer det fra å fungere. En vanlig framgangsmåte for FMEA og FMECA kan være som følger:

1. Definerings og avgrensning av systemet med hensyn til hvilke komponenter som er innenfor grensene og hvilke som ligger utenfor.
2. Definerings av hovedfunksjonene til systemet.
3. Beskrivelse av de operasjonelle modusene til systemet.
4. Nedbrytning til mindre undersystem som kan håndteres mest mulig effektivt.
5. Gjennomgang av funksjonsdiagram og tegninger for å finne sammenkoblinger mellom undersystemene. Sammenkoblingene kan deretter illustreres av blokkdiagram hvor hver blokk representerer et undersystem. For mer informasjon om blokkdiagram vises det til Rausand, M. et al. ([6], s.93-101).
6. Forberedelse av en komplett komponentliste for hvert undersystem.
7. Forklaring av de operasjonelle og miljømessige belastningene som kan påvirke systemet og de ulike funksjonene. Disse gjennomgås for å påpeke uheldige effekter de kan påføre systemet.

### 2.3.2 Feiltreanalyse

Denne teknikken ble utviklet av Bell Telephone Laboratories i 1962 i forbindelse med en sikkerhetsevaluering av utskytningssystemet for den interkontinentale ”Minuteman” raketten.

Teknikken baserer seg på en topp-bunn tilnærming av problemet hvor *topphendelsen* vil være en potensiell uønsket hendelse. Man kan da gå nedover i feiltreet og finne alle mulige årsaker til at hendelsen oppstår. Slike uønskede hendelser kan være miljøforhold, menneskelige feil, normale forhold som for eksempel slitasje, og spesielle komponentfeil. Resultatene av en slik analyse vil derfor kunne si noe om hvilke mulige kombinasjoner av uønskede hendelser som kan oppstå, men også sannsynligheten for at en kritisk hendelse skal oppstå i løpet av et gitt tidsrom.

I denne oppgaven er feiltreet forenklet til å inneholde kun de to logiske portene ”OG” og ”ELLER”:

- ”OG”-port: Hendelsene som går inn i denne porten må inntreffe samtidig for at man skal kunne få en gyldig avgitt effekt oppover i feiltreet som tilslutt kan føre til den uønskede topphendelsen.
- ”ELLER”-port: Porten gir ut gyldig avgitt effekt oppover hvis kun én av hendelsene som går inn i denne porten inntreffer.

Feiltreanalyse er et godt verktøy for å kartlegge systemtilgjengelighet. Det gir en god oversikt av sammenhengene i systemet. Man kan sette opp et feiltre for hver hendelse som får systemet til å stanse opp. Deretter kan man bruke likningene (4.33) og (4.34) til å sette opp en strukturfunksjon (kapitel 4.1.1) for systemet.

En feiltreanalyse utføres normalt i 5 trinn:

1. Definisjon av problemet og grensebetingelser.
2. Oppbygging av feiltreet.
3. Identifisering av minimum ”cut sets” og ”path sets”.
4. Kvalitativ analyse av feiltreet.
5. Kvantitativ analyse av feiltreet.

Det er veldig viktig å ha en klar definisjon på hva topphendelsen (uønsket hendelse) er. En slik analyse vil være av begrenset verdi hvis ikke topphendelsen gir svar på spørsmål som hva, hvor og når. Man ønsker å definere en topphendelse som ”gassutslipp i turbin A ved normal drift” i stedet for det mer diffuse ”gassutslipp i anlegget”. Dette vil gjøre det enklere å finne klare årsakssammenhenger til ”ulykken” enn om topphendelsen skulle være veldig generelt definert.

For å få en klarere oversikt over hvordan en topphendelse inntreffer er det vanlig å definere såkalte ”cut sets” og ”path sets”:

- ”Cut set”: Et sett med grunnleggende hendelser som vil føre til at topphendelsen inntreffer gitt at alle de grunnleggende hendelsene inntreffer samtidig.
- ”Path set”: Et sett av grunnleggende hendelser som vil føre til at topphendelsen ikke inntreffer hvis alle hendelsene samtidig ikke inntreffer.

Man kan da finne ut hvor mange grunnleggende hendelser som må inntreffe samtidig for at topphendelsen skal få utslag. Antallet grunnleggende hendelser i et ”cut set” vil da være ordenen i et ”cut set”.

Det finnes en algoritme man kan benytte seg av for å finne minimum ”cut sets” i et feiltre. Denne metoden kalles MOCUS (metode for å oppnå ”cut sets”) og blir grundigere forklart i vedlegg B. Denne metoden vil senere bli benyttet ved analysen av feiltrærne i kapittel 6.

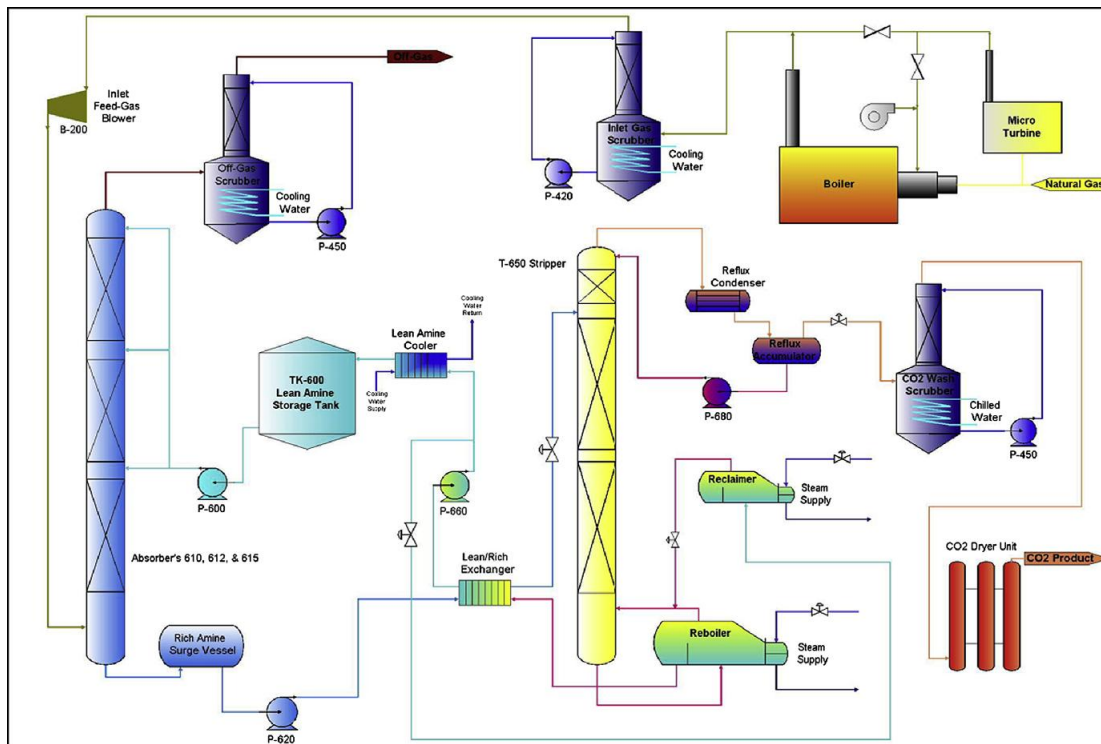
### 3 Prosessmodell

Selv om fangst av CO<sub>2</sub> er en velkjent teknologi er den ikke tatt i bruk i fullskala størrelse i et kraftproduserende anlegg enda. Ifølge IPCC sin spesialrapport om CO<sub>2</sub>-fangst og –lagring [10] fins det i dag en rekke gassprosesseringsanlegg og ammoniakkanlegg som separerer ut CO<sub>2</sub> for å tilfredsstille produktkrav, men det er svært lite relevant informasjon å finne om feilårsaker og systemoppbygning. Det vil derfor si at det er få industrielle erfaringer å støtte seg på om man skal gjøre en tilgjengelighetsanalyse av CO<sub>2</sub>-fangst fra kraftproduserende industri.

#### 3.1 CO<sub>2</sub>-rensaneanlegg for rensing av eksosgass

I denne oppgaven er det valgt å bruke et CO<sub>2</sub>-rensaneanlegg for rensing av eksosgass som utgangspunkt for analysen. Det er denne teknologien som er mest aktuell for CO<sub>2</sub>-fangst i kraftproduserende industri per i dag.

Det er viktig å si noe om sammenhengene i prosessen for å kunne få til et nøyaktig og godt resultat av simuleringene. Prosessmodellen som er vist i *figur 3.1* er hentet fra en artikkel skrevet av Wu, Y. et al. [11] og er benyttet i forsøksanlegget for CO<sub>2</sub>-fangst ved det internasjonale testsenteret for CO<sub>2</sub>-fangst på Reginauniversitetet i Canada. En av hovedårsakene til at nettopp denne prosessmodellen er benyttet i denne oppgaven er at den kjennetegner en typisk aminrensing for CO<sub>2</sub>-rensing av eksosgass. I tillegg fins det en beskrivelse av komponentoppbygningen av anlegget beskrevet av Zhou, Q. et al. [12] som er viktig når man skal sette opp en analyse av systemet.



Figur 3.1: Flyttdiagram for aminrensing av eksosgass.

Når man skal utføre en systemanalyse må man i utgangspunktet ha inngående kjennskap til hvordan systemet er bygd opp og hvordan det virker. I kapittel 2.2.2 gikk man gjennom en rekke krav til hvordan man skal kunne utføre en regularitetsanalyse. Innenfor omfanget av denne oppgaven har det hverken vært nok ressurser tilgjengelig, i form av tilgjengelig data som for eksempel systemskisser av anlegget, eller nok tid til å gjennomføre en så omfattende analyse av et renseanlegg for CO<sub>2</sub>-fangst. Det er likevel gjort et forsøk på å bygge en prosessmodell slik at man kan få vist metodikken utarbeidet i denne oppgaven på en god måte. Systemgrensene for prosessmodellen som blir brukt i denne oppgaven er gitt som følgende:

- *Startpunkt:* Definerer starten på systemet ved innløpet til gassviften (*figur 3.1*).
- *Sluttpunkt:* Det er definert flere sluttpunkt for systemet. Grensene går fra der eksosgassen går ut på toppen av absorberer og der CO<sub>2</sub>-gassen går ut av refluksopsamleren mot gassvaskeren (*figur 3.1*).
- I tillegg er dampkretsen regnet med i systemet på grunn av at dampleveransen er svært viktig for driften av anlegget.

I følge Zhou, Q. et al. [12] er komponentene i prosessmodellen i *figur 3.1* delt inn i tre ulike klasser hvorav alle gassvaskerne med tilhørende pumper, mikroturbinen og CO<sub>2</sub>-tørkeren er definert til å være utenfor systemgrensene:

#### **(1) Reaksjonskomponenter**

Disse består av 16 komponenter i forsøksanlegget. De er igjen delt inn i tre undergrupper basert på funksjonaliteten (tabell 3.1):

- *CO<sub>2</sub>-målgruppen:* Disse komponentene benyttes hovedsakelig til å prosessere karbondioksidgassen. Absorpsjonskolonnen absorberer CO<sub>2</sub> inn i den tynne aminløsningen fra røykgassen mens stripperkolonnen løser ut CO<sub>2</sub> fra den rike aminløsningen og regenererer den til å bli en tynn aminløsning igjen.
- *Amin-målgruppen:* Hovedfunksjonen til disse komponentene er å lagre og behandle den tynne og rike aminløsningen mens den sirkulerer gjennom hele CO<sub>2</sub>-fangstanlegget. Kjølere til den tynne aminløsningen vil kjøle ned den gjenvunne tynne aminløsningen til en påkrevd temperatur før den returneres til lagertanken.
- *Varme-målgruppen:* Dampkjelen og kokeren skal sørge for varmen som kreves til CO<sub>2</sub>-fangstprosessen. Dampkjelen varmer opp røykgassen under forbehandlingen og leverer damp til kokeren og regeneratoren. Kokeren sørger for varme til stripperen for å løse ut CO<sub>2</sub> fra den rike aminløsningen.

Reaksjonskomponenter		
CO <sub>2</sub> -målgruppen	Amin-målgruppen	Varme-målgruppen
Innløpsgassvasker	Lagertank for tynn amin	Dampkjel
Eksosgassvasker	Trykksvingningstank for rik amin	koker
Mikroturbin	Varmeveksler for rik/tynn amin	
Absorber	Kjøler for tynn amin	
Stripper	Regenerator	
Tilbakekjøler		
Refluks samletank		
CO <sub>2</sub> -vasker		
CO <sub>2</sub> -tørker		

Tabell 3.1: Oversikt av de ulike reaksjonskomponentene i aminreanseanlegget.

### (2) Pumper

Pumpene er delt inn i to kategorier basert på hvilket fluid som behandles:

- *Væskefordelingspumper*: Disse pumpene kan igjen deles inn i tre undergrupper som er avhengig av hvilken strømning de kontrollerer. Man har pumper som kontrollerer vannstrømning, aminløsningsstrømning og kjemikaliestrømning.
- *Gassviftepumper*.

### (3) Ventiler

Avhengig av kontrollmekanismen til ventilene kan man dele de inn i to grupper:

- *PID (proporsjonal-integral-derivat) kontrollventiler*: Disse kan igjen deles inn i fire undergrupper avhengig av hvilke typer fluid de håndterer. De fire gruppene er kontrollventiler for damplevering, amin, vann og gass.
- *Sylinderspoleventiler*: Alle sylinderspoleventiler brukes for å kontrollere vannstrømning.

Det er ikke lagt vekt på å forklare aminprosessen i denne oppgaven. For en generell prosessbeskrivelse for en aminreanseprosess av eksosgass kan man for eksempel benytte seg av Abu-Zahra, M.R.M. et al. [13].

## 3.2 Feilårsaker i aminrenseanlegget

I kapittel 2.2.2 ble det slått fast at man måtte ha en grundig forståelse av prosessene i det systemet som skulle analyseres for å kunne få et godt resultat av analysen. I denne oppgaven er det dessverre ikke tilstrekkelig med informasjon tilgjengelig til å kunne utføre en grundig og god FME(C)A. Man har likevel bruk for å vite hva som er de mest sentrale og kritiske feilårsakene i et CO<sub>2</sub>-fangstanlegg for å kunne sette opp et feiltre for systemet. Det er i denne sammenhengen ikke lagt vekt på å utarbeide en fullstendig FME(C)A i og med at det sentrale i denne oppgaven er å vise frem metodikken og ikke nødvendigvis gi konklusjoner på sammenhenger i aminanlegget. Det er gjort en del antakelser underveis for å kunne si noe om de ulike feilårsakene.

Det er vanskelig å utarbeide en regularitetsanalyse etter de kravene som er gitt i NORSOK standard Z-016 (kapittel 2.2) uten inngående kjennskap og forståelse av alle faktorer som fører til feil i anlegget. Derfor vil ikke resultatene som metodikken i denne oppgaven viser gi noen konkrete svar på tilgjengeligheten til et aminrenseanlegg. Det vil likevel være nødvendig å definere et sett med feilårsaker slik at man kan få satt opp en feiltrestruktur å basere metodikken på.

Feiltrestrukturen inkluderer for eksempel ikke mulige sikkerhetssystem på grunn av at det ikke var detaljerte systemskisser tilgjengelige for aminrenseanlegget. Man har også begrenset modellen til utelukkende å gjelde for den *funksjonelle driftsfasen* på grunn av at dataene som er hentet fra OREDA baserer seg på feilhendelser i denne fasen. Dette innskrenker bredden av metodikken ved at man har sett bort fra feilhendelser som kan oppstå ved oppstart eller når utstyr begynner å oppleve slitasje.

Identifiseringen av feilhendelsene og sammenhengene i aminanlegget er delvis basert på en forenklet FMECA (vedlegg C) som ble gjennomført av rapportforfatter høsten 2008 i tilknytning til en prosjektoppgave. FMECA-metoden er basert på en analyse som ble gjennomført av Nord, L.O. et al. [14] for et kraftverk med CO<sub>2</sub>-rensing. Metoden som er brukt ser nærmere på følgende kategorier: Undersystem, komponent, funksjon, funksjonelle krav, feilmodus, feilårsak, effekter på samme utstyr, effekter på annet utstyr, effekter på systemets helhetlige funksjon, deteksjon, feilrate, konsekvens og risiko.

Risiko er beregnet som et produkt av kategoriene deteksjon, feilrate og konsekvens hvorav hver av de tre kategoriene er delt inn i en skala fra 1 til 3:

Deteksjon:

- 1 = Lett påviselig
- 2 = Moderat påviselighet
- 3 = Upåviselig

Feilrate:

- 1 = Feil usannsynlig
- 2 = Sjelden feil



3 = Ofte feil

Konsekvens:

1 = Ingen effekt (eller veldig liten)  
2 = Anlegg opererer på delvis last  
3 = Stans av anlegget

Følgende antakelser og vurderinger ligger til grunn for analysen:

- Graden av deteksjon er vurdert ut ifra logiske betraktninger av anlegget. Et eksempel er at graden av deteksjon for stans av pumpe er satt til 1 i og med at det anses som enkelt å finne ut om en pumpe har stanset eller ikke.
- Graden av feilraten for komponenter som er representert i OREDA er gitt ut i fra en enkel vurdering av tallmaterialet til hver enkelt komponent i OREDA.
- Feilårsaker er i stor grad hentet fra OREDA.
- De funksjonelle kravene, som uttrykker konkrete tallverdier, til de ulike komponentene er hentet fra Bolland, O. [15] og Tobiesen, F.A. et al. [16].

I følge den forenklete FMECA (vedlegg C) som ble gjennomført for aminreanseanlegget kan det se ut som pumpene er de mest risikoutsatte komponentene. Det er ikke uvanlig at det er roterende utstyr som blir utsatt for de største påkjenningene i et anlegg slik at de også vil være mest utsatt for feil. Man har derfor i de fleste tilfeller også redundans på pumper slik at det vil være en eller flere pumper i reserve. Det er viktig å presisere at den analysen som er utført med tanke på feilmodus, effekter og kritikalitet av aminreanseanlegget er veldig generell og kan ikke brukes til å trekke gode konklusjoner på reell pålitelighet og tilgjengelighet.

Man kan trekke slutninger med at feilårsaksidentifisering og feiltreoppbyggingen i denne oppgaven er en nødvendig brikke som må på plass for at metodikken skal kunne være gjennomførbar. Skal man derimot vise reelle og troverdige resultater fra en analyse må identifisering og feiltreoppbyggingen gjennomføres med riktige data.

Feilhendelsene som er identifisert her er definert som ”alvorlig”. Det innebærer at de vil ha en innvirkning på systemet eller omgivelsene til systemet som resulterer i stans av anlegget. Følgende generelle feilårsaker er identifisert på bakgrunn av gjennomført FMECA (vedlegg C) og antakelser fra OREDA:

*Mekanisk svikt*

Dette er en feilhendelse som hovedsakelig inntreffer på roterende komponenter som pumper og vifter. I dette tilfellet velges det også å plassere ”feil ved åpning/lukking av ventiler” i denne kategorien.

*Korrosjon*

Korrosjon på varmevekslere, absorber og stripper er definert som en stor risiko for anlegget slik at det medfører stans. I virkeligheten vil det være forskjell i alvorlighetsgraden av

korrosjon avhengig av hvilken type korrosjon det er og hvor lenge det har vart.

*Groing*

I likhet med korrosjon er dette noe som hovedsakelig inntreffer i varmevekslere, absorber og stripper og vil også ha ulik grad av alvorlighet generelt sett.

*Ekstern lekkasje av prosessmedium*

Dette er i alle tilfeller antatt kritisk og fører til kontrollert stans av anlegget. Amin er et giftig stoff som man ikke ønsker skal lekke ut til omgivelsene, men i virkeligheten vil man ikke stanse et anlegg på grunnlag av mindre lekkasjer.

### 3.3 Datagrunnlag

Det er veldig viktig å ha gode og nøyaktige data tilgjengelig når man skal utføre en regularitetsanalyse. Problemet i denne oppgaven er at det i praksis ikke fins noen erfaringsdata fra CO<sub>2</sub>-renseanlegg. Det har blant annet vært diskusjoner med Det Norske Veritas [17] hvor det ble bekreftet at det stort sett er et fåtall av industriselskap som for eksempel StatoilHydro og Aker Clean Carbon i Norge som har data tilgjengelig på dette området. Aker Clean Carbon har foreløpig bare gjort forsøk på småskala anlegg. Det har som nevnt innledningsvis i denne oppgaven blitt opprettet kontakt med StatoilHydro [2] i forbindelse med å skaffe erfaringsdata fra aminprosesser, men de har ikke hatt mulighet til å gi ut noen opplysninger om dette.

På grunnlag av at det har vært vanskelig å skaffe gode feil- og reparasjonsdata er OREDA brukt som en kilde til erfaringsdata. OREDA består av en samling av pålitelighetsdata fra olje- og gassvirksomhet fra åtte forskjellige aktører. Hovedfokuset til OREDA er å fremstille gjennomsnittlige feilrater sammen med estimerte reparasjonstider for en rekke feilhendelser i ulike olje- og gassinstallasjoner.

I OREDA er det ikke tatt hensyn til feil i den tidlige fasen, eller installasjonsfasen som den også kalles (kapittel 4.1). I tillegg er det også antatt at mye av det utstyret som er undersøkt i OREDA er underlagt en vedlikeholds- eller bytteprosess. Det betyr at gjenstandene ofte byttes ut eller blir reparert før de når slitasjefasen. Dette resulterer i at hoveddelen av feilhendelsene i OREDA har sitt utspring fra den fasen med konstant feilrate. OREDA sier derfor at alle feilrateestimatene som gjengis i håndboken er basert på antakelsen om at feilratefunksjonen er konstant og uavhengig av tiden. Dette betyr igjen at en gjenstand er regnet som ”så god som ny” så lenge den er i drift (kapittel 4.2). Det må samtidig understrekes at det ikke er utført noen statistiske tester for å bekrefte antakelsen om konstant feilrate i OREDA.

Alle feilhendelsene i denne oppgaven er regnet for å ha en kritisk effekt på systemet som på et eller annet vis fører til stans av anlegget. Det medfører at feildataene som er hentet fra OREDA i *tabell 3.2* er basert på de dataene som går under kategorien ”critical” i OREDA.

Feilhendelsene som er identifisert i *tabell 3.2* er underlagt en rekke antakelser og forenklinger på grunn av at det bare er en liten andel av utstyret fra aminreanseanlegget som er representert i OREDA:

- Alle feilhendelsene for pumpene er antatt å være tilknyttet pumpetypen ”condensate processing centrifugal pump” bortsett fra feilhendelsen som innebærer ”ekstern lekkasje av prosessmedium” som er tilknyttet pumpetypen ”combined function centrifugal pump”.
- Feildata for gassviften er hentet fra typen ”Gas processing centrifugal pump”.
- Både for absorberen og stripperen er feildata hentet fra typen ”contactor”.
- Feildataene til feilhendelsene ”groing i absorber” og ”groing i stripper” er hentet fra feilhendelsen ”plugged/choked”.
- Både varmeveksleren for rik/tynn amin og kjøleren for tynn amin er antatt å være av typen ”shell and tube heat exchanger”.
- Reflukskondenseren er antatt å være av typen ”flash drum”.
- Kontrollventilen for damp til kokeren er antatt å være av typen ”globe valve”.
- Det var ikke oppført noen reparasjonstider for feilhendelsen ”ekstern aminlekkasje” for begge vekslerne. Derfor er det bare angitt en verdi på 75 timer for reparasjonstidene til vekslerne.

De konstante feilratene for de ulike feilhendelsene er presentert sammen med reparasjonstidene i *tabell 3.2*:

Nr	Feilhendelse	Feilrate [feil/år]	Reparasjons- tid [timer]	Feil per 1e6 timer
1	Utilstrekkelig varmetilførsel fra koker	0,1005648	2,8	11,48
2	Feil ved åpning av dampventil til koker	0,010512	15,9	1,2
3	Groing i absorber	0,027156	9,2	3,1
4	Groing i stripper	0,027156	9,2	3,1
5	Havari av reflukspumpe	0,028908	45	3,3
6	Vibrasjon i reflukspumpe	0,0454644	5	5,19
7	Korrosjon i Tynn/Rik varmeveksler	0,0184836	81,2	2,11
8	Korrosjon i kjøler for tynn aminløsning	0,0184836	81,2	2,11
9	Ekstern aminlekkasje i Pumpe for rik aminløsning	0,1146684	70,4	13,09
10	Ekstern aminlekkasje i boosterpumpe for tynn aminløsning	0,1146684	70,4	13,09
11	Ekstern aminlekkasje i sirkulasjonspumpe for tynn aminløsning	0,1146684	70,4	13,09
12	Ekstern aminlekkasje i Tynn/Rik varmeveksler	0,0090228	75	1,03
13	Ekstern aminlekkasje i kjøler for tynn aminløsning	0,0090228	75	1,03
14	Ekstern aminlekkasje i absorber	0,0677148	41	7,73
15	Ekstern aminlekkasje i stripper	0,0677148	41	7,73
16	Havari av Pumpe for rik aminløsning	0,028908	45	3,3
17	Kritisk vibrasjon i Pumpe for rik aminløsning	0,0454644	5	5,19
18	Havari av boosterpumpe for tynn aminløsning	0,028908	45	3,3
19	Kritisk vibrasjon i boosterpumpe for tynn aminløsning	0,0454644	5	5,19
20	Havari av sirkulasjonspumpe for tynn aminløsning	0,028908	45	3,3
21	Kritisk vibrasjon i sirkulasjonspumpe for tynn aminløsning	0,0454644	5	5,19
22	Havari av gassvifte	0,0706056	20,2	8,06
23	Kritisk vibrasjon i gassvifte	0,0281196	51	3,21

*Tabell 3.2: Nøkkeltall for ulike feilhendelser utarbeidet fra OREDA.*



## 4 Bakgrunnsteori til metodikken

I denne oppgaven er målet å komme fram til en metodikk som ser på tilgjengeligheten til et CO<sub>2</sub>-renseanlegg. Det er ønskelig å måle tilgjengeligheten som oppetid for anlegget over et gitt tidsrom. Det er derfor sett på tilgjengeligheten til et reparerbart system som vil veksle mellom å være i drift og ute av drift. Når anlegget er i drift vil det fungere slik det er påkrevd å gjøre helt til det feiler. I det øyeblikket en feil inntreffer som får systemet til å gå ut av drift vil anlegget gå inn i dødtiden. Systemet forblir i denne tilstanden helt til det er reparert og aktivert igjen. Tiden systemet er i operasjonell fase kalles *gjennomsnittlig tid til feil (MTTF)*, mens tiden systemet er nede kalles *gjennomsnittlig tid til reparasjon (MTTR)*.

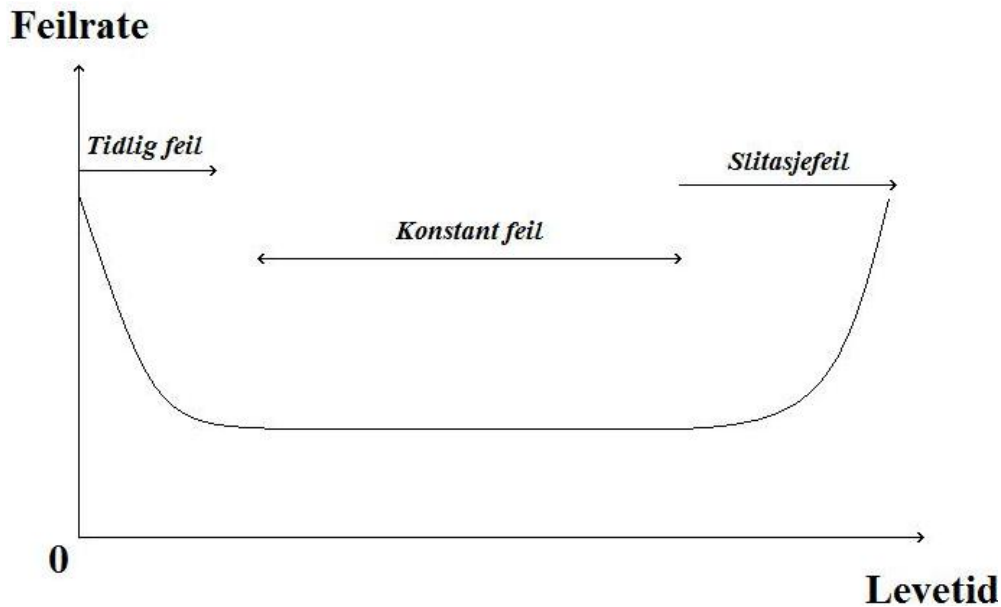
Dette kapittelet går gjennom en del bakgrunnsteori som må ligge til grunn for å forstå metodikken som blir presentert i kapittel 5. Det er hovedsakelig ulike fordelingsfunksjoner, systemanalyse og Monte Carlo simulering som blir forklart i dette kapittelet. Teorien i dette kapittelet er hovedsakelig hentet fra Rausand, M. et al. ([6], s.26-42,127-185).

### 4.1 Fordelingsfunksjoner

I dette delkapittelet blir ulike fordelingsfunksjoner for feilrater gjennomgått. I kapittel 3.3 ble det slått fast at feildataene som er benyttet i denne oppgaven har en konstant feilrate. Det vil si at observerte feil i anlegget kan beskrives rimelig godt med en eksponentialfordeling (kapittel 4.1.3). Det betyr likevel ikke at man kan si at levetiden til komponenten er eksponentialfordelt. Det er presentert flere fordelingsfunksjoner i dette kapittelet for å gi et mer helhetlig bilde av hvilke fordelingsfunksjoner som kan egne seg godt til bruk i modellering av tilgjengelighetsprosesser.

En feilfordeling har som mål å beskrive livsløpet til et materiale, en konstruksjon eller en komponent ved hjelp av en matematisk fremstilling. Det er mange ulike fysiske faktorer som har innvirkning på feil til en komponent. Det er derfor nærmest umulig å isolere disse fysiske årsakene og samtidig kunne redegjøre for dem alle rent matematisk. Et mål vil derfor være å finne et konsept som gjør det mulig å skille mellom de ulike fordelingsfunksjonene på bakgrunn av et fysisk utgangspunkt. Et slikt konsept vil være basert på feilratefunksjonen som i pålitelighetslitteratur er kjent som feilintensitet ([18], s.116). Utledning av feilratefunksjon er gjennomgått i kapittel 2.1 og feilraten er gitt av likning (2.4).

For lettere å kunne velge den funksjonelle formen på  $z(t)$  er det vanlig å dele feil inn i tre ulike klasser avhengig av alderen. I starten ved installasjon vil feilraten være høy, så vil den falle i løpet av normal operasjonell fase før den tilslutt øker når utstyret opplever forringelse. For mange typer av utstyr vil feilraten følge en ”badekar-kurve” (figur 4.1). Denne kurven har 3 forskjellige områder: (1) *tidlige feil*, (2) *konstante feil* og (3) *slitasjefeil*. Tidlige feil er hovedsakelig forårsaket av defekt utstyr og unøyaktig installasjon. Konstante feil skyldes ofte av tilfeldige svingninger i last som overgår den dimensjonerte styrken til utstyret. Slitasjefeil er logisk nok forårsaket av at utstyret blir forringet og slitt ([19], kap.7 s.11).



Figur 4.1: "Badekar-kurve" for feilraten

I denne oppgaven er det antatt at alle utstyrskomponenter kan inneha en av to mulige tilstander: funksjonell tilstand eller ute av drift. Man kan da sette opp en *strukturfunksjon* der tilstanden til komponent  $i$ ,  $i = 1, 2, \dots, n$ , kan beskrives med en binær variabel  $x_i$ :

$$X_i = \begin{cases} 1 & \text{hvis komponent } i \text{ er i funksjonell tilstand} \\ 0 & \text{hvis komponent } i \text{ er ute av drift} \end{cases} \quad (4.1)$$

I kapittel 4.2 vil man komme tilbake til strukturfunksjonen og sammenhengene i et system.

Man kan dele de ulike livsløpsfordelingene inn i to forskjellige familier avhengig av om feilraten er stigende eller avtakende:

- *IFR-fordelinger*: Stigende feilratefordelinger. Dette gjelder for livsløpsfordelingen  $F$  hvis  $-\ln(1 - F(t))$  er konveks for  $0 < t < F^{-1}(1)$ .
- *DFR-fordelinger*: Avtakende feilratefordelinger. Dette gjelder for livsløpsfordelingen  $F$  hvis  $-\ln(1 - F(t))$  er konkav for  $0 < t < F^{-1}(1)$ .

Når livsløpsfordelingen er kontinuert kan man skrive feilraten som:

$$z(t) = \frac{f(t)}{1-F(t)} = \frac{d}{dt} (-\ln(1 - F(t))) \quad \text{for } t > 0 \quad (4.2)$$

En differensiert konveks funksjon vil alltid ha en stigende verdi av den deriverte på samme måte som en differensiert konkav funksjon alltid har en avtakende verdi av den deriverte.

#### 4.1.1 Poissonprosess

Et poissoneksperiment er et eksperiment som resulterer i numeriske verdier av en tilfeldig variabel  $X$ , der  $X$  er antall utfall som inntreffer i løpet av et gitt tidsintervall eller i et

spesifisert område. Det gitte tidsintervallet kan være av ulik tidsskala på samme måte som det spesifiserte området kan være alt fra et areal, et volum eller for eksempel en del av et materiale. Et poissoneksperiment er utledet fra en *poissonprosess* som kjennetegnes av følgende egenskaper ([20], s.161-162):

- Antall utfall som inntreffer i et tidsintervall eller spesifisert område er uavhengig av antallet utfall som inntreffer i et annet tidsintervall eller område. På denne måten kan man si at poissonprosessen ikke har hukommelse.
- Sannsynligheten for at et enkelt utfall vil inntreffe i løpet av et veldig kort tidsintervall eller i et lite område er proporsjonalt med lengden av tidsintervallet eller størrelsen på området og vil ikke være avhengig av antall utfall som oppstår utenfor dette tidsintervallet eller området.
- Sannsynligheten for at mer enn et utfall vil inntreffe i et så lite tidsintervall eller område som nevnt i forrige punkt er neglisjerbart.

Om man studerer komponenter hvor reparasjonstiden er neglisjerbar vil  $X$  kunne være hendelsen som gjør at komponenten feiler. Prosessen har intensiteten  $\lambda$ .

En *poissonfordeling* kan skrives på følgende måte:

$$P(N(t) = n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \quad \text{for } n = 0, 1, 2, \dots \quad (4.3)$$

Denne fordelingen har forventningsverdi gitt som:

$$E(N(t)) = \sum_{n=0}^{\infty} n \cdot \frac{(\lambda t)^n}{n!} e^{-\lambda t} = \lambda t \quad (4.4)$$

Siden forventet antall hendelser av  $X$  per tidsenhet ( $t=1$ ) er  $\lambda$  sier man at  $\lambda$  uttrykker intensiteten til prosessen.

#### 4.1.2 Normalfordeling

Normalfordelingen er den mest benyttede fordelingen innenfor statistikken. I begynnelsen ble den benyttet som en tilnærming til binomialfordelingen. I starten av 1800-tallet ble normalfordelingen ukritisk akseptert som basis for mye praktisk statistisk arbeid, spesielt innenfor astronomi.

Normalfordelingen har en unik posisjon innenfor sannsynlighetsteorien og kan brukes som en tilnærming til andre fordelinger. Man kan i praksis benytte seg av ”normal teori” i tilfeller hvor ikke-normale fordelinger sammenfaller mest med observerte verdier, uten stor risiko for alvorlige feil ([21], s.45).

Det fins flere eksempler på at feildata tilsynelatende kan anta en form som tilsier at de er normalfordelte, men de fleste livsløpsfordelinger som oppstår i praktiske tilfeller er ikke normalfordelte på grunn av at de er skjevfordelte mens normalfordelingen er symmetrisk ([22], s.16).

En tilfeldig variabel  $T$  er normalfordelt med middelværdi  $\nu$  og varians  $\tau^2$ ,  $T \sim N(\nu, \tau^2)$ , når sannsynlighetstettheten til  $T$  er gitt som:

$$f(t) = \frac{1}{\sqrt{2\pi}\tau} e^{-\frac{(t-\nu)^2}{2\tau^2}} \quad \text{for } -\infty < t < \infty \quad (4.5)$$

Fordelingen gitt som  $N(0,1)$  kalles for standardnormalfordelingen og har sannsynlighetstetthetsfunksjonen:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \quad (4.6)$$

Sammenhengen mellom normalfordelingen gitt av  $T \sim N(\nu, \tau^2)$  og standardnormalfordelingen gitt som  $\Phi(\bullet)$  kan skrives på følgende form:

$$F(t) = P(T \leq t) = \Phi\left(\frac{t-\nu}{\tau}\right) \quad (4.7)$$

Normalfordelingen kan benyttes som livsløpsfordeling selv om den tillater negative verdier med positiv sannsynlighet. Viktige variable i tilknytning til pålitelighetsstudier er henholdsvis påliteligheten  $R(t)$  og feilraten  $z(t)$ :

$$R(t) = 1 - \Phi\left(\frac{t-\nu}{\tau}\right) \quad (4.8)$$

$$z(t) = \frac{1}{\tau} \cdot z_{\phi}\left(\frac{t-\nu}{\tau}\right) \quad (4.9)$$

Her er  $z_{\phi}(t)$  gitt som feilraten for standardnormalfordelingen.

Hvis en tilfeldig variabel er normalfordelt har den i mange tilfeller en øvre eller nedre avgrensning for verdiene i fordelingen. Denne fordelingen kalles en trunkert normalfordeling. Det er tre ulike typer trunkerte normalfordelinger:

1. Når det kun er en nedre grense er fordelingen *venstretrunkert*.
2. Når det kun er en øvre grense er fordelingen *høyretrunkert*.
3. Hvis det både er en nedre og øvre grense er fordelingen *dobbeltrunkert*.

Det er vanlig å benytte en venstretrunkert normalfordeling med grense i 0 som en livsløpsfordeling. Årsaken til dette er at det er normalt å starte tidsberegning i verdien 0 slik at negative verdier ikke vil ha noen mening i denne sammenhengen.

### 4.1.3 Eksponentialfordeling

I pålitelighetsstudier har eksponentialfordelingen en like viktig rolle som normalfordelingen har i andre områder av statistikken. Eksponentialfordelingen kan ikke tilstrekkelig beskrive alle tilfeldige livsløpsvariable, men i og med at teorien er lettfattelig vil den egne seg godt til behandling av generelle tilfeller. På grunn av fordelingen sin enkle oppbygning og tilknytning til den velutviklede teorien om poissonprosesser (kapittel 4.1.1) er den et godt valg for livsløpsstudier. Tilknytningen til poissonprosesser kjennetegnes ved at eksponentialfordelingen har et fullstendig tilfeldig feilmønster. Dette innebærer at, uavhengig



av feilårsak, vil feilen inntreffe i henhold til forutsetningene gitt av en poissonprosess med en parameter  $\lambda$  ([18], s.120).

Sannsynlighetstetthetsfunksjonen (PDF)  $f_X(t)$  til en eksponentialfordeling kan enten utledes ut i fra feilintensitetskonseptet eller ved å se på ventetid mellom hendelser i en poissonprosess. Feilintensitetskonseptet har den antakelsen at støt som inntreffer tilfeldig med parameteren  $\lambda$  medfører en konstant feilrate,  $z(t) = \lambda$ , for  $t \geq 0$  ([18], s.123-125).

Fra kapittel 2.1 er det vist at  $f_X(t)$  er gitt av følgende likning:

$$f_X(t) = z(t)e^{-\int_0^t z(s)ds} \quad (4.10)$$

Dette innebærer at man får følgende PDF for eksponentialfordelingen:

$$f_X(t) = \lambda e^{-\lambda t} \quad (4.11)$$

Mer generelt kan likning (4.11) skrives på følgende form ([18], s.123-125):

$$f_X(t) = \begin{cases} \lambda e^{-\lambda(t-A)} & \text{for } t \geq Y \\ 0 & \text{for } t < Y \end{cases} \quad (4.12)$$

Her er Y referert til som en skiftparameter og vil i de fleste tilfeller være satt lik 0.

Pålitelighetsfunksjonen blir da følgende:

$$R(t) = P(X \geq t) = \int_t^\infty f(s)ds = e^{-\lambda t} \quad (4.13)$$

Dette gir igjen den akkumulerte fordelingsfunksjonen (CDF):

$$F(t) = 1 - R(t) = 1 - e^{-\lambda t} \quad (4.14)$$

En nøkkelvariabel i denne oppgaven, MTTF, kan utledes på følgende måte:

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t} dt = \frac{1}{\lambda} \quad (4.15)$$

Man ser at hvis  $\lambda$  er stor vil MTTF bli liten og motsatt hvis  $\lambda$  er liten blir MTTF stor. Dette er årsaken til at  $\lambda$  blir kalt for intensiteten til prosessen.

Man ser at en komponent med et eksponentialfordelt livsløp har en konstant feilrate. Dette vil i mange tilfeller være realistisk for den funksjonelle levetiden til en komponent.

Sannsynligheten for at en komponent fungerer ved tiden  $t + \delta$ , gitt at den fungerer ved tiden  $t$ , er lik sannsynligheten for at en ny komponent har en tid til feil som er lengre enn  $\delta$ . Det vil si at gjenstående levetid for en enhet som virker ved tiden  $t$  vil være uavhengig av  $t$ .

Eksponentialfordelingen har derfor ikke noe "hukommelse".

Ved å anta at en komponent har et eksponentialfordelt livsløp må enkelte forutsetninger ligge til grunn:

- En enhet i bruk vil til enhver tid være definert som "så god som ny" slik at det ikke er noen grunn til å skifte ut en komponent som fungerer.
- Alderen til en komponent har ingen som helst betydning for å kunne estimere pålitelighetsfunksjonen, MTTF og feilraten.

Ekspontialfordelingen er den mest brukte livsløpsfordelingen innenfor tekniske pålitelighetsanalyser. Årsaken til dette er blant annet at den er veldig enkelt bygd opp rent matematisk og fører også til svært realistiske livsløpsmodeller for enkelte typer komponenter.

Ekspontialfordelingen hører hjemme både under IFR- og DFR-fordelingene på grunn av at den har en konstant feilrate. Det er faktisk den eneste kontinuerlige fordelingen som tilhører begge familiene.

#### 4.1.4 Gammafordeling

Hvis man tar utgangspunkt i en poissonprosess med intensitet  $\lambda$  vil tidsintervallet  $T_1, T_2, T_3, \dots$  mellom påfølgende støt på enheten være uavhengig og eksponentielt fordelt med parameteren  $\lambda$ . Antar man at enheten feiler akkurat ved støt  $k$  vil tiden til feil for komponenten være:

$$T = T_1 + T_2 + \dots + T_k$$

Da kan man si at  $T$  er gammafordelt  $(\alpha, \lambda)$  med PDF:

$$f(t) = \frac{\lambda}{\Gamma(\alpha)} (\lambda t)^{\alpha-1} e^{-\lambda t} \quad (4.16)$$

$\Gamma(\alpha)$  refererer til gammafunksjonen  $(\alpha, \lambda)$ ,  $t > 0$ ,  $\lambda > 0$ , og  $\alpha$  er et positivt heltall. Parameteren  $\lambda$  betegner støtintensiteten og er en ekstern parameter for enheten.  $\alpha$  kan tolkes som et mål for evnen til å motstå støt. Det kan også vises at MTTF og variansen blir:

$$MTTF = \frac{\alpha}{\lambda} \quad (4.17)$$

$$var(T) = \frac{\alpha}{\lambda^2} \quad (4.18)$$

For heltallsverdier av  $k$  kan man uttrykke CDF og feilraten som:

$$F(t) = 1 - R(t) = 1 - \sum_{x=0}^{\alpha-1} \frac{(\lambda t)^x}{x!} e^{-\lambda t} \quad (4.19)$$

$$z(t) = \frac{f(t)}{R(t)} = \frac{\frac{\lambda}{\Gamma(\alpha)} (\lambda t)^{\alpha-1} e^{-\lambda t}}{\sum_{j=0}^{\alpha-1} \frac{(\lambda t)^j}{j!} e^{-\lambda t}} \quad (4.20)$$

Gammafordelinger har den egenskapen at den kan etterlikne en normalfordeling ved å sette en stor verdi for  $\alpha$  samtidig som fordelingen representerer en positiv tilfeldig variabel ([21], s.172).

Når  $\alpha$  er definert som en heltallsverdi vil feilraten alltid være stigende (IFR) som en funksjon av  $t$ . Det refereres til Rausand, M. et al. ([6], s.62,471-472) for ytterligere forklaringer om gammafunksjonen.

### 4.1.5 Weibullfordeling

I mange tilfeller vil feilraten for mekanisk utstyr være svakt stigende i den perioden som kjennetegnes som konstant i ”badekar-kurven” i figur 4.1. I slike tilfeller vil ikke den eksponentielle fordelingen være en realistisk modell. Det er derfor vanlig å benytte seg av weibullfordelingen når feilraten ikke er konstant. Tiden til feil  $T$  for en enhet er weibullfordelt med parametre  $\alpha$  og  $\lambda$  (begge større enn 0) hvis fordelingsfunksjonen er gitt som:

$$F(t) = P(T \leq t) = \begin{cases} 1 - e^{-(\lambda t)^\alpha} & \text{for } t > 0 \\ 0 & \text{ellers} \end{cases} \quad (4.21)$$

Man kan finne sannsynlighetstettheten som:

$$f(t) = \frac{d}{dt} F(t) = \begin{cases} \alpha \lambda (\lambda t)^{\alpha-1} e^{-(\lambda t)^\alpha} & \text{for } t > 0 \\ 0 & \text{ellers} \end{cases} \quad (4.22)$$

Her er  $\lambda$  en skaleringsparameter og  $\alpha$  en formparameter. Pålitelighetsfunksjonen og feilraten er videre gitt som:

$$R(t) = P(T > 0) = e^{-(\lambda t)^\alpha} \quad \text{for } t > 0 \quad (4.23)$$

$$z(t) = \frac{f(t)}{R(t)} = (\alpha \lambda) (\lambda t)^{\alpha-1} \quad \text{for } t > 0 \quad (4.24)$$

MTTF er gitt som:

$$MTTF = \int_0^\infty R(t) dt = \frac{1}{\lambda} \Gamma\left(\frac{1}{\alpha} + 1\right) \quad (4.25)$$

Når  $\alpha = 1$  blir feilraten konstant og weibullfordelingen blir da redusert til en eksponentialfordeling. Når  $\alpha > 1$  er feilraten stigende, mens den er avtakende når  $0 < \alpha < 1$ . Dette kan vises ved å se på følgende sammenheng:

$$-\ln(1 - F(t)) = -\ln(e^{-(\lambda t)^\alpha}) = (\lambda t)^\alpha \quad (4.26)$$

$(\lambda t)^\alpha$  er konveks i tiden  $t$  når  $\alpha \geq 1$  og konkav i tiden  $t$  når  $\alpha \leq 1$ .

Hovedstyrken til weibullfordelingen er at den er fleksibel og kan derfor benyttes til å modellere livsløpsfordelinger hvor feilraten er avtakende, konstant eller stigende. Et annet navn på weibullfordelingen er ”svakeste-ledd-fordelingen”. Årsaken til dette er at den virker som en grensefordeling for den minste variabelen av en stor mengde uavhengige og identisk fordelte variable.

I følge Johnson, N.L. et al. ([21], s.251) er det ikke uvanlig å bytte ut en eksponentialfordeling med en weibullfordeling for en teoretisk modell som beskriver variasjonen av en observerbar størrelse som for eksempel ”levetiden”, når man ønsker å ta hensyn til mulige unøyaktigheter i modellen. Årsaken til dette er at man ved å introdusere parameteren  $\alpha$  får tillagt modellen mer fleksibilitet

Weibullfordelingen blir ofte benyttet innefor pålitelighetsanalyser av halvledere, kulelager, motorer, punktsveiser og biologiske organismer.

### 4.1.6 Lognormalfordeling

Tiden til feil  $T$  til en enhet er lognormalfordelt hvis  $Y = \ln T$  er normalfordelt.

Sannsynlighetstettheten er gitt som:

$$f(t) = \begin{cases} \frac{1}{\sqrt{2\pi}} \frac{1}{\tau t} e^{-\frac{(\ln t - \nu)^2}{2\tau^2}} & \text{for } t > 0 \\ 0 & \text{ellers} \end{cases} \quad (4.27)$$

Her er  $\nu$  og  $\tau^2$  henholdsvis middelveien og variansen til  $Y$ . Videre kan det vises at MTTF er gitt som:

$$MTTF = e^{\nu + \frac{\tau^2}{2}} \quad (4.28)$$

Pålitelighetsfunksjonen og feilraten av lognormalfordelingen kan ved hjelp av standardnormalfordelingen skrives som:

$$R(t) = P(T > t) = P(\ln T > \ln t) = P\left(\frac{\ln T - \nu}{\tau} > \frac{\ln t - \nu}{\tau}\right) = \Phi\left(\frac{\nu - \ln t}{\tau}\right) \quad (4.29)$$

$$z(t) = -\frac{d}{dt} \left( \ln \Phi\left(\frac{\nu - \ln t}{\tau}\right) \right) = \frac{\phi\left(\frac{\nu - \ln t}{\tau}\right)}{\tau t \cdot \Phi\left(\frac{\nu - \ln t}{\tau}\right)} \quad (4.30)$$

$\Phi(\bullet)$  og  $\phi(\bullet)$  er henholdsvis fordelingsfunksjonen og sannsynlighetstettheten til standardnormalfordelingen. I lognormalfordelingen vil  $z(t) \rightarrow 0$  når  $t \rightarrow \infty$ .

Lognormalfordelingen er ofte benyttet som fordeling for reparasjonstider. Reparasjonsraten er definert på samme måte som feilraten og det er vanlig å anta at reparasjonsraten øker i startfasen. Det betyr at sannsynligheten for å fullføre reparasjonen innenfor et kort tidsintervall øker med forløpt reparasjonstid. Når reparasjonen har pågått over en lengre periode indikerer det alvorlige problemer, som for eksempel at det ikke fins reservedeler tilgjengelige på området. Man kan derfor si at reparasjonsraten avtar etter en bestemt tid.

## 4.2 Systemanalyse

Det er vanlig å utføre en pålitelighetsanalyse av systemer for å kunne si noe om systempåliteligheten i tillegg til å identifisere kritiske deler av systemet. Det er spesielt viktig å identifisere de ulike undersystemene som har størst påvirkning på påliteligheten til hele systemet. I en slik analyse er det nødvendig å ha data om både feilrater så vel som reparasjonstider ([19], kap.7, s.21).

Det er ikke like lett å forutsi med sikkerhet om en komponent er i operasjonell tilstand etter  $t$  tidsenheter. Når man utfører slike undersøkelser er det derfor viktig å se etter statistisk regularitet. Man velger derfor å tolke tilstandsvariablene til de  $n$  komponentene ved tid  $t$  som tilfeldige variable. Tilstandsvariablene er gitt som:

$$X_1(t), X_2(t), \dots, X_n(t)$$

Tilstandsvektoren  $\mathbf{X}(t) = (X_1(t), \dots, X_n(t))$  og strukturfunksjonen  $\phi(\mathbf{X}(t))$  er andre viktige funksjoner. I tillegg vil følgende sannsynligheter være relevante i denne sammenhengen:

$$P(X_i(t) = 1) = p_i(t) \quad \text{for } i = 1, 2, \dots, n \quad (4.31)$$

$$P(\phi(\mathbf{X}(t)) = 1) = p_s(t) \quad (4.32)$$

Det vil i denne oppgaven være fokus på å studere systemer hvor feil på de individuelle komponentene kan tolkes som uavhengige hendelser. Dette er en forenkling som er tatt for å kunne benytte seg av det metodiske grunnlaget som er gitt i dette kapittelet. Hvis man skulle antatt at alle hendelser var avhengige av hverandre hadde metodikken blitt veldig omfattende rent matematisk og i tillegg mindre oversiktlig.

Måten man undersøker systemene på er avhengig av hvordan man tolker livsløpet til systemet og de ulike komponentene. Det er hovedsakelig to måter å tolke dette på:

- 1 *Ikke-reparerbare komponenter og systemer:* Dette karakteriserer et system som stanser helt opp slik at undersøkelsen avsluttes når en feil oppstår. I dette tilfellet vil likning (4.31) og (4.32) uttrykke *påliteligheten* ved tid  $t$  til henholdsvis komponent  $i$  og systemet.
- 2 *Reparerbare komponenter og systemer:* Dette karakteriserer komponenter og et system som blir byttet ut eller reparert med en gang det oppstår en feil. I dette tilfellet vil likning (4.31) og (4.32) uttrykke *tilgjengeligheten* ved tid  $t$  til henholdsvis komponent  $i$  og systemet.

Strukturfunksjonen til systemet kan uttrykkes som en seriestruktur, parallellstruktur eller en kombinasjon av begge. Serie- og parallellstrukturen har følgende funksjoner:

$$\phi(\mathbf{X}(t)) = \prod_{i=1}^n X_i(t) \quad (4.33)$$

$$\phi(\mathbf{X}(t)) = \prod_{i=1}^n X_i(t) = 1 - \prod_{i=1}^n (1 - X_i(t)) \quad (4.34)$$

I denne oppgaven er det sett på et system som har reparerbare komponenter. Årsaken til dette er at det vil være en mer realistisk tilnærming til virkeligheten. Man kan på denne måten finne ut hvor mye enkelte feilhendelser har å si for tilgjengeligheten til hele systemet.

#### 4.2.1 Reparerbare systemer

Vedlikehold på reparerbare systemer er normalt delt inn i to grupper:

1. *Korrektivt vedlikehold:* Dette er vedlikehold som man vanligvis forbinder med *reparasjon* etter en feil har inntruffet. Hensikten med denne typen vedlikehold er å sette komponenten, så raskt som mulig, tilbake til den tilstanden den var i før feil inntraff. Denne typen vedlikehold innebærer også i enkelte tilfeller bytte av komponenter.

2. *Forebyggende vedlikehold*: Dette er vedlikehold som har til hensikt å redusere sannsynligheten for feil i en komponent før feil inntreffer. Det kan innebære alt fra små endringer til bytte av komponenter som begynner å bli slitt ut.

Når en komponenten eventuelt feiler vil den bli byttet ut med en ny komponent eller reparert til "så god som ny" tilstand. Dette innebærer at man får en sekvens av levetider  $T_1, T_2, \dots$  som er både uavhengige og identisk fordelte med fordelingsfunksjon  $F_T(t) = P(T_i \leq t)$ ,  $i = 1, 2, \dots$ , og *gjennomsnittlig tid til feil* MTTF. Når en komponent feiler vil den være ute av drift i en gitt periode kalt *dødtiden*. På samme måte som med levetidene er reparasjonstidene  $D_1, D_2, \dots$  både uavhengige og identisk fordelte med fordelingsfunksjon  $F_D(t) = P(D_i \leq t)$ ,  $i = 1, 2, \dots$ , og *gjennomsnittlig tid til reparasjon* MTTR. Videre antas det at  $T_i + D_i$  for  $i = 1, 2, \dots$  er uavhengige. For et reparerbart system der reparasjonstiden ikke kan neglisjeres kan man definere *gjennomsnittlig til mellom feil* MTBF som:

$$MTBF_i = MTTF_i + MTTR_i \quad \text{for } i = 1, 2, \dots \quad (4.35)$$

Etter å ha observert en komponent helt til reparasjon  $n$  er gjennomført har man fått sekvensen  $T_1, T_2, \dots, T_n$  av levetider og sekvensen  $D_1, D_2, \dots, D_n$  av reparasjonstider. Man kan da anta følgende sammenhenger:

$$\frac{1}{n} \sum_{i=1}^n T_i \xrightarrow{P} E(T) = MTTF \quad \text{når } n \rightarrow \infty \quad (4.36)$$

$$\frac{1}{n} \sum_{i=1}^n D_i \xrightarrow{P} E(D) = MTTR \quad \text{når } n \rightarrow \infty \quad (4.37)$$

Ut i fra dette kan man se at andelen av tiden komponenten har vært i drift er gitt som:

$$\frac{\left(\frac{1}{n}\right) \sum_{i=1}^n T_i}{\left(\frac{1}{n}\right) \sum_{i=1}^n T_i + \left(\frac{1}{n}\right) \sum_{i=1}^n D_i} = \frac{E(T)}{E(T)+E(D)} \quad \text{når } n \rightarrow \infty \quad (4.38)$$

I pålitelighetslitteratur er likning (4.38) ofte benyttet som definisjon på *gjennomsnittlig tilgjengelighet*  $A_{av}$ :

$$A_{av} \stackrel{\text{def}}{=} \frac{E(T)}{E(T)+E(D)} = \frac{MTTF}{MTTF+MTTR} \quad (4.39)$$

Den begrensede tilgjengeligheten  $A$  er definert som:

$$A = \lim_{t \rightarrow \infty} A(t) \quad (4.40)$$

Vanligvis vil  $A(t)$  konvergere raskt mot  $A$  og man kan i de fleste tilfeller anta  $A = A_{av}$ .

For et system er det ulike sammenhenger som gjelder for MTTF og MTTR. Hvis man antar at reparasjonstiden er neglisjerbar kan man benytte seg av følgende sammenheng for *gjennomsnittlig tid til systemfeil*  $MTTF_s$  i en eksponentialfordelt seriestruktur:

$$\begin{aligned} MTTF_s &= \int_0^\infty R_s(t) dt = \int_0^\infty e^{-\int_0^t \sum_{i=1}^n \lambda_i(u) du} dt \\ &= \int_0^\infty e^{-\sum_{i=1}^n \lambda_i t} dt = \frac{1}{\sum_{i=1}^n \lambda_i} \end{aligned} \quad (4.41)$$

Hvis man har redundans av komponenter vil man få en parallellkobling som gir følgende sammenheng for  $MTTF_s$ :

$$MTTF_s = \sum_{i=1}^n MTTF_i \quad (4.42)$$

For et reparerbart system der man ikke kan se bort fra reparasjonstiden kan man benytte seg av feiltreorien rundt "cut sets" (kapittel 2.3.2) for å finne  $MTTR_s$  og  $MTBF_s$ . Anta at man ser på en spesifikk "minste-cut-parallellstruktur"  $K_j$ , for  $j=1,2, \dots k$ . Når alle komponentene i "cut set"  $K_j$  er i feilstadiet kan man si at man har en "cut set feil". På grunnlag av antakelsen om uavhengighet og "hukommelsesfrihet" for eksponentialfordelingen (kapittel 4.1.3) kan man uttrykke gjennomsnittlig tid til reparasjon for "minste-cut-parallellstruktur"  $K_j$  som:

$$MTTR_j = \frac{1}{\sum_{i \in K_j} \mu_i} \quad (4.43)$$

Her uttrykker  $\mu_i$  reparasjonsraten til den individuelle komponenten  $i$  i "cut set"  $K_j$ .

Videre kan man uttrykke raten av forekomster av feilhendelser (ROCOF) i "cut set"  $K_j$  som:

$$\omega_j = \frac{1}{MTBF_j} \quad (4.44)$$

Hvis man antar at alle  $k$  "minste-cut-parallellstrukturer" er uavhengige og alle dødtidene er neglisjerbare kan ROCOF for systemet uttrykkes som:

$$\omega_s = \sum_{j=1}^k \omega_j \quad (4.45)$$

For systemer med høy tilgjengelighet vil likning (4.45) være en god tilnærming til ROCOF for systemet selv om man ikke kan neglisjere dødtidene. Da kan man uttrykke gjennomsnittlig tid mellom feil for systemet som:

$$MTBF_s \approx \frac{1}{\omega_s} \quad (4.46)$$

Tilslutt kan man benytte seg av likning (4.45) til å sette opp et uttrykk for gjennomsnittlig tid til reparasjon for systemet:

$$MTTR_s \approx \frac{\sum_{j=1}^k \omega_j MTTR_j}{\sum_{j=1}^k \omega_j} \quad (4.47)$$

## 4.2.2 Monte Carlo simulering

I modellering av reparerbare systemer antas det som regel at komponentene er uavhengige av hverandre. Dette blir nokså urealistisk i mange sammenhenger og man må benytte seg av andre tilnæringsmåter for å anslå tilgjengeligheten til et system. I tillegg er enkelte systemer for komplekse og uoversiktlige til at man kan beregne sannsynlighetssammenhenger analytisk. Monte Carlo simulering er en teknikk som baserer seg på generering av ulike tilfeldige og diskontinuerlige hendelser i en datamaskinmodell for så å skape et realistisk livsløpsscenario for et system. Man skaper dermed en realisering av livssyklusen til systemet. Etter å ha observert den simulerte prosessen over tid kan man danne seg et bilde av viktige ytelsesfaktorer som gjennomsnittlig tilgjengelighet og gjennomsnittlig antall feil per tidsenhet. Simuleringen er altså en serie av reelle eksperimenter. Man kan enten simulere hendelsene med fastlåste eller variable tidskritt. I denne oppgaven blir metoden med fastlåste

tidsskritt  $\Delta t$  benyttet og simuleringen starter i tidspunktet  $t_0$ . Hvis en hendelse inntreffer i intervallet  $(t_0, t_0 + \Delta t)$  vil systemtilstanden lagres og nødvendig statistisk informasjon hentes ut. Deretter blir simuleringen økt med  $\Delta t$  igjen helt til totaltiden  $T$  har utløpt.



## 5 Metodikk for beskrivelse av tilgjengelighet

I denne oppgaven er det utviklet en metodikk som skal identifisere operasjonell tilgjengelighet til ulike feiltrestrukturer. Metodikken er delvis basert på en lignende metode som er vist i Thangamani, G. et al. [1]. Simuleringsprogrammet MATLAB [23] er benyttet til å utføre Monte Carlo simuleringen og til å beregne og vise resultatene. Dette kapitlet viser i hovedsak hvordan metodikken er bygd opp og vil i tillegg gå inn på sensurering av data. Helt til slutt kan man i *figur 5.1* se et flytskjema for simuleringsprosessen.

### 5.1 Hovedstruktur

Metodikken i denne oppgaven har som hovedmål å identifisere operasjonell tilgjengelighet til feiltrestrukturer basert på feilhendelsene som ble identifisert i *tabell 3.2*. Når man skal beregne tilgjengeligheten til en komponent eller et system er det viktig å definere hva som skal inngå av vedlikeholdsarbeid og ikke minst hva som anses som total tid for simuleringen. I denne oppgaven har man valgt å ta med forebyggende vedlikehold (kapittel 4.2.1) i form av en revisjonsstans hvert andre år i tillegg til det korrigerende vedlikeholdet som skjer underveis i prosessen. Det medfører at anlegget blir ”så godt som nytt” etter hver revisjonsstans. Samtidig er det viktig å understreke at for å kunne få mest mulig nøyaktige verdier av MTTF, må simuleringstiden være mye lengre enn det som anses som driftstid for anlegget mellom hver revisjonsstans. Man skiller derfor mellom total simuleringstid  $T_t$  og total driftstid mellom revisjonsstans  $T_d$  for anlegget. Kapittel 5.3 går nærmere inn på hvordan man har kommet fram til total simuleringstid for prosessen.

Man kan dele metodikken inn i to hoveddeler:

1. *Oppsett av feiltrær for systemet:* I denne prosessen settes feilhendelsene fra *tabell 3.2* i sammenheng til en feiltrestruktur (kapittel 2.3.2) avhengig av tilknytningen mellom årsakssammenhengene som fører til den uønskede topphendelsen. I denne oppgaven er det tre feiltrær med en topphendelse som fører til stans av systemet. De tre topphendelsene er: *Systemstans som følge av for lav effektivitet/slitasje*, *systemstans som følge av aminutslipp til omgivelsene* og *systemstans som følge av stopp i prosesstrømning*. I feiltreet kontrollerer man om de enkelte feilhendelsene er koblet i parallell (redundans) eller serie ved hjelp av logiske porter (kapittel 2.3.2). Tilslutt beregner man hvert enkelt feiltre sitt system av ”cut sets” på samme måte som man har gjort i eksempelet i vedlegg B.
2. *Monte Carlo simulering av prosessen:* I denne delen benytter man seg av simuleringsprogrammet MATLAB [23] til å simulere en livsløpsprosess for systemet. Kildekoden for simuleringsprosessen ligger i vedlegg D. Flytskjema for simuleringsprosessen er vist i *figur 5.1* i kapittel 5.5. Hovedsakelig går simuleringsprosessen ut på å sammenlikne feilraten til hver enkelt feilhendelse i feiltrærne med et tilfeldig generert tall i intervallet [0,1] for hvert tidsskritt som

forløper. Ut ifra resultatet av denne sammenlikningen vil programmet registrere feil eller ikke feil for den gitte feilhendelsen i det gitte tidsskrittet. Tilslutt kan man benytte ulike likninger og sammenhenger for å komme fram til operasjonell tilgjengelighet for totalsystemet og gjennomsnittlig tid til feil for enkelthendelser, feiltrær og totalsystemet. Detaljert beskrivelse av hvordan man kommer fram til de ulike leddene i simuleringsprosessen er gitt i kapittel 5.5.

På grunn av at feilratene som er benyttet i denne oppgaven er konstante (kapittel 3.3) kan man anta at observasjonen av feilhendelser er tilnærmet eksponentialfordelt. I teoridelen om systemanalyse i kapittel 4.2 er de fleste likninger gjeldende for ”komponenter”. I denne oppgaven er strukturen til feiltrærne bygd opp av ”feilhendelser” og ikke ”komponenter”. Det vil si at det blir vanskelig å se på levetiden til en komponent i seg selv. I og med at man i en eksponentialfordeling antar at alle feil er uavhengige (kapittel 4.1.3) kan det antas at alle feilhendelser er eksponentialfordelte. Det innebærer at man kan benytte seg av formelverket beskrevet i kapittel 4.2 i beregningen av MTTF, MTTR og MTBF for systemet. Man kan bruke simulerte MTTF for systemet til å anslå en fordeling for levetiden til anlegget når man ser bort fra forebyggende vedlikehold. Denne empiriske fordelingsfunksjonen kan en sammenlikne med fordelingsfunksjoner av de typene som er beskrevet i kapittel 4.1 for å se om man finner en sammenheng.

En svakhet i metodikken er at det ikke er utarbeidet en god modell for simulering av reparasjonstider. Reparasjonstidene til de ulike komponentene er i simuleringen satt lik en konstant verdi for hver feilhendelse (kapittel 3.3). Dette er en forenkling som er gjort hovedsakelig på grunn av at det har vært vanskelig å finne nødvendig informasjon om hvordan selve simuleringen og modelloppbygningen skulle gjennomføres. I utgangspunktet ville for eksempel en lognormalfordelt reparasjonstid være et godt alternativ (kapittel 4.1.6). Prosessen med å finne gode estimater for parametrene i lognormalfordelingen krever derimot stor forståelse for de ulike statistiske modellene. Det har derfor ikke blitt lagt vekt på å finne gode modeller for reparasjonstider i denne oppgaven. Som en forenkling av simuleringsprosessen har man antatt at reparasjonstidene er konstante verdier for hver enkelt feilhendelse og derfor ikke underlagt en eksponentiell fordeling slik som feilhendelsene.

Metodikken er satt opp slik at man ser på problemer som fører til full stans av anlegget. Det er imidlertid ofte minst like interessant å se på problemer som fører til en reduksjon i driftskapasiteten, slik at anlegget for eksempel kjører på 80% eller 60% av maksimal kapasitet. Slik metodikken er satt opp er det ikke mulig å endre driftsgraden uten å endre feilratene. Feilhendelsene og tilhørende feilrater er definert med utgangspunkt i at de fører til feil som stanser anlegget. Det er selvfølgelig mulig å endre feilhendelsene og feilratene til å omfatte hendelser som fører til degradering av driften. Det er derimot ikke like enkelt å definere feilhendelser som medfører lavere driftskapasitet, der man i tillegg skal vite i hvor stor grad driften reduseres. Spesielt innenfor kraftproduserende industri er kapasiteten til anleggene avgjørende for om anlegget er drivverdig eller ikke. På samme måte kan man si at rensegraden av CO<sub>2</sub> vil være viktig for om man skal kunne drive renseanlegget ”lønnsomt” eller ikke. Kostnadene knyttet til bygging og drift av et CO<sub>2</sub>-reanseanlegg er svært høye i tillegg til at man sitter igjen med et uønsket produkt som må transporteres bort for lagring.

Derfor er det viktig å identifisere og forbedre de elementene som fører til størst dødtid av anlegget.

## 5.2 Ulike oppsett av systemet

Det vil i denne oppgaven simuleres ulike oppsett av feiltrærne for å se om man kan finne sammenhenger mellom komponentoppsett og tilgjengeligheten til systemet. Det vil bli sett på tre forskjellige oppsett av systemet:

### 1. Hovedoppsett:

Her vil alle feilhendelser være koblet i serie slik at det er nok at bare en feilhendelse inntreffer for at topphendelsen skal inntreffe. Dette vil i utgangspunktet være det systemet med lavest tilgjengelighet i og med at det ikke er redundans av noen komponenter i systemet.

### 2. Redundans av alle pumper i aminkretsen:

I dette oppsettet vil man sette inn en pumpe i reserve (redundans) for alle de tre hovedpumpene i aminkretsen: *Pumpe for rik aminløsning*, *boosterpumpe for tynn aminløsning* og *sirkulasjonspumpe for tynn aminløsning*. Det er valgt å sette pumpene i redundans fordi det typisk er denne typen utstyr som er utsatt for flest feil i et anlegg (kapittel 3.2). Dette vil i utgangspunktet gi en høyere systemtilgjengelighet enn hovedoppsettet i og med at en ny pumpe overtar hvis en annen feiler.

### 3. Seriestructur med feilhendelser som har redusert feilrate:

I dette oppsettet antas det at man har klart å forbedre designet til enkelte komponenter i systemet som gjør at disse har en mye lavere feilrate. Poenget med dette oppsettet er å se hvor mye feilratene til ulike feilhendelser må reduseres i hovedoppsettet for at systemet skal ha en best mulig tilgjengelighet.

## 5.3 Total simuleringstid og antall simuleringer

Når man opererer med feilhendelser som har en relativt lav feilrate som i denne oppgaven (*tabell 3.2*) er det vanskelig å få fram en realistisk verdi for MTTF til feilhendelsene hvis man ikke har en lang simuleringstid. I denne oppgaven er det i likhet med vurderingene gjort i Thangamani, G. et al. [1] antatt at aminanlegget gjennomgår en revisjonsstans hvert andre år med en reparasjonstid på  $T_s = 1460$  timer (=2mnd).

Feilratene i *tabell 3.2* viser at feilhendelsen med størst feilrate antyder én feil i løpet av ca 9 år. Det vil derfor være nødvendig å ha en simuleringstid som er mye lengre enn to år for å kunne få realistiske verdier for MTTF til feilhendelsene og systemet. I *tabell 5.1* kan man se

hvor stor andel av feilhendelsene som har lengre MTTF enn den totale simuleringstiden ved ulike simuleringstidspunkt for hovedoppsettet.

**Andelen av feilhendelser som har høyere MTTF enn totaltiden for simuleringen (*cr*)**

<b>HOVEDOPPSETT</b>						
Simuleringsperiode	2 år		10 år		50 år	
Antall simuleringer, <i>p</i>	1000	10000	1000	10000	1000	10000
%-andel, <i>cr</i>	<b>91,35</b>	<b>91,46</b>	<b>65,56</b>	<b>65,33</b>	<b>21,3</b>	<b>21,29</b>
Simuleringsperiode	100år		150 år		300 år	
Antall simuleringer, <i>p</i>	1000	10000	1000	10000	1000	10000
%-andel, <i>cr</i>	<b>8,66</b>	<b>8,33</b>	<b>4,13</b>	<b>4,11</b>	<b>0,75</b>	<b>0,77</b>

*Tabell 5.1: Andelen feilhendelser som har MTTF lengre enn den totale simuleringstiden ved ulike simuleringstidspunkt og ulikt antall simuleringer.*

*Tabell 5.1* viser at andelen feilhendelser som ”overlever” simuleringstidspunktet er forholdsvis høy frem til simuleringstidspunktet passerer 100 år. Da vil ”overlevelsesandelen” synke under 10%. Ved en simuleringstidspunkt på 300 år vil nesten alle beregningene av MTTF for feilhendelsene være innenfor total simuleringstid slik at resultatene blir mest mulig nøyaktige. Derfor velges det å kjøre simuleringen med en total simuleringstid på  $T_t = 2628000$  timer (=300 år).

I tillegg kan man variere antall simuleringer som skal utføres for hvert system i en simuleringstidspunkt. Beregningstiden for simuleringen blir naturligvis lengre med et større antall simuleringer. Ut ifra *tabell 5.1* kan man slå fast at det er marginal forskjell i resultatet for *cr* om man velger  $p = 10000$  simuleringer i stedet for  $p = 1000$  simuleringer. Man har derfor valgt å kjøre simuleringsalgoritmen med  $p = 1000$  simuleringer for hver simuleringstidspunkt. Det er ikke lagt stor vekt på effektiviteten til simuleringsalgoritmen i denne oppgaven utover det at den må kunne beregne resultatene innenfor ”rimelig” tid.

## 5.4 Beregning av operasjonell tilgjengelighet

Det er viktig å definere hvilken type tilgjengelighet man er ute etter å finne når man skal analysere et system. Tilgjengeligheten som er gitt av likning (4.39) er en god definisjon på tilgjengelighet til enkeltkomponenter og systemer hvis man ser bort fra forebyggende vedlikehold. I de fleste tilfeller vil det derimot være en eller annen form for forebyggende vedlikehold på et anlegg som er såpass komplisert som et CO<sub>2</sub>-rensaneanlegg. I denne oppgaven velger man derfor å inkludere forebyggende vedlikehold i beregningen av operasjonell tilgjengelighet. Beregningsmodellen som er benyttet er hentet fra Thangamani, G. et al. [1]. Den går ut på å benytte seg av MTBF i stedet for MTTF og inkluderer i tillegg revisjonsstansen  $T_s$  i utregningen av tilgjengeligheten. Nedenfor er viktige parametere i metodikken listet opp og definert:

$m_i$  : Gjennomsnittlig tid mellom feil for den uønskede topphendelsen  $i$ .

- $t_i$ : Gjennomsnittlig tid til reparasjon for den uønskede topphendelsen  $i$ .
- $n$ : Totalt antall uønskede topphendelser.
- $T_d$ : Antall timer med systemdrift mellom hver revisjonsstans. I dette tilfellet er systemdriften satt lik 2 år ( $T = 17520$  timer).
- $T_0$ : Total kalendertid med korrektivt vedlikehold.
- $T_s$ : Total kalendertid med forebyggende vedlikehold i forbindelse med revisjonsstans ( $T_s = 1460$  timer).
- $T_r$ : Total kalendertid med vedlikehold.
- $A_{sys}$ : Systemtilgjengeligheten.

Total korrektivt vedlikeholdstid er definert på følgende måte:

$$T_0 = \sum_{i=1}^n \frac{t_i}{m_i} T_d \quad (5.1)$$

Videre kan man uttrykke den totale vedlikeholdstiden som  $T_r = T_0 + T_s$  som gir følgende sammenheng for systemtilgjengeligheten:

$$A_{sys} = \frac{T_d}{T_r + T_d} \quad (5.2)$$

For å kunne benytte seg av likning (5.2) må man anta at alle de uønskede hendelsene er uavhengige.

## 5.5 Beskrivelse av simuleringsprosessen

I denne oppgaven er det utviklet en egenkomponert simuleringskode i MATLAB (vedlegg D) som består av en Monte Carlo simulering (kapittel 4.2.2) med konstante tidsskritt  $\Delta t$ . Når man skal kartlegge operasjonell tilgjengelighet til et komplisert system som for eksempel et CO<sub>2</sub>-rensaneanlegg må man benytte en metodikk som kan håndtere mange typer data på en gang. Monte Carlo simulering er et godt verktøy å ta i bruk hvis kompleksiteten er stor. Man får simulert levetiden til de ulike feilhendelsene i systemet over en ønsket tidsperiode og med gitte tidsskritt. På denne måten får man god kontroll på hvilke rammer som skal gjelde for simuleringen. Metoden til Thangamani G. et al. [1] benyttet seg også av Monte Carlo simulering. Det var egentlig aldri foreslått noen alternativer til Monte Carlo simulering i denne oppgaven. Derfor er det vanskelig å si noe om hva eventuelle forskjeller ville vært ved valg av en annen simuleringsteknikk.

For at det skal bli mer oversiktlig å få med seg hva som skjer i simuleringsprosessen deles den inn i 4 hoveddeler: *Inndata*, *beregning av enkeltdata*, *beregning av systemdata* og *utdata*.

### ***Inndata:***

Det første man må gjøre i en simuleringsprosess er å spesifisere de ulike inputverdiene som skal være med i prosessen. Nedenfor er disse listet opp punktvis:

- Feilraten til de ulike feilhendelsene i feiltrærne (*tabell 3.2*) ved vektoren  $x$ .
- Reparasjonstiden til de ulike feilhendelsene i feiltrærne (*tabell 3.2*) ved vektoren  $y$ .
- Verdien på de konstante tidsskrittene  $\Delta t = 1/12$  år.
- Tidsperioden for den totale simuleringstiden  $T_t = 2628000$  timer (=300 år).
- Tidsperioden for total systemdrift mellom revisjonsstans  $T_d = 17520$  timer (=2 år).
- Tiden det tar å utføre forebyggende vedlikehold i form av revisjonsstans  $T_s = 1460$  timer (=2 mnd).

I tillegg til alle disse gitte verdiene må man definere alle matriser og vektorer som skal være med i simuleringsprogrammet. For detaljer om de ulike matrisene og vektorene, og simuleringen generelt, vises det til simuleringskoden i vedlegg D.

### ***Beregning av enkeltdata:***

Simuleringsprogrammet består av tre ”for-løkker” som utfører en gitt prosess et visst antall ganger før den går videre. De tre ”for-løkkene” har henholdsvis grenser for  $n = \text{antall tidsskritt} = n_1 + n_2$ ,  $m = \text{antall feilhendelser} = m_1 + m_2$  og  $p = \text{antall simuleringer}$ .

Sannsynligheten for at feil skal oppstå i et tidsskritt  $q$  for hver feilhendelse er definert av følgende likning ([19], kap.7, s.29-31):

$$q = x \cdot \Delta t \quad (5.3)$$

For hvert tidsskritt  $n$  sammenlignes sannsynligheten  $q$  med et tilfeldig generert tall  $r \in (0,1)$ , som er generert fra en uniform fordelingsfunksjon. Hvis det tilfeldige tallet er mindre eller lik sannsynligheten for feil i det gitte tidsskrittet vil feilhendelsen inntreffe.

Hvis feilhendelsen inntreffer beregnes tid til feil TTF for feilhendelsen. Dette skjer ved hjelp av en tidsskrittindikator  $dt\_ind$  som passer på i hvilket tidsskritt siste feil inntraff. I tillegg er det en feilteller  $dt\_teller$  som teller hvor mange feil som har oppstått for en feilhendelse i løpet av totaltiden. Deretter beregnes MTTF for feilhendelsen over hele totaltiden  $T_t$  ved hjelp av likning (4.36). Hvis det ikke har inntruffet feil på feilhendelsen i løpet av totaltiden settes MTTF lik totaltiden  $T_t$ . Hele denne prosedyren gjentas  $m$  antall ganger slik at man nå har sjekket  $n$  ulike tilfeldige tall opp mot hver feilsannsynlighet i  $q$ -vektoren (vedlegg D). Alle feilhendelser regnes som uavhengige av hverandre i den grad at en feilhendelse ikke har innvirkning på andre feilhendelser. Dette er en forenkling som er gjort for å kunne benytte likningssystemet fra teoridelen i kapittel 4.2 i simuleringen.

### ***Beregning av systemdata:***

Hver gang simuleringen har regnet seg gjennom alle tidsskrittene for alle feilhendelsene beregnes ulike parametere for feiltrærne og totalsystemet. Denne prosedyren gjentas  $p$  antall ganger. Ved hjelp av informasjon fra ulike ”cut sets” (kapittel 2.3.2) i de forskjellige feiltrærne kan man beregne følgende verdier:

- MTTF for feiltrærne og systemet gitt av likningene (4.15), (4.41) og (4.42).
- MTBF for feiltrærne og systemet gitt av likningene (4.35) og (4.44-4.46).
- MTTR for feiltrærne gitt av likning (4.47).
- Total kalendertid av korrektivt vedlikehold  $T_0$  gitt av likning (5.1).
- Systemtilgjengeligheten  $A_{\text{sys}}$  gitt av likning (5.2).

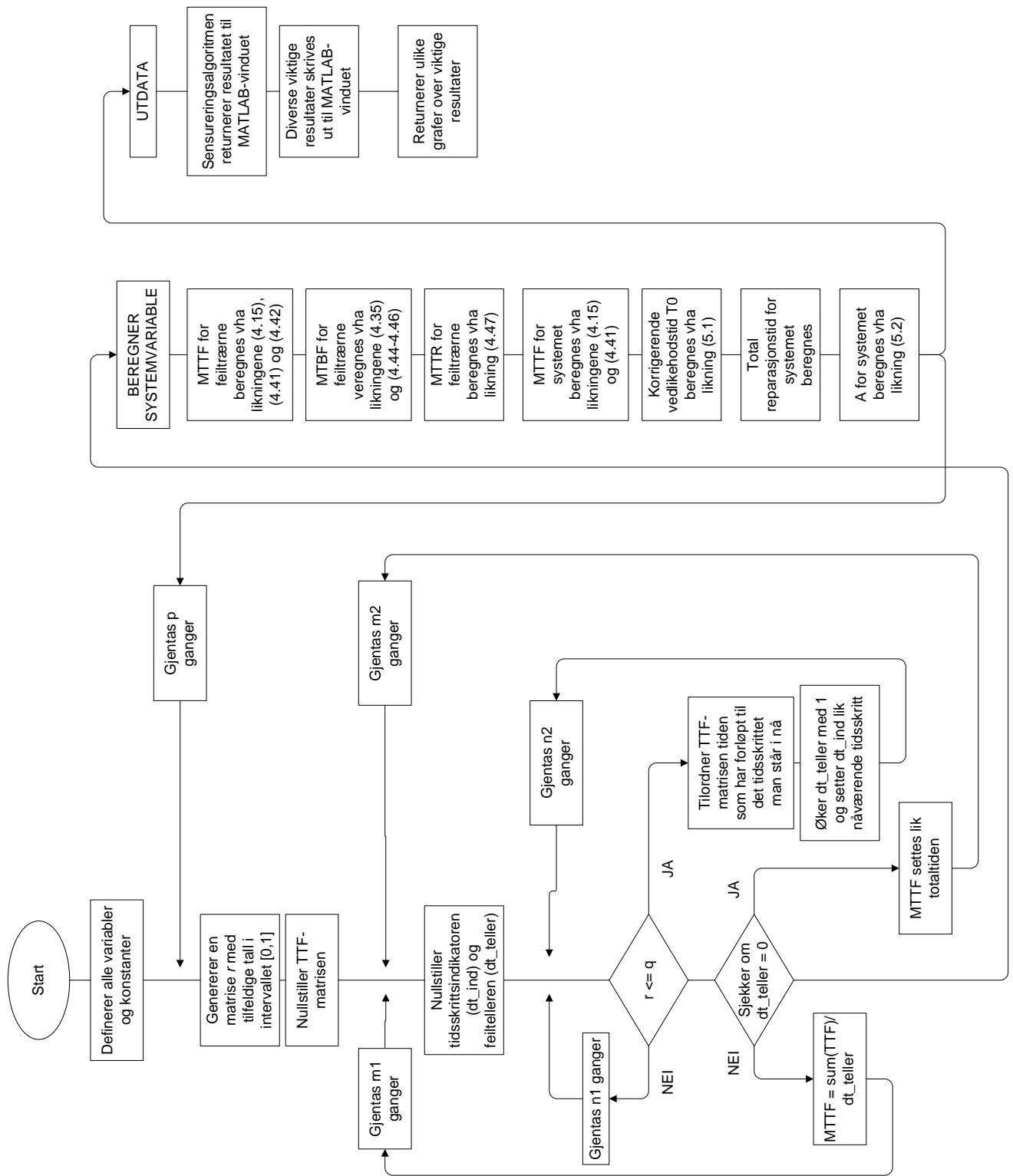
MTTR-verdier for de ulike feilhendelsene settes lik den konstante verdien gitt av  $y$ -vektoren uavhengig av om feilhendelsen har en levetid som er større eller mindre enn totaltiden.

#### ***Utdata:***

Når alle systemsimuleringene er fullført er det ønskelig å hente ut flere av de beregnede verdiene for å se på sammenhenger mellom de ulike oppsettene. Først er det satt opp en algoritme som beregner andelen feilhendelser som har MTTF lengre enn den totale simuleringen. Denne  $cr$ -verdien gir en indikasjon på hvor nøyaktige resultatene er (kapittel 5.3). I vedlegg D kan man se detaljer over hvilke resultater som hentes ut i MATLAB-vinduet. Enkelte av de viktigste resultatene plottes i ulike grafer for å kunne se en sammenheng mellom resultatene. MATLAB-kodene *wblfit*, *gamfit*, *normfit* og *exffit* (vedlegg A) benyttes for de simulerte dataene for MTTF til systemet. Dette utføres for å vise hvilke parametere, for de ulike fordelingsfunksjonene (kapittel 4.1), som passer best opp i mot de simulerte verdiene.

De simulerte verdiene for MTTF til feilhendelsene med lavest og høyest feilrate, og MTTF til systemet, plottes i en empirisk CDF sammen med en eksponential-, gamma-, normal- og weibull-CDF basert på de maksimale sannsynlighetsestimatene (MLE) beregnet av *wblfit*, *gamfit*, *normfit* og *exffit*. På denne måten kan man se hvilken type fordelingsfunksjon som likner mest på de simulerte verdiene og dermed få dannet et statistisk utgangspunkt for "levetiden" som kan benyttes til å kvantifisere usikkerheter (kapittel 2.2.2). For nærmere detaljer om beregning av maksimale sannsynlighetsestimater vises det til Mann, N.R. et al. ([18], s.81-85).

Det plottes også en IFR/DFR-test som benyttes for å se om de ulike fordelingene har økende eller minkende feilrate. Denne testen baserer seg på å plote  $-\ln(1-F(t))$  for de forskjellige fordelingene mot tiden. Hvis kurven er konveks er feilratene økende (IFR) og hvis kurven er konkav er feilraten minkende (DFR) i følge likning (4.2).



Figur 5.1: Flytskjema for simuleringsprosessen.



## 6 Resultater og diskusjon

### 6.1 Hovedoppsett

Dette oppsettet har følgende ”cut set”-struktur, hvor tallene i klammene referer til feilhendelsen. I vedlegg E.1 kan man se hvordan feiltrærne er bygd opp for dette oppsettet. Det vises også til vedlegg B for detaljer rundt utregning av ”cut sets”:

- Feiltre 1: {1},{2},{3},{4},{5},{6},{7},{8}.
- Feiltre 2: {9},{10},{11},{12},{13},{14},{15}.
- Feiltre 3: {16},{17},{18},{19},{20},{21},{22},{23}.

Feiltrestrukturen viser at dette oppsettet utelukkende består av en stor seriekobling. Det medfører at hele systemet vil stoppe opp hvis en komponent har en feilhendelse som inntreffer.

I *tabell 6.1* er de simulerte verdiene for MTTF og MTTR for enkelte feilhendelser, feiltrærne og totalsystemet presentert. De tabellererte verdiene for MTTF til feiltrærne er gjennomsnittet av de  $p = 1000$  simulerte verdiene. Tabellert verdi for systemet er beregnet ut fra de gitte verdiene for feiltrærne ved hjelp av likningene (4.15) og (4.41).

Årsaken til at man har valgt å ta med feilhendelse 9 og 12 er fordi de har henholdsvis størst og minst feilrate (*tabell 3.2*). I følge antakelsen om eksponentialfordelte feilhendelser viser likning (4.15) at feilhendelse 9 og 12 da har henholdsvis lavest og høyest teoretisk MTTF. *Tabell 6.1* viser også hvor stor andel av feilhendelsene som har en MTTF høyere enn total simuleringstid,  $cr$ . En total simuleringstid på  $T_t = 300$  år er tilstrekkelig for at alle de  $p = 1000$  simuleringene, for alle MTTF-verdier for feilhendelse 9, skal falle innenfor totaltiden. Ser man derimot på feilhendelse 12 er fortsatt 8% av MTTF-verdiene større enn totaltiden etter en simuleringstid på 300 år. Dette medfører at middelverdien av simulert tid til feil for feilhendelse 12 blir litt unøyaktig i forhold til den teoretiske verdien (*tabell 6.4*).

I totalsystemet er ca 1% av MTTF for feilhendelsene større enn den totale simuleringstiden. Dette er en såpass liten andel at det ikke vil føre til store unøyaktigheter i resultatene og man kan benytte de simulerte levetidene til å lage en empirisk fordeling og beregne middelverdier uten å måtte se bort fra enkelte levetider som ”faller utenfor” simuleringsperioden.

<b>Hovedoppsett</b>						
$p = 1000, T = 2628000$						
	FH 9	FH 12	FT 1	FT 2	FT 3	Systemet
MTTF [timer]	<b>76430</b>	<b>939080</b>	<b>28590</b>	<b>16659</b>	<b>25170</b>	<b>7422</b>
MTTR [timer]	<b>70,4</b>	<b>75</b>	<b>21,3</b>	<b>62,76</b>	<b>23,66</b>	<b>40,46</b>
cr [%]	<b>0</b>	<b>8</b>	<b>0,71</b>	<b>2,01</b>	<b>0</b>	<b>0,87</b>

*Tabell 6.1: MTTF og MTTR for enkelte feilhendelser, feiltrær og totalsystem.*

Tabell 6.2 viser nøkkelverdier for den empiriske fordelingsfunksjonen basert på simulerte verdier for feilhendelse 9 og 12 og systemet. Både feilhendelse 9 og systemet har en middelvei og medianverdi som er nokså like. Dette kan tyde på at fordelingen er forholdsvis normalfordelt. Feilhendelse 12 har derimot en medianverdi som er 26% lavere enn middelveien. I tillegg er standardavviket like stort som medianverdien. Dette kan bety at det er veldig mange høye verdier i fordelingen. Dette er også underbygd ved at man har påvist at 8% av de simulerte verdiene for feilhendelse 12 er satt lik totaltiden.

Empirisk CDF			
	FH 9	FH 12	Systemet
Middelvei [timer]	<b>76430</b>	<b>939080</b>	<b>7373</b>
Minimumsverdi [timer]	<b>46212</b>	<b>15330</b>	<b>4831</b>
Maksimalverdi [timer]	<b>167900</b>	<b>2628000</b>	<b>8821</b>
Medianverdi [timer]	<b>74064</b>	<b>696055</b>	<b>7355</b>
Standardavvik [timer]	<b>13995</b>	<b>696450</b>	<b>457,59</b>

Tabell 6.2: Maksimum-, minimum-, middel- og medianverdi i tillegg til standardavvik for feilhendelse 9 og 12 og systemet basert på en empirisk CDF av de simulerte verdiene.

### 6.1.1 Feilhendelsene med lavest og høyest feilrate

I tabell 6.3 og tabell 6.4 kan man sammenlikne teoretisk verdi av MTTF for henholdsvis feilhendelse 9 og feilhendelse 12 opp imot den simulerte verdien. Avviket er gitt som %-forskjell mellom den teoretiske og den simulerte verdien.

På grunn av at feilhendelse 9 har en lav teoretisk MTTF i forhold til totaltiden (3% av totaltiden) har simuleringssystemet beregnet en simulert verdi for MTTF som avviker minimalt fra teoretisk verdi (tabell 6.3). Feilhendelse 12 har en teoretisk MTTF som er 37% av totaltiden og den simulerte verdien for MTTF avviker 3% fra den teoretiske verdien.

Som et resultat av at simuleringssystemet velger ut helt tilfeldige tall i intervallet [0,1] hver gang man kjører en simuleringrunde vil resultatene variere noe mellom hver runde algoritmen kjøres. Tabell F.1 og tabell F.2 i vedlegg F viser hvor mye de enkelte resultatene gitt i hovedoppsettet avviker fra tilsvarende resultater beregnet i en annen simuleringrunde. Det er resultatene for feilhendelse 12 som vil variere mest på grunn av at det også er disse resultatene som er mest usikre.

Feilhendelse #9		
	Teoretisk verdi	Simulert verdi
MTTF [timer]	<b>76394</b>	<b>76430</b>
Avvik [%]	<b>0</b>	<b>0,05</b>

Tabell 6.3: Teoretisk verdi for MTTF til feilhendelse 9 sammenliknet med simulert verdi.

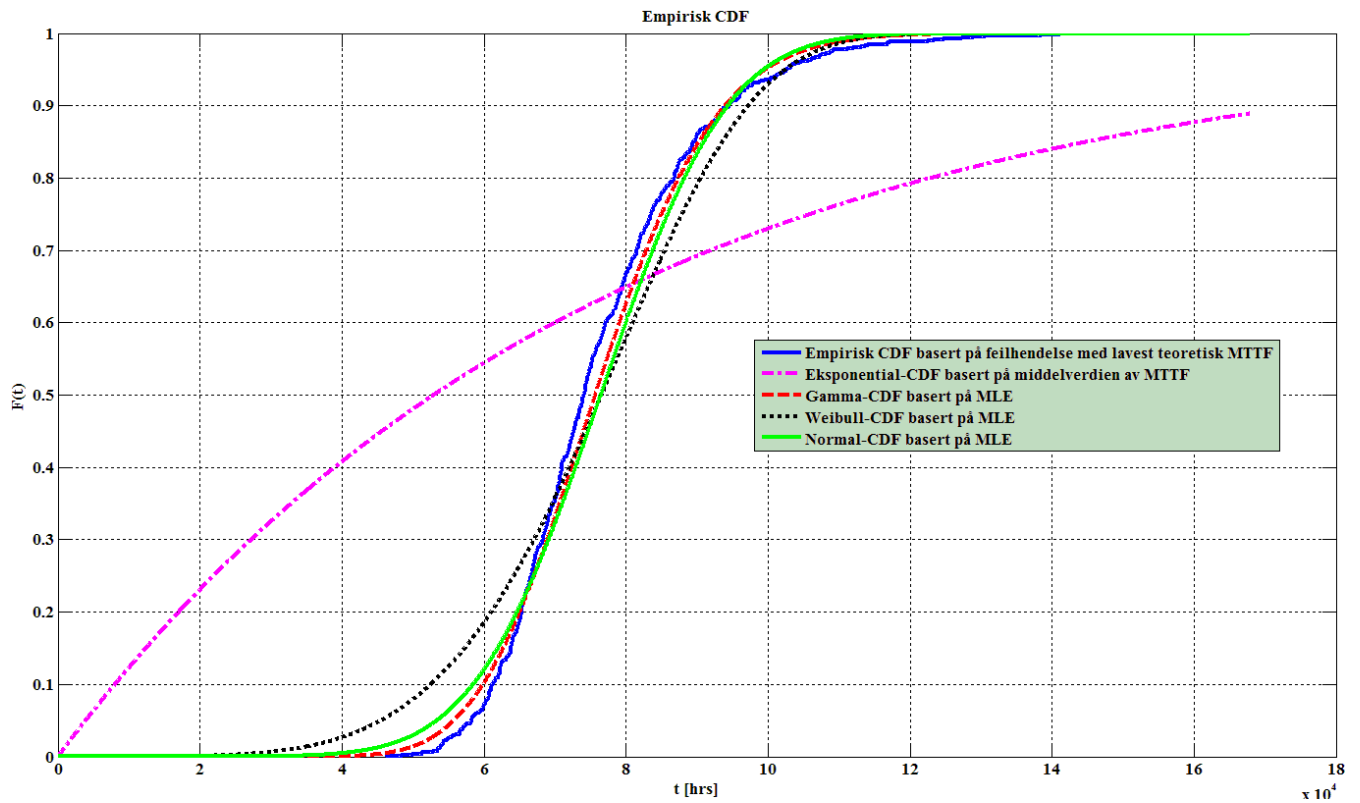
### Feilhendelse #12

	Teoretisk verdi	Simulert verdi
MTTF [timer]	<b>970874</b>	<b>939080</b>
Avvik [%]	<b>0</b>	<b>-3</b>

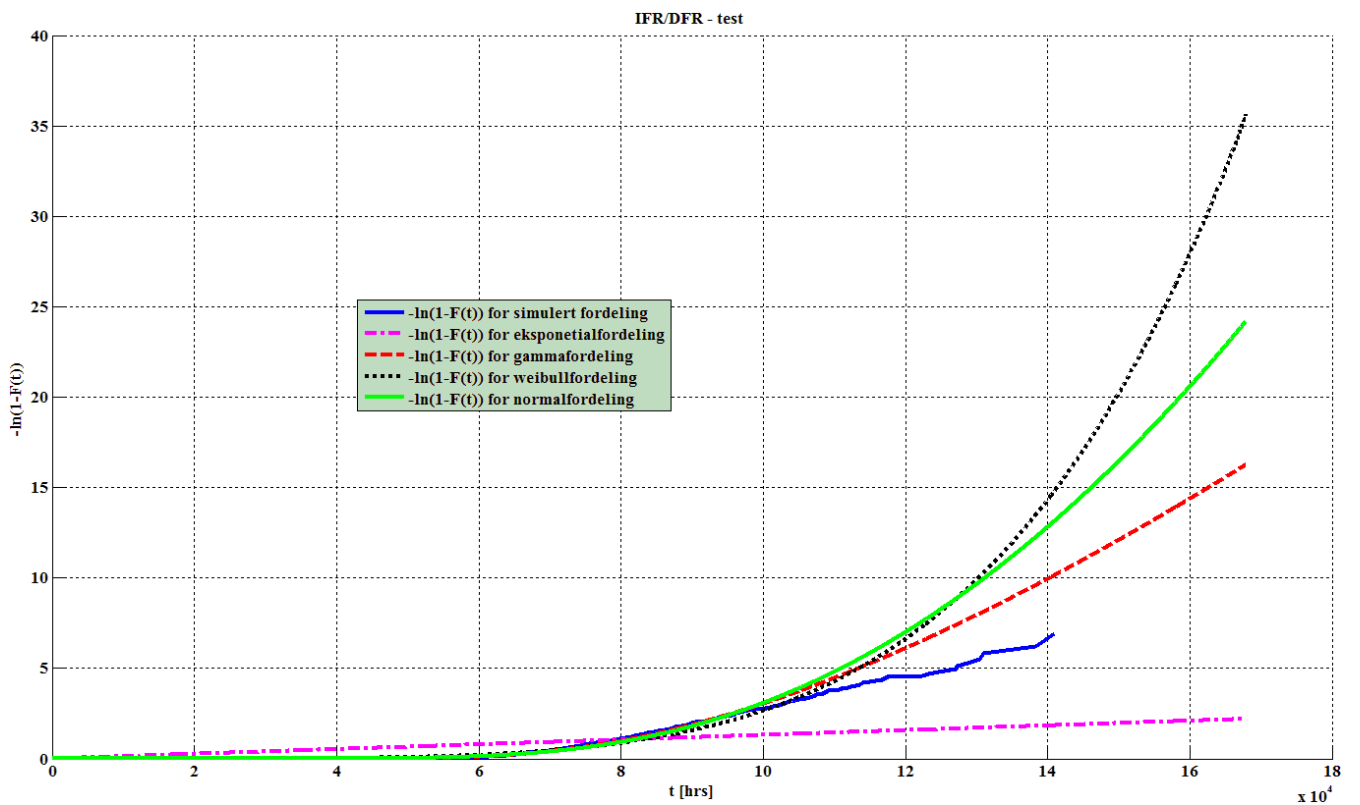
Tabell 6.4: Teoretisk verdi for MTTF til feilhendelse 12 sammenliknet med simulert verdi.

De  $p = 1000$  levetidene som er simulert for feilhendelse 9 og 12 og systemet er benyttet til å plote empiriske fordelingsfunksjoner for levetidene i henholdsvis figur 6.1, figur 6.3 og figur 6.5.

Figur 6.1 viser ulike fordelinger for feilhendelse 9. X-aksen uttrykker tiden i antall timer mens y-aksen uttrykker den akkumulerte sannsynligheten  $F(t)$  for at feilhendelsen skal ha inntruffet i løpet av tiden  $t$ . Den blå kurven viser den empiriske fordelingsfunksjonen basert på de simulerte levetidene for feilhendelse 9. Denne kurven er sammenlignet med andre tilpassede fordelingsfunksjoner. Man har benyttet de simulerte levetidene til å beregne maksimale sannsynlighetsestimater for parametrene i gamma-, weibull- og normalfordelingen. I tillegg har man brukt middelveiden av de simulerte levetidene som en estimator for parameteren i eksponentialfunksjonen (kapittel 4.1.2-4.1.5). Deretter har man plottet de tilpassede fordelingene inn i samme graf som den empiriske fordelingen for å se om den kan tilnærmes en eksponential-, weibull-, gamma- eller normalfordeling. Resultatet fra figur 6.1 viser at weibullfordelingen (sort kurve), gammafordelingen (rød kurve) og normalfordelingen (grønn kurve) er nokså like med den empiriske fordelingen til de simulerte levetidene. Et viktig resultat å merke seg er at eksponentialfordelingen (rosa kurve) ikke passer med den empiriske fordelingen.



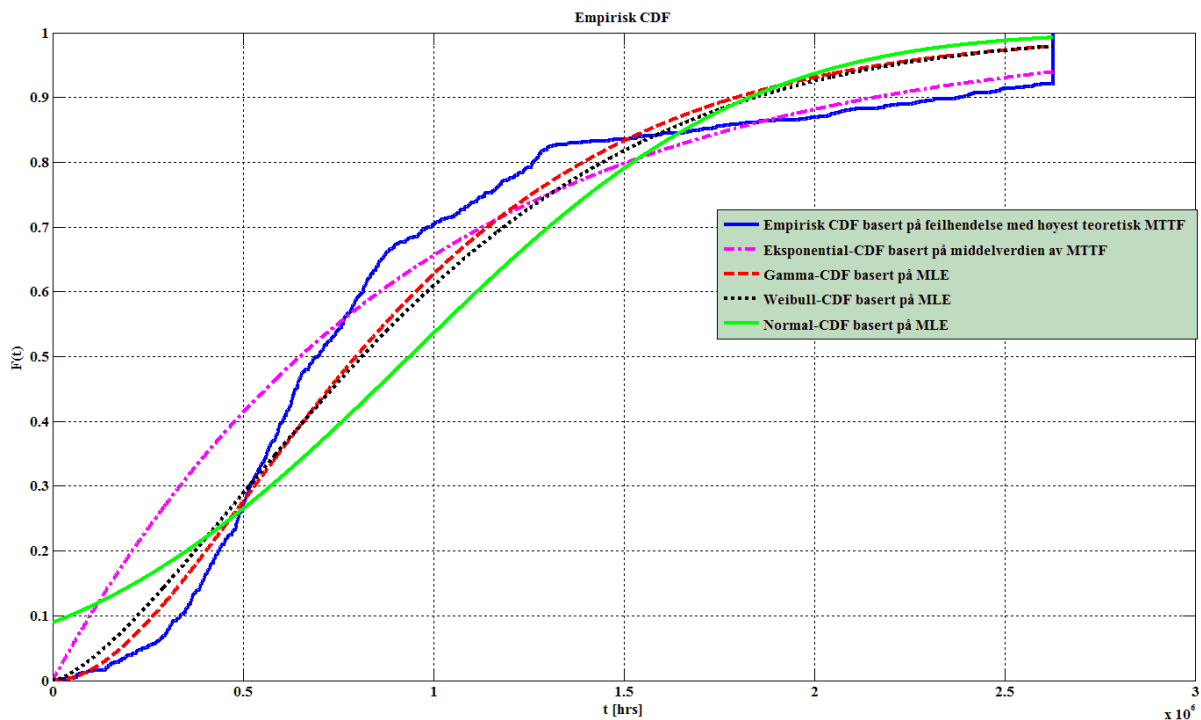
Figur 6.1: Empirisk CDF for feilhendelse 9 sammenliknet med empiriske CDF for flere fordelingsfunksjoner.



Figur 6.2: IFR/DFR-test for feilhendelse 9.

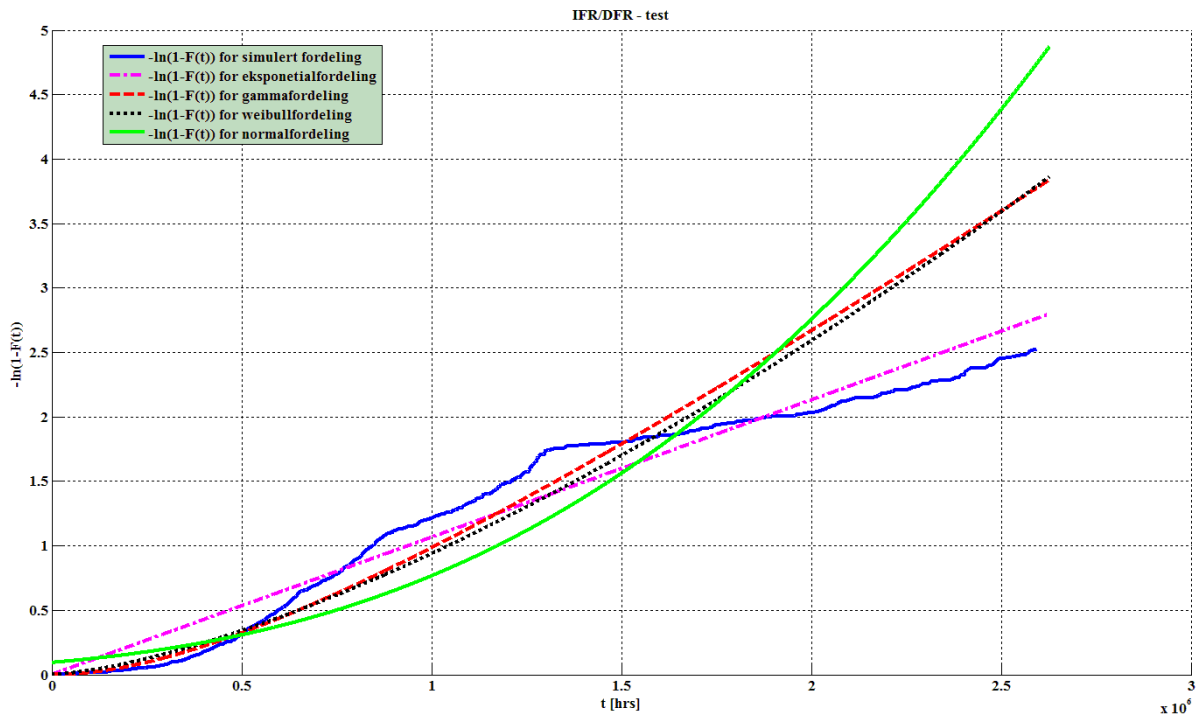
Figur 6.2 viser en IFR/DFR-test for feilhendelse 9. Den tilpassede gamma-, weibull- og normalfordelingen har en kurve med konveks form som i følge likning (4.2) betyr at feilraten er økende. Eksponentialfordelingen er lineær som gir en konstant feilrate. Den empiriske fordelingen (blå kurve) har hovedsakelig en konveks form som tyder på at den har en økende feilrate.

Figur 6.3 viser resultatene fra de simulerte verdiene for feilhendelse 12. I denne figuren er det ikke så lett å kunne anslå en spesifikk fordelingsfunksjon for de simulerte verdiene (blå kurve). Feilhendelse 12 har en veldig lav feilrate som igjen fører til at 8% av de simulerte verdiene for levetiden vil være høyere enn den totale simuleringstiden. Dette kan man også se i figuren ved at kurven knekker av rett opp ved totaltiden  $T_T=2628000$  timer. De tilpassede gamma-, og weibullfordelingene (rød og sort kurve) er veldig like og den tilpassede eksponentialfordelingen (rosa kurve) ligger nærmere de andre fordelingene enn tilfelle var for feilhendelse 9. Normalfordelingen ser ikke ut til å passe med de simulerte verdiene.



Figur 6.3: Empirisk CDF for feilhendelse 12 sammenliknet med empiriske CDF for flere fordelingsfunksjoner.

I figur 6.4 har man en graf med en tilsvarende IFR/DFR-test som ble gjort for feilhendelse 9 i figur 6.3. Selv om den empiriske fordelingen (blå kurven) er nokså ujevn enkelte steder i grafen kan man antyde at den er konveks med ulik grad av stigning i hele simuleringstiden. Det betyr at den empiriske fordelingen for feilhendelse 12 har en økende feilrate.



Figur 6.4: IFR/DFR-test for feilhendelse 12.

### 6.1.2 Feiltrærne og systemet

Tabell 6.5 viser resultatene for beregnet MTBF og total korrektiv vedlikeholdstid  $T_0$ , for de ulike topphendelsene som fører til systemstans, og for totalsystemet. MTBF til feiltrærne i tabellen er gjennomsnittet av de  $p = 1000$  simulerte verdiene. Systemverdien er beregnet ut fra de gitte verdiene for feiltrærne ved hjelp av likningene (4.15) og (4.41).  $T_0$  er beregnet ut i fra likning (5.1) hvor verdier for MTTR og MTBF er hentet fra henholdsvis tabell 6.1 og tabell 6.5.

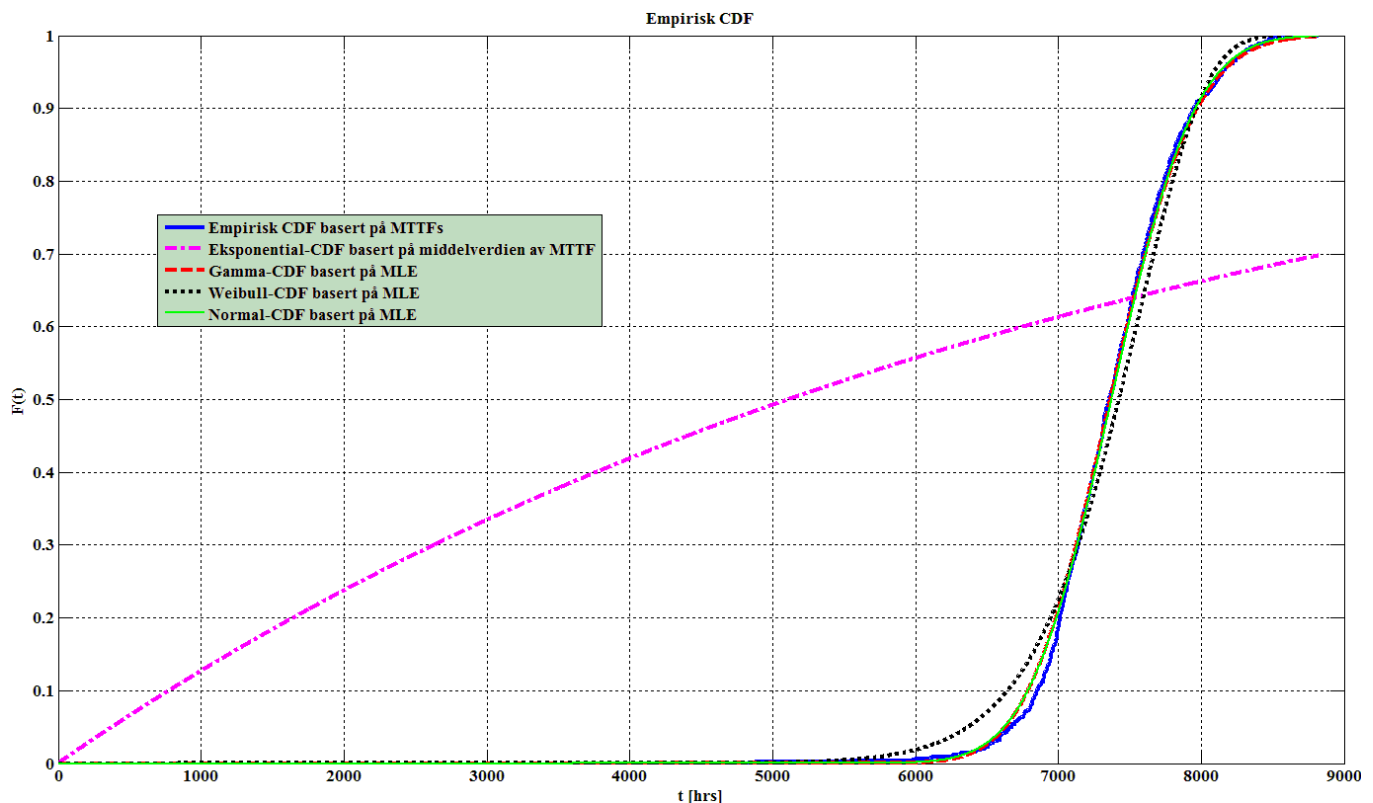
Uønskede topphendelser		
Systemstans som følge av:	MTBF [timer]	$T_0$ [timer]
<i>for lav effektivitet/slitasje</i>	<b>28356</b>	<b>13,18</b>
<i>aminutslipp til omgivelsene</i>	<b>16654</b>	<b>66,00</b>
<i>stopp i prosesstrømning</i>	<b>25057</b>	<b>16,52</b>
<b>Totalsystemet</b>	<b>7395</b>	<b>95,70</b>

Tabell 6.5: MTBF og MTTR for de ulike uønskede topphendelsene i systemet.

De simulerte resultatene indikerer at det er feiltre 2 ("Systemstans som følge av aminutslipp til omgivelsene") som er det minst pålitelige undersystemet med en MTBF på 16654 timer eller i underkant av 2 år. Feiltre 1 ("Systemstans som følge av for lav effektivitet/slitasje") og

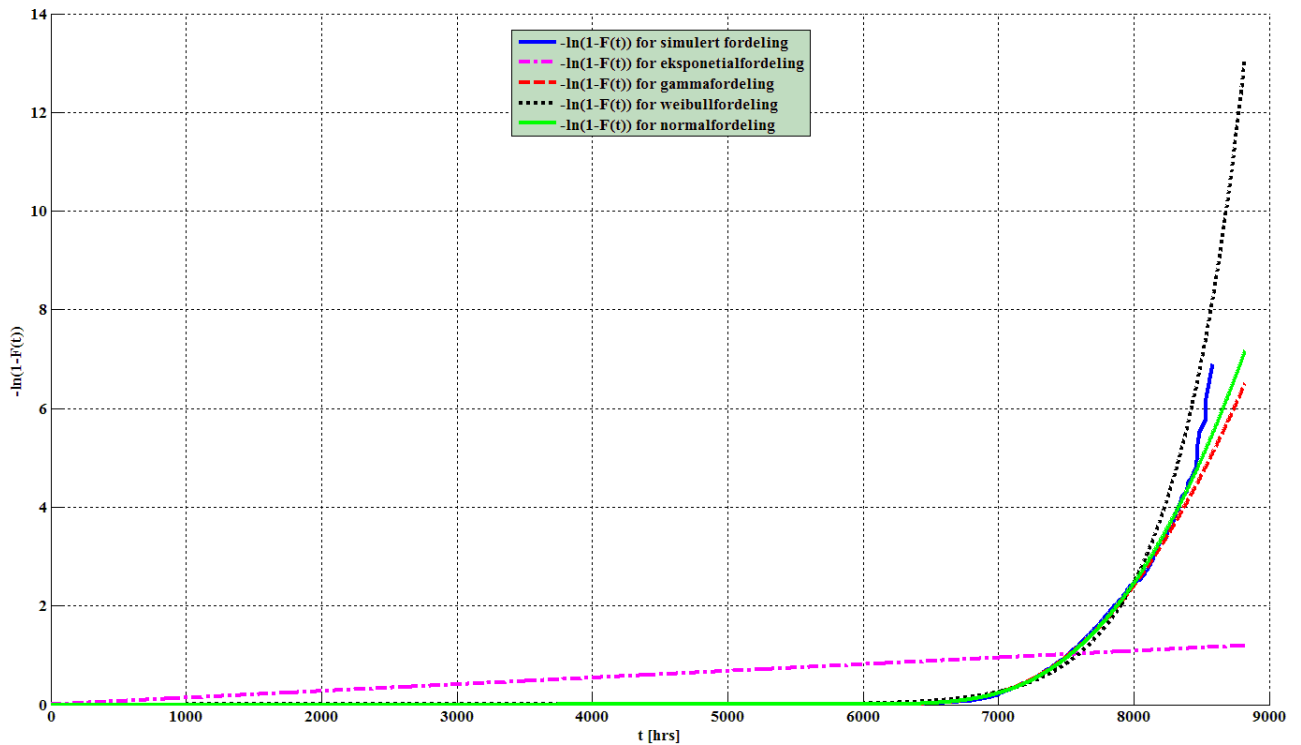
feiltre 3 ("Systemstans som følge av stopp i prosessstrømning") har henholdsvis MTBF på i overkant av 3 år og i underkant av 3 år. Det utføres forebyggende vedlikehold i form av en revisjonsstans en gang hvert andre år hvor alt av utstyr etter revisjonsstansen betegnes som "så godt som nytt". Det vil derfor være naturlig å anta at man vil få flere stans i anlegget som følge av feil i feiltre 2 enn av de andre feiltrærne som har en MTBF høyere enn driftstiden mellom hver revisjonsstans  $T_d=17520$  timer. Totalsystemet har en MTBF på 7395 timer eller 10 måneder.

Figur 6.5 viser en empirisk CDF for systemet. Akkurat som i figur 6.1 kan man se at den eksponentielle fordelingen (rosa kurve) ikke er i nærheten av å passe med den empiriske fordelingen (blå kurve). Det kan se ut som gammafordelingen (rød kurve) og normalfordelingen (grønn kurve) er de fordelingene som gir en best fremstilling av de simulerte verdiene for levetidene til systemet. Weibullfordelingen (sort kurve) er også nokså lik den empiriske fordelingen.



Figur 6.5: Empirisk CDF for systemet sammenliknet med empiriske CDF for flere fordelingsfunksjoner.

Figur 6.6 viser en IFR/DFR-test for systemet. Man kan se at den empiriske fordelingen (blå kurve) er konveks og følger samme linje som gamma-, weibull- og normalfordelingen i starten. På slutten er det vanskelig å se hvordan den oppfører seg.



Figur 6.6: IFR/DFR-test for systemet.

Det har blitt antydnet at verdiene til de simulerte "levetidene" til feilhendelsene kan være gamma-, weibull- eller normalfordelt. I utgangspunktet antok man at feilratene til de ulike feilhendelsene var konstante og at dermed "levetiden" var eksponentielt fordelt. Når man gjennomfører en Monte Carlo simulering vil middelverdien av simulerte "levetider" etter hvert bli tilnærmet lik MTTF gitt av den teoretiske feilraten. Dette stemmer også bra med resultatene oppnådd i denne oppgaven for feilhendelse 9 og 12. Det er likevel ikke ensbetydende med at de simulerte "levetidene" er eksponentialfordelte. De tilfeldig genererte variablene som benyttes i simuleringen vil resultere i "levetider" som kan variere en del fra "teoretiske" verdier. Man kan se en tendens til at "levetidene" blir normalfordelte etter simuleringen på grunn av at man vil få enkelte avvik i forhold til middelverdi i begge retninger når man simulerer over en lengre periode. Det er heller ikke usannsynlig at en weibullfordeling vil inneha den nødvendige fleksibiliteten, ved parameteren  $\alpha$ , som skal til for å fremstille den simulerte "levetiden" på en best mulig måte. I kapittel 4.1.4 ble det vist at en sum av uavhengige eksponentialfordelte tilfeldige variable har en gammafordeling. En gammafordeling kan derfor også være et godt alternativ slik oppsettet er nå. Det er derfor ikke helt klart om det er en normal-, weibull- eller gammafordeling som ville vært det beste alternativet for de simulerte verdiene, men i tilgjengelighetsstudier er det stort sett weibullfordelingen som benyttes, på grunn av sine fleksible egenskaper. I tillegg er normalfordelingen symmetrisk mens det er mulig med skjevhet i fordelingen i en weibullfordeling. Det er generelt vanskelig å si noe om fordelingen til resultatene i og med at hverken dataene, feilårsakene eller feiltreoppbygningen i denne oppgaven har gjennomgått en tilfredsstillende kvalitativ vurdering. Simuleringsresultatene kan derfor ses på som et praktisk



eksempel på hva man kan beregne og hvordan man kan utføre beregningene hvis man har de riktige dataene tilgjengelig for analysen.

## 6.2 Redundans av alle pumper i aminkretsen

Dette oppsettet har ”cut set”-strukturen vist under, hvor tallene i klammene referer til feilhendelsene. Det er viktig å merke seg at i dette tilfellet har det ikke blitt 9 ”nye” feilhendelser, men flere av de samme feilhendelsene er satt opp i parallell slik at de numrene som står her ikke nødvendigvis korresponderer med numrene som står i *tabell 3.2*. Det vises derfor til feiltrestrukturen i vedlegg E.2 for en oversikt av hvilke feilhendelser som står til hvert nummer. Det vises også til vedlegg B for detaljer rundt utregning av ”cut sets”:

- Feiltre 1: {1},{2},{3},{4},{5},{6},{7},{8}.
- Feiltre 2: {9,10},{11,12},{13,14},{15},{16},{17},{18}.
- Feiltre 3: {19,20},{21,22},{23,24},{25,26},{27,28},{29,30},{31},{32}.

Feiltre 1 i dette oppsettet er likt som feiltre 1 i hovedoppsettet (vedlegg E.1). De andre feiltrærne har parallellkoblinger som følge av redundans av aminpumpene (vedlegg E.2). I og med at hver pumpe har tre ulike feilhendelser vil man få 9 parallelle ”cut sets” i totalsystemet. Feiltre 2 har 3 parallelle ”cut sets” mens feiltre 3 har 6 parallelle ”cut sets”.

Det er ikke noe poeng å vise resultater for feilhendelse 9 og 12 i dette oppsettet i og med at utregningen av MTTF for alle enkeltfeilhendelser foregår på samme måte i alle oppsettene. I *tabell 6.6* er MTTF og MTTR for feiltrærne og totalsystemet listet opp for feiltrærne og totalsystemet.

<b>Oppsett med redundans i aminpumper</b>				
$p = 1000$	$T = 2628000 \text{ hrs}$			
	FT 1	FT 2	FT 3	Systemet
MTTF [timer]	<b>28717</b>	<b>25417</b>	<b>39426</b>	<b>10047</b>
MTTR [timer]	<b>21,06</b>	<b>41,03</b>	<b>19,42</b>	<b>28,54</b>
cr [%]	<b>0,75</b>	<b>1,27</b>	<b>0</b>	<b>0,6</b>

*Tabell 6.6: MTTF og MTTR for feiltrær og totalsystemet i oppsettet med redundans i aminpumper.*

I totalsystemet er nesten ingen av de simulerte levetidene for feilhendelsene større enn den totale simuleringstiden. Det vil si at man kan benytte de simulerte levetidene til å lage en empirisk fordeling og beregne middelverdier uten å måtte se bort fra enkelte levetider som ”faller utenfor” simuleringssperioden.

## 6.2.1 Feiltrærne og systemet

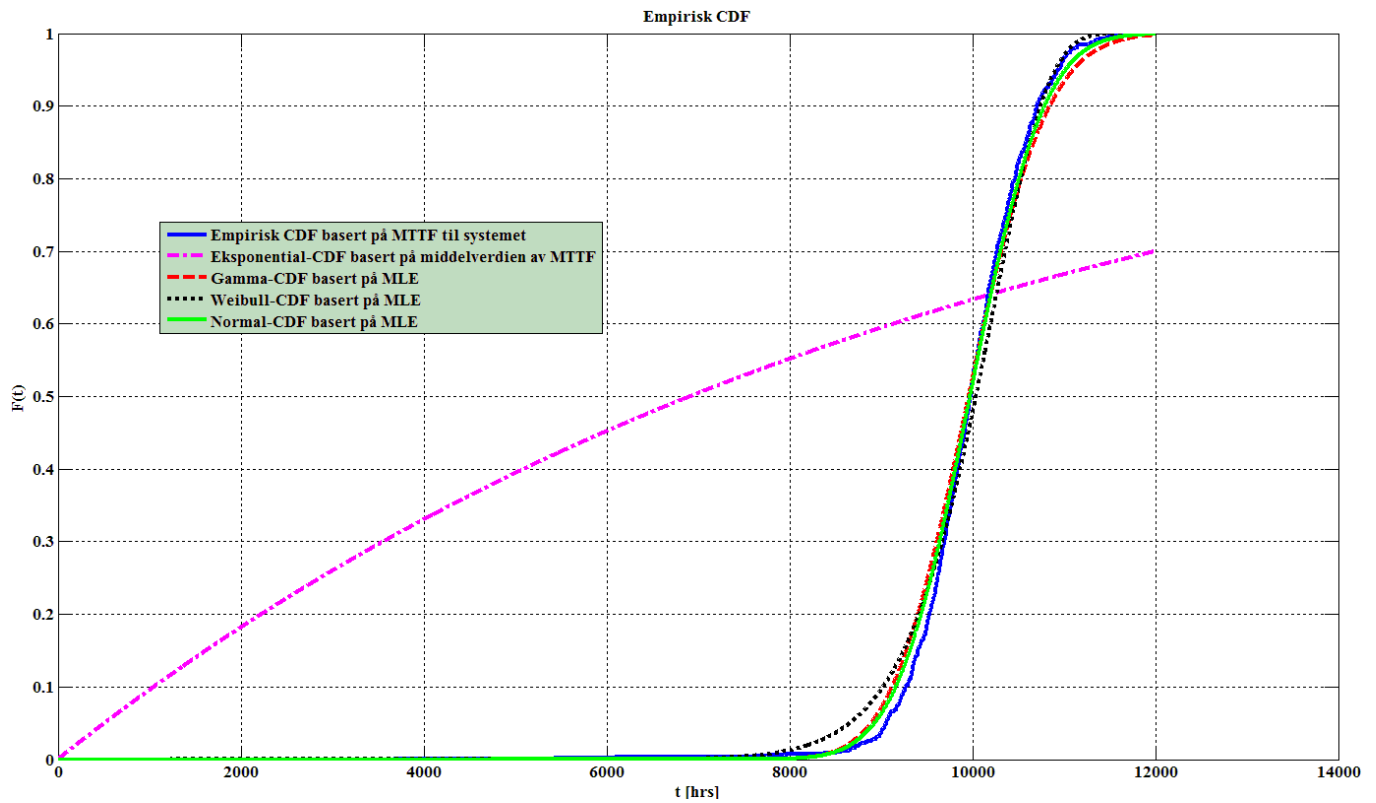
Tabell 6.7 viser resultater fra beregnet MTBF og MTTR for de ulike topphendelsene som fører til systemstans for dette oppsettet. Disse størrelsene er beregnet ut ifra feiltrestrukturen gitt i vedlegg E.2.

Uønskede topphendelser		
Systemstans som følge av:	MTBF [timer]	T0[timer]
<i>for lav effektivitet/slitasje</i>	<b>28719</b>	<b>12,85</b>
<i>aminutslipp til omgivelsene</i>	<b>25424</b>	<b>28,27</b>
<i>stopp i prosessstrømning</i>	<b>39430</b>	<b>8,63</b>
<b>Totalsystemet</b>	<b>10049</b>	<b>49,75</b>

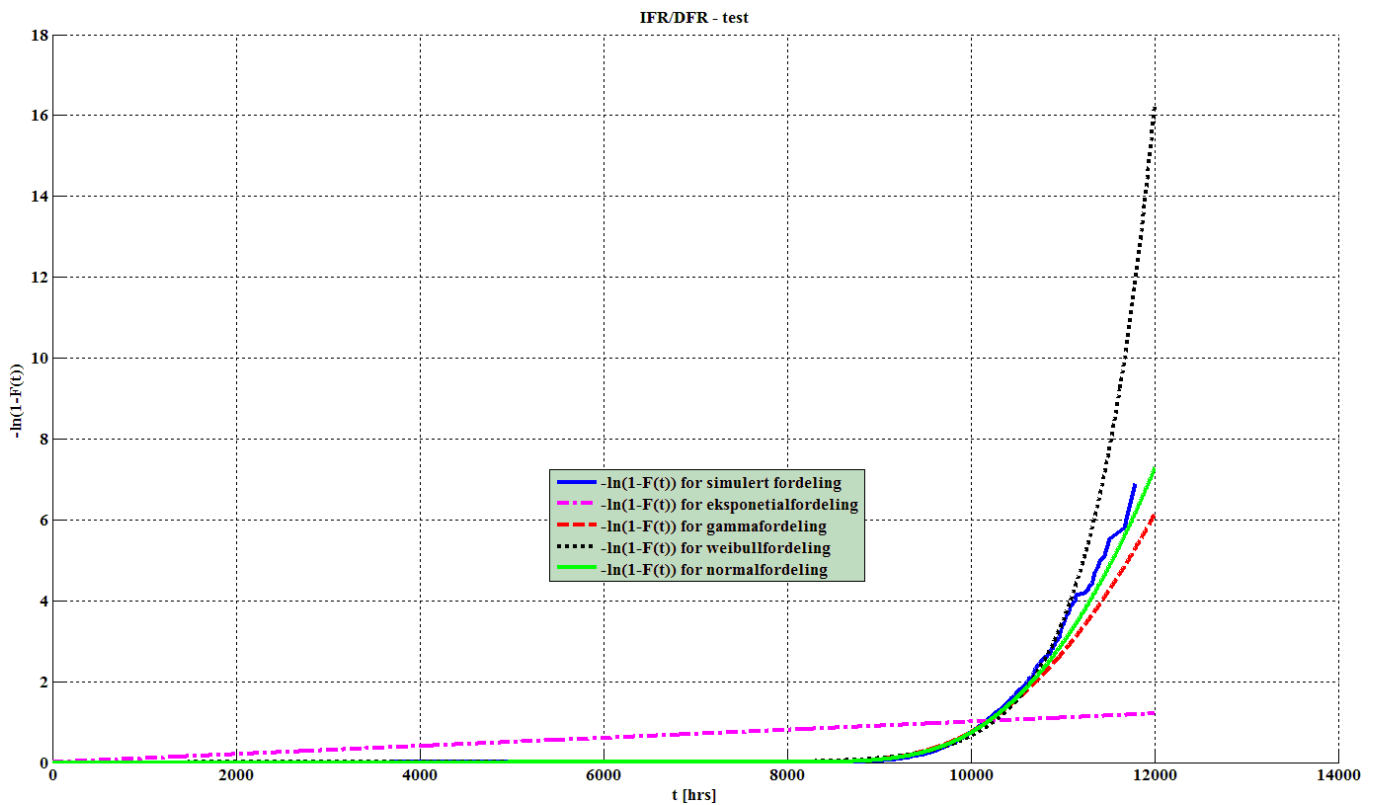
Tabell 6.7: MTBF og MTTR for de ulike uønskede topphendelsene i systemet.

På grunn av redundans av aminpumpene har både feiltre 2 ("Systemstans som følge av aminutslipp til omgivelsene") og feiltre 3 ("Systemstans som følge av stopp i prosessstrømning") fått en betydelig høyere MTBF sammenliknet med hovedoppsettet. De simulerte resultatene indikerer at det fortsatt er feiltre 2 som er det minst pålitelige undersystemet. Nå er derimot MTBF for feiltre 2 økt med 8752 timer eller ca 1 år sammenliknet med hovedoppsettet (tabell 6.5). Feiltre 1 ("Systemstans som følge av for lav effektivitet/slitasje") har tilnærmet samme MTBF på grunn av at strukturen for dette feiltreet er identisk med tilsvarende feiltre i hovedoppsettet. Feiltre 3 har fått en økt MTBF på 14257 timer eller i overkant av 1,5 år. Årsaken til at MTBF har økt mer for feiltre 3 enn for feiltre 2 er fordi feiltre 3 har fått flere parallelle "cut sets" sammenliknet med feiltre 2. På grunn av redundans i systemet har hvert feiltre en MTBF som er vesentlig høyere enn driftstiden mellom hver revisjonsstans  $T_d=17520$  timer. Totalsystemet har en MTBF på 10049 timer eller i underkant av 14 mnd. Dette er 2624 timer eller 35% høyere enn for hovedoppsettet.

Figur 6.7 viser en empirisk CDF for systemet. Akkurat som i figur 6.5 ser man at den eksponentielle fordelingen (rosa kurve) ikke er i nærheten av å passe med den empiriske fordelingen (blå kurve). Det kan se ut som gammafordelingen (rød kurve) og normalfordelingen (grønn kurve) er de fordelingene som gir en best fremstilling av de simulerte verdiene for levetidene til systemet. Weibullfordelingen (sort kurve) er også nokså lik den empiriske fordelingen.



Figur 6.7: Empirisk CDF for systemet sammenliknet med empiriske CDF for flere fordelingsfunksjoner.



Figur 6.8: IFR/DFR-test for systemet.

Figur 6.8 viser en IFR/DFR-test for systemet. Man kan se at den empiriske fordelingen (blå kurve) er konveks og følger samme linje som gamma-, weibull- og normalfordelingen i starten. På slutten er det litt vanskeligere å se hvordan den oppfører seg.

Fordelingen for MTTF til systemet i dette oppsettet ser ut til å være nokså lik det som ble vist for hovedoppsettet i kapittel 6.1. Det er derfor ikke diskutert noe nærmere rundt disse resultatene her, og det vises derfor til kapittel 6.1.2 for diskusjon rundt fordelingen til MTTF for systemet.

### 6.3 Operasjonell tilgjengelighet

Tabell 6.8 viser resultater for den operasjonelle systemtilgjengeligheten og den totale korrektive vedlikeholdstiden for hovedoppsettet og oppsettet med redundans av alle aminpumper.

Systemtilgjengelighet			
	Hovedoppsett	Redundansoppsett	Maks teoretisk
Asys	<b>0,9185</b>	<b>0,9207</b>	<b>0,9231</b>
T0 [timer]	<b>95,47</b>	<b>49,75</b>	<b>0</b>
% av maks	<b>99,50</b>	<b>99,74</b>	<b>100</b>

Tabell 6.8: Systemtilgjengelighet og total korrektiv vedlikeholdstid sammenliknet med maksimal teoretisk verdi uten korrektivt vedlikehold.

Man kan se at det minst pålitelige systemet, hovedoppsettet med seriestruktur, har en operasjonell tilgjengelighet  $A_{sys} = 0,9185$ . I tabell 6.8 har man beregnet hvor mye den maksimale teoretiske operasjonelle tilgjengeligheten ville vært hvis anlegget var i drift uten stans gjennom hele driftsperioden  $T_d = 17520$  timer. Man kan se at den maksimale operasjonelle systemtilgjengeligheten er på  $A_{sys,maks} = 0,9231$ . Hovedoppsettet har en systemtilgjengelighet som er 99,50% av den maksimale mens redundansoppsettet har en systemtilgjengelighet som er 99,74% av den maksimale.

Generelt sett for alle simuleringsresultatene har den operasjonelle tilgjengeligheten endret seg svært lite for de to oppsettene som er gjennomgått til nå. Årsaken til dette er at man har en forebyggende vedlikeholdsperiode  $T_s = 1460$  timer hvert andre år, som vil være en dominerende faktor i dødtiden til anlegget. Selv om man får redusert den korrektive vedlikeholdsperioden  $T_0$  med 52%, fra hovedoppsettet med seriestruktur til redundansoppsettet, har dette ingen betydelig innvirkning på tilgjengeligheten. I seriestrukturen er  $T_0 = 96$  timer, eller 6,5% av  $T_s$ . Dette indikerer at det ikke er mange timer å spare i korrektivt vedlikeholdstid med det datagrunnlaget som er gitt i denne oppgaven. Virkeligheten kan derimot være helt forskjellig fra det som er funnet i denne oppgaven på grunn av at data brukt i denne oppgaven er uegnet for kvalitative vurderinger.

## 6.4 Seriestruktur med feilhendelser som har redusert feilrate

I dette oppsettet er det sett på i hvor stor grad man kan bedre systemtilgjengeligheten ved å redusere feilraten for noen av feilhendelse med høyest feilrate. Det er da tatt utgangspunkt i hovedoppsettet slik at ”cut set”-strukturen er identisk med oppsettet i kapittel 6.1. Simuleringsalgoritmen vil også være identisk med den som er gitt for hovedoppsettet bortsett fra at enkelte feilrater i x-vektoren blir endret (vedlegg D.1). *Tabell 3.2* viser at det er feilhendelse 1,9,10,11 og 22 som har høyest teoretisk feilrate.

### 6.4.1 Reduksjon av feilraten til feilhendelse 1

I første omgang er det gjort et forsøk med å endre feilraten til feilhendelse 1 som ifølge *tabell 3.2* har den klart største feilraten i feiltre 1. *Tabell 6.9* viser hvor mye MTBF,  $T_0$  og systemtilgjengeligheten endrer seg når man reduserer feilraten med et visst antall prosentpoeng.

<b>Hovedoppsett med høyere pålitelighet for FH 1</b>						
$p = 1000, T = 2628000$						
Reduksjon i %	MTBF FT1	MTBF FT2	MTBF FT3	MTBF SYS	$T_0$ sys	Asys
0	28702	16728	25183	7444	95,11	0,9185
10	29366	16728	25183	7495	94,95	0,9185
25	30808	16728	25183	7594	94,75	0,9185
50	34020	16728	25183	7752	94,88	0,9185
70	36998	16728	25183	7931	94,41	0,9185
90	40284	16728	25183	8032	94,70	0,9185

*Tabell 6.9: Seriestruktur med forbedret feilrate for feilhendelse 1.*

*Tabell 6.9* viser at en forbedring av feilhendelse 1 har ingen innvirkning på systemtilgjengeligheten  $A_{sys}$ . Årsaken til dette er at den totale tiden for korrektiv reparasjon  $T_0$  bare er redusert med ca 0,5 time fra det originale oppsettet til man har redusert feilraten med 90%. I og med at det forebyggende vedlikeholdet i form av revisjonsstans består av 1460 timer vil ikke et par timer fra eller til i total reparasjonstid ha noe å si for systemtilgjengeligheten. MTBF for feiltre 1 øker med 11582 timer eller 40% fra originaloppsettet til dette oppsettet hvor feilraten er redusert med 90%. På grunn av at de andre to feiltrærne ikke har endret MTBF vil ikke MTBF til systemet endre seg mer enn 588 timer eller 8%.

Det er også viktig å poengtere at feilhendelse 1 har en svært lav korrektiv vedlikeholdstid på bare 2,8 timer (*tabell 3.2*). Det betyr at den totale korrektive vedlikeholdstiden for systemet ikke blir mye påvirket av at denne feilhendelsen inntreffer sjeldnere.

## 6.4.2 Reduksjon av feilraten til feilhendelsen med høyest feilrate i hvert feiltre

I dette delkapittelet skal man se på et tilfelle hvor man reduserer feilraten til feilhendelsen som har høyest feilrate i hvert feiltre. Alle feilratene blir redusert med en lik prosentandel samtidig. Feilhendelsene som er redusert er i feiltre 1: feilhendelse 1, feiltre 2: feilhendelse 9 og feiltre 3: feilhendelse 22. *Tabell 6.10* viser resultatene for MTBF til feiltrærne og systemet sammen med total korrektiv reparasjonstid  $T_0$  og systemtilgjengeligheten  $A_{\text{sys}}$ .

<b>Hovedoppsett med høyere pålitelighet for FH 1, 9 og 22</b>						
$p = 1000, T = 2628000$						
Reduksjon i %	MTBF FT1	MTBF FT2	MTBF FT3	MTBF SYS	$T_0$ sys	$A_{\text{sys}}$
0	28580	16664	25162	7422	95,40	0,9185
10	29455	17144	25516	7606	92,89	0,9186
25	31000	17637	26325	7878	90,54	0,9187
50	34003	18730	27858	8425	85,64	0,9189
70	36717	19803	29148	8925	81,40	0,9191
90	40101	20711	30348	9419	78,05	0,9193

*Tabell 6.10: Seriestructur med forbedret feilrate for feilhendelse 1,9 og 22.*

*Tabell 6.10* viser at en forbedring av feilraten til feilhendelsen med høyest feilrate i hvert feiltre samtidig, naturligvis gir bedre resultater enn det som ble vist i kapittel 6.4.1. MTBF for feiltre 1 har økt like mye som det gjorde i kapittel 6.4.1. MTBF for feiltre 2 har økt med 4047 timer eller 24%. MTBF for feiltre 3 har økt med 5186 timer eller 21% og MTBF for systemet har økt med 1997 timer eller 27%. Samtidig ser man at total korrektiv vedlikeholdstid  $T_0$  for systemet er redusert med 17 timer eller 18%. Systemtilgjengeligheten er på  $A_{\text{sys}} = 0,9193$ .

## 6.4.3 Reduksjon av feilraten til de tre feilhendelsene med høyest feilrate

Nå har man sett på et tilfelle hvor man har redusert feilraten for en feilhendelse i ett feiltre (kapittel 6.4.1) og et tilfelle hvor man har redusert feilraten til den feilhendelsen med høyest feilrate i hvert feiltre (kapittel 6.4.2). I dette delkapittelet blir de tre feilhendelsene med høyest feilrate i systemet generelt sett forbedret samtidig slik som tilfellet var for feilhendelsene i kapittel 6.4.2. De tre feilhendelsene det gjelder er feilhendelse 9, 10 og 11, som alle er i feiltre 2. *Tabell 6.11* viser resultatene for MTBF til feiltrærne og systemet sammen med total korrektiv vedlikeholdstid  $T_0$  og systemtilgjengeligheten  $A_{\text{sys}}$ .

<b>Hovedoppsett med høyere pålitelighet for FH 9,10 og 11</b>						
<i>p = 1000, T = 2628000</i>						
Reduksjon i %	MTBF FT1	MTBF FT2	MTBF FT3	MTBF SYS	T0 sys	Asys
0	<b>28482</b>	<b>16701</b>	<b>25366</b>	<b>7440</b>	<b>94,99</b>	<b>0,9185</b>
10	<b>28482</b>	<b>17791</b>	<b>25366</b>	<b>7649</b>	<b>90,45</b>	<b>0,9187</b>
25	<b>28482</b>	<b>20060</b>	<b>25366</b>	<b>8040</b>	<b>82,70</b>	<b>0,9191</b>
50	<b>28482</b>	<b>24915</b>	<b>25366</b>	<b>8721</b>	<b>70,72</b>	<b>0,9197</b>
70	<b>28482</b>	<b>30904</b>	<b>25366</b>	<b>9355</b>	<b>61,15</b>	<b>0,9201</b>
90	<b>28482</b>	<b>40090</b>	<b>25366</b>	<b>10053</b>	<b>52,04</b>	<b>0,9206</b>

*Tabell 6.11: Seriestruktur med forbedret feilrate for feilhendelse 9,10 og 11.*

Tabell 6.11 viser at når man forbedrer feilraten til de tre feilhendelsene med høyest feilrate samtidig vil det ha relativ stor innvirkning på totalsystemet. MTBF for feiltre 1 og 3 holder seg konstant gjennom hele reduksjonsprosessen fordi det kun er feilhendelser i feiltre 2 som er involvert. MTBF for feiltre 2 har en økning på hele 23389 timer eller 140% etter at man har redusert feilraten til feilhendelse 9, 10 og 11 med 90% hver. MTBF for systemet har økt med 2613 timer eller 35%. Samtidig ser man at total korrektiv vedlikeholdstid  $T_0$  for systemet er redusert med 43 timer eller 45%.

Systemtilgjengeligheten som er oppnådd her er like høy som den som ble oppnådd i systemet med redundans på alle aminpumper. Nå er det ikke sikkert at det er praktisk mulig å redusere feilraten til enkelte feilhendelser med 90%, men å installere ekstra pumper i reserve er fullt mulig. Dette viser at hvis man skal ha et komplisert system med høy tilgjengelighet er man avhengig av å ha mange utstyrskomponenter i en redundanskobling.





## 7 Konklusjon og videre arbeid

### 7.1 Konklusjon

Gjennom arbeidet med dette temaet har det vist seg at det foreløpig ikke er gjort mye utvikling innenfor området som tar for seg RAM-analyser av CO<sub>2</sub>-rensing. Dette er ikke veldig overraskende med tanke på at det ikke fins storskala anlegg for separasjon av CO<sub>2</sub> fra eksosgass. Det som er gjort i denne oppgaven er derfor basert på mange antakelser og forenklinger og man kan ikke ta i bruk metodikken slik den er nå uten å endre på feilhendelser og feiltreoppbygning, hvis man skal analysere et CO<sub>2</sub>-renseanlegg.

Metodikken er likevel egnet som et fundament for videre RAM-studier av CO<sub>2</sub>-rensing. Det som er beskrevet i denne oppgaven vil være et godt utgangspunkt for andre som ønsker å utvikle metoder for tilgjengelighetsstudier av separasjonsprosesser for CO<sub>2</sub>-fangst. Metodikken beskrevet i denne rapporten er forholdsvis enkelt oppbygd i den forstand at det er antatt uavhengighet for alle feilhendelsene. Dette er gjort for å kunne benytte seg av likningssettene gitt i teoridelen. I mer avanserte og kommersielle RAM-verktøy vil det i mange tilfeller være mulig å gjøre mer kompliserte beregninger. I denne oppgaven er metodikken utarbeidet fra bunn av med utgangspunkt i en artikkel av Thangamani, G. et al. [1], i tillegg til generell teori om systemanalyse og Monte Carlo simulering.

Det er ikke utviklet en god modell for simulering og beregning av reparasjonsrater, som nok vil være den største svakheten til metodikken. Det har dessverre vært vanskelig å finne gode løsninger for hvordan reparasjonsratene kunne implementeres med det informasjonsgrunnlaget og datagrunnlaget som fins. Det har derfor blitt lagt lite vekt på utarbeidelse av modell for reparasjonsratene. Man må likevel understreke at det er viktig å kunne modellere gode og realistiske reparasjonsrater med tanke på å få riktige anslag for dødtiden til anlegget. Metodikken har til tross for en manglende modell for reparasjonsraten en generell struktur som kan være en god og enkel modell for simulering av ”levetider” og beregning av operasjonell tilgjengelighet.

På grunn av at det er benyttet mye feildata for utstyr som ikke er tilknyttet et CO<sub>2</sub>-renseanlegg er det vanskelig å gi noen spesifikke konklusjoner for tilgjengeligheten til et slikt anlegg. Man kan på et generelt grunnlag se at redundans av utstyrskomponenter har en stor effekt på reduksjonen av korrektiv vedlikeholdstid, sammenliknet med et seriekoblet system hvor man har redusert feilraten til enkelte feilhendelser betraktelig. Slik som anlegget er satt opp i denne oppgaven vil ikke den korrektive vedlikeholdstiden ha mye å si for dødtiden til anlegget. Dette er fordi man har implementert en forebyggende vedlikeholdsfase hvert andre år som varer i to måneder og dermed er en dominerende faktor for dødtiden.

Det er gjort et forsøk på å sammenlikne fordelingen for ”levetidene” til simuleringsresultatene for systemet og enkelte feilhendelser med spesifikke teoretiske fordelinger. Resultatene viste at fordelingen til systemet kunne være weibull-, gamma- eller normalfordelt. I praksis er det nok mest sannsynlig at en weibullfordeling vil gi den beste fordelingen på grunn av at den er

mest fleksibel. Normalfordelingen er symmetrisk mens det i praksis vil oppstå enkelte skjevheter i fordelingen som den ikke kan takle like godt. Det er likevel ikke gjort grundige nok analyser av det statistiske grunnlaget til systemene til å gi noen konklusjoner om hvilken fordelingsfunksjon som egner seg best. Det betyr at det også ikke er gjort noen forsøk på å kvantifisere usikkerheter ved hjelp av fordelingsfunksjoner. Metoden som er vist for sammenlikning av ulike fordelinger gir en grov indikasjon på hvilke tendenser som er fremtredende for de simulerte verdiene.

Det har vært vanskelig å verifisere om metodikken som er benyttet gir riktige resultater sammenliknet med et reelt anlegg. Hovedårsaken til dette er at det ikke har vært mulig å skaffe erfaringsdata fra eksisterende anlegg som man kunne sammenliknet de simulerte verdiene med. I tillegg kreves det veldig god prosessforståelse og gode systemskisser av eksisterende anlegg for å kunne sette opp en riktig feiltrestruktur med reelle feilårsaker for systemet.

## 7.2 Videre arbeid

Det må bli et større fokus på utarbeidelsen av gode RAM-analyser av CO<sub>2</sub>-fangst for at man skal kunne ta i bruk fangstteknologien for praktiske formål. Det er enorme investeringer som ligger til grunn for denne teknologien og da er man nødt til å være sikker på at effektiviteten til et anlegg er på topp. Problemet er at det er veldig vanskelig å utføre gode analyser og trekke riktige konklusjoner når man ikke har erfaringsdata å benytte seg av. Det er derfor viktig at det blir bygget et storskala anlegg for separasjon av CO<sub>2</sub> fra eksosgass slik at man kan samle informasjon som kan benyttes til å forbedre teknologien ytterligere.

Metodikken utarbeidet i denne oppgaven hadde fått et stort løft hvis det var databaser av typen OREDA med feildata for komponenter benyttet til aminrensing. Det er fullt mulig å få samlet denne typen data i dag på grunn av at det fins mange anlegg rundt omkring i verden som renses gasstrømmen for CO<sub>2</sub> for å møte produktspesifikasjonene til salgsgassen. En feildatabase ville vært et godt utgangspunkt for videre utvikling av RAM-studier ved at man da ville hatt et nyttig verktøy til bruk i analysen.

Bærebjelken i tilgjengelighetsstudien av fangstanlegget er en grundig og utfyllende analyse av feilårsaker og feilsammenhenger i systemet. For å utføre slike analyser må man ha gode systemskisser og god innsikt i prosessen og hvilke feil som kan oppstå. Dette kan være et svært omfattende arbeid, men like fullt noe av det viktigste arbeidet som må gjøres. Systemskisser av kompliserte anlegg er noe som svært sjelden er tilgjengelig for offentligheten slik at det må inngås en avtale med en industribedrift som kan skaffe denne informasjonen. I tillegg vil nok en bedrift som har utviklet denne typen teknologi ha erfaringsdata om feil og reparasjoner hvis et slikt anlegg er satt i drift.

Når man har erfaringsdata på plass kan det være fordelaktig å utvikle det statistiske grunnlaget gitt i denne oppgaven videre. På denne måten får man kvantifisert usikkerheter i

systemet på en god måte. Det kan beregnes fordelinger for systemet ved å anslå en fordelingsfunksjon for enkeltkomponenter og sette dette i en sammenheng. Beregningene kan sammenliknes med simuleringsresultater fra metodikken for å kunne verifisere om simuleringen gir gode resultater. For å kunne ta i bruk metodikken i en større sammenheng må man vite om den gir riktige resultater, og en sammenlikning med konkrete erfaringsdata vil være et godt utgangspunkt for dette.

Total driftstid mellom hver revisjonsstans  $T_d$  og forebyggende vedlikeholdstid  $T_s$  er i denne oppgaven valgt på grunnlag av hva som er gjort i referansematerialet av Thangamani, G. et al. [1]. Det er anbefalt å legge en del ressurser i å finne riktige estimater for  $T_d$  og  $T_s$  i og med at dette er viktige parametere for beregning av operasjonell tilgjengelighet. Hvis det for eksempel viser seg at man ikke behøver å utføre en revisjonsstans mer enn hvert tredje år og at anlegget ikke er ute av drift mer enn en måned for hver revisjonsstans, vil den korrektive vedlikeholdstiden  $T_0$  ha større innvirkning på systemtilgjengeligheten enn hva den har med nåværende oppsett.

Den største mangelen som er identifisert med metodikken er mangelen av en god modell for reparasjonsrater. Et steg videre kan være å benytte lognormalfordelingen til å sette opp en modell for reparasjonsrater. Dette er nokså krevende ved at man er nødt til å ha god innsikt i det statistiske grunnlaget, men hvis metodikken skal bli realistisk er det viktig å få på plass en god modell for reparasjonsrater. I tillegg ville det vært gunstig om også erfaringsdata for reparasjonstider kunne samles i et statistisk oppslagsverk av typen OREDA.



## 8 Referanser

1. Thangamani, G., T.T. Narendran, and R. Subramanian, *Assessment of availability of a Fluid Catalytic Cracking Unit through simulation*. Reliability Engineering & System Safety, 1995. **47**(3): p. 207-220.
2. Solbraa, E., *Muntlig og skriftlig korrespondanse*, R. StatoilHydro R&D, Editor. 2009: Trondheim.
3. Aspen HYSYS. 2008, Aspen Technology, inc.: Burlington, MA
4. Pro/II. 2004, Invensys Systems, Inc/ SimSci-Esscor: Lake Forest, CA.
5. SINTEF Industrial Management, *Offshore Reliability Data Handbook*. 4th Edition ed. 2002, Trondheim: OREDA Participants: ENI S.p.A/AGIP Exploration & Production, BP Exploration Operating Company Ltd., ExxonMobil International Ltd., Norsk Hydro ASA, Phillips Petroleum Company Norway, Statoil ASA, Shell Exploration & Production, TotalFinaElf.
6. Rausand, M. and A. Høyland, *System Reliability Theory: Models and statistical methods*. first ed. Wiley series in probability and mathematical statistics, ed. Wiley. 1994, New York: John Wiley & Sons, Inc. 518.
7. Norwegian Technology Standards Institution, N., *Regularity Management & Reliability Technology in Z-016*. 1998, Norwegian Technology Standards Institution: Oslo, Norway.
8. International Organization for Standardization, *Petroleum, petrochemical and natural gas industries - Production assurance and reliability mangement*, in ISO 20815. 2008, TC 67 Materials, equipment and offshore structures for petroleum, petrochemical and natural gas industries: online.
9. McDermott, R.E., R.J. Mikulak, and M.R. Beauregard, *The Basics of FMEA*. 1996: Productivity.
10. Intergovernmental Panel on Climate Change, I., *Carbon Dioxide Capture and Storage*, O.D. Bert Metz, Heleen de Coninck, Manuela Loos, Leo Meyer Editor. 2005, IPCC: Cambridge, UK.
11. Wu, Y. and C.W. Chan, *A data analysis decision support system for the carbon dioxide capture process*. Expert Systems with Applications, 2009. **36**(6): p. 9949-9960.
12. Zhou, Q., C.W. Chan, and P. Tontiwachiwuthikul, *A monitoring and diagnostic expert system for carbon dioxide capture*. Expert Systems with Applications, 2009. **36**(2, Part 1): p. 1621-1631.
13. Abu-Zahra, M.R.M., et al., *CO<sub>2</sub> capture from power plants: Part I. A parametric study of the technical performance based on monoethanolamine*. International Journal of Greenhouse Gas Control, 2007. **1**(1): p. 37-46.
14. Nord, L.O., et al., *A qualitative reliability and operability analysis of an integrated reforming combined cycle plant with CO<sub>2</sub> capture*. International Journal of Greenhouse Gas Control, 2009. **In Press, Corrected Proof**.
15. Bolland, O., *Power Generation: CO<sub>2</sub> Capture and Storage*, in *Kompendie i modulemnet TEP04*. 2008, NTNU: Trondheim.
16. Tobiesen, F.A., H.F. Svendsen, and K.A. Hoff, *Desorber energy consumption amine based absorption plants*. International Journal of Green Energy 2, 2005: p. 201-215.
17. Myhrvold, T., *Muntlig korrespondanse*, D.N.V. Senior forsker, Editor. 2009: Trondheim.

18. Mann, N.R., R.E. Schafer, and N.D. Singpurwalla, *Methods for statistical analysis of reliability and life data*. Wiley series in probability and mathematical statistics, ed. Wiley. 1974, New York: John Wiley & Sons. 545.
19. Lees, F.P., *Lees' loss prevention in the process industries: Hazard identification, assessment and control*. Third ed. Vol. 1. 2005, Burlington, MA: Elsevier Butterworth-Heinemann.
20. Walpole, R.E., et al., *Probability and statistics for engineers & scientists*. eighth ed. Pearson Education International, ed. P.P. Hall. 2007, New Jersey: Peatson Eduction, Inc. 816.
21. Johnson, N.L. and S. Kotz, *Continuous univariate distributions*. The Houghton Mifflin Series in Statistics, ed. H. Chernoff. Vol. 1. 1970, Boston: Houhton Mifflin Company. 300.
22. Barlow, R.E. and F. Proschan, *Mathematical theory of reliability* 1965, New York: John Wiley and Sons.
23. The MathWorks, I., *MATLAB*. 1984-2008.

## VEDLEGG

### A: Tilpasning av simulerte verdier til spesifikke fordelinger i MATLAB

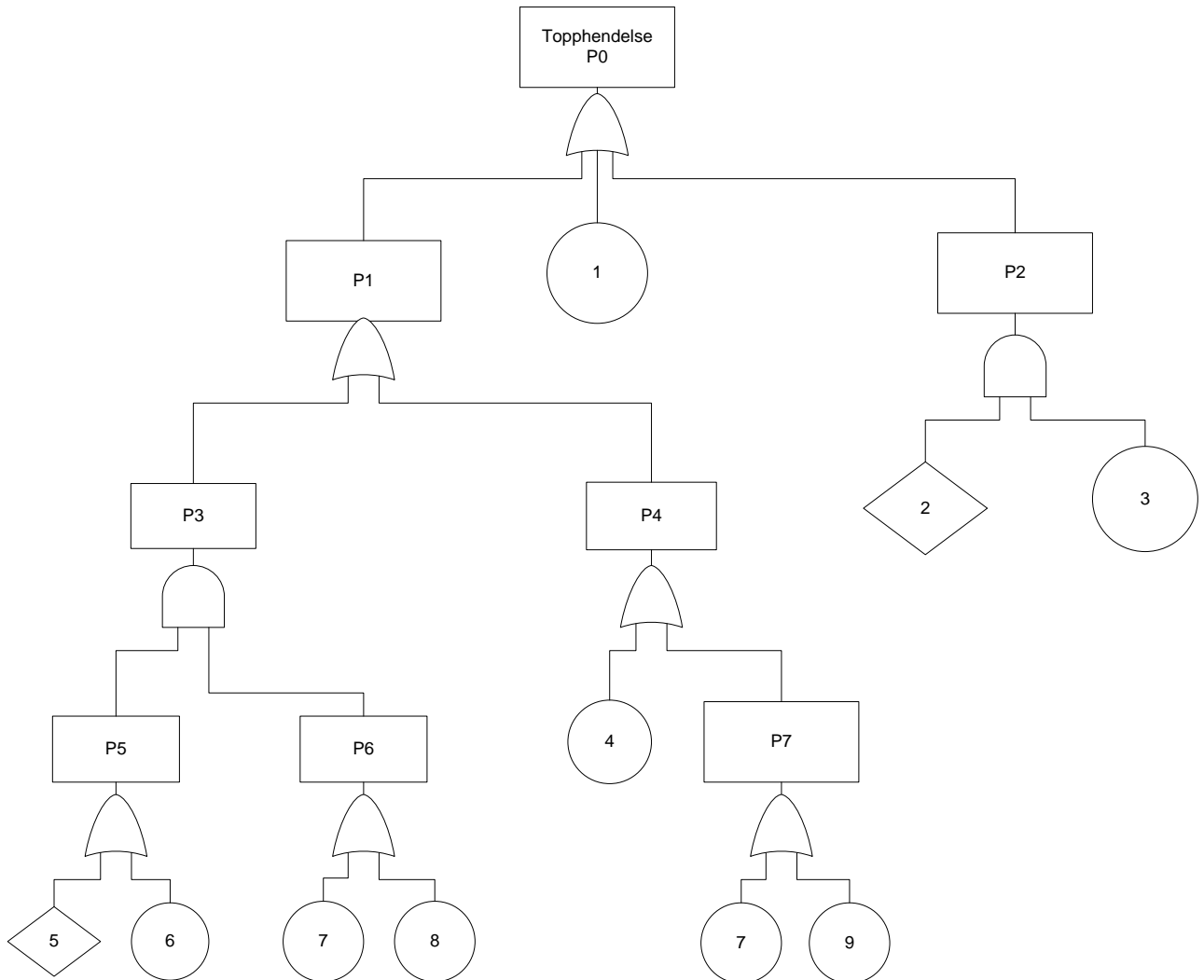
I MATLAB benyttes algoritmene  $gamfit(DATA)$ ,  $normfit(DATA)$  og  $wblfit(DATA)$  til å finne maksimale sannsynlighetsestimater av parametrene i henholdsvis gammafordelingen, normalfordelingen og weibullfordelingen gitt av utvalget i DATA.

Dette vil si at man kan ta de estimerte verdiene av MTTF beregnet av simuleringialgoritmen for å få fram estimatorene  $a = \alpha^*$ ,  $b = 1/\lambda^*$ ,  $c = v^*$  og  $d = \tau^*$ . Disse kan man igjen benytte til å plote en gamma-CDF,  $gamcdf(f,a,b)$ , en normal-CDF,  $normcdf(f,c,d)$  og en weibull-CDF,  $wblcdf(f,a,b)$ , sammen med den empiriske fordelingsfunksjonen.  $f$  uttrykker x-verdiene i utregningen av funksjonen. På denne måten kan man se om fordelingen til de simulerte verdiene kan tilnærmes en spesifikk fordelingsfunksjon basert på simulerte verdier for levetider.

For eksponentialfordelingen kan man benytte seg av middelverdien til simulerte MTTF-verdier for å plote en eksponential-CDF. Man kan ikke benytte denne middelverdien som inputverdi i gamma- og weibullfunksjonen på grunn av at det ikke vil gi riktige estimater. Dette kommer av hvordan MTTF er definert i disse to fordelingene som man kan se av likningene (4.17) og (4.25).

## B: MOCUS (Metode for å oppnå "cut sets")

Den beste måten å forklare MOCUS på er gjennom et eksempel. Figur 2 viser et feiltre hvor portene er nummerert fra P0 til P7.



*Figur B.1: Eksempel på feiltre*

Algoritmen starter på P0-porten rett under topphendelsen. Hvis dette er en "ELLER" port så vil hver inndata til porten bli satt opp i forskjellige rekker. Hvis P0 er en "OG" port vil inndataene settes opp i forskjellige kolonner. I dette eksempelet er P0 en "ELLER" port så vi starter med følgende:

P1

1

P2

66



Siden alle de tre inndataene P1, 1 og P2 vil føre til at topphendelsen inntreffer vil hver av dem utgjøre et "cut set". Videre må man erstatte hver port med inndataene til porten helt til man har gått gjennom hele feiltreet og kun står igjen med grunnleggende hendelser.

- Siden P1 er en "ELLER" port:

P3

P4

1

P2

- Siden P2 og P3 er "OG" porter:

P5, P6

P4

1

2, 3

- Siden P4 og P5 er "ELLER" porter:

5, P6

6, P6

4

P7

1

2, 3

- Siden P6 er en "ELLER" port:

5, 7

5, 8

6, 7

6, 8

4

P7

1

67

2, 3

- Siden P7 er en "ELLER" port:

5, 7

5, 8

6, 7

6, 8

4

7

9

1

2, 3

Radene som står igjen representerer "cut sets" i feiltreet. Dette eksempelet består da av følgende 9 "cut sets":  $\{1\}$ ,  $\{4\}$ ,  $\{7\}$ ,  $\{9\}$ ,  $\{2, 3\}$ ,  $\{5, 7\}$ ,  $\{5, 8\}$ ,  $\{6, 7\}$  og  $\{6, 8\}$ . Siden  $\{7\}$  er et "cut set" vil ikke  $\{5, 7\}$  og  $\{6, 7\}$  være minimale. Følgende 7 "cut sets" vil derfor være minimum "cut sets" for feiltreet i eksempelet:

$\{1\}$ ,  $\{4\}$ ,  $\{7\}$ ,  $\{9\}$ ,  $\{2, 3\}$ ,  $\{5, 8\}$ ,  $\{6, 8\}$

Det er altså 4 minimum "cut sets" av 1.orden og 3 minimum "cut sets" av 2.orden. Årsaken til at algoritmen ikke fører til minimum "cut sets" er at grunnleggende hendelse 7 inntreffer flere steder i feiltreet.

## C: Forenklet FMECA av et aminrenseanlegg

Undersystem	Komponent	Funksjon	Funksjonelt krav
Aminanlegg	Vifte	Overvinne trykktap i absorberen	Levere ønsket trykkøkning
	Absorber	Rense røykgassen for CO <sub>2</sub> vha amin	40°C ≥ T <sub>abs</sub> ≥ 60°C
	Sentrifugalpumpe (tynn amin/rik amin)	Overvinne trykktap i nedstrøms varmeveksler	Leverer ønsket trykkøkning
	Aminvarmeveksler	Varme tynn amin og kjøle rik amin	15°C ≥ ΔT <sub>pinch</sub> ≥ 5°C
	Stripper	Rense tynn amin for CO <sub>2</sub> vha høy temp	130°C ≥ T <sub>str</sub> ≥ 100°C
	Kjølevannsvexler (rik amin)	Kjøle rik amin før det går inn på absorber	TH, ut ≤ 60°C
	Koker	Leverer varme til stripper	130°C ≥ T <sub>str</sub> ≥ 100°C
	Kjølevannsvexler CO <sub>2</sub>	Kjøle ned CO <sub>2</sub> som går ut på toppen av stripper før det går inn i flashtanken	Senke temp på CO <sub>2</sub> -mix nok til å få ut ønsket mengde amin/H <sub>2</sub> O i flashtank
	Flashtank for CO <sub>2</sub> -kompresjon	Skille ut CO <sub>2</sub> fra amin/H <sub>2</sub> O før CO <sub>2</sub> går til kompresjon	Kondensere ut ønsket mengde amin/H <sub>2</sub> O

Figur C.1: Forenklet FMECA av aminrenseanlegg del 1

Feilmodus	Feilårsak	Effekt på samme komponent
Utilstrekkelig trykkøkning	Mekanisk skade	Stans
Tabbs utenfor funksjonelt område	Korrosjon	Lavere absorpsjonsevne av CO2
	Ekstern lekkasje av prosessmedium	Lavere absorpsjonsevne av CO2
Utilstrekkelig trykkøkning	Mekanisk skade	Stans
	Ekstern lekkasje av prosessmedium	Reduksjon i effektivitet
	Uønsket stans	Stans
	Uregelmessig instrumentavlesning	Leverer for mye/for lite effekt
$\Delta T$ pinch utenfor funksjonelt område	Korrosjon	Lavere varmeoverføring
	Ekstern lekkasje av prosessmedium	Lavere varmeoverføring
	Groing	Lavere varmeoverføring
	Korrosjon	Lavere gjennvinningsvne av amin
Tstr < 100°C	Ekstern lekkasje av prosessmedium	Lavere gjennvinningsvne av amin
	Korrosjon	Lavere varmeoverføring
TH, ut > 60°C	Ekstern lekkasje av prosessmedium	Lavere varmeoverføring
	Groing	Lavere varmeoverføring
	Strukturell mangel	Lavere gjennvinningsvne av amin
Tstr < 100°C	Uregelmessig instrumentavlesning	Leverer for mye/for lite effekt
	Korrosjon	Lavere effektivitet
	Ekstern lekkasje av prosessmedium	Lavere effektivitet
Utilstrekkelig senking av temp på CO2-mix	Korrosjon	Lavere varmeoverføring
	Ekstern lekkasje av CO2-blanding	Lavere varmeoverføring
	Groing	Lavere varmeoverføring
Utilstrekkelig kondensasjon av amin/H2O	Uregelmessig instrumentavlesning	Ingen
	Korrosjon	Ingen

Figur C.2: Forenklet FMECA av aminrenseanlegg del 2

Effekt på andre komponenter	Effekter på systemets helhetlige funksjon	Deteksjon	Feilrate	Konsekvens	Risiko
Stans hele anlegget	Stans hele anlegget	1	1	3	3
Kan føre til slitasje på nedstrøms utstyr pga partikkelavsetning	Generelt dårligere effektivitet	2	1	2	4
Lavere effektivitet pga tap av amin	Stans pga lekkasje av amin	1	1	3	3
Stans hele anlegget	Stans hele anlegget	1	1	3	3
Generell dårligere effektivitet i anlegg	Stans pga lekkasje av amin	1	1	3	3
Stans hele anlegget	Stans hele anlegget	1	2	3	6
Ikke optimal drift av nedstrøms utstyr	Generelt dårligere effektivitet	2	3	1	6
Lavere desorbsjonsevne i stripper, større krav til kjølevannsvexler (rik amin)	Generelt dårligere effektivitet	2	1	2	4
og slitasje av nedstrøms utstyr pga partikkelavsetning					
Lavere effektivitet pga tap av amin	Stans pga lekkasje av amin	1	1	3	3
Lavere desorbsjonsevne i stripper og større krav til kjølevannsvexler (rik amin)	Generelt dårligere effektivitet	2	1	2	4
Krever mer effekt av koker / slitasje av nedstrøms utstyr pga partikkelavsetning	Generelt dårligere effektivitet	2	1	2	4
Krever mer effekt av koker / lavere effektivitet pga tap av amin	Stans pga lekkasje av amin	1	1	3	3
Lavere absorpsjonsevne i absorber / slitasje av nedstrøms utstyr pga partikkelavsetning	Generelt dårligere effektivitet	2	1	2	4
Lavere absorpsjonsevne i absorber	Stans pga lekkasje av amin	1	1	3	3
Krever mer effekt av koker	Generelt dårligere effektivitet	2	1	2	4
Stans hele anlegget	Stans hele anlegget	1	1	3	3
Ikke optimal drift av stripper	Generelt dårligere effektivitet	2	1	1	2
Krever mer effekt fra dampen / slitasje av nedstrøms utstyr pga partikkelavsetning	Generelt dårligere effektivitet	2	1	2	4
Lavere desorbsjonsevne i stripper	Stans pga lekkasje av amin	1	1	3	3
Mindre utskilling av CO2 i flashtank	Mindre CO2 til kompresjon	2	1	2	4
Mindre utskilling av CO2 i flashtank	Stans pga lekkasje av amin	1	1	3	3
Mindre utskilling av CO2 i flashtank	Mindre CO2 til kompresjon	2	1	2	4
Ikke optimal effekt på nedstrøms utstyr	Ueffektiv CO2-kompresjonstog/stripper	2	2	1	4
Slitasje av nedstrøms utstyr pga partikkelavsetning	Generelt dårligere effektivitet	2	1	2	4

Figur C.3: Forenklet FMECA av aminrenseanlegg del 3

## D: MATLAB-kode

### D.1: Hovedoppsett

```
clc

% Det er ønskelig å generere ulike tilfeldige tall for hver gang man
% kjører programmet. Denne algoritmen gjør at rand returnerer ulike
% tilfeldige verdier i intervallet [0,1] avhengig av hva systemklokka er

RandStream.setDefaultStream(RandStream('mt19937ar','seed',sum(1000*clock)))
;

%% CASE 1: HOVEDOPPSETT

% Monte Carlo simulering av tre feiltrær der man ikke har redundans av
% noen komponenter slik at alt er seriekoblet

% Det antas at feilraten for de ulike feilhendelsene er eksponentialt
% fordelt mens reparasjonstiden er satt konstant. Det antas videre at en
% testperiode strekker seg over 2 år. Likevel må man fortsatt simulere
% prosessen over en mye lengre tid enn 2 år for å kunne oppnå realistiske
% verdier for MTTF for de ulike feilhendelsene. Det er derfor utarbeidet
% en sensureringsalgoritme som regner ut hvor stor andel av
% feilhendelsene som har en levetid som er lengre enn simuleringstiden

% Simuleringen beregner MTTF for feilhendelser, feiltrær og systemet som
% helhet. De beregnede verdiene for gjennomsnittlig tid til feil for
% systemet MTTFs benyttes til å vise en empirisk cdf for levetiden til
% systemet. Man kan da benytte seg av middelverdien til de p utregnede
% MTTFs som en inputverdi i beregningen av en cdf som er eksponential-,
% weibull-, normal- og gammafordelt. På denne måten kan man se hvilken
% fordeling som passer best til de empiriske verdiene

% Her anslås systemtilgjengeligheten som en funksjon av totaltiden før
% revisjonsstans Td, og reparasjonstiden Tr som inkluderer forebyggende
% og korrektivt vedlikehold. Korrektivt vedlikeholdstid er en
% funksjon av MTBF og MTTR for de ulike feiltrærne (kapitel 5.3)

% - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * -
% * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * -

x = [0.1005648, 0.010512, 0.027156, 0.027156, 0.028908,
     0.0454644,0.0184836, 0.0184836, 0.1146684, 0.1146684, 0.1146684,
     0.0090228, 0.0090228, 0.0677148, 0.0677148, 0.028908, 0.0454644,
     0.028908, 0.0454644, 0.028908, 0.0454644, 0.0706056, 0.0281196];

y = [2.8, 15.9, 9.2, 9.2, 45, 5, 81.2, 81.2, 70.4, 70.4, 70.4, 75,
     75, 41, 41, 45, 5, 45, 5, 45, 5, 20.2, 51];

% x er en vektor med feilraten til hver feilhendelse i [feil/år]
% y er en vektor med reparasjonstiden til hver feil xi som inntreffer i
% [timer]
```

```

dt = 1/12;          % dt er tidsskritt tilsvarende 1 mnd der man regner tiden
                   % i [år]
t = dt:dt:600;     % Antall tidsskritt fra det første tidsskrittet til
                   % totaltiden av simuleringen [år]

n = length(t);    % Lengden av t-vektoren
m = length(x);    % Lengden av x-vektoren

p = 1000;         % Antall simuleringer for hvert system

Tt = n*dt*24*365; % Totaltiden av simuleringen [timer]
TTF = zeros(n,m); % Matrise som tar seg av tid til feil for hver
                 % feilhendelse delt inn i de ulike tidsstegene
MTTF = zeros(m,p); % Matrise som tar seg av gjennomsnittlig tid til feil
                 % for hver feilhendelse
MTTFsm = zeros(3,p); % Matrise som tar seg av gjennomsnittlig tid til feil
                 % for systemet i alle de tre feiltreene
MTTFsv = zeros(1,p); % Vektor som tar seg av gjennomsnittlig tid til feil
                 % for totalsystemet
MTBFsv = zeros(1,p); % Vektor som tar seg av gjennomsnittlig tid mellom
                 % feil for totalsystemet
omega1 = zeros(8,p); % Matrise som tar seg av verdiene til ROCOF for det
                 % første feiltreet med 8 "cut sets"
omega2 = zeros(7,p); % Matrise som tar seg av verdiene til ROCOF for det
                 % andre feiltreet med 7 "cut sets"
omega3 = zeros(8,p); % Matrise som tar seg av verdiene til ROCOF for det
                 % tredje feiltreet med 8 "cut sets"
MTBFsm = zeros(3,p); % Matrise som tar seg av gjennomsnittlig tid mellom
                 % feil for systemet i alle de tre feiltreene
MTTRsm = zeros(3,p); % Matrise som tar seg av gjennomsnittlig tid til
                 % reparasjon for systemet i alle de tre feiltreene
T0 = zeros(1,p);    % Vektor som tar seg av den totale tiden brukt på
                 % korrigert vedlikehold for hver av de angitte
                 % feilhendelsene [timer]
Tr = zeros(1,p);    % Vektor som tar seg av total reparasjonstid inkludert
                 % forebyggende vedlikehold ifm revisjonsstans
Ts = 2*dt*24*365;   % Total reparasjonstid ifm forebyggende vedlikehold
                 % ved revisjonsstans (lik 2 mnd) [timer]
Td = 2*24*365;      % Total tid mellom revisjonsstanser for anlegget (lik
                 % 2 år) [timer]
Asv = zeros(1,p);   % Vektor som tar seg av totalsystemtilgjengeligheten

q = x.*dt;          % Sannsynligheten for feil i et tidsskritt for hver
                   % feilhendelse

%% FORLØKKER SOM SIMULERER PROSESSEN FOR n ANTALL TIDSSKRITT, m ANTALL
%% FEILHENDELSER OG p ANTALL SIMULERINGER

for c3 = 1:p        % TELLER FOR ANTALL SIMULERINGER

    TTF = 0;        % Nullstiller tid til feil vektoren ved utregningen
                   % av en ny komponent
    r = rand(n,m);  % Genererer en vektor med tilfeldige variable for den
                   % aktuelle feilhendelsen ved hvert tidsskritt

```

```

for c2 = 1:length(x) % TELLER FOR FEILHENDELSE

    dt_ind = 0; % Nullstiller variabel som indikerer hvilket
                % tidsskritt siste feil inntraff
    dt_teller = 0; % Nullstiller teller som angir hvor mange feil
                % som inntreffer for en feilhendelse ila
                % totaltiden

        for c1 = 1:length(t) % TELLER FOR TIDSSKRITT

            if (r(c1,c2)<= q(c2)) % Sjekker om sannsynligheten for feil
                % i en feilhendelse er større enn de
                % tilfeldig genererte variablene

                TTF(c1,c2) = (c1-dt_ind)*dt*24*365; % Tilordner tiden det
                    % tar til feil
                    % inntreffer for den
                    % gitte feilhendelse
                    % ved det gitte
                    % tidsskritt i [timer]

                dt_teller = dt_teller+1; % Antall ganger feil inntreffer
                    % for angitt feilhendelse ila
                    % totaltiden adderes med 1

                dt_ind = c1; % Teller som passer på i hvilket
                    % tidsskritt siste feil for angitt
                    % feilhendelse inntraff

            end

        end

    if dt_teller == 0 % If-setning som angir hva som skjer hvis det
        % ikke inntreffer feil på angitt feilhendelse
        % ila totaltiden

        MTTF(c2,c3) = Tt; % Gjennomsnittlig tid til feil er satt lik
            % totaltiden pga at feil ikke har
            % inntruffet for angitt feilhendelse.

    else % Hvis det har oppstått feil ila totaltiden

        MTTF(c2,c3) = sum(TTF(:,c2))/dt_teller; % Funksjon som regner ut
            % middelveidien av tid til
            % feil ved hjelp av
            % likning (4.36)

    end

end

end

%% UTREGNING FOR DE ULIKE FEILTRÆRNE

% FEILTRE 1 har følgende 8 "cut sets": {1}, {2}, {3}, {4}, {5}, {6},
% {7}, {8}. Det betyr at alle feilhendelser er koblet i serie

```



```

% Funksjon som regner ut gjennomsnittlige tid til feil for systemet til
% det første feiltreet vha likningene (4.15) og (4.41)
MTTFsm(1,c3)=1/((1/MTTF(1,c3))+(1/MTTF(2,c3))+(1/MTTF(3,c3))+(1/MTTF(4,c3))
+(1/MTTF(5,c3))+(1/MTTF(6,c3))+(1/MTTF(7,c3))+(1/MTTF(8,c3)));

% Regner ut omega-verdiene for alle 8 "cut sets" i det første feiltreet
% vha likningene (4.35) og (4.44)
omega1(1,c3) = 1/(MTTF(1,c3)+y(1)); omega1(2,c3) = 1/(MTTF(2,c3)+y(2));
omega1(3,c3) = 1/(MTTF(3,c3)+y(3)); omega1(4,c3) = 1/(MTTF(4,c3)+y(4));
omega1(5,c3) = 1/(MTTF(5,c3)+y(5)); omega1(6,c3) = 1/(MTTF(6,c3)+y(6));
omega1(7,c3) = 1/(MTTF(7,c3)+y(7)); omega1(8,c3) = 1/(MTTF(8,c3)+y(8));

% Funksjon som regner ut gjennomsnittlige tid mellom feil for systemet
% til det første feiltreet vha likningene (4.45) og (4.46)
MTBFsm(1,c3) = 1 / sum(omega1(:,c3));

% Funksjon som regner ut gjennomsnittlige tid til reparasjon for systemet
% til det første feiltreet vha av likning (4.47)
MTTRsm(1,c3)=(omega1(1,c3)*y(1)+(omega1(2,c3)*y(2)+(omega1(3,c3)*y(3))+
(omega1(4,c3)*y(4)+(omega1(5,c3)*y(5)+(omega1(6,c3)*y(6))+
(omega1(7,c3)*y(7)+(omega1(8,c3)*y(8)))/sum(omega1(:,c3));

% FEILTRE 2 har følgende 7 "cut sets": {9}, {10}, {11}, {12}, {13},
% {14}, {15}. Det betyr at alle feilhendelser er koblet i serie.

% Funksjon som regner ut gjennomsnittlige tid til feil for systemet til
% det andre feiltreet vha likningene (4.15) og (4.41)
MTTFsm(2,c3)=1/((1/MTTF(9,c3))+(1/MTTF(10,c3))+(1/MTTF(11,c3)))+
(1/MTTF(12,c3))+(1/MTTF(13,c3))+(1/MTTF(14,c3))+(1/MTTF(15,c3)));

% Regner ut omega-verdiene for alle 7 "cut sets" i det andre feiltreet
% vha likningene (4.35) og (4.44)
omega2(1,c3) = 1/(MTTF(9,c3)+y(9)); omega2(2,c3) = 1/(MTTF(10,c3)+y(10));
omega2(3,c3) = 1/(MTTF(11,c3)+y(11)); omega2(4,c3) = 1/(MTTF(12,c3)+y(12));
omega2(5,c3) = 1/(MTTF(13,c3)+y(13)); omega2(6,c3) = 1/(MTTF(14,c3)+y(14));
omega2(7,c3) = 1/(MTTF(15,c3)+y(15));

% Funksjon som regner ut gjennomsnittlige tid mellom feil for systemet
% til det andre feiltreet vha likningene (4.45) og (4.46)
MTBFsm(2,c3) = 1 / sum(omega2(:,c3));

% Funksjon som regner ut gjennomsnittlige tid til reparasjon for systemet
% til det andre feiltreet vha av likning (4.47)
MTTRsm(2,c3)=(omega2(1,c3)*y(9)+(omega2(2,c3)*y(10)+(omega2(3,c3)*y(11))
+(omega2(4,c3)*y(12)+(omega2(5,c3)*y(13)+(omega2(6,c3)*y(14))+
(omega2(7,c3)*y(15)))/sum(omega2(:,c3));

% FEILTRE 3 har følgende 8 "cut sets": {16}, {17}, {18}, {19}, {20},
% {21}, {22}, {23}. Det betyr at alle feilhendelser er koblet i serie

% Funksjon som regner ut gjennomsnittlige tid til feil for systemet til
% det tredje feiltreet vha likning (4.15) og (4.41)
MTTFsm(3,c3)=1/((1/MTTF(16,c3))+(1/MTTF(17,c3))+(1/MTTF(18,c3)))+
(1/MTTF(19,c3))+(1/MTTF(20,c3))+(1/MTTF(21,c3))+(1/MTTF(22,c3))+
(1/MTTF(23,c3)));

```

```

% Regner ut omega-verdiene for alle 8 "cut sets" i det tredje feiltreet
% vha likningene (4.35) og (4.44)
omega3(1,c3) = 1/(MTTF(16,c3)+y(16)); omega3(2,c3) = 1/(MTTF(17,c3)+y(17));
omega3(3,c3) = 1/(MTTF(18,c3)+y(18)); omega3(4,c3) = 1/(MTTF(19,c3)+y(19));
omega3(5,c3) = 1/(MTTF(20,c3)+y(20)); omega3(6,c3) = 1/(MTTF(21,c3)+y(21));
omega3(7,c3) = 1/(MTTF(22,c3)+y(22)); omega3(8,c3) = 1/(MTTF(23,c3)+y(23));

% Funksjon som regner ut gjennomsnittlige tid til reparasjon for systemet
% til det tredje feiltreet vha likningene (4.45) og (4.46)
MTBFsm(3,c3) = 1 / sum(omega3(:,c3));

% Funksjon som regner ut gjennomsnittlige tid til reparasjon for systemet
% til det tredje feiltreet vha av likning (4.47)
MTTRsm(3,c3)=( omega3(1,c3)*y(16))+(omega3(2,c3)*y(17))+
(omega3(3,c3)*y(18))+(omega3(4,c3)*y(19))+(omega3(5,c3)*y(20))+
(omega3(6,c3)*y(21))+(omega3(7,c3)*y(22))+
(omega3(8,c3)*y(23))/sum(omega3(:,c3));

%% UTREGNING FOR TOTALSYSTEMET

% Funksjon som regner ut gjennomsnittlig tid til feil og
% gjennomsnittlig tid mellom feil for totalsystemet vha linkningene
% (4.15), (4.41) og (4.46)
MTTFsv(c3) = 1 / ((1/MTTFsm(1,c3))+(1/MTTFsm(2,c3))+(1/MTTFsm(3,c3)));
MTBFsv(c3) = 1 / ((1/MTBFsm(1,c3))+(1/MTBFsm(2,c3))+(1/MTBFsm(3,c3)));

% Funksjon som regner ut tiden det tar å utføre korrigerende
% vedlikehold på systemet vha likning (5.1)
T0(c3)=( (MTTRsm(1,c3)/MTBFsm(1,c3)+(MTTRsm(2,c3)/MTBFsm(2,c3))+
(MTTRsm(3,c3)/MTBFsm(3,c3)))*Td);

% Funksjon som regner ut den totale tiden brukt på systemreparasjon
% inkludert forebyggende vedlikehold
Tr(c3) = T0(c3) + Ts;

% Funksjon som regner ut systemtilgjengeligheten vha likning (5.2)
Asv(c3) = (Td / (Tr(c3) + Td));

end

```

```
%% SENSURERINGSALGORITMER
```

```
% Her lages en test for å se hvor stor del av de simulerte verdiene av MTTF  
% for alle feilhendelsene som er større enn totaltiden til simuleringen
```

```
obstime = sort(min(Tt, MTTF));  
fail = obstime(obstime<Tt);  
nfail = length(fail);  
surv= obstime(obstime>=Tt);  
nsurv = length(surv);  
censored_rate = nsurv/(m*p)
```

```
% Her lages det en test for å se hvor stor andel av de simulerte verdiene  
% for MTTF til feilhendelsene som inngår i det 1. feiltreet som er større  
% enn totaltiden til simuleringen
```

```
obstimef1 = sort(min(Tt,MTTF(1:8,:)));  
failf1 = obstimef1(obstimef1<Tt);  
nfailf1 = length(failf1);  
survf1= obstimef1(obstimef1>=Tt);  
nsurvf1 = length(survf1);  
censored_ratef1 = nsurvf1/(8*p)
```

```
% Her lages det en test for å se hvor stor andel av de simulerte verdiene  
% for MTTF til feilhendelsene som inngår i det 2. feiltreet som er større  
% enn totaltiden til simuleringen
```

```
obstimef2 = sort(min(Tt,MTTF(9:15,:)));  
failf2 = obstimef2(obstimef2<Tt);  
nfailf2 = length(failf2);  
survf2= obstimef2(obstimef2>=Tt);  
nsurvf2 = length(survf2);  
censored_ratef2 = nsurvf2/(7*p)
```

```
% Her lages det en test for å se hvor stor andel av de simulerte verdiene  
% for MTTF til feilhendelsene som inngår i det 3. feiltreet som er større  
% enn totaltiden til simuleringen
```

```
obstimef3 = sort(min(Tt,MTTF(16:23,:)));  
failf3 = obstimef3(obstimef3<Tt);  
nfailf3 = length(failf3);  
survf3= obstimef3(obstimef3>=Tt);  
nsurvf3 = length(survf3);  
censored_ratef3 = nsurvf3/(8*p)
```

```
% Her lages det en test for å se hvor stor andel av de simulerte verdiene  
% for MTTF for feilhendelse #9 som er større enn totaltiden til  
% simuleringen
```

```
obstime9 = sort(min(Tt, MTTF(9,:)));  
fail9 = obstime9(obstime9<Tt);  
nfail9 = length(fail9);  
surv9= obstime9(obstime9>=Tt);  
nsurv9 = length(surv9);  
censored_rate9 = nsurv9/p
```

```
% Her lages det en test for å se hvor stor andel av de simulerte verdiene  
% for MTTF for feilhendelse #12 som er større enn totaltiden til  
% simuleringen
```

```
obstime12 = sort(min(Tt, MTTF(12,:)));  
fail12 = obstime12(obstime12<Tt);  
nfail12 = length(fail12);  
surv12= obstime12(obstime12>=Tt);  
nsurv12 = length(surv12);  
censored_rate12 = nsurv12/p
```

```
%% SIMULERINGSRESULTATER
```

```
% Diverse resultater som er nyttige å se på
```

```
MTTF_feiltre1 = mean(MTTFsm(1,:))  
MTTF_feiltre2 = mean(MTTFsm(2,:))  
MTTF_feiltre3 = mean(MTTFsm(3,:))  
MTTF_systemet = mean(MTTFsv)  
MTTF_komponent9 = mean(MTTF(9,:))  
MTTF_komponent12 = mean(MTTF(12,:))  
MTTR_feiltre1 = mean(MTTRsm(1,:))  
MTTR_feiltre2 = mean(MTTRsm(2,:))  
MTTR_feiltre3 = mean(MTTRsm(3,:))  
MTBF_feiltre1 = mean(MTBFsm(1,:))  
MTBF_feiltre2 = mean(MTBFsm(2,:))  
MTBF_feiltre3 = mean(MTBFsm(3,:))  
MTBF_systemet = mean(MTBFsv)  
T0_sys = mean(T0)  
A_systemet = mean(Asv)
```

```
% Resultater som gir ut de karakteristiske parametrene for weibull-,  
% gamma- og normalfordelingen og som passer best mulig til den empiriske  
% cdf-funksjonen til dataene som er simulert for MTTF til feilhendelse 9,  
% 12 og systemet
```

```
weibullfit9 = wblfit(MTTF(9,:))  
weibullfit12 = wblfit(MTTF(12,:))  
weibullfitsys = wblfit(MTTFsv)  
gammafit9 = gamfit(MTTF(9,:))  
gammafit12 = gamfit(MTTF(12,:))  
gammafitsys = gamfit(MTTFsv)  
[mu9,sigma9] = normfit(MTTF(9,:))  
[mu12,sigma12] = normfit(MTTF(12,:))  
[musys,sigmasys] = normfit(MTTFsv)
```

```
clf
```

```
%% PLOT AV SIMULERINGSRESULTATER
```

```
% Figur som viser CDF funksjon for tiden til feil for den feilhendelsen som  
% har lavest tid til feil, basert på data fra simulerte verdier av  
% gjennomsnittlig tid til feil for systemet. Sammenliknes med en  
% "teoretisk" weibull-, gamma-, normal- og eksponential-cdf
```

```
figure(1), [h1,stats] = cdfplot(MTTF(9,:))  
hold on  
f1 = 0:1:stats.max;  
g1 = expcdf(f1,stats.mean);  
i1 = gamcdf(f1,gammafit9(1),gammafit9(2));  
j1 = wblcdf(f1,weibullfit9(1),weibullfit9(2));  
k1 = normcdf(f1,mu9,sigma9);  
plot(f1,g1,'-.m','LineWidth',3)  
plot(f1,i1,'--r','LineWidth',3)  
plot(f1,j1,':k','LineWidth',3)  
plot(f1,k1,'g','LineWidth',3)  
legend('Empirisk CDF basert på feilhendelse med lavest teoretisk  
MTTF','Eksponential-CDF basert på middelverdien av MTTF','Gamma-CDF basert  
på MLE','Weibull-CDF basert på MLE','Normal-CDF basert på  
MLE','Location','SE')  
hold off
```

```

% Figur som viser CDF funksjon for tiden til feil for den feilhendelsen som
% har lavest tid til feil, basert på data fra simulerte verdier av
% gjennomsnittlig tid til feil for systemet. Sammenliknes med en
% "teoretisk" weibull-, gamma-, normal- og eksponential-cdf
figure(2), [h2,stats] = cdfplot(MTTF(12,:))
hold on
f2 = 0:1:stats.max;
g2 = expcdf(f2,stats.mean);
i2 = gamcdf(f2,gammafit12(1),gammafit12(2));
j2 = wblcdf(f2,weibullfit12(1),weibullfit12(2));
k2 = normcdf(f2,mu12,sigma12);
plot(f2,g2,'-.m','LineWidth',3)
plot(f2,i2,'--r','LineWidth',3)
plot(f2,j2,':k','LineWidth',3)
plot(f2,k2,'g','LineWidth',3)
legend('Empirisk CDF basert på feilhendelse med høyest teoretisk
MTTF','Eksponential-CDF basert på middelverdien av MTTF','Gamma-CDF basert
på MLE','Weibull-CDF basert på MLE','Normal-CDF basert på
MLE','Location','SE')
hold off

% Figur som viser CDF funksjon for tiden til feil for systemt basert på
% data fra simulerte verdier av gjennomsnittlig tid til feil for systemet.
% Sammenliknes med en teoretisk weibull-, gamma-, normal- og eksponential-
% cdf
figure(3), [h3,stats] = cdfplot(MTTFsv)
hold on
f3 = 0:1:stats.max;
g3 = expcdf(f3,stats.mean);
i3 = gamcdf(f3,gammafitsys(1),gammafitsys(2));
j3 = wblcdf(f3,weibullfitsys(1),weibullfitsys(2));
k3 = normcdf(f3,musys,sigmasys);
plot(f3,g3,'-.m','LineWidth',3)
plot(f3,i3,'--r','LineWidth',3)
plot(f3,j3,':k','LineWidth',3)
plot(f3,k3,'g','LineWidth',3)
legend('Empirisk CDF basert på MTTFs','Eksponential-CDF basert på
middelverdien av MTTF','Gamma-CDF basert på MLE','Weibull-CDF basert på
MLE','Normal-CDF basert på MLE','Location','SE')
hold off

% Lager et plot for å sjekke om funksjonen er IFR eller DFR (kapittel 4.1)
% Hvis funksjonene er konvekse er de IFR og hvis de er konkave er de DFR

% Kode for å sjekke om cdf for feilhendelse 9 er IFR eller DFR
figure(4), [f,z]=ecdf(MTTF(9,:), 'function', 'survivor');
hold on
plot(z,-log(f), 'b', 'LineWidth', 3)
plot(f1,-log(1-g1), '-.m', 'LineWidth', 3)
plot(f1,-log(1-il), '--r', 'LineWidth', 3)
plot(f1,-log(1-j1), ':k', 'LineWidth', 3)
plot(f1,-log(1-k1), 'g', 'LineWidth', 3)
legend('-ln(1-F(t)) for simulerte fordeling', '-ln(1-F(t)) for
eksponetialfordeling', '-ln(1-F(t)) for gammafordeling', '-ln(1-F(t)) for
weibullfordeling', '-ln(1-F(t)) for normalfordeling', 'Location', 'SE')
hold off

% Kode for å sjekke om cdf for feilhendelse 12 er IFR eller DFR
figure(5), [g,r]=ecdf(MTTF(12,:), 'function', 'survivor');
hold on
plot(r,-log(g), 'b', 'LineWidth', 3)

```

```

plot(f2,-log(1-g2),'-.m','LineWidth',3)
plot(f2,-log(1-i2),'--r','LineWidth',3)
plot(f2,-log(1-j2),':k','LineWidth',3)
plot(f2,-log(1-k2),'g','LineWidth',3)
legend('-ln(1-F(t)) for simulerte fordeling','-ln(1-F(t)) for
eksponetialfordeling','-ln(1-F(t)) for gammafordeling','-ln(1-F(t)) for
weibullfordeling','-ln(1-F(t)) for normalfordeling','Location','SE')
hold off

% Kode for å sjekke om cdf for feilhendelse 12 er IFR eller DFR
figure(6),[h,s]=ecdf(MTTFsv,'function','survivor');
hold on
plot(s,-log(h),'b','LineWidth',3)
plot(f3,-log(1-g3),'-.m','LineWidth',3)
plot(f3,-log(1-i3),'--r','LineWidth',3)
plot(f3,-log(1-j3),':k','LineWidth',3)
plot(f3,-log(1-k3),'g','LineWidth',3)
legend('-ln(1-F(t)) for simulerte fordeling','-ln(1-F(t)) for
eksponetialfordeling','-ln(1-F(t)) for gammafordeling','-ln(1-F(t)) for
weibullfordeling','-ln(1-F(t)) for normalfordeling','Location','SE')
hold off

```

## D.2: Oppsett med redundans av alle aminpumper

```

clc

% Det er ønskelig å generere ulike tilfeldige tall for hver gang man
% kjører programmet. Denne algoritmen gjør at rand returnerer ulike
% tilfeldige verdier i intervallet [0,1] avhengig av hva systemklokka er

RandStream.setDefaultStream(RandStream('mt19937ar','seed',sum(1000*clock)));

%% CASE 2: OPPSETT MED REDUNDANS AV ALLE AMINPUMPER

% Monte Carlo simulering av tre feiltrær der man har redundans av de tre
% hovedpumpene i aminsysteet

% Det antas at feilraten til de ulike feilhendelsene er
% eksponetialfordelt mens reparasjonstiden er satt konstant. Det antas
% videre at en testperiode strekker seg over 2 år. Likevel må man
% fortsatt simulere prosessen over en mye lengre tid enn 2 år for å kunne
% oppnå realistiske verdier for MTTF for de ulike feilhendelsene. Det er
% derfor utarbeidet en sensureringsalgoritme som regner ut hvor stor
% andel av feilhendelsene som har en levetid som er lengre enn
% simuleringstiden

% Simuleringen beregner MTTF for feilhendelser, feiltrær og systemet som
% helhet. De beregnede verdiene for gjennomsnittlig tid til feil for
% systemet MTTFs benyttes til å vise en empirisk cdf for levetiden til
% systemet. Man kan da benytte seg av middelverdien til de p utregnede
% MTTFs som en inputverdi i beregningen av en cdf som er eksponetial-,
% weibull-, normal- og gammafordelt. På denne måten kan man se hvilken
% fordeling som passer best til de empiriske verdiene

% Her anslås systemtilgjengeligheten som en funksjon av totaltiden før
% revisjonsstans Td, og reparasjonstiden Tr som inkluderer forebyggende
% og korrektivt vedlikehold. Korrektivt vedlikeholdstid er en

```

```

% funksjon av MTBF og MTTR for de ulike feiltrærne (kapitel 5.3)

% - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * -
* - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * - * -

%% TILORDNING AV INPUTVERDIER OG DEFINERING AV SIMULERINGSPARAMETER

x = [0.1005648, 0.010512, 0.027156, 0.027156, 0.028908, 0.0454644,
     0.0184836, 0.0184836, 0.1146684, 0.1146684, 0.1146684, 0.1146684,
     0.1146684, 0.1146684, 0.0090228, 0.0090228, 0.0677148, 0.0677148,
     0.028908, 0.028908, 0.0454644, 0.0454644, 0.028908, 0.028908,
     0.0454644, 0.0454644, 0.028908, 0.028908, 0.0454644, 0.0454644,
     0.0706056, 0.0281196];

y = [2.8, 15.9, 9.2, 9.2, 45, 5, 81.2, 81.2, 70.4, 70.4, 70.4, 70.4, 70.4,
     70.4, 75, 75, 41, 41, 45, 45, 5, 5, 45, 45, 5, 5, 45, 45, 5, 5, 20.2,
     51];

% x er en vektor med feilraten til hver feilhendelse i [feil/år]
% y er en vektor med reparasjonstiden til hver feil xi som inntreffer i
[timer]

dt = 1/12;          % dt er tidsskritt tilsvarende 1 mnd der man regner tiden
                   % i [år]
t = dt:dt:600;     % Antall tidsskritt fra det første tidsskrittet til
                   % totaltiden av simuleringen [år]

n = length(t);    % Lengden av t-vektoren
m = length(x);    % Lengden av x-vektoren

p = 1000;         % Antall simuleringer for hvert system

Tt = n*dt*24*365; % Totaltiden av simuleringen [timer]
TTF = zeros(n,m); % Matrise som tar seg av tid til feil for hver
                 % feilhendelse delt inn i de ulike tidsstegene
MTTF = zeros(m,p); % Matrise som tar seg av gjennomsnittlig tid til feil
                 % for hver feilhendelse
MTTFsm = zeros(3,p); % Matrise som tar seg av gjennomsnittlig tid til feil
                 % for systemet i alle de tre feiltreene
MTTFsv = zeros(1,p); % Vektor som tar seg av gjennomsnittlig tid til feil
                 % for totalsystemet
MTBFsv = zeros(1,p); % Vektor som tar seg av gjennomsnittlig tid mellom
                 % feil for totalsystemet
omega1 = zeros(8,p); % Matrise som tar seg av verdiene til ROCOF for det
                 % første feiltreet med 8 "cut sets"
omega2 = zeros(7,p); % Matrise som tar seg av verdiene til ROCOF for det
                 % andre feiltreet med 7 "cut sets"
omega3 = zeros(8,p); % Matrise som tar seg av verdiene til ROCOF for det
                 % tredje feiltreet med 8 "cut sets"
MTBFsm = zeros(3,p); % Matrise som tar seg av gjennomsnittlig tid mellom
                 % feil for systemet i alle de tre feiltreene
MTTRsm = zeros(3,p); % Matrise som tar seg av gjennomsnittlig tid til
                 % reparasjon for systemet i alle de tre feiltreene
T0 = zeros(1,p);    % Vektor som tar seg av den totale tiden brukt på
                 % korrigert vedlikehold for hver av de angitte
                 % feilhendelsene [hrs]
Tr = zeros(1,p);    % Vektor som tar seg av total reparasjonstid inkludert
                 % forebyggende vedlikehold ifm revisjonsstans

```

```

Ts = 2*dt*24*365;      % Total reparasjonstid ifm forebyggende vedlikehold
                        % ved revisjonsstans (lik 2 mnd) [timer]
Td = 2*24*365;         % Total tid mellom revisjonsstanser for anlegget (lik
                        % 2 år) [timer]
Asv = zeros(1,p);     % Vektor som tar seg av totalsystemtilgjengeligheten

q = x.*dt;            % Sannsynligheten for feil i et tidsskritt for hver
                        % feilhendelse

%% FORLØKKER SOM SIMULERER PROSESSEN FOR n ANTALL TIDSSKRITT, m ANTALL
%% FEILHENDELSER OG p ANTALL SIMULERINGER

for c3 = 1:p          % TELLER FOR ANTALL SIMULERINGER

    TTF = 0;          % Nullstiller tid til feil vektoren ved utregningen
                    % av en ny komponent
    r = rand(n,m);   % Genererer en vektor med tilfeldige variable for den
                    % aktuelle feilhendelsen ved hvert tidsskritt

    for c2 = 1:length(x) % TELLER FOR FEILHENDELSE

        dt_ind = 0;   % Nullstiller variabel som indikerer hvilket
                    % tidsskritt siste feil inntraff
        dt_teller = 0; % Nullstiller teller som angir hvor mange feil
                    % som inntreffer for en feilhendelse ila
                    % totaltiden

        for c1 = 1:length(t) % TELLER FOR TIDSSKRITT

            if (r(c1,c2)<= q(c2)) % Sjekker om sannsynligheten for feil
                % i en feilhendelse er større enn de
                % tilfeldig genererte variablene

                TTF(c1,c2) = (c1-dt_ind)*dt*24*365; % Tilordner tiden det
                % tar til feil
                % inntreffer for den
                % gitte feilhendelse
                % ved det gitte
                % tidsskritt i [timer]

                dt_teller = dt_teller+1; % Antall ganger feil inntreffer
                % for angitt feilhendelse ila
                % totaltiden adderes med 1.

                dt_ind = c1; % Teller som passer på i hvilket
                % tidsskritt siste feil for angitt
                % feilhendelse inntraff

            end

        end

    end

    if dt_teller == 0 % If-setning som angir hva som skjer hvis det
                    % ikke inntreffer feil på angitt feilhendelse
                    % ila totaltiden

```



```

MTTF(c2,c3) = Tt; % Gjennomsnittlig tid til feil er satt lik
                % totaltiden pga at feil ikke har inntruffet
                % for angitt feilhendelse.

else            % Hvis det har oppstått feil ıla totaltiden

MTTF(c2,c3) = sum(TTF(:,c2))/dt_teller; % Funksjon som regner ut
                % middelverdien av tid til
                % feil ved hjelp av likning
                % (4.36)

end

end

%% UTREGNING FOR DE ULIKE FEILTRÆRNE

% FEILTRE 1 har følgende 8 "cut sets": {1}, {2}, {3}, {4}, {5}, {6},
% {7}, {8}. Det betyr at alle feilhendelser er koblet i serie

% Funksjon som regner ut gjennomsnittlige tid til feil for systemet til
% det første feiltreet vha likningene (4.15) og (4.41)
MTTFsm(1,c3)=1/((1/MTTF(1,c3))+(1/MTTF(2,c3))+(1/MTTF(3,c3))+(1/MTTF(4,c3))
+(1/MTTF(5,c3))+(1/MTTF(6,c3))+(1/MTTF(7,c3))+(1/MTTF(8,c3)));

% Regner ut omega-verdiene for alle 8 "cut sets" i det første feiltreet
% vha likningene (4.35) og (4.44)
omega1(1,c3) = 1/(MTTF(1,c3)+y(1)); omega1(2,c3) = 1/(MTTF(2,c3)+y(2));
omega1(3,c3) = 1/(MTTF(3,c3)+y(3)); omega1(4,c3) = 1/(MTTF(4,c3)+y(4));
omega1(5,c3) = 1/(MTTF(5,c3)+y(5)); omega1(6,c3) = 1/(MTTF(6,c3)+y(6));
omega1(7,c3) = 1/(MTTF(7,c3)+y(7)); omega1(8,c3) = 1/(MTTF(8,c3)+y(8));

% Funksjon som regner ut gjennomsnittlige tid mellom feil for systemet
% til det første feiltreet vha likningene (4.45) og (4.46)
MTBFsm(1,c3) = 1 / sum(omega1(:,c3));

% Funksjon som regner ut gjennomsnittlige tid til reparasjon for systemet
% til det første feiltreet vha av likning (4.47)
MTTRsm(1,c3)=((omega1(1,c3)*y(1))+(omega1(2,c3)*y(2))+(omega1(3,c3)*y(3))+
(omega1(4,c3)*y(4))+(omega1(5,c3)*y(5))+(omega1(6,c3)*y(6))+
(omega1(7,c3)*y(7))+(omega1(8,c3)*y(8)))/sum(omega1(:,c3));

% FEILTRE 2 har følgende 7 "cut sets": {9,10}, {11,12}, {13,14}, {15},
% {16}, {17}, {18}. Det er altså 3 parallellkoblinger (Det er her man har
% redundans av pumpene)

% Funksjon som regner ut gjennomsnittlige tid til feil for systemet til det
% andre feiltreet vha linkningene (4.15), (4.16) og (4.41)
MTTFsm(2,c3)=1/((1/(MTTF(9,c3)+MTTF(10,c3)))+(1/(MTTF(11,c3)+MTTF(12,c3)))+
(1/(MTTF(13,c3)+MTTF(14,c3)))+(1/MTTF(15,c3))+(1/MTTF(16,c3))+
(1/MTTF(17,c3))+(1/MTTF(18,c3)));

% Regner ut omega-verdiene for alle 7 "cut sets" i det andre feiltreet vha
% likningene (4.35), (4.43) og (4.44)
omega2(1,c3) = 1/((MTTF(9,c3)+MTTF(10,c3))+(y(9)*y(10)/(y(9)+y(10))));
omega2(2,c3) = 1/((MTTF(11,c3)+MTTF(12,c3))+(y(11)*y(12)/(y(11)+y(12))));

```

```

omega2(3,c3) = 1 / ((MTTF(13,c3)+MTTF(14,c3))+(y(13)*y(14)/(y(13)+y(14))));
omega2(4,c3) = 1 / (MTTF(15,c3)+y(15));
omega2(5,c3) = 1 / (MTTF(16,c3)+y(16));
omega2(6,c3) = 1 / (MTTF(17,c3)+y(17));
omega2(7,c3) = 1 / (MTTF(18,c3)+y(18));

% Funksjon som regner ut gjennomsnittlige tid mellom feil for systemet til
% det andre feiltreet vha likningene (4.45) og (4.46)
MTBFsm(2,c3) = 1 / sum(omega2(:,c3));

% Funksjon som regner ut gjennomsnittlige tid til reparasjon for systemet
% til det andre feiltreet vha av likning (4.47)
MTTRsm(2,c3)=(omega2(1,c3)*(1/((1/y(9))+1/y(10))))+
(omega2(2,c3)*(1/((1/y(11))+1/y(12))))+(omega2(3,c3)*(1/((1/y(13))+
(1/y(14)))))+(omega2(4,c3)*y(15))+omega2(5,c3)*y(16))+
(omega2(6,c3)*y(17))+omega2(7,c3)*y(18))/sum(omega2(:,c3));

% FEILTRE 3 har følgende 8 "cut sets": {19,20}, {21,22}, {23,24}, {25,26},
% {27,28}, {29,30}, {31}, {32}. Det er altså 6 parallellkoblinger (Det er
% her man har redundans av pumpene)

% Funksjon som regner ut gjennomsnittlige tid til feil for systemet til det
% tredje feiltreet vha likningene (4.15), (4.16) og (4.41)
MTTFsm(3,c3)=1/((1/(MTTF(19,c3)+MTTF(20,c3)))+(1/(MTTF(21,c3)+MTTF(22,c3))
+(1/(MTTF(23,c3)+MTTF(24,c3)))+(1/(MTTF(25,c3)+MTTF(26,c3)))+
(1/(MTTF(27,c3)+MTTF(28,c3)))+(1/(MTTF(29,c3)+MTTF(30,c3)))+(1/MTTF(31,c3))
+(1/MTTF(32,c3)));

% Regner ut omega-verdiene for alle 8 "cut sets" i det tredje feiltreet vha
% likningene (4.35), (4.43) og (4.44)
omega3(1,c3) = 1 / ((MTTF(19,c3)+MTTF(20,c3))+(y(19)*y(20)/(y(19)+y(20))));
omega3(2,c3) = 1 / ((MTTF(21,c3)+MTTF(22,c3))+(y(21)*y(22)/(y(21)+y(22))));
omega3(3,c3) = 1 / ((MTTF(23,c3)+MTTF(24,c3))+(y(23)*y(24)/(y(23)+y(24))));
omega3(4,c3) = 1 / ((MTTF(25,c3)+MTTF(26,c3))+(y(25)*y(26)/(y(25)+y(26))));
omega3(5,c3) = 1 / ((MTTF(27,c3)+MTTF(28,c3))+(y(27)*y(28)/(y(27)+y(28))));
omega3(6,c3) = 1 / ((MTTF(29,c3)+MTTF(30,c3))+(y(29)*y(30)/(y(29)+y(30))));
omega3(7,c3) = 1 / (MTTF(31,c3)+y(31));
omega3(8,c3) = 1 / (MTTF(32,c3)+y(32));

% Funksjon som regner ut gjennomsnittlige tid mellom feil for systemet til
% det tredje feiltreet vha likningene (4.45) og (4.46)
MTBFsm(3,c3) = 1 / sum(omega3(:,c3));

% Funksjon som regner ut gjennomsnittlige tid til reparasjon for systemet
% til det tredje feiltreet vha av likning (4.47)
MTTRsm(3,c3)=(omega3(1,c3)*(1/((1/y(19))+1/y(20))))+
(omega3(2,c3)*(1/((1/y(21))+1/y(22))))+(omega3(3,c3)*(1/((1/y(23))+
(1/y(24)))))+(omega3(4,c3)*(1/((1/y(25))+1/y(26)))))+(
omega3(5,c3)*(1/((1/y(27))+1/y(28)))))+(omega3(6,c3)*(1/((1/y(29))+
(1/y(30)))))+(omega3(7,c3)*y(31))+omega3(8,c3)*y(32))/sum(omega3(:,c3));

%% UTREGNING FOR TOTALSYSTEMET

% Funksjon som regner ut gjennomsnittlig tid til feil og
% gjennomsnittlig tid mellom feil for totalsystemet vha linkningene (4.15),
% (4.41) og (4.44)
MTTFsv(c3) = 1 / ((1/MTTFsm(1,c3))+(1/MTTFsm(2,c3))+(1/MTTFsm(3,c3)));
MTBFsv(c3) = 1 / ((1/MTBFsm(1,c3))+(1/MTBFsm(2,c3))+(1/MTBFsm(3,c3)));

```

```

% Funksjon som regner ut tiden det tar å utføre korrigerende
% vedlikehold på systemet vha likning (5.1)
T0(c3)=( (MTTRsm(1,c3)/MTBFsm(1,c3)+(MTTRsm(2,c3)/MTBFsm(2,c3))+
(MTTRsm(3,c3)/MTBFsm(3,c3))) *Td);

% Funksjon som regner ut den totale tiden brukt på systemreparasjon
% inkludert forebyggende vedlikehold
Tr(c3) = T0(c3) + Ts;

% Funksjon som regner ut systemtilgjengeligheten vha likning (5.2)
Asv(c3) = (Td / (Tr(c3) + Td));

end

%% SENSURERINGSALGORITMER

% Her lages det en test for å se hvor stor andel av de simulerte verdiene
% for MTTF for alle feilhendelsene som er større enn totaltiden til
% simuleringen
obstime = sort(min(Tt, MTTF));
fail = obstime(obstime<Tt);
nfail =length(fail);
surv= obstime(obstime>=Tt);
nsurv = length(surv);
censored_rate = nsurv/(m*p)

% Her lages det en test for å se hvor stor andel av de simulerte verdiene
% for MTTF til feilhendelsene som inngår i det 1. feiltreet som er større
% enn totaltiden til simuleringen
obstimef1 = sort(min(Tt,MTTF(1:8,:)));
failf1 = obstimef1(obstimef1<Tt);
nfailf1 = length(failf1);
survf1= obstimef1(obstimef1>=Tt);
nsurvf1 = length(survf1);
censored_ratef1 = nsurvf1/(8*p)

% Her lages det en test for å se hvor stor andel av de simulerte verdiene
% for MTTF til feilhendelsene som inngår i det 2. feiltreet som er større
% enn totaltiden til simuleringen
obstimef2 = sort(min(Tt,MTTF(9:18,:)));
failf2 = obstimef2(obstimef2<Tt);
nfailf2 = length(failf2);
survf2= obstimef2(obstimef2>=Tt);
nsurvf2 = length(survf2);
censored_ratef2 = nsurvf2/(10*p)

% Her lages det en test for å se hvor stor andel av de simulerte verdiene
% for MTTF til feilhendelsene som inngår i det 3. feiltreet som er større
% enn totaltiden til simuleringen
obstimef3 = sort(min(Tt,MTTF(19:32,:)));
failf3 = obstimef3(obstimef3<Tt);
nfailf3 = length(failf3);
survf3= obstimef3(obstimef3>=Tt);
nsurvf3 = length(survf3);
censored_ratef3 = nsurvf3/(14*p)

```

```

% Her lages det en test for å se hvor stor andel av de simulerte verdiene
% for MTTF for feilhendelse #9 som er større enn totaltiden til
% simuleringen
obstime9 = sort(min(Tt, MTTF(9,:)));
fail9 = obstime9(obstime9<Tt);
nfail9 = length(fail9);
surv9= obstime9(obstime9>=Tt);
nsurv9 = length(surv9);
censored_rate9 = nsurv9/p

% Her lages det en test for å se hvor stor andel av de simulerte verdiene
% for MTTF for feilhendelse #12 som er større enn totaltiden til
% simuleringen (OBS! Feilhendelse #12 referer til tabell 3.2 og ikke til
% nummereringen i feiltreet)
obstime12 = sort(min(Tt, MTTF(15,:)));
fail12 = obstime12(obstime12<Tt);
nfail12 = length(fail12);
surv12= obstime12(obstime12>=Tt);
nsurv12 = length(surv12);
censored_rate12 = nsurv12/p

%% SIMULERINGSRESULTATER

% Diverse resultater som er nyttige å se på
MTTF_feiltre1 = mean(MTTFsm(1,:))
MTTF_feiltre2 = mean(MTTFsm(2,:))
MTTF_feiltre3 = mean(MTTFsm(3,:))
MTTF_systemet = mean(MTTFsv)
MTTF_komponent9 = mean(MTTF(9,:))
MTTF_komponent12 = mean(MTTF(15,:))
MTTR_feiltre1 = mean(MTTRsm(1,:))
MTTR_feiltre2 = mean(MTTRsm(2,:))
MTTR_feiltre3 = mean(MTTRsm(3,:))
MTBF_feiltre1 = mean(MTBFsm(1,:))
MTBF_feiltre2 = mean(MTBFsm(2,:))
MTBF_feiltre3 = mean(MTBFsm(3,:))
MTBF_systemet = mean(MTBFsv)
T0_sys = mean(T0)
A_systemet = mean(Asv)

% Resultater som gir ut de karakteristiske parametrene for weibull-,
% gamma- og normalfordelingen og som passer best mulig til den empiriske
% cdf-funksjonen til dataene som er simulert for MTTF til feilhendelse 9,
% 12 og systemet
weibullfit9 = wblfit(MTTF(9,:))
weibullfit12 = wblfit(MTTF(15,:))
weibullfitsys = wblfit(MTTFsv)
gammafit9 = gamfit(MTTF(9,:))
gammafit12 = gamfit(MTTF(15,:))
gammafitsys = gamfit(MTTFsv)
[mu9,sigma9] = normfit(MTTF(9,:))
[mu12,sigma12] = normfit(MTTF(15,:))
[musys,sigmasys] = normfit(MTTFsv)

clf

```

```
%% PLOT AV SIMULERINGSRESULTATER
```

```
% Figur som viser CDF funksjon for tiden til feil for den feilhendelsen som  
% har lavest tid til feil, basert på data fra simulerte verdier av  
% gjennomsnittlig tid til feil for systemet. Sammenliknes med en  
% "teoretisk" weibull-, gamma-, normal- og eksponential-cdf
```

```
figure(1),[h1,stats] = cdfplot(MTTF(9,:))
```

```
hold on
```

```
f1 = 0:1:stats.max;
```

```
g1 = expcdf(f1,stats.mean);
```

```
i1 = gamcdf(f1,gammafit9(1),gammafit9(2));
```

```
j1 = wblcdf(f1,weibullfit9(1),weibullfit9(2));
```

```
k1 = normcdf(f1,mu9,sigma9);
```

```
plot(f1,g1,'-.m','LineWidth',3)
```

```
plot(f1,i1,'--r','LineWidth',3)
```

```
plot(f1,j1,':k','LineWidth',3)
```

```
plot(f1,k1,'g','LineWidth',3)
```

```
legend('Empirisk CDF basert på feilhendelse med lavest teoretisk  
MTTF','Eksponential-CDF basert på middelverdien av MTTF','Gamma-CDF basert  
på MLE','Weibull-CDF basert på MLE','Normal-CDF basert på  
MLE','Location','SE')
```

```
hold off
```

```
% Figursom viser CDF funksjon for tiden til feil for den feilhendelsen som  
% har lavest tid til feil, basert på data fra simulerte verdier av  
% gjennomsnittlig tid til feil for systemet. Sammenliknes med en  
% "teoretisk" weibull-, gamma-, normal- og eksponential-cdf
```

```
figure(2),[h2,stats] = cdfplot(MTTF(15,:))
```

```
hold on
```

```
f2 = 0:1:stats.max;
```

```
g2 = expcdf(f2,stats.mean);
```

```
i2 = gamcdf(f2,gammafit12(1),gammafit12(2));
```

```
j2 = wblcdf(f2,weibullfit12(1),weibullfit12(2));
```

```
k2 = normcdf(f2,mu12,sigma12);
```

```
plot(f2,g2,'-.m','LineWidth',3)
```

```
plot(f2,i2,'--r','LineWidth',3)
```

```
plot(f2,j2,':k','LineWidth',3)
```

```
plot(f2,k2,'g','LineWidth',3)
```

```
legend('Empirisk CDF basert på feilhendelse med høyest teoretisk  
MTTF','Eksponential-CDF basert på middelverdien av MTTF','Gamma-CDF basert  
på MLE','Weibull-CDF basert på MLE','Normal-CDF basert på  
MLE','Location','SE')
```

```
hold off
```

```
% Figur som viser CDF funksjon for tiden til feil for systemt basert på  
% data fra simulerte verdier av gjennomsnittlig tid til feil for systemet.  
% Sammenliknes med en teoretisk weibull-, gamma-, normal- og eksponential-  
% cdf
```

```
figure(3),[h3,stats] = cdfplot(MTTFsv)
```

```
hold on
```

```
f3 = 0:1:stats.max;
```

```
g3 = expcdf(f3,stats.mean);
```

```
i3 = gamcdf(f3,gammafitsys(1),gammafitsys(2));
```

```
j3 = wblcdf(f3,weibullfitsys(1),weibullfitsys(2));
```

```
k3 = normcdf(f3,musys,sigmasys);
```

```
plot(f3,g3,'-.m','LineWidth',3)
```

```
plot(f3,i3,'--r','LineWidth',3)
```

```
plot(f3,j3,':k','LineWidth',3)
```

```
plot(f3,k3,'g','LineWidth',3)
```

```

legend('Empirisk CDF basert på MTTFs','Ekspontial-CDF basert på
middelverdien av MTTF','Gamma-CDF basert på MLE','Weibull-CDF basert på
MLE','Normal-CDF basert på MLE','Location','SE')
hold off

% Lager et plot for å sjekke om funksjonen er IFR eller DFR (kapittel 4.1)
% Hvis funksjonene er konvekse er de IFR og hvis de er konkave er de DFR

% Kode for å sjekke om cdf for feilhendelse 9 er IFR eller DFR
figure(4),[f,z]=ecdf(MTTF(9,:), 'function', 'survivor');
hold on
plot(z,-log(f), 'b', 'LineWidth',3)
plot(f1,-log(1-g1), '-.m', 'LineWidth',3)
plot(f1,-log(1-i1), '--r', 'LineWidth',3)
plot(f1,-log(1-j1), ':k', 'LineWidth',3)
plot(f1,-log(1-k1), 'g', 'LineWidth',3)
legend('-ln(1-F(t)) for simulerte fordeling', '-ln(1-F(t)) for
eksponetialfordeling', '-ln(1-F(t)) for gammafordeling', '-ln(1-F(t)) for
weibullfordeling', '-ln(1-F(t)) for normalfordeling', 'Location', 'SE')
hold off

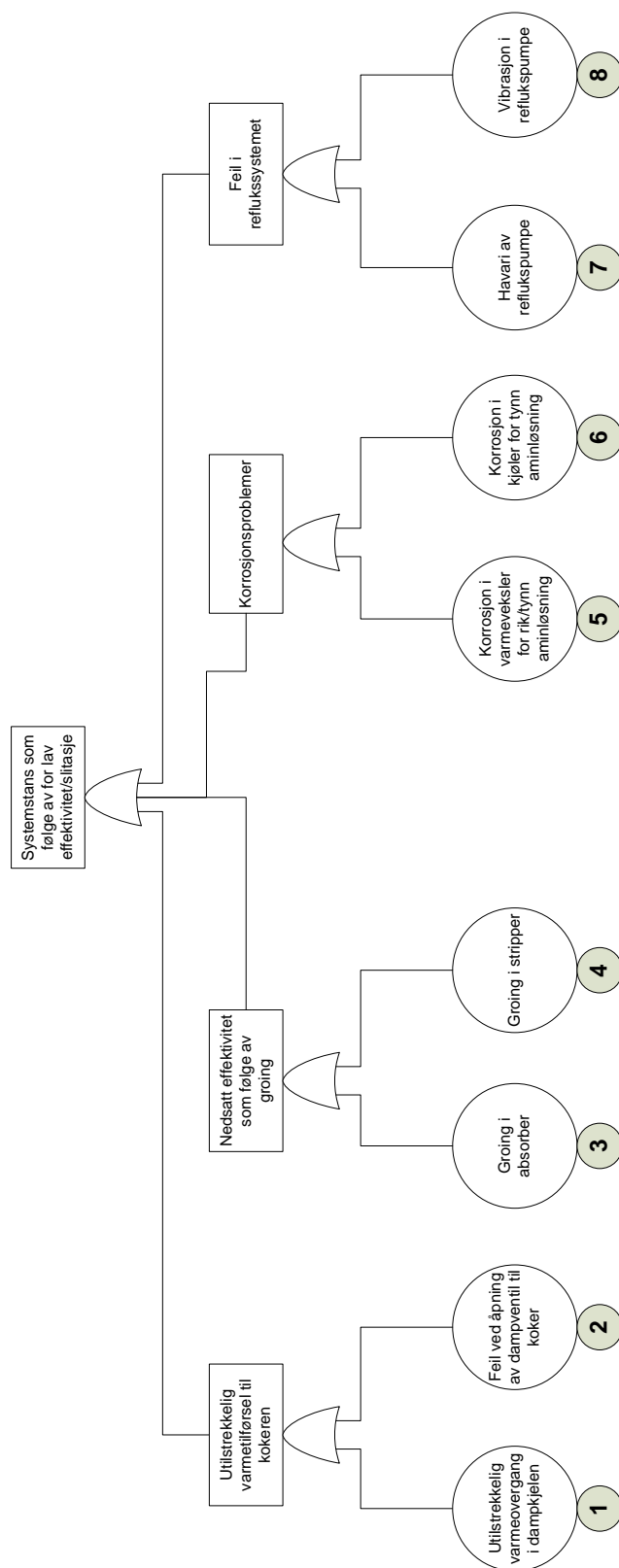
% Kode for å sjekke om cdf for feilhendelse 12 er IFR eller DFR
figure(5),[g,r]=ecdf(MTTF(15,:), 'function', 'survivor');
hold on
plot(r,-log(g), 'b', 'LineWidth',3)
plot(f2,-log(1-g2), '-.m', 'LineWidth',3)
plot(f2,-log(1-i2), '--r', 'LineWidth',3)
plot(f2,-log(1-j2), ':k', 'LineWidth',3)
plot(f2,-log(1-k2), 'g', 'LineWidth',3)
legend('-ln(1-F(t)) for simulerte fordeling', '-ln(1-F(t)) for
eksponetialfordeling', '-ln(1-F(t)) for gammafordeling', '-ln(1-F(t)) for
weibullfordeling', '-ln(1-F(t)) for normalfordeling', 'Location', 'SE')
hold off

% Kode for å sjekke om cdf for systemet er IFR eller DFR
figure(6),[h,s]=ecdf(MTTFsv, 'function', 'survivor');
hold on
plot(s,-log(h), 'b', 'LineWidth',3)
plot(f3,-log(1-g3), '-.m', 'LineWidth',3)
plot(f3,-log(1-i3), '--r', 'LineWidth',3)
plot(f3,-log(1-j3), ':k', 'LineWidth',3)
plot(f3,-log(1-k3), 'g', 'LineWidth',3)
legend('-ln(1-F(t)) for simulerte fordeling', '-ln(1-F(t)) for
eksponetialfordeling', '-ln(1-F(t)) for gammafordeling', '-ln(1-F(t)) for
weibullfordeling', '-ln(1-F(t)) for normalfordeling', 'Location', 'SE')
hold off

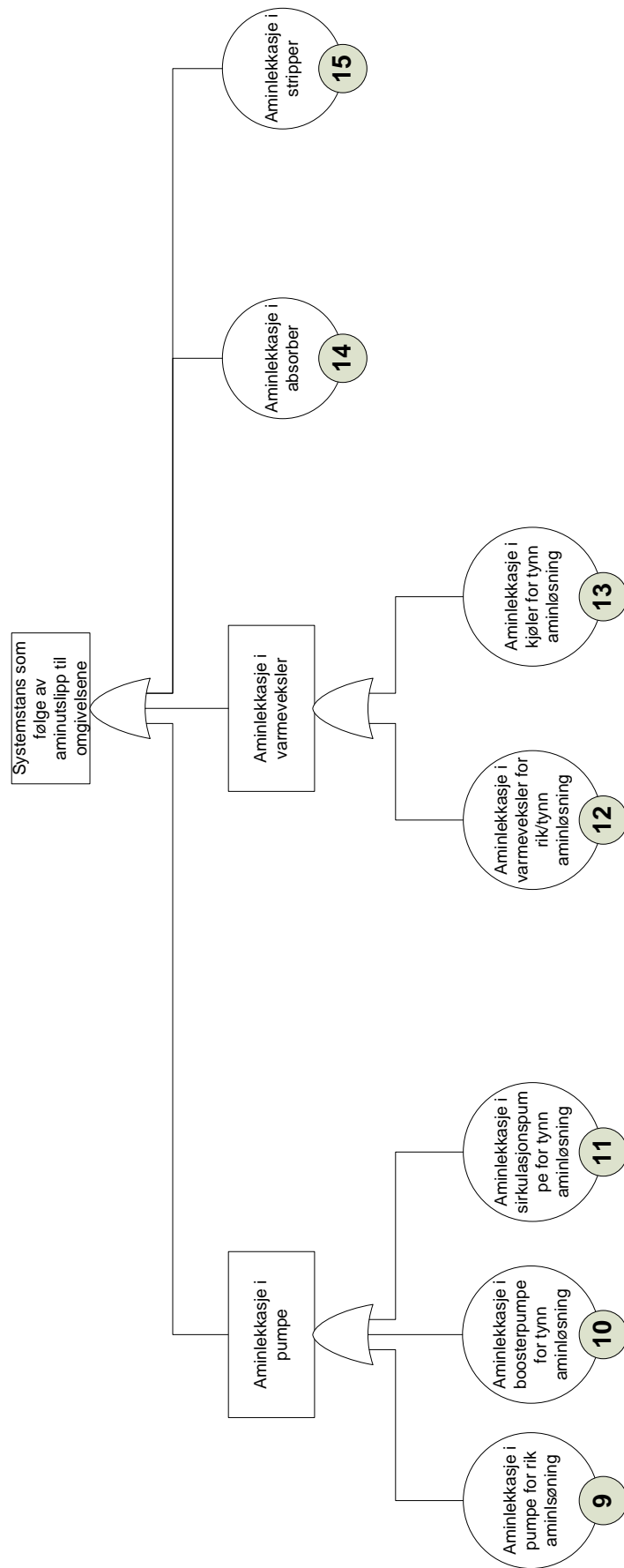
```

# E: Feiltreoppbygning

## E.1: Hovedoppsett

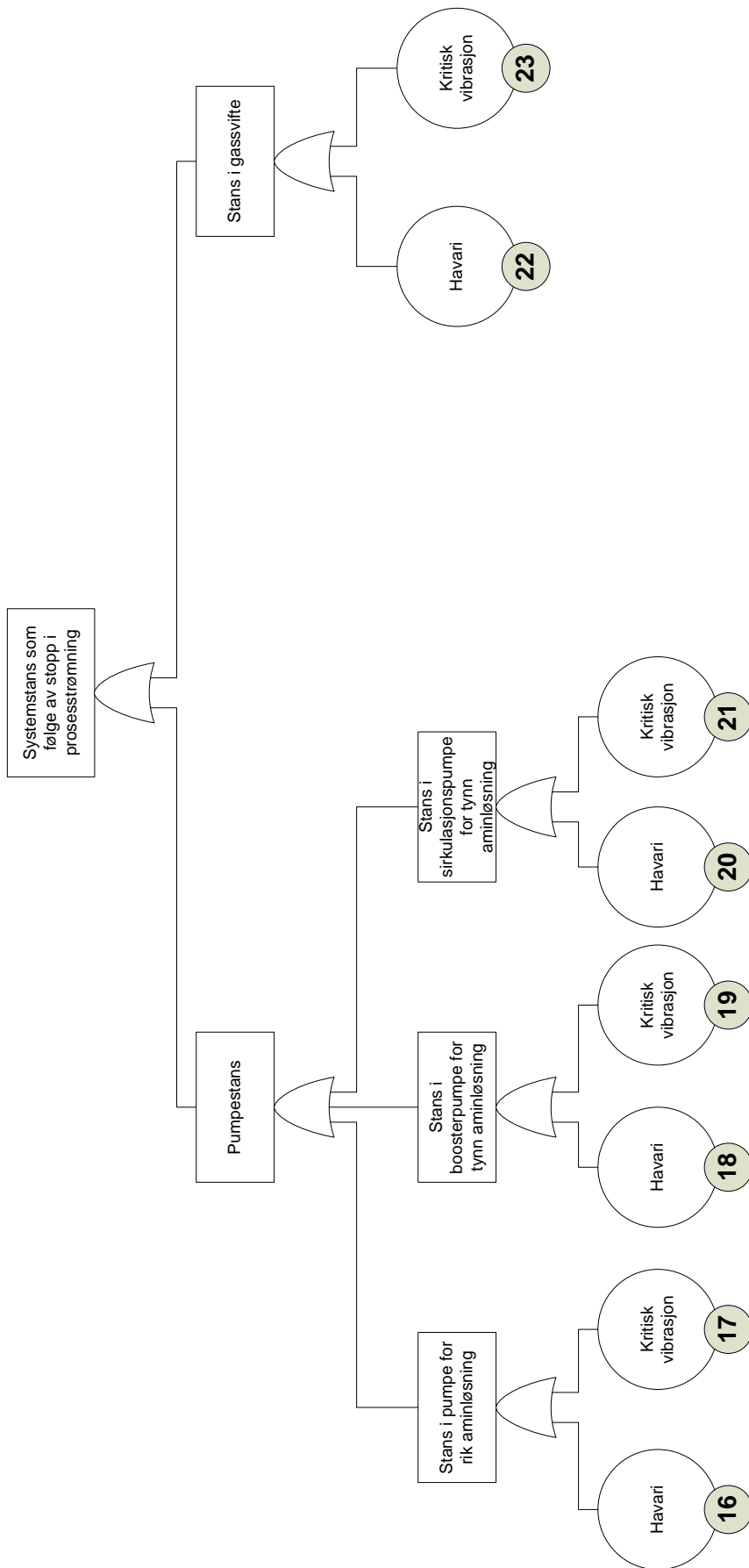


Figur E.1: Feiltre 1 i hovedoppsett



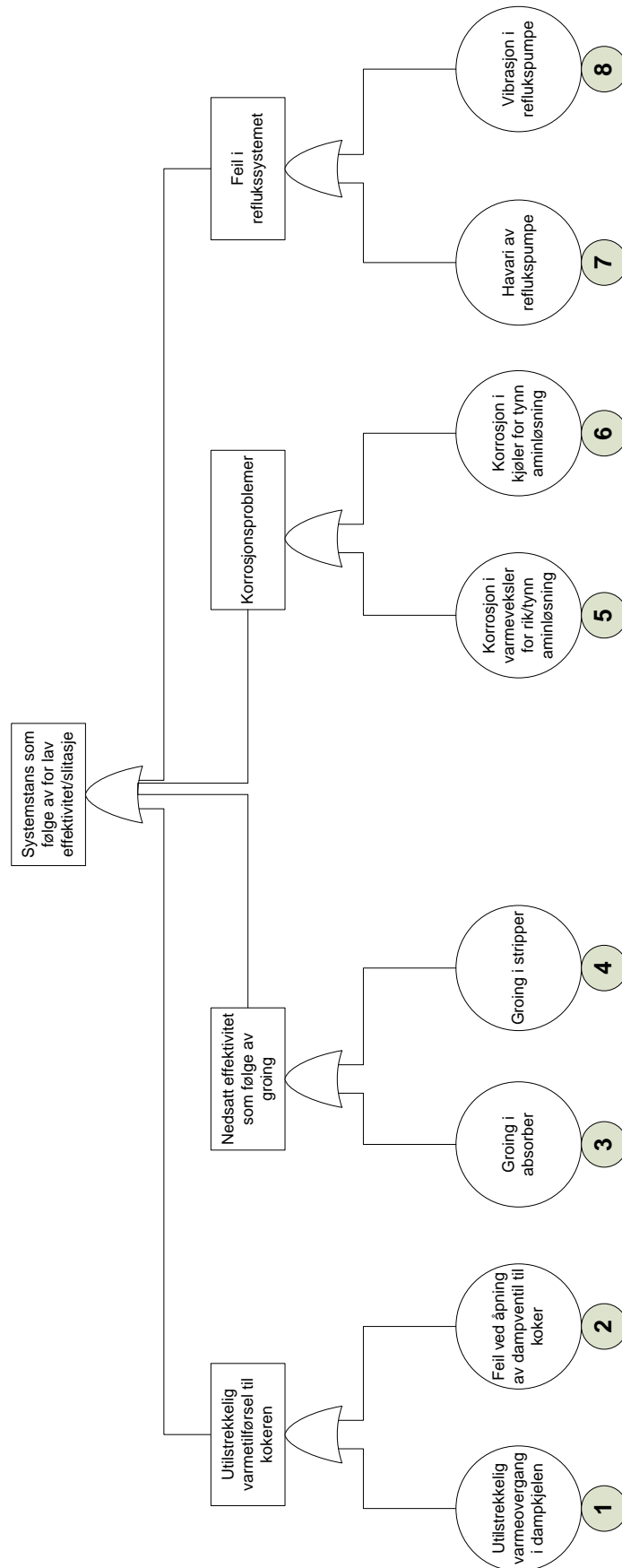
Figur E.2: Feiltre 2 i hovedoppsett



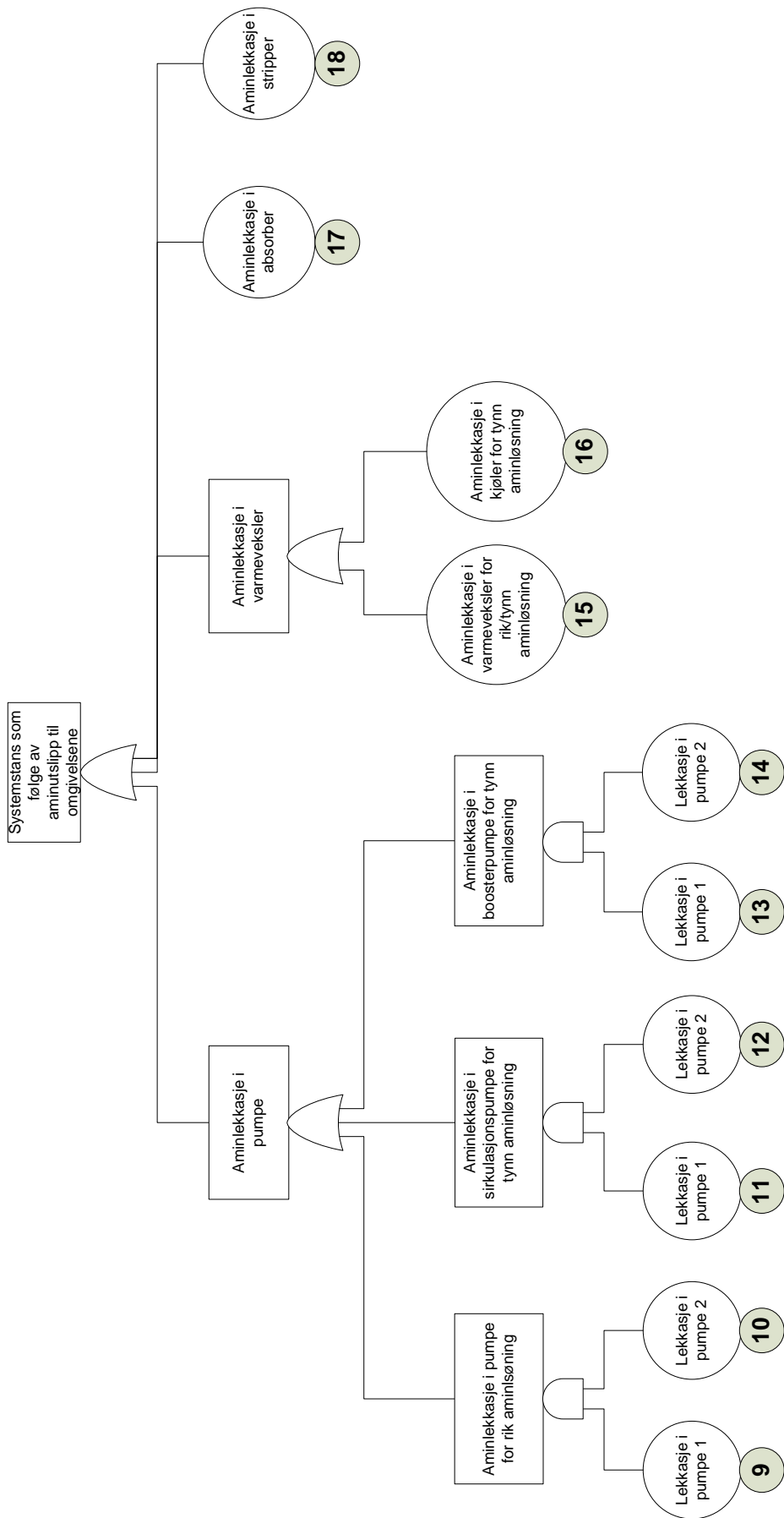


Figur E.3: Feiltre 3 i hovedoppsett

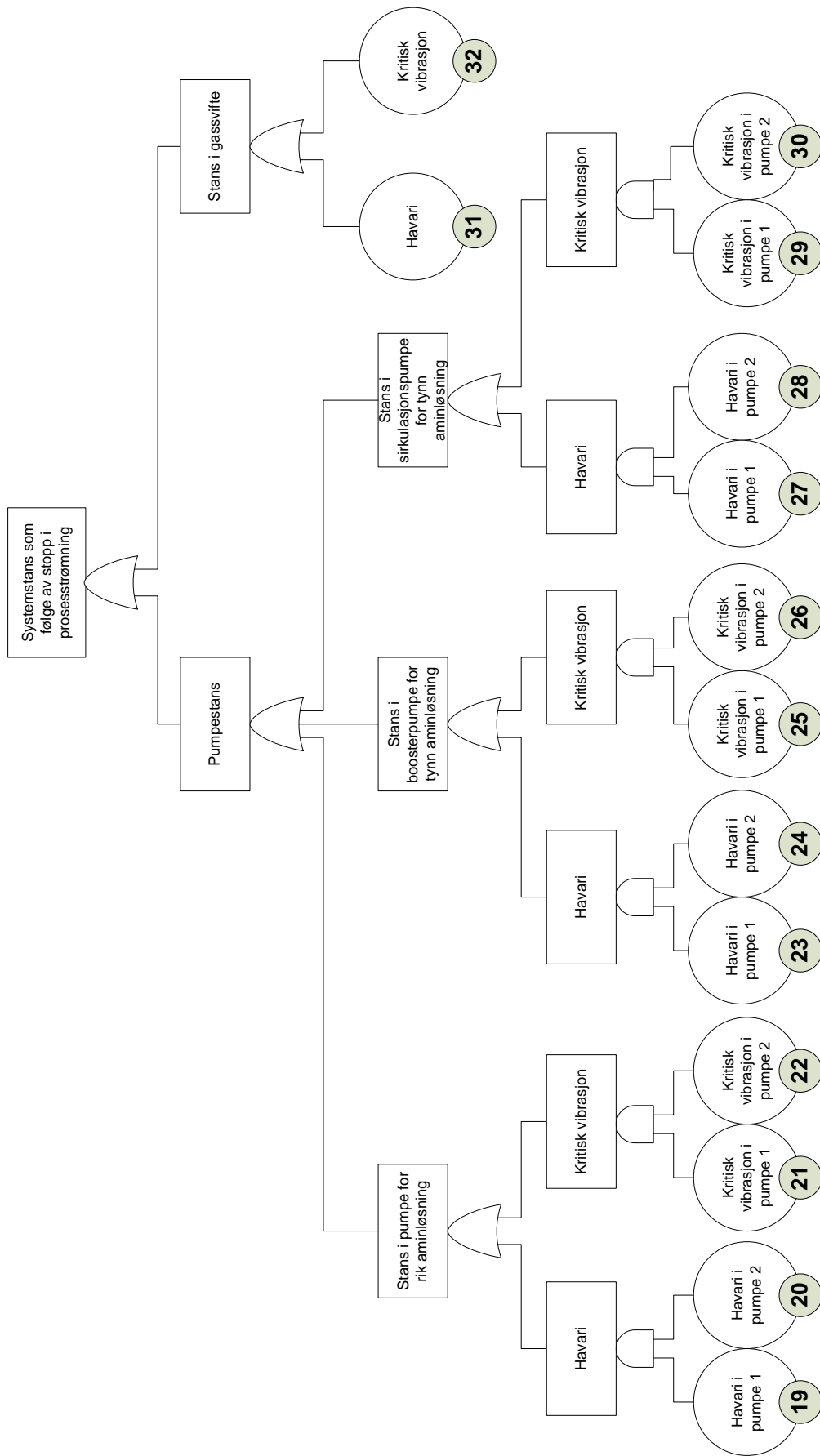
## E.2: Oppsett med redundans av alle aminpumper



Figur E.4: Feiltre 1 i oppsett med redundans av aminpumper



Figur E.5: Feiltre 2 i oppsett med redundans av aminpumper



Figur E.6: Feiltre 3 i oppsett med redundans av aminpumper

## F: Avvik mellom simuleringene

### Avvik mellom simuleringene

$p = 1000, Tt = 2628000$

Gjennom- kjøring #	MTTF feilhendelse 9		MTTF feilhendelse 12		MTTF systemet	
	MTTF [hrs]	Avvik fra #1 [%]	MTTF [hrs]	Avvik fra #1 [%]	MTTF [hrs]	Avvik fra #1 [%]
1	76430	0	939080	0	7422	0
2	76473	0,06	885570	-6,04	7397	-0,34
3	76147	-0,37	934540	-0,49	7380	-0,57
4	76319	-0,15	894920	-4,93	7347	-1,02
5	75851	-0,76	900340	-4,30	7333	-1,21
6	76769	0,44	895500	-4,87	7361	-0,83
7	76405	-0,03	900490	-4,29	7392	-0,41
8	75509	-1,22	906210	-3,63	7349	-0,99
9	76833	0,52	871390	-7,77	7345	-1,05
10	76231	-0,26	860260	-9,16	7391	-0,42

Tabell F1: Avvik mellom de ulike simuleringsrundene ved  $p = 1000$  simuleringer.

### Avvik mellom simuleringene

$p = 10000, Tt = 2628000$

Gjennom- kjøring #	MTTF feilhendelse 9		MTTF feilhendelse 12		MTTF systemet	
	MTTF [hrs]	Avvik fra #1 [%]	MTTF [hrs]	Avvik fra #1 [%]	MTTF [hrs]	Avvik fra #1 [%]
1	76161	0	919580	0	7377	0
2	76257	0,13	899040	-2,28	7369	-0,11
3	76452	0,38	910520	-1,00	7376	-0,01
4	76232	0,09	910600	-0,99	7371	-0,08
5	76464	0,40	903250	-1,81	7375	-0,03
6	76335	0,23	908200	-1,25	7362	-0,20
7	76161	0,00	908360	-1,24	7372	-0,07
8	76335	0,23	914310	-0,58	7370	-0,09
9	76729	0,74	903500	-1,78	7378	0,01
10	76497	0,44	895410	-2,70	7366	-0,15

Tabell F2: Avvik mellom de ulike simuleringsrundene ved  $p = 10000$  simuleringer.