



Høgskulen
på Vestlandet

BACHELOROPPGÅVE

Reinseanlegg RA200 Sande

Roar Bøyum

Vegard Aven Ullebø

Peter Søreide Skaar

Automatiseringsteknikk med robotikk
Institutt for datateknologi, elektroteknologi og
realfag

Rettleiarar: Olav Sande, Joar Sande og
Bjarte Pollen

Innleveringsdato: 21.05.2024

Eg stadfestar at arbeidet er sjølvstendig utarbeida, og at referansar/kjeldetilvisingar til alle kjelder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 12-1.



Reinseanlegget på Sande
Foto: Christopher Sveen

1 Forord

Denne bacheloroppgåva er utarbeida som ein del av vår utdanning i Automatiseringsteknikk med robotikk ved Høgskulen på Vestlandet, campus Førde. Vi starta arbeidet tidleg januar, og oppgåva vert avslutta med ein presentasjon i juni.

Oppgåva er skreven på vegne av Renasys [1] og Sunnfjord kommune [2] og omhandlar avlaupsreinseanlegget på Sande i Sunnfjord. Ein spesiell takk til dykk som har gjort denne oppgåva mogleg og bidratt med gode faglege diskusjonar samt omvisningar på reinseanlegget på Sande og i Førde.

Vi ønskjer å rette ein stor takk til rettleiarane våre Olav Sande, Bjarte Pollen og Joar Sande ved HVL Førde, som har bidratt med gode råd gjennom prosjektet. Takk også til HVL [3] og MIDTechnologies [4] for programvarelisensar nytta i oppgåva [5] [6]. Figurar nytta i oppgåva er laga med Draw.io [7].

Familie og vener fortener ein spesiell takk for deira støtte gjennom denne perioden.

Vi håpar sluttresultatet vil bidra til betre forståelse og kan løyse nokon av utfordringane Sunnfjord kommune har hatt med reinseanlegget dei siste åra. For oss har det vore ein lærerik prosess der vi har fordjupa oss innan avlaupsreinsing samt at vi har utvikla ferdigheitane i LaTeX [8] [9], PLS programmering i Codesys [10] og teikning av blokkdiagram med SCD. Vi er sikre på at dette vil vere gode erfaringar å ta med vidare.

2 Samandrag

Bacheloroppgåva, som vi har løyst saman med Renasys og Sunnfjord kommune, handlar om utbetring av styresystemet på avlaupsreinseanlegget i Sande i Sunnfjord. Anlegget er teknisk utdatert noko som gjer at ei oppgradering av styresystemet må undersøkast.

Rapporten presenterer fire ulike løysningsalternativ der ulik grad av dybde blir vurdert.

Rapporten legg til grunn val av løysningsalternativ C som presenterer planlegging av eit nytt styresystem. Vidare undersøkar rapporten generell verkemåte til eit avlaupsreinseanlegg og forklarar og dokumenterer reinseanlegget på Sande.

Oppgåva gir innblikk i dei relevante stega i planlegging av eit nytt styresystem der eigen dokumentasjon er nytta som grunnlag for vidare arbeid. Utføring av programmering, simulering og testing er løyst med eigne funksjonsblokker og metodar i programmeringsverktøyet Codesys, der anerkjende IEC standardar er undersøkt og tatt i bruk. Vidare blir styresystemet og programmeringsarbeidet skildra og dokumentert.

Resultatet av bacheloroppgåva gir vår arbeidsgivar ny og forbetra dokumentasjon av anlegget og grunnlaget for eit nytt, fleksibelt og teknisk moderne styresystem. Rapporten undersøker også generelle oppgraderingar og ytterlegare forbetningspotensial for reinseanlegget.

Innholdsliste

1 Forord	III
2 Samandrag	IV
3 Innleiing	1
3.1 Om oss	1
3.2 Oppdragsgivar	2
3.3 Vedlegg	2
4 Analyse av oppgåva	3
4.1 Problemstilling	3
4.2 Løsningsforslag	4
4.2.1 Løsningsalternativ A: Bytte PLS	4
4.2.2 Løsningsalternativ B: Bytte PLS, konvertere styresystemet	4
4.2.3 Løsningsalternativ C: Bytte PLS, nytt styresystem	4
4.2.4 Løsningsalternativ D: Nytt anlegg, ny teknologi	5
4.3 Val og drøfting	5
5 Krav og mål	6
5.1 Krav	6
5.2 Mål	6
6 Verkemåten til anlegget	7
6.1 Generell verkemåte	7
6.2 Teknisk verkemåte	8
6.3 Praktisk verkemåte	9
6.3.1 Grovrist	9
6.3.2 Mottakstank	10
6.3.3 Reaktor	12
6.3.4 P&ID	14
6.4 Spesifikke anleggsvariasjonar	16
6.5 Dokumentasjon	20
7 Tillrettelegging for programmering	21
7.1 IEC	21
7.2 Codesys	22
7.3 SCD	23
7.4 Tilstandsmaskin	24

8	Programmering	25
8.1	IEC funksjonsblokker	25
8.1.1	Monitor Binary	26
8.1.2	Monitor Analogue	27
8.1.3	Switch Binary Electrical	28
8.1.4	Switch Binary Valve	30
8.2	Generelle funksjonsblokker	32
8.2.1	fbTimer	32
8.2.2	fbAnalogueAlarm	32
8.2.3	fbDigitalAlarm	32
8.2.4	fbSwap	32
8.2.5	fbCalculations	32
8.2.6	fbTimeMeter	32
8.2.7	fbHighLoad	33
8.2.8	fbSivbedRotation	33
8.2.9	fbDataprocessing	33
8.2.10	fbProcessedWater	33
8.3	Tilstandsmaskin	34
8.4	Tilstandslogikk	36
8.5	Oppbygging av programmet	37
8.5.1	Programmeringsmetode	37
8.5.2	Hovuddel	38
8.5.3	Styring tilstandslogikk	39
8.6	Alarm og feilhandtering	40
8.7	Utfordringar	41
8.7.1	Utgangsblokker med signalkonflikt	41
8.7.2	Mangel på sensorikk	41
9	Dokumentasjon	42
9.1	Dokumentasjon av funksjonsblokker	42
9.2	Forrigling	42
9.3	Tilstandsovergangsbetingelsar	43
9.4	IO-liste	44
9.5	Objektliste	44
9.6	SCD	46
10	Simulering og verifisering	48
10.1	Kontinuerlig simulering	48
10.2	Simuleringsblokker	49
10.3	Simuleringsvindaug	50

10.4 Fullskala simulering	51
10.4.1 Oppsett	51
10.4.2 Resultat	52
11 Diskusjon	53
11.1 Vegen vidare for anlegget	53
11.1.1 Ombygging	53
11.1.2 Anbefalingar sensorikk	54
11.2 Vegen vidare for programmet	55
11.2.1 Programmering	55
11.2.2 Simulering	55
11.3 Måloppnåing	56
11.3.1 Krav	56
11.3.2 Mål	56
11.3.3 Forprosjekt	56
11.4 Lærdom og forbettringspotensial	57
12 Konklusjon	58
Ordliste	59
Referansar	64
Liste av URL	65
Liste av vedlegg	66

Figurliste

3.1	Logo oppdragsgivar	2
4.1	Styresystem	3
6.1	Generell verkemåte for eit avlaupsreinseanlegg	7
6.2	SBR-prosessen	8
6.3	RA200 flytskjema	9
6.4	Grovrist levert av Huber	9
6.5	P&ID mottakstank	10
6.6	Matepumper	11
6.7	Diffuser system	12
6.8	Reaktor	13
6.9	Illustrasjon av reaktorsoner	14
6.10	P&ID	15
6.11	Illustrasjon sivbed	16
6.12	Satelittfoto frå Google Earth [11]	17
6.13	P&ID pumpehus	18
6.14	Sivbedventilar	19
7.1	Codesys programvare	22
7.2	Eksempel av SCD	23
7.3	Prinsipp tilstandsmaskin	24
8.1	Bokstavmatrise basert på Table A.1 i IEC PAS 63131:2017 [12]	25
8.2	Monitor Binary	26
8.3	Monitor Analogue	27
8.4	Switch Binary Electrical	29
8.5	Prinsippskisse SBE tilstandsmaskin standard parameteroppsett	29
8.6	Switch Binary Valve	31
8.7	Prinsippskisse SBV tilstandsmaskin standard parameteroppsett	31
8.8	Enkel model av tilstandsmaskin	34
8.9	Tilstandsmaskin implementert i programmet	35
8.10	Tilstandslogikkblokker implementert i programmet	36
8.11	Eksempel CFC - Styring reaktor 1	37
8.12	Illustrasjon oppbygging program	38
8.13	Eksempel CFC - styring innpumping	39
8.14	Varslingar implementert i simulering	40
9.1	Grafisk presentasjon av overgangsbetingelsar	43

9.2	Objektliste	45
9.3	SCD av innpumpingssekvens	47
10.1	Kontinuerleg testing ved manipulasjon av verdiar	48
10.2	Simuleringsblokker	49
10.3	Eksempel Codesys visualiseringselement	50
10.4	Oppsett av simuleringsvindauge	51
11.1	Første gjennomgangen av anlegget	53

3 Innleiing

3.1 Om oss

Vi er tre studentar som studerar Automatiseringsteknikk med robotikk ved Høgskulen på Vestlandet (HVL), campus Førde. Vi har alle fagbrev som elektrikar og fann lett tonen i starten av studiet.

Gjennom tre år har vi nytta vår breie kompetanse innen industri, programmering og elektronikk til å danne ei god gruppe.

Roar Bøyum bur i Sogndal og har arbeidd med teknologi og prosjektstyring i maritime sektor.

Vegard Aven Ullebø bur i Vadheim og har variert erfaring frå industri og offshore-arbeid.

Peter Søreide Skaar er busatt i Førde og balanserar studiet med ei deltidsstilling i Renasys.

Peter si tilknytning til oppdragsgivar har gitt oss viktig innsikt i bransjen og Vegard og Roar sine tekniske kunnskapar og praktisk tilnærming har vore nyttige for prosjektet og hjelpt oss undervegs.

3.2 Oppdragsgivar

Renasy AS i samarbeid med Sunnfjord kommune

Renasy er ei innovativ og nyskapande oppstartsbedrift som arbeider med banebrytande teknologi innan mekanisk finpartikkelfiltrering av avlaupsvatn. Bedrifta har 15 tilsette fordelt på forskjellige kontor. Dei har kontor på Øyrane i Førde og Sandnes i Rogaland. Etter å ha arbeidd konfidensielt over lengre tid gjekk Renasy offentleg ut med teknologien sin i løpet av 2023. Dei tilbyr reinsetenester til kommunar og interkommunale selskap innan avlaup og maritime sektor.

Samarbeidet med Sunnfjord kommune er retta mot “Mission Zero” som er eit ambisiøst mål om null utslepp, null avfall, null energi og ein generell forbetring av avlaupssektoren i Noreg. [13]

Sunnfjord kommune er ansvarleg for vann, veg og avlaup i sitt område og har engasjert Renasy for å utforske forbetringar ved reinseanlegget på Sande.



Sunnfjord
kommune

Figur 3.1: Logo oppdragsgivar

3.3 Vedlegg

Til denne rapporten er det utarbeidd eit eige dokument som inneheld alle relevante vedlegg. Desse vedlegga omfattar ytterlegare tekniske detaljar knytt til oppgåva vår.

Vedlegga er organiserte for å gje lesaren ein djupare innsikt i arbeidet vi har utført.

Vedleggsliste er tilgjengeleg på side 66.

4 Analyse av oppgåva

Sande reinseanlegg ligg 25 minuttar med bil frå Førde sentrum og er ein av dei større reinseanlegga i Sunnfjord kommune. Anlegget er dimensjonert for 1500 personekvivalentar og vart bygd i 2003. Anlegget har fleire utfordringar, og ein av desse utgjør ei problemstilling som eignar seg godt til ei bacheloroppgåve innan automasjon.

4.1 Problemstilling

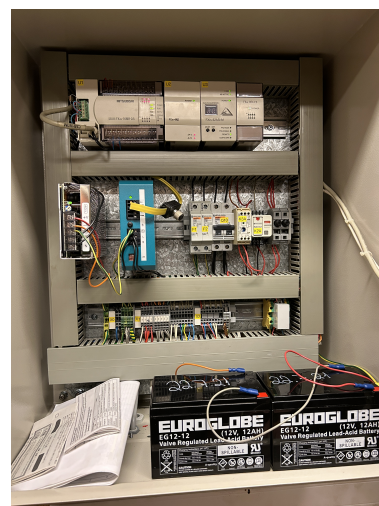
Reinseanlegget er teknisk utdatert og treng fornying. Styresystemet, som no er over tjue år gammalt, består hovudsakeleg av eldre og utgåtte komponentar. Med eldre komponentar aukar risikoen for svikt, og det kan vere vanskeleg å finne passende reservedelar.

WaterCare AS [14], som opphavleg leverte styresystemet har i seinare tid blitt avvikla. Dette gjer at kompetansen innan styresystemet og moglegheita for å gjere endringar i det, er utfordrande. Grunna desse utfordringane har ikkje anlegget klart å halde tritt med den teknologiske utviklinga, og mindre problem har gradvis auka til større utfordringar.

Samstundes med desse faktorane er dokumentasjonen til reinseanlegget mangelfull, noko som gjer at enkle arbeidsoppgåver blir utfordrande og tidkrevjande. I verste tilfelle kan styresystemet til anlegget svikte og med dei utfordringane nemnt ovanfor vil det være krevjande å få anlegget tilbake i drift. Dette utgjør ei kritisk utfordring innan offentleg infrastruktur og kan ikkje oversjåast.



(a) Beijer HMI



(b) Styreskap

Figur 4.1: Styresystem

4.2 Løysningsforslag

Fellesfaktor for løysningsalternativa til problemstillinga vil vere å fornye styresystemet med ein ny styringseining (PLS). Dette vil bidra til å løyse fleire av utfordringane som anlegget har i dag.

Det er fleire måtar å utføre dette på, men vi vel å fokusere på dei mest relevante.

4.2.1 Løysningsalternativ A: Bytte PLS

Den enklaste løysninga er å oppgradere eksisterande PLS ein til ein. Dette inneber å bytte ut den eksisterande PLS-en med ein nyare PLS frå same leverandør, som inneheld lik programvare.

Dette vil redusere faren for kritisk komponentsvikt, men sidan PLS-programmet er det same, vil det ikkje bli enklare å gjere endringar i styresystemet.

4.2.2 Løysningsalternativ B: Bytte PLS, konvertere styresystemet

Eit steg vidare frå løysningsalternativ A, er å konvertere det eksisterande styresystemet over til eit meir framtidsretta programmeringsspråk der kompetansen er meir tilgjengeleg. Dette alternativet inneber å konvertere logikk for logikk, med dei nødvendige tilpassingane som trengst for det nye språkets unike krav og funksjonar.

Utfordringa med dette alternativet er at ein risikerar å overføre eventuelle feil frå eksisterande styresystem over til det nye programmeringsspråket. Det kan også vere problematisk å gjere endringar i det konverterte programmet, ettersom det er vanskeleg og behalde den overordna strukturen.

4.2.3 Løysningsalternativ C: Bytte PLS, nytt styresystem

Eit tredje og meir komplett løysningsalternativ bygger på dei to førre alternativa, men forheld seg til problematikken med endringar i styresystemet. Dette alternativet involverer ikkje valet av PLS eller bruk av eksisterande PLS-program.

Ved å sette seg inn i anleggets teknologiske verkemåte vil vi kunne danne eit bilete av korleis anlegget skal fungere teoretisk. På denne måten vil ein kunne opprette ein ny funksjonsbeskrivelse for anlegget og bygge eit nytt program basert på denne beskrivelsen. Programmeringsmessig vil dette vere som å starte på ny, men dette vil gi sikkerheit for at anlegget driftast optimalt og at det nye programmet følgjer dagens standard. Dette tilsvara ein arbeidsprosess for eit nytt anlegg.

4.2.4 Løysningsalternativ D: Nytt anlegg, ny teknologi

I tillegg til ein full gjennomgang av styresystemet, kan ein undersøke nye løysningar for å optimalisere, forbetre og utvide heile reinseanlegget. I samråd med Renasys arbeider Sunnfjord kommune med å undersøke moglegheitene og kostnadane for eit slikt alternativ. Første steg i ein slik prosess er uansett eit løysningsalternativ som forholder seg til problema rundt styresystemet.

4.3 Val og drøfting

Då vi starta med å definere dei ulike løysningsalternativa i forprosjektet, var vi ikkje heilt sikre på kva retning vi skulle velje. Alle løysningsalternativa synast å vere gode, men med varierende arbeidsmengde. Vi kom fram til at løysningsalternativ C hadde ei arbeidsmengd som best passa ei bacheloroppgåve.

Dette var ein av grunnane til at vi valte løysningsalternativ C. I tillegg til dette er det fleire aspekt som gjer denne løysinga til den beste for anlegget.

Ved å følgje eit løysningsalternativ som bygger på ein arbeidsprosess for eit nytt anlegg kan vi forhindre å overføre feil og manglar frå eksisterande program. Dette gir oss også moglegheita til å utforske og finne betre løysningar, samtidig som vi oppgraderer styresystemet med ein forbetra oppbygging og heilskap. Vidare legg dette til rette for ein enklare og meir fullstendig dokumentasjonsjobb, som er essensiell for å oppretthalde kvaliteten og berekrafta til prosjektet over tid.

Med tanke på arbeidsmengd, valde vi å halde oppgåva teoretisk. Det var uvisst om reinseanlegget følgde gjeldande sikkerheitskrav og normer, og om starten av eit praktisk inngrep ville utløyse krav om forbetringar. Dette var noko vi ikkje ønska å ta stilling til i vår bacheloroppgåve, og det blei avklart med både Renasys og Sunnfjord kommune at oppgåva skulle haldast teoretisk.

5 Krav og mål

Renasys hadde som ønske at vi etablerte kravspesifikasjonen til oppgåva i samråd med Sunnfjord kommune.

Vi hadde eit møte med kommunen der dei kom med nokre konkrete ønske til oppgåva og løysningsforslaget vi hadde presentert.

1. Unngå høge lisenskostandar i planlegginga av PLS og val av programmeringsverktøy.
2. Opprette ein funksjonsbeskrivelse som forklarar korleis anlegget verkar.
3. Undersøke og planleggje forbetringar av anlegget ved hjelp av ny sensorikk.

5.1 Krav

Vi tok utgangspunkt i ønska frå Sunnfjord kommune for å danne kravspesifikasjonen. Deretter utarbeida vi ei liste med krav som naturleg delte seg i tre hovuddelar: dokumentasjon, programmering og simulering med verifisering. Denne oppdelinga vart framlagt for oppdragsgivarane og godkjent.

1. Dokumentasjon:
 - Dokumentere anlegget og opprette ein funksjonsbeskrivelse.
 - Dokumentere det nye styresystemet.
2. Programmering, vi skal:
 - Programmere etter den nyoppretta funksjonsbeskrivelsen.
 - Anvende open kjeldekode og ikkje låse anlegget til ein leverandør.
3. Simulering med verifisering
 - Simulere og teste det nye programmet

5.2 Mål

Utanom kravspesifikasjonane som var avklart med arbeidsgivar sat vi oss også nokre personlege mål for oppgåva.

- Utvide eigen kompetansen ved bruk av eit nytt programmeringsverktøy.
- Programmere eigne funksjonsblokker som anvender relevante industristandardar.
- Levere eit program som er lett å anvende og enkelt å vedlikehalde.
- Implementere tilstandsmaskin som styringsform.

6 Verkemåten til anlegget

For å sette oss inn i korleis Sande reinseanlegg fungerer var vi nøydd til å forstå korleis eit generelt avlaupsreinseanlegg er oppbygd.

6.1 Generell verkemåte

Eit avlaupsreinseanlegg er bygd opp av tre hovuddelar, primær, sekundær og tertiarreinsing, samt ein del for behandling av slam [15]. Desse delane kan løysast på forskjellige måtar, men hovudoppgåvene er dei same i alle reinseanlegg.

“Primærreinsing” handlar om å skilje organisk og uorganisk materiale. I eit avlaupsreinseanlegg tilsvarer dette å skilje avlaupsvatn, som ein vil behandle, frå sand, Q-tips, våtserviettar og anna material som ein ikkje ønskjer vidare i prosessen.

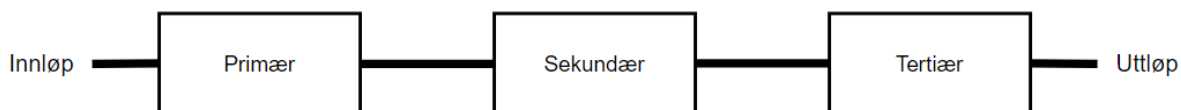
“Primærreinsing” er eit viktig steg for å bevare pumper og anna prosessutstyr.

“Sekundærreinsing” handlar om å fjerne mest mogleg suspenderte stoff og organisk materiale frå vatnet.

“Sekundærreinsing” er i kvart anlegg avhengig av kva “reinseprinsipp” som er nytta. Dette tilsvarer kva teknologisk metode som nyttast for å utføre steget. “Sekundærreinsing” refererast generelt som biologisk reinsing.

“Tertiarreinsing” handlar om å fjerne resterande forureiningar i vatnet. Dette steget varierer frå anlegg til anlegg og er avhengig av kva krav reinseanlegget har på sitt utsleppsvatn.

Slamebehandling handler om å fjerne og behandle det oppbygde organiske materialet (slam) som skiljast ut i sekundær og tertiarreinsing



Figur 6.1: Generell verkemåte for eit avlaupsreinseanlegg

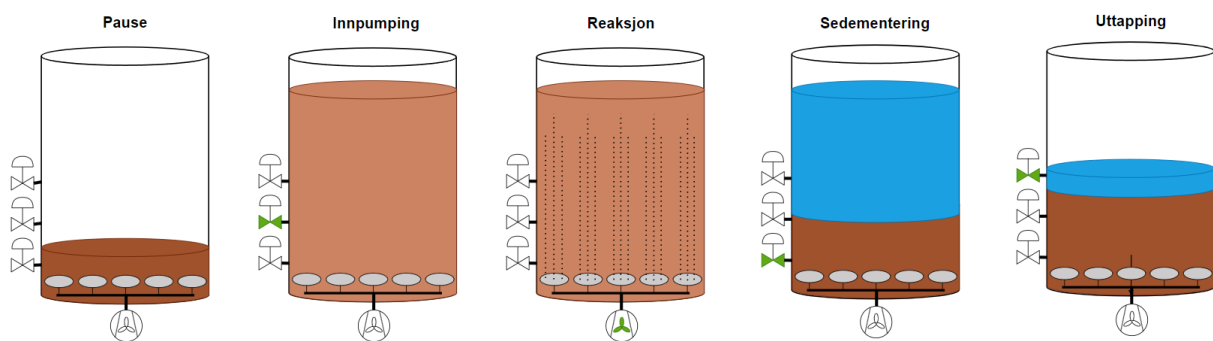
Då vi hadde fått oversikt over verkemåten til eit generelt avlaupsreinseanlegg, retta vi fokuset mot Sande. Dette valde vi å dele opp i tre hovuddelar. Kva reinseprinsipp er nytta, korleis Sande reinseanlegg fungerer i praksis og om anlegget har nokre særreigna preg.

6.2 Teknisk verkemåte

Sande reinseanlegg er konstruert og basert på SBR-teknologi.

SBR står for “Sequence Batch Reactor”, på norsk “sekvensiell batchreaktor”.

SBR er en reinsem metode der alle prosessar føregår i same reaktortank. Reaktor nyttar biologisk reinsing, ved hjelp av aktivert slam som inneheld mikroorganismar, for å koagulere og å fjerne løyste og ikkje sedimenterbare partiklar, samt stabilisere organisk materiale. Avlaupsvatn tilførast reaktor i “batcher” for å bli behandla. Kvar avlaups-batch går gjennom ein reaktorsyklus som består av følgjande fem delsekvensar [16].



Figur 6.2: SBR-prosessen

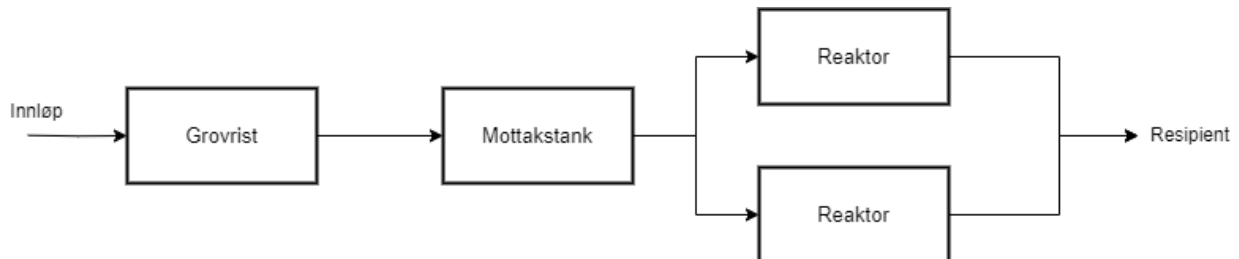
1. **Pause** : Reaktor er klar og ventar.
2. **Innpumping** : Reaktor mottar avlaupsvatn, normalt frå ein utjamningstank.
3. **Reaksjon** : Reaktor luftast periodisk for å tilføre oksygen til mikroorganismane.
4. **Sedimentering** : Reaktor sedimenterer ved hjelp av gravitasjon. Overskuddslam fjernast.
5. **Uttapping** : Reaktor drenerar reinsa vatn mot resipient.

Meir detaljert informasjon om SBR og anleggets teknologiske prinsipp er tilgjengeleg, via vedlegg, i anleggets nye funksjonsbeskrivelse. (Vedlegg G)

6.3 Praktisk verkemåte

Sande reinseanlegg består av “primærreinsing” via grovrisk, ein mottakstank som samlar varierende tilstrøymingar for å gi resten av anlegget homogene forhold og “sekundær” og “tærtierreinsing” ved to reaktorar som anvender SBR-teknologi.

Avlaupsvatn vil opphalde seg i eller på veg mot ein av desse fire hovuddelane medan det er i anlegget. Ferdig behandla avlaupsvatn blir drenert ut til resipient (elva Gaula).



Figur 6.3: RA200 flytskjema

6.3.1 Grovrisk

Innløpet på anlegget renn først gjennom grovrista. Grovrista på reinseanlegget er ein “HUBER ROTAMAT Ro9” som er ei grovrisk som nyttar ein mekanisk skru. Den fungerer som ein liten tank og ein intern nivågivar startar skruen ved innkommande avlaupsvatn. Skruen tek med uorganisk materiale og fjernar det til eigen avfallshandtering. Dersom grovrista er ute av drift vil vatn renne vidare til mottakstanken via eit overløpsrør.



(a) Motor og avfallshandtering



(b) Gods og skru

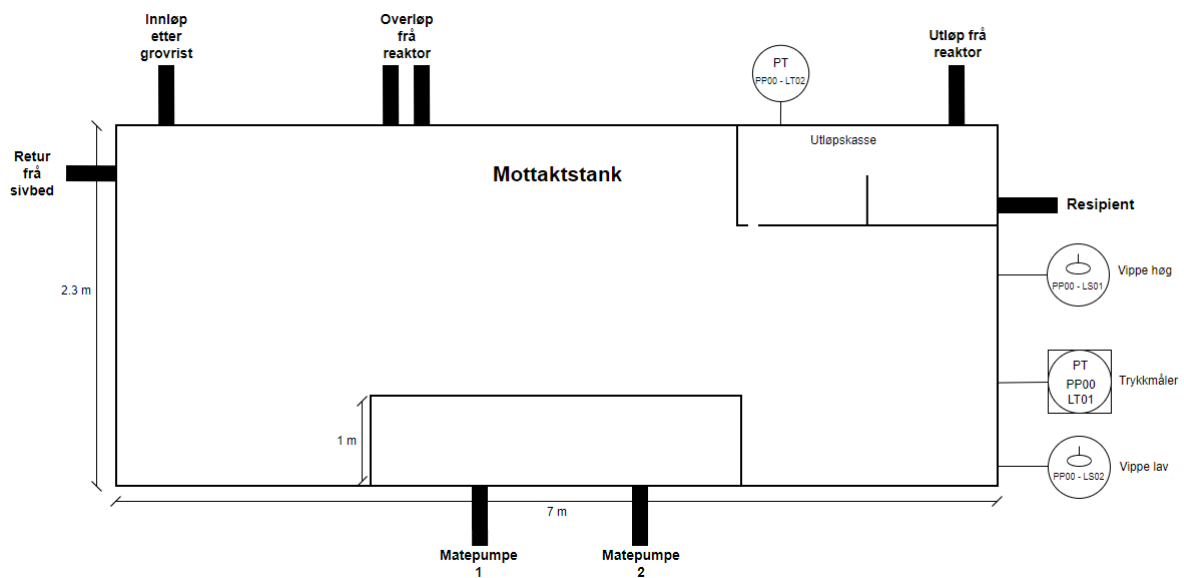
Figur 6.4: Grovrisk levert av Huber

6.3.2 Mottakstank

Frå grovrasta renn vatn med sjølvfall mot mottakstanken som ligg som lågaste punkt på anlegget. Mottakstanken er 120 m^3 og er ein felles lagringsplass for vatn før det går vidare mot reaktorane. Mottakstanken har fire sensorar som heng frå taket.

Nivået i mottakstanken blir primært målt med trykkgivar LT01. For at vatn skal pumpast vidare mot ein reaktor i riktig sekvens må trykkgivar indikere at nivået er høgt nok. LS02 fungerer som backup.

I toppen av mottakstanken er det ei overløpskasse som drenerer mot resipient, her vil det ved normale omstendigheter ikkje renne anna ein reinsa vatn. Trykkgivar for overløp måler dersom ureinsa vatn renner i resipientrøyret.



Figur 6.5: P&ID mottakstank

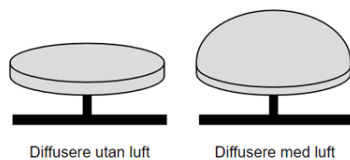


Figur 6.6: Matepumper

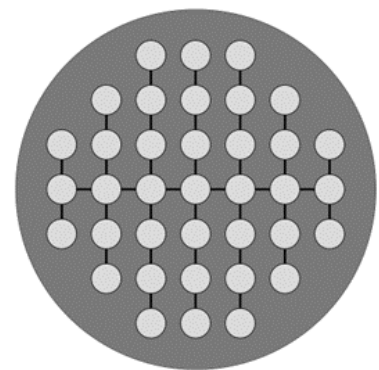
6.3.3 Reaktor

Frå mottakstanken blir vatn pumpa opp til reaktor dersom den er i riktig sekvens. Vatn blir pumpa ved hjelp av to matepumper som rullerer, der kvar pumpe kan levere til kvar reaktor. Reaktorane er $165 m^3$ og står på bakkenivå og strekker seg opp mot taket på bygget. Reaktorane er utstyrt med nivåmåling via trykk og sensor er plassert to meter over botn.

I reaksjonssekvensen blir reaktorane periodisk lufta. Dette er for å lage aerobe og anaerobe fasar for mikroorganismene som vidare gir betre reinsing. For å best spreie oksygenet i den aerobe fasen er det satt inn eit diffuser-oppsett i botn. Diffuserane er laga av ein membran med små hol som dannar bobler når lufta kjem i kontakt med avlaupsvatn. Lufting av reaktoren gir også effektiv omrøring utan behov for ekstra mekanisk inngrep.



(a) Illustrasjon diffuser



(b) Oppsett av diffuser

Figur 6.7: Diffuser system

På reinseanlegget er det tilleggskrav for fjerning av fosfor [15]. På grunn av desse tilleggskrava er det sett inn eit tertiærreinsteg ved hjelp av simultanfelling. Simultanfelling er ein fellesbetegnelse på kombinert biologisk og kjemisk reinsing.

I slutten på reaksjonssekvensen tilsettest polyaluminium klorid, og kjemikaliet binder seg til løyst fosfor og danner sedimenterbare partiklar [17]. Desse sedimenterbare partiklane synk så i sedimenteringssekvensen og utsleppskravet på fosfor oppretthaldast.

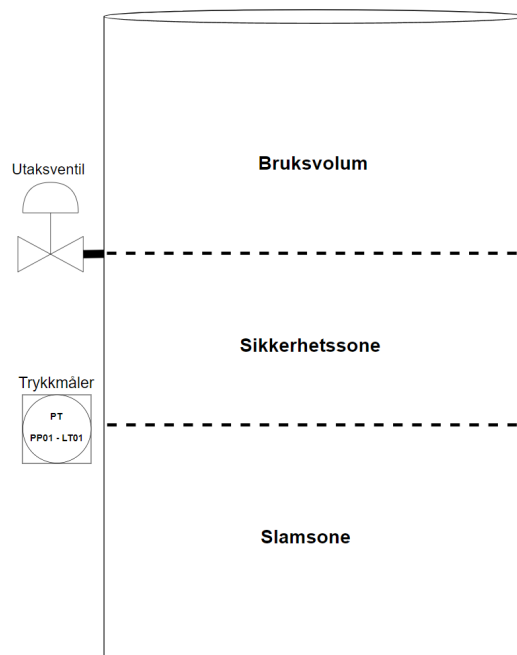


Figur 6.8: Reaktor

Reaktorane er delt opp i tre forskjellige soner. Desse er med på å skilje dei forskjellige substansane når reaktoren er ferdig med ein syklus. Alle SBR-reaktorar har lagra aktivert slam i botn [16]. Det er her alle mikroorganismane akkumulert og gir grunnlag for god biologisk reinsing. Denne sona blir kalla slamsona.

Sjølve bruksvolumet til reaktoren er alt over uttaksventilen, og det er dette volumet som blir fylt og behandla under kvar innpumpingsekvens.

Mellom dei to sonene er det ei sikkerheitssone. Denne sona er med for å ta hand om varierende sedimenteringsegenskapar og eventuelt overskottsram.

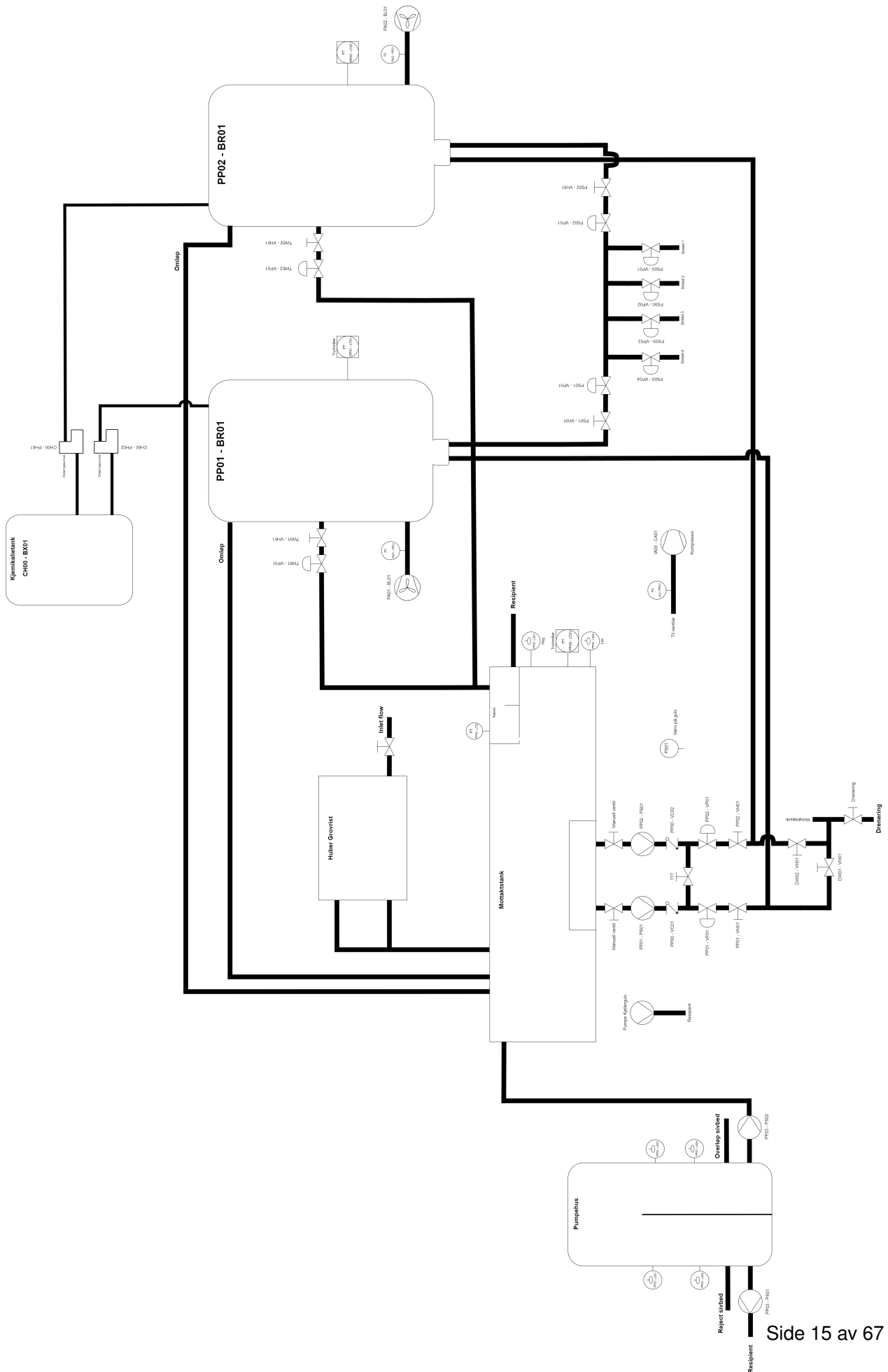


Figur 6.9: Illustrasjon av reaktorsoner

6.3.4 P&ID

Arbeidet rundt anleggets verkemåte var meir problematisk enn forventa. Hovudgrunnen til dette var at det ikkje fantest noko røyr gateskjema eller teknisk planteikning.

Vi såg det som naudsynt å etablere ein P&ID for å betre dokumentere og vise samanhengen til anlegget.



Figur 6.10: P&ID

6.4 Spesifikke anleggsvariasjonar

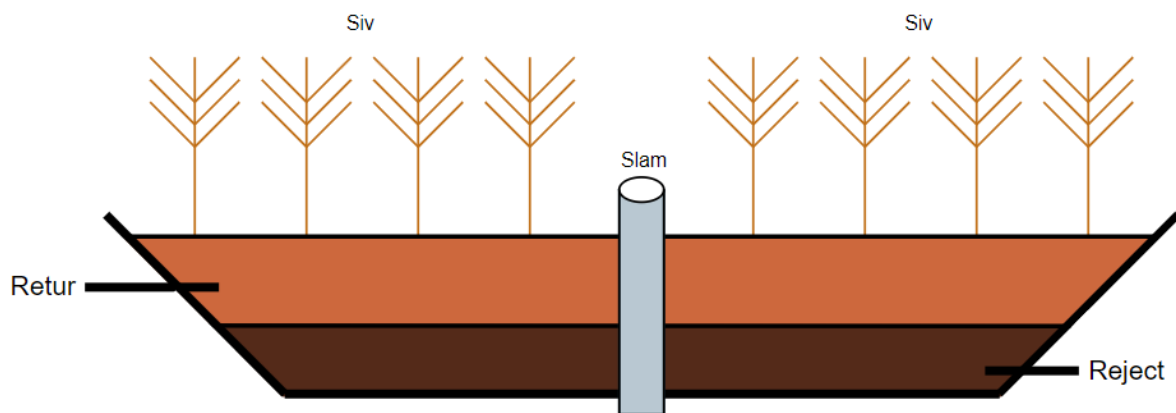
Sjølv om Sande reinseanlegg anvender SBR-teknologi så er det enkle spesifikke punkt der dette reinseanlegget avviker frå normalen. Sande reinseanlegg nyttar eksempelvis ein spesiell form for slambehandling sett i norsk perspektiv.

Vanleg slambehandling er deponering av slam som handterast og sendast til hygienisering [18]. På Sande blir ikkje slammet lagra, men jamnlig spreidd ut over eit bestemt område. På dette området er det planta siv som skal ta opp slammet og resterande vatn blir naturleg filtrert og drenert. Desse områda kallast sivbed.

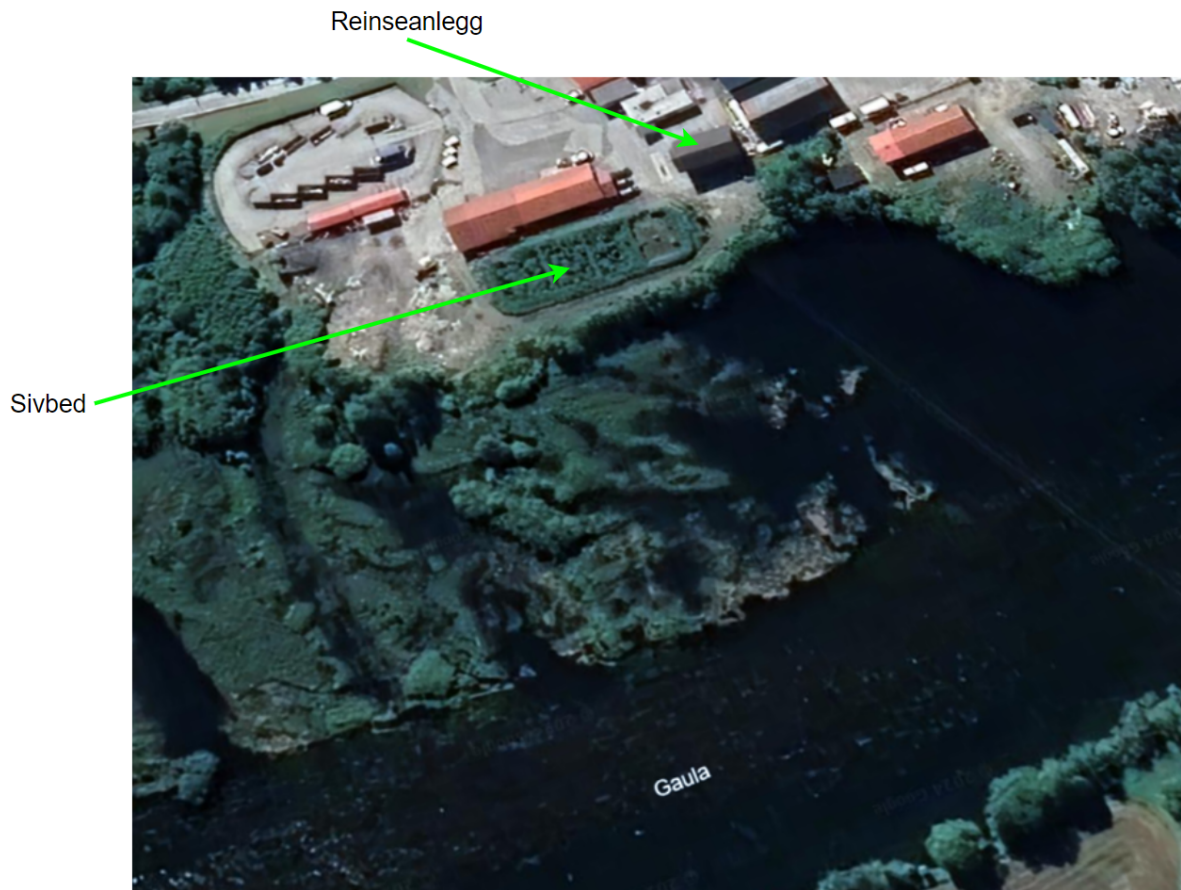
Sivboda er konstruert med fleire dreneringslag som gjer at resterande vatn skiljast ut i forskjellige soner. Her kan desse handterast vidare etter ønska behov. Grunna denne slambehandlingsmetoden er det på Sande reinseanlegg heller slutfjerning frå reaktor i reaksjonssekvens. Dette blir gjort for å ha mindre konsentrert slam [19].

Sande reinseanlegg har fire sivbed og den kombinerte størrelsen er 676 m^2 . Sivboda er lokalisert på utsida av reinseanlegget (sjå figur 6.12).

Vidare har kvart sivbed sin eigen respektive ventil (sjå figur 6.14), og under uttapping av slam vil kun eit sivbed og ein ventil vere aktiv.



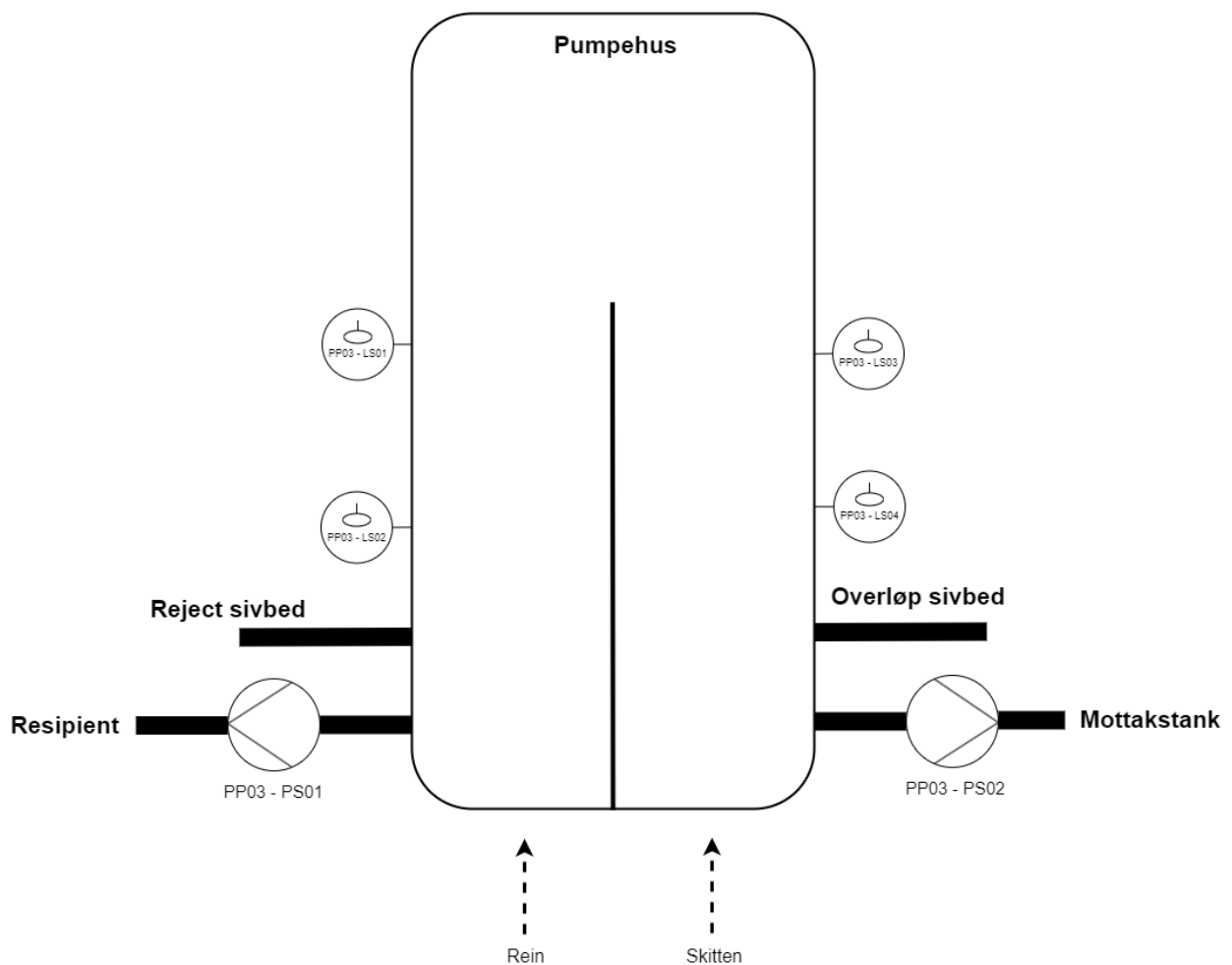
Figur 6.11: Illustrasjon sivbed



Figur 6.12: Satelittfoto frå Google Earth [11]

Vatn frå desse forskjellige dreneringslaga renn vidare til eit pumpehus/kumme. Pumpehuset er lokalisert tjue meter frå sjølve reinseanlegget og er utstyrt med to pumper og nokre nivåvipper. Pumpehuset er delt i to og skiljer vatn som kjem frå dei forskjellige dreneringslaga.

I sivbedet er vatn frå djupaste dreneringssone klassifisert som reinsa vatn (sivbed "reject") og blir sendt ut til resipient. Den øvre dreneringssona er fortsatt klassifisert som skittent og blir returnert til mottakstanken.



Figur 6.13: P&ID pumpehus



Figur 6.14: Sivbedventilar

6.5 Dokumentasjon

Som tidlegare skildra i kapittel 5 var ein stor del av oppgåva å dokumentere verkemåten til anlegget. Vi bestemte oss for å gjere ei dokumentasjonsfornying der vi henta inn alt det som var av tidlegare dokumentasjon og kombinerte det med vår nyerfarte kunnskap.

Vi oppretta ein funksjonsbeskrivelse som bygger vidare på driftsinstruksen (Vedlegg F) som var levert av Watercare i 2003. Dokumentet er tiltenkt ein slags bruksanvisning for heile anlegget, der sikkerheit, prosess, verkemåte og styresystem er sentrale tema.

Funksjonsbeskrivelsen skal være forståeleg for alle intresserte parter og inneheld mange av våre nye figurar og avsnitt. Dokumentet inneheld detaljert kvifor og korleis ting heng saman og gir derfor ein forståelse av anlegget som ikkje var mogleg med den opphavlege dokumentasjonen.

Funksjonsbeskrivelsen inneheld alt om reinseanlegget på Sande men er delt opp i kapittel for å ikkje overvelde lesar med unødvendig informasjon. Kompleksiteten auker gradvis gjennom dokumentet og programmering av anlegget ligg under utdjupa teknisk beskrivelse.

Dokumentet er delt opp i desse kapittela.

1. Introduksjon
2. Verkemåte
3. Teknisk beskrivelse
4. Drift og vedlikehald
5. Feilsøking
6. Utdjupa teknisk beskrivelse
7. Teknisk underlag

Alt som er nemnt her i kapittel 6 står meir detaljert under kapittel 3 i funksjonsbeskrivelsen og er sentralt for å best forstå korleis anlegget fungerer.

Funksjonsbeskrivelsen ligg som vedlegg. (Vedlegg G)

7 Tillrettelegging for programmering

Tidlegare skildra under kapittel 5, ønskja vi å nytte ei løysning som ikkje hadde høge lisenskostnadar for vår arbeidsgivar. Sunnfjord kommune var interessert å ikkje låse seg til ein fast leverandør, men heller ha moglegheita til å ha valet mellom fleire leverandørar innan PLS. Dette ilag med moglegheiten til å utvide vår eigen kompetanse gjorde at vi såg vekk frå Siemens TIA-portal [20], som vi hadde lært gjennom PLS emnet ELE304. Programmet ønskja vi å programmere i strukturert tekst (ST), men undersøkte også typar av grafiske diagrambaserte språk for å betre vise samanhengar.

Det var også viktig at alle på gruppa kunne programmere samtidig og at ein felles programmeringsstandard skulle nyttast. Det var viktig at parallelt arbeid ikkje skulle gi synkroniseringsproblem og at det fantest ei god løysning for dette.

7.1 IEC

IEC [21] er ein internasjonal, ikkje statleg organisasjon som utviklar og publiserer tekniske standardar innan elektrofag. Noreg er representert i IEC ved Norsk Elektrotekniske Komité (NEK) [22]. IEC har ein standard som dekker programmering av PLS som går heilt tilbake til 1993 [23]. Den nåverande standarden som omfattar PLS er IEC 61131 [24]. Dette er ein standard spesielt laga for programmerbare kontrollarar, og er delt opp i ti delar, der del tre tar for seg programmeringsspråk.

Vårt program er i hovudsak tiltenkt programmert etter IEC 61131-3 og IEC PAS 63131 [25]. IEC PAS 63131 er ein standard som gir oss grunnlag for å nytte forhandsdefinerte funksjonstemplat for funksjonsblokker. IEC PAS 63131 er laga med formål at leverandørindustrien og oljeselskap skal ha eit felles rammeverk for bruk på norsk sokkel, og er utarbeida etter NORSOK I-005:2013 [26]. Ved å nytte desse standardane gir det oss eit robust og fleksibelt rammeverk for å programmere anlegget. Ved bruk av forhandsdefinerte funksjonstemplat har vi moglegheit til å enkelt knytte saman fleire delar av programmet. Dette gir oss fleksibilitet ved å enkelt kunne endre og leggje til funksjonar.

Ved bruk av IEC PAS 63131-standardens sikrar vi at begge oppdragsgivarar får tilgang til funksjonsblokker som kan brukast i alle PLS-styresystem i industrielle samanhengar. Dette gir fleksibilitet og meirverdi utover det som blir implementert i det planlagde styresystemet, ettersom funksjonsblokkene kan gjenbrukast og tilpassast ulike applikasjonar og framtidige behov.

7.2 Codesys

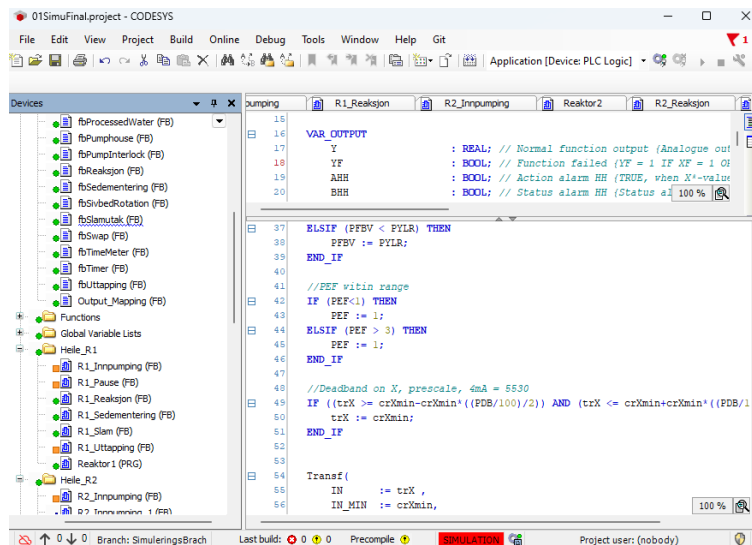
Codesys laga av Codesys Group tilbyr ein open kjeldekodeløysning for prosjektet og har ingen lisens-kostnader. [27]. I tillegg så kan prosjektfilene nyttast på fleire typar PLS einingar [28]. Dette gir vår oppdragsgivar fleksibilitet i korleis dei ønskjer å implementere løysningsforslaget.

Codesys nyttar programmeringsspråkstandarden satt av IEC 61131-3 som omfattar ST, “Sequential Function Chart” (SFC) og “Ladder Diagram” (LD).

Codesys har nyleg fått støtte for integrering av GitHub i programvaren [29]. Dette gjer det enklare å halde versjonskontroll og for gruppe medlem å programmere saman. GitHub har vi nytta på andre prosjekt, og det har vi god erfaring med.

Vidare så har Codesys støtte for bibliotek gjennom CODESYS Store [30]. Ved å nytte kjende bibliotek, får vi tilgang til ei samling av gjenbrukbare kodeblokker og funksjonar. Dette er bilbioteka vi ønskjer å nytte.

- CODESYS Building Automation [31]
- SysTime [32]
- Util [33]



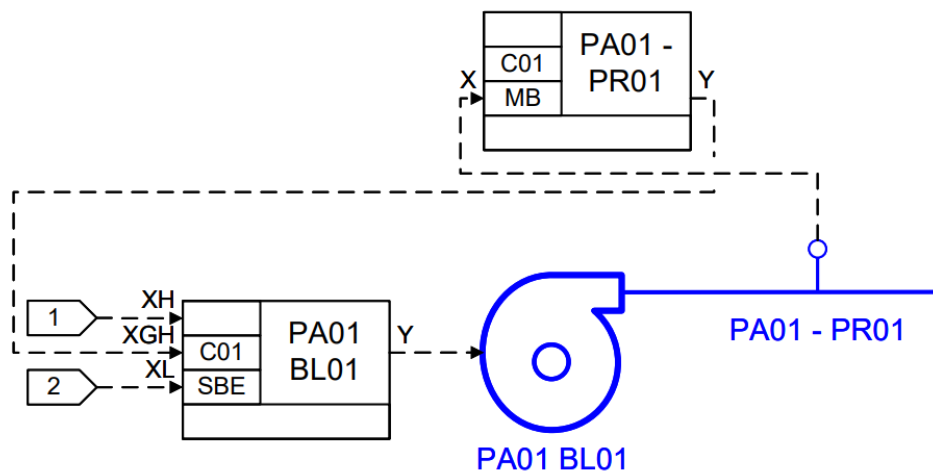
Figur 7.1: Codesys programvare

7.3 SCD

“System Control Diagram” (SCD) er eit grafisk dokumentasjonsverktøy skildra i IEC PAS 63131. SCD blir nytta for å vise relasjon mellom prosessens komponentar og programmet som styrer dei.

Vi ønskjer å nytte SCD som eit planleggingsverktøy for programstrukturen. SCD inneheld IEC funksjons-templata, noko som gjer at ein kan planleggje programmeringa av anlegget før ein har programmert blokk-koden. Diagrammet gir oss moglegheit til å visualisere, teikne og kople IEC-blokkene mot komponentane på anlegget. SCD gir oss også ein unik moglegheit for å kunne dokumentere arbeidet og vil gi ein grafisk representasjon av styringsform og løysningar som blir valt.

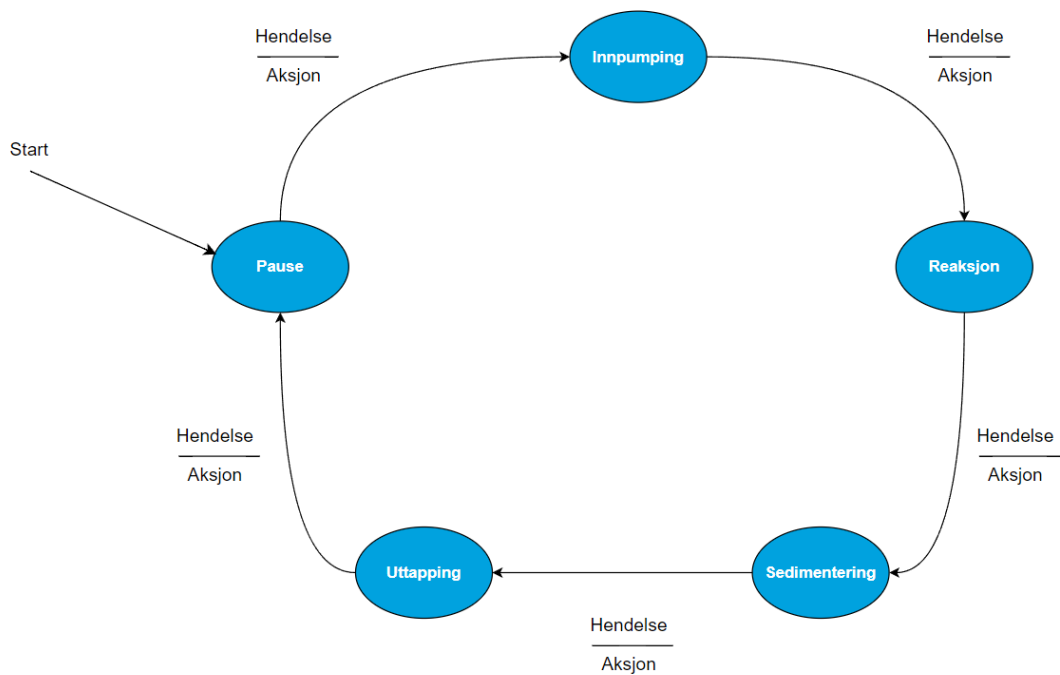
Vi har kontakta MIDTechnology [4] som er eit selskap som utviklar programvare for SCD og vi har fått utlevert studentlisensar. Vi vil nytte programvaren til planlegging og dokumentering av styresystemet til reinseanlegget.



Figur 7.2: Eksempel av SCD

7.4 Tilstandsmaskin

Vi identifiserte tidleg i prosessen eit ønske om å lage ei tilstandsmaskin som hadde overordna styring. Sidan ein SBR-reaktor er oppbygd av sekvensar verka det logisk å implementere ei tilstandsmaskin. Tilstandsmaskina vil ha ansvar for å oppretthalde korrekt tilstand og avansere vidare når gitte kriterium er nådde.



Figur 7.3: Prinsipp tilstandsmaskin

8 Programmering

Etter å ha danna oss eit godt grunnlag av korleis vi ønskja programmet i kapittel 7 byrja vi programmeringa. Vi starta eit tomt prosjekt i Codesys med kun undervisningsmatriell i ELE304 som erfaring. [34] [35] Første steget var å setje oss inn i, og programmere etter IEC funksjonstemplata.

8.1 IEC funksjonsblokker

Vi gjorde eit utval av moglege funksjonsblokktemplat basert på dei komponentane vi identifiserte i reinseanlegget. For å lage eit robust program valde vi å fokusere på to inngangsfunksjonar, MA, MB, og to utgangsfunksjonar SBE, SBV, som måtte programmerast frå grunn.

Funksjonsblokktemplata inneholdt avansert funksjonalitet som vi visste ville ta lang til å programmere. Sjølv om ikkje all denne funksjonaliteten var naudsynt for reinseanlegget, valde vi likevett å programmere alt, ettersom blokkene gir potensiell meirverdi i mogleg utvidning og eventuell bruk i andre styresystem. Det gav oss også ein tydelig retning å arbeide mot og vi visste at blokka ville dekke alle aktuelle behov.

IEC har sentrale begrep som vi ønskjer å utdjupe nærmare. Grunna mangel på gode norske begrep og for å forhindre forvirring vel vi å beskrive begrepa slik dei er skildra i normen [36]

- **Lock:** Action overruling any other signal while being true
- **Force:** Action overruling any other signal
- **Disable transition:** Transition high/low function not available
- **Blocking:** Prevention of certain functions or operations
- **Suppression:** Disable alarm annunciation as well as any associated automatic actions

Det er også andre genrelle forkortingar som er viktige i skildring av desse blokkene.

Bokstav	Første karakter	Andre karakter
A	Action alarm	Auto mode
B	Binary status	Blocked mode
C		Confirmed
D		Disable transistion mode
E		Enable status
F		Fault/Failed/Forward
G		Position
H		High
I		Internal setpoint mode
J		
K		
L	Lock command	Low
M		Manual mode
N		Hold
O		Outside mode
P	Parameter	Reversed if required
Q		Quantity
R	Reset command	Refrence signal
S	Set command	Safeguarding mode
T		Track mode
U		Supressed mode
V		Variance/deviation
W	Warning arlarm	Warning alarm
X	External input	Multifunction
Y	Normal function output	
Z		

Figur 8.1: Bokstavmatrise basert på Table A.1 i IEC PAS 63131:2017 [12]

8.1.1 Monitor Binary

Inngangsfunksjon MB blir nytta til automatisk overvaking, alarmhandtering, framvising og låsing av binære prosessvariablar [25]. Funksjonsblokka er nytta i programmet for å overvake alle digitale inngangar t.d. pressostat og nivåvipper.

MB mottar binær verdi på X og returnerer verdi på Y.

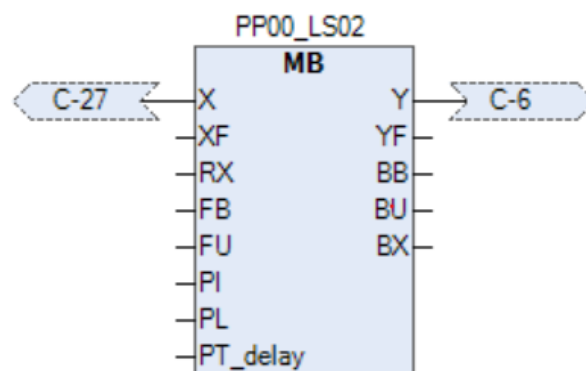
Denne utgangsverdien har moglegheit for tidsforseinking, invertering, låsning, “blocking” og “suppression” via “force” inngangsvariablar og parameter.

Blokka har tre statusvariablar som indikerer modus og område (BU, BB, BX), ein utgang for indikasjon av feil (YF) med moglegheit for “suppression” og ein inngang (RX) for reset av eventuell låst verdi.

A.4.11.2.1 Function template schematic

Inputs		MB		Outputs	
Normal function input	X	Y	Normal function output		
External fault	XF	YF	Function failed		
Reset latched output	RX	BB	Status blocked		
Force blocking	FB	BU	Status suppressed		
Force suppression	FU	BX	Status normal function input		
<u>Operator station:</u>		<u>Operator station:</u>			
Blocking on/off		Alarms and faults			
Suppression on/off		Latched value			
Reset latched output		Blocked			
		Suppressed			

(a) IEC PAS 63131:2017 [37]



(b) MB nytta i programmet

Figur 8.2: Monitor Binary

Meir informasjon om blokka, inngangar, utgangar, og parameter er tilgjengeleg i vedlegg. (Vedlegg C.1)

8.1.2 Monitor Analogue

Inngangsfunksjon MA er nytta for skalering, visning, overvaking og alarmhandtering av analoge prosess og kontrollvariablar [25]. Funksjonsblokka er nytta for å overvake analoge trykknivåivarar, og å skalere å vise desse som ein fyllingsgrad i prosent.

MA hentar rå analog verdi på X og returnerer verdi på Y. Denne utgangsverdien blir skalert basert på parameter og har hysteresis og dødband tilgjengeleg.

Hysteresis og dødband definerer områder rundt eit gitt setpunkt der systemet ikkje endrar tilstand eller reagerer på små variasjonar. Dette gjer at systemet unngår hyppige veksingar og unødvendige justeringar.

Funksjonsblokka har to forvarsel (WH og WL) og to handlingar med alarm (AHH og ALL). Det er moglegheit for tidsforsinking, "blocking" og "suppression" av desse via "force" inngangsvariablar og parameter.

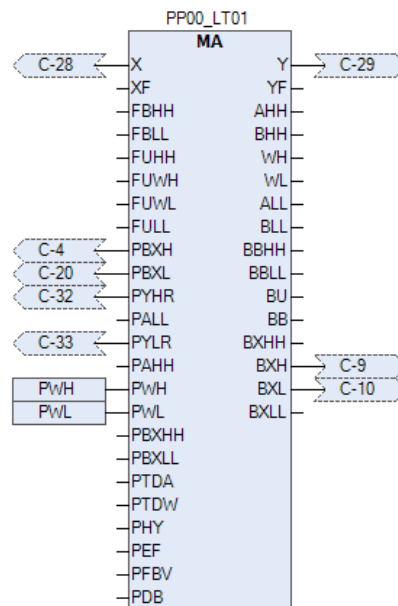
Blokka har fem statusvariablar som indikerer modus og område (BU, BB, BHH, BLL, BBLL, BBHH), ein utgang for indikasjon av feil (YF) og fire hendingar (BXHH, BXH, BXLL, BXL) utan "blocking" og "suppression" som nyttast for styring.

Grenser på varsling og hendigar er justerbare via parameter.

A.4.9.2.1 Function template schematic

Inputs	MA	Outputs
Normal function input	X	Y Normal function output
External fault	XF	YF Function failed
Force blocking alarm HH	FBHH	AHH Action alarm HH
Force blocking alarm LL	FBLL	BHH Status alarm HH
Force suppression alarm	FUHH	WH Warning alarm H ²⁾
HH		
Force suppression alarm	FUWH	WL Warning alarm L ²⁾
WH		
Force suppression alarm	FUWL	ALL Action alarm LL
WL		
Force suppression alarm LL	FULL	BLL Status alarm LL
		BBHH Action alarm HH is blocked
		BBLL Action alarm LL is blocked
		BU Status suppressed
		BB Status blocked
		BXHH Status event HH
		BXH Status event H
		BXL Status event L
		BXLL Status event LL
Operator station: Blocking HH on/off Blocking LL on/off Suppression on/off		Operator station: Alarms and faults Alarm and event limits Blocked Suppressed

(a) IEC PAS 63131:2017 [38]



(b) MA nytta i programmet

Figur 8.3: Monitor Analogue

Meir informasjon om blokka, inngangar, utgangar, og parameter er tilgjengeleg i vedlegg. (Vedlegg C.2)

8.1.3 Switch Binary Electrical

Utgangsfunksjon SBE blir nytta for binærkontroll (av/på) av strøymingselement for elektrisitet, varme eller væske. Den kontrollerte komponenten kan vere t.d. motor, pumpe, varmeelement, vifte [25].

Funksjonsblokka er nytta til å styre motorar, pumper og blåserar.

Funksjonsblokka har tre modus, auto, manuell og lokal. Lokal betyr lokal HMI eller direktekøyring. SBE hentar start/stopp signal på.

1. **Auto:** XH og XL
2. **Manuell:** HMI
3. **Lokal:** XOH og XOL

Aktiveringssignal er tilgjengeleg via inngang XE.

Blokka sender køyrsignal på Y og via pulsmodulerte utgangar YH og YL.

SBE inneheld ei tilstandsmaskin med fire funksjonstilstandar.

1. **Høg:** Komponent køyrer
2. **Lav:** Komponent stoppa
3. **Transisjon mot lav:** Fått stoppsignal
4. **Transisjon mot høg:** Fått køyrsignal.

I transisjonstilstand ventar blokka på eksternt tilbakemelding (XGH) og sett tilstanden til høg/lav etter at korrekt tilbakemelding er mottatt. Dette vises via to statusvariablar (BCL, BCH)

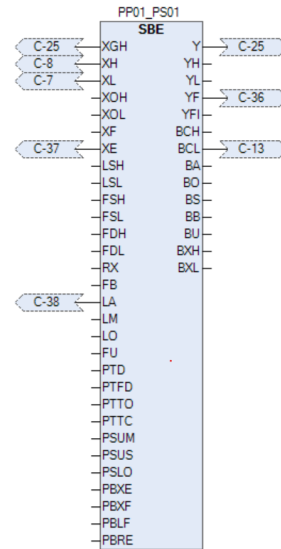
Blokka har sju andre statusvariablar som indikerer modus og område (BA, BO, BS, BB, BU, BXH, BXL) og ein utgang for indikasjon av feil (YF) med tilhøyrande heiltallsverdi (YFI) som indikerer feiltype. Feilstatus har moglegheit for "suppression".

Blokka har fire signal for høg/lav tvangskøyring, der "lock" endrar modus til manuell (LSH, LSL) og "force" gjer ikkje (FSL, FSH) og to signal for høg/lav "force disable transition" (FDH, FDL). Desse signala har moglegheit for "blocking".

A.4.17.2.1 Function template schematic

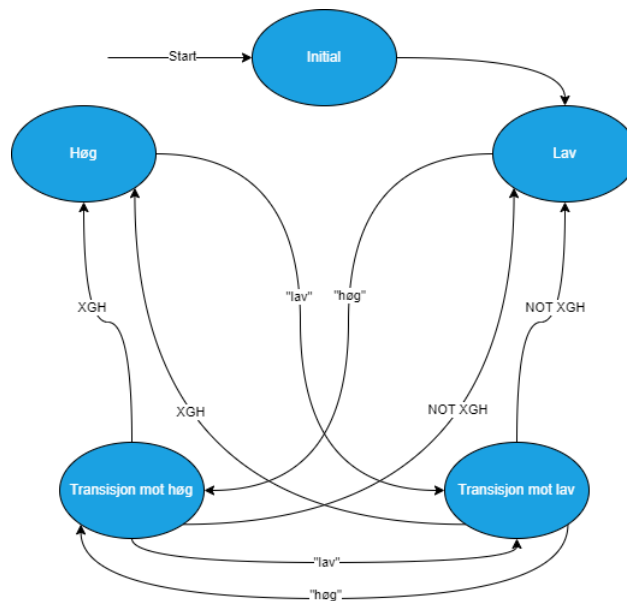
Inputs	SBE	Outputs
Pos high feedback (MCC)	XGH	Y
External set high	XH ⁽⁷⁾	YH
External set low	XL ⁽⁹⁾	YL
External outside set high	XOH	YF
External outside set low	XOL	BCH
External fault	XF	BCL
Externally enabled (MCC)	XE	BA
Lock safeguarding high	LSH	BO
Lock safeguarding low	LSL	BS
Force safeguarding high	FSH	BB
Force safeguarding low	FSL	BU
Force disable transition high	FDH	BXH
Force disable transition low	FDL	BXL
Force blocking	FB	
Force suppression	FU	
Lock auto	LA	
Lock manual	LM	
Lock outside	LO	
<u>Operator station:</u>		<u>Operator station:</u>
Auto/Manual/Outside		Alarms and faults
Set output on(high)/off(low)		Running/Stopped
Blocking on/off		Auto/Manual/Outside
Suppression on/off		Blocked
		Suppressed
		Disabled
		Safeguarding
		Conflict
		Priority
<u>Information from SBC⁽⁸⁾:</u>		<u>Information to SBC⁽⁷⁾:</u>
External set high		Run
External set low		Fault
Priority provided by SBC		Available (SBE in auto)
		Start disabled (FDH)
		Stop disabled (FDL)
		Safeguarding

(a) IEC PAS 63131:2017 [39]



(b) SBE nytta i programmet

Figur 8.4: Switch Binary Electrical



Figur 8.5: Prinsippskisse SBE tilstandsmaskin standard parameteroppsett

Meir informasjon om blokka, inngangar, utgangar, og parameter er tilgjengeleg i vedlegg. (Vedlegg C.3)

8.1.4 Switch Binary Valve

Utgangsfunksjon SBV skal nyttast til binær (av/på) kontroll av eit strøymingselement ved å endre strøymen av medium (varme eller væske). Typisk komponentar som styrast er bl.a. ventilar og spjeld [25]. Funksjonsblokka er nytta til å styre ventilar.

Samanlikna med SBE har SBV har tre modus, auto, manuell og lokal som nyttar inngangar med same namn. SBV hentar opne/stenge signal på.

1. **Auto:** XH og XL
2. **Manuell:** HMI
3. **Lokal:** XOH og XOL

Blokka sender signal på Y og via pulsmodulerte utgangar YH og YL.

SBV inneheld ei tilstandsmaskin med fire funksjonstilstandar.

1. **Høg:** Komponent open
2. **Lav:** Komponent stengd
3. **Transisjon mot lav:** Fått stengesignal
4. **Transisjon mot høg:** Fått opnesignal.

I transisjonstilstand ventar blokka på eksternt tilbakemelding (XGH, XGL) og sett tilstanden til høg/lav etter at korrekt tilbakemelding er mottatt. Dette vises via to statusvariablar (BCL, BCH)

Blokka har fem andre statusvariablar som indikerer modus og område (BA, BO, BS, BB, BU) og ein utgang for indikasjon av feil (YF) med tilhøyrande heiltallsverdi (YFI) som indikerer feiltype. Feilstatus har moglegheit for "suppression".

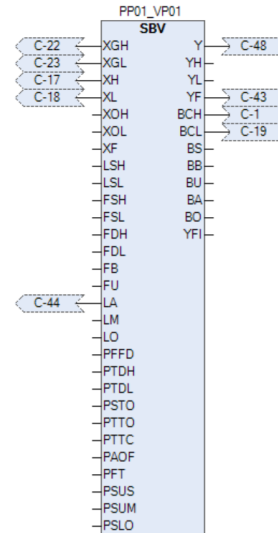
Blokka har fire signal for høg/lav tvangskøyring, der "lock" endrar modus til manuell (LSH, LSL) og "force" gjer ikkje (FSL, FSH) og to signal for høg/lav "force disable transition" (FDH, FDL). Desse signala har moglegheit for "blocking".

A.4.18.2.1 Function template schematic

Inputs	SBV	Outputs
Position high feedback	XGH	Y Normal function output
Position low feedback	XGL	YH Pulsed normal function output high
External set high	XH	YL Pulsed normal function output low
External set low	XL	YF Function failed
External outside set high	XOH	BCH Output position high confirmed
External outside set low	XOL	BCL Output position low confirmed
External fault	XF	BS Status safeguarding
Lock safeguarding high	LSH	BB Status blocked
Lock safeguarding low	LSL	BU Status suppressed
Force safeguarding high	FSH	BA Status auto/man
Force safeguarding low	FSL	BO Status outside
Force disable transition high	FDH	
Force disable transition low	FDL	
Force blocking	FB	
Force suppression	FU	
Lock auto	LA	
Lock manual	LM	
Lock outside	LO	

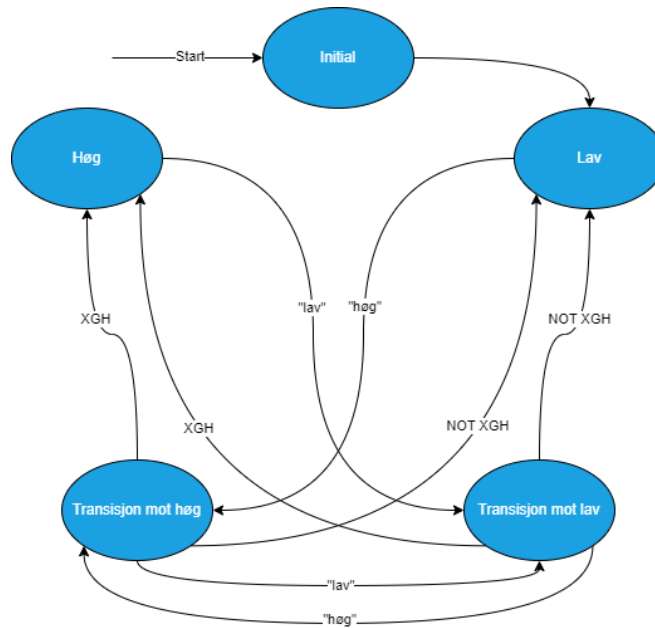
Operator station:	Operator station:
Auto/Manual/Outside	Alarms and faults
Set output open(high)/close(low)	Open/Opening/Closed/Closing
Blocking on/off	Auto/Manual/Outside
Suppression on/off	Blocked
	Suppressed
	Disabled
	Safeguarding
	Conflict

(a) IEC PAS 63131:2017 [40]



(b) SBV nytta i programmet

Figur 8.6: Switch Binary Valve



Figur 8.7: Prinsippskisse SBV tilstandsmaskin standard parameteroppsett

Meir informasjon om blokka, inngangar, utgangar, og parameter er tilgjengeleg i vedlegg. (Vedlegg C.4)

8.2 Generelle funksjonsblokker

Undervegs i programmering av IEC-blokkene såg vi det naudsynt av nokre generelle funksjonsblokker som kunne gjenbrukast fleire gonger. Dette er hensiktsmessig då ein slepp å programmere lik funksjonalitet fleire gonger.

Sjå vedlegg for meir dokumentasjon av funksjonsblokkene. (Vedlegg B)

8.2.1 fbTimer

Denne funksjonsblokka gir moglegheit for tidsforsinking. Her kan ein nytte tidsforsinking på, tidsforsinking av, eller ein kombinasjon av begge.

8.2.2 fbAnalougeAlarm

Denne funksjonsblokka nyttast til å overvake, behandle grenseverdiar, gi alarmar og leggje til hysteresse på ferdig skalerte analoge inngangsverdiar. Funksjonsblokka nyttast fleire gonger i MA-blokka.

8.2.3 fbDigitalAlarm

Denne funksjonsblokka nyttast til å overvake og gi alarmar på digitale inngangsverdiar. Det er valbart om blokka skal aktiverast på høg eller låg inngang basert på ein parameter.

8.2.4 fbSwap

Denne funksjonsblokka mottar ein inngangsverdi og rullerar mellom å nytte to utgangsverdiar. Blokka hugsar på kva utgang som vart nytta sist, og vil nytte den andre utgangen ved neste kall. Blokka har også moglegheit for feilhandtering.

8.2.5 fbCalculations

Denne funksjonsblokka utfører nokre rekneoppgåver. Det er hovudsakleg utrekningar av volum i reaktora-
ne, mottakstank og drenert volum i frå reaktortankane.

8.2.6 fbTimeMeter

Denne funksjonsblokka tel tida så lenge den er kalla på, og lagrar verdien i forskjellige tidsformat for driftsdata. Den blir nytta til å telle gangtid for forskjellige elektriske komponentar.

8.2.7 fbHighLoad

Denne funksjonsblokka blir nyttast til å overvake den antatte tilstrøyming på anlegget basert på nivå- endringar i mottakstank. Blokka bereknar gjennomsnittleg tilstrøyming ved å måle tanknivået kvart minutt over totalt 30 minutt. Denne berekninga blir samanlikna med eit parameter for å sette anlegget i høgbelastningsmodus.

8.2.8 fbSivbedRotation

Denne funksjonsblokka nyttast for å styre og rotere mellom anleggets fire sivbedetventilar under uttapping av slam. Parameter angir antall uttappingar per sivbed/ventil. Det er mogleg å ta eit sivbed ut av drift, slik at ventil ikkje blir inkludert i rotasjonen.

8.2.9 fbDataprocessing

Denne funksjonsblokka blir nytta i samband med innsamlinga av driftsdata.

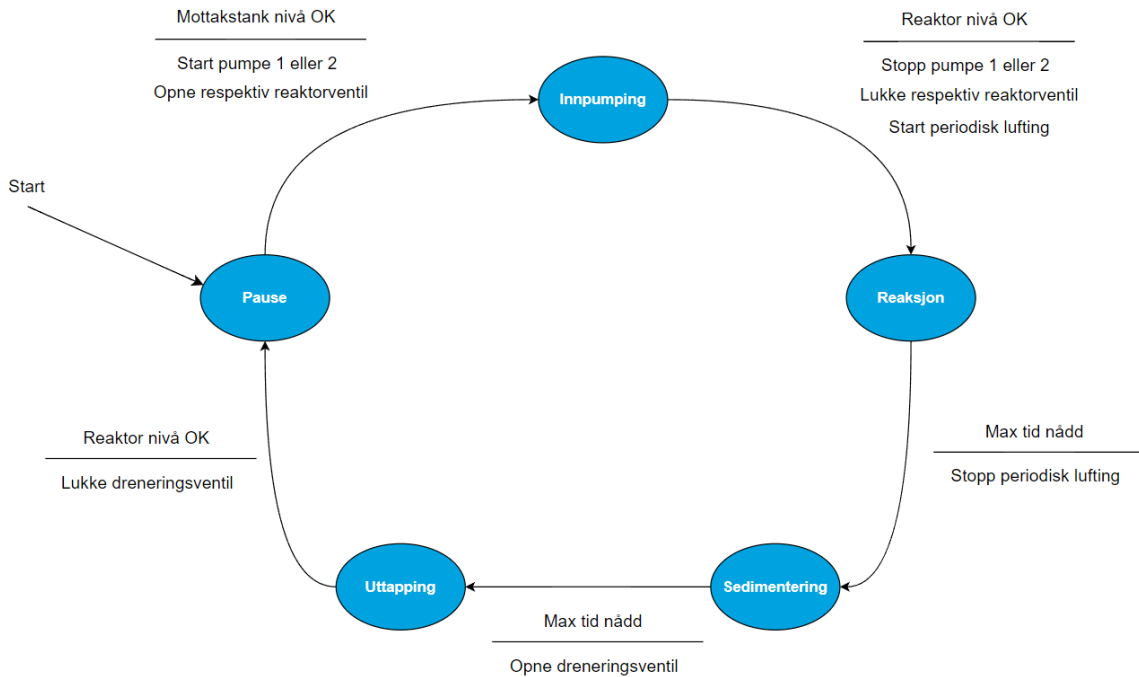
8.2.10 fbProcessedWater

Denne funksjonsblokka nyttast for oversikt over driftsdata for behandla vatn. Blokka gir informasjon i forskjellige tidsformat.

8.3 Tilstandsmaskin

Då IEC-blokkene var ferdige, byrja vi på tilstandsmaskina som skulle styre sjølve SBR-prosessen. Vi hadde allereie danna oss eit bilete, men kunne no byrje å nytte kunnskapen frå anleggets verkemåte til å grovt fylle inn dei hendelsane og aksjonane som foregikk mellom tilstandane. Tilstandsmaskina er bygd opp av dei fem reaktorsekvensane som eksisterar i eit SBR-anlegg.

Dette er ein enkel modell av korleis tilstandsmaskina er programmert, men gir eit godt innblikk i funksjonaliteten.



Figur 8.8: Enkel model av tilstandsmaskin

I sjølve programmeringa av tilstandsmaskina blei den oppretta som ei eiga funksjonsblokk, noko som gav oss moglegheita å nytte blokka for begge reaktorane. Tilstandsmaskina er laga med fem inngangar og seks utgangar, og baserer seg på "switch/case" logikk.

Tilstandsmaskina sender ut høg på den respektive utgangen som samsvarer med reaktortilstanden den er i. Dersom tilstandsmaskina får tilbake høg på den respektive tilstandsinngangen avanserer tilstandsmaskina. Det er også mogleg å hente ut aktiv tilstand ved hjelp av ein heiltallsverdi (1-5).



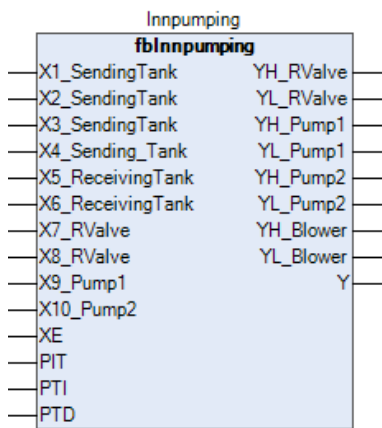
Figur 8.9: Tilstandsmaskin implementert i programmet

8.4 Tilstandslogikk

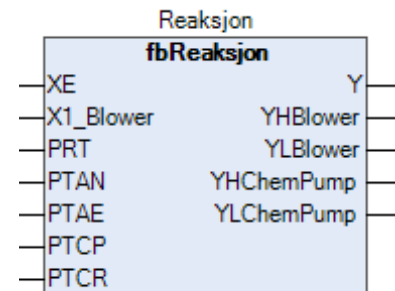
Styring av logikk som skulle skje i kvar tilstand, valde vi å samle i ei funksjonsblokk som vart definert som tilstandslogikk, og fikk navn etter tilstanden den skulle styre t.d. reaksjon.

Kvar tilstandslogikkblokk får inn "external enable" XE frå tilstandsmaskina som startar tilstandslogikken. Når sekvensen er ferdig sender funksjonsblokka høg på utgang Y som returnerast til tilstandsmaskina som avanserer til neste tilstand og tilstandslogikkblokk.

Det er tilstandslogikken som har ansvar for å samarbeide med IEC-blokkene som handterer inngangssignaler start/stopp av elektrisk utstyr og kontrollerer feil og tilbakemeldingar.



(a) Innpumping



(b) Reaksjon

Figur 8.10: Tilstandslogikkblokker implementert i programmet

8.5 Oppbygging av programmet

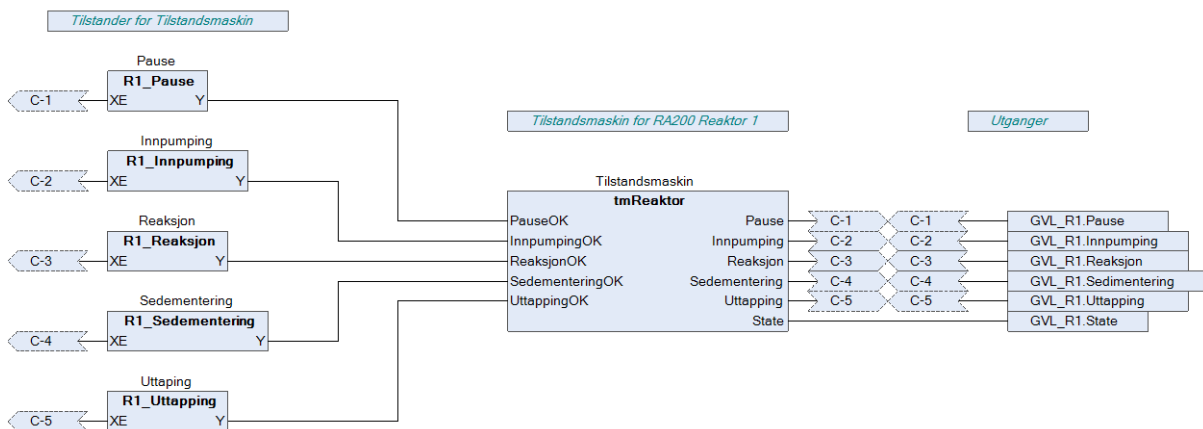
8.5.1 Programmeringsmetode

For å sette saman alle funksjonsblokkene vi hadde programmert i ST, valde vi å nytte Codesys “Continuous Function Chart” (CFC). CFC er eit grafisk programmeringsspråk som nyttar symbol og koplingar for å gjere programmet meir visuelt.

Alle samankoplingar av blokker valde vi å gjere i CFC. Ved å nytte ein grafisk metode sikra vi oss god lesbarheit og visuell forståelse av programmet.

Alle inngangar og utgangar er leselege og enkle og forstå. CFC i lag med god dokumentasjon vil kunne gi personar utan programmeringsbakgrunn god forståelse av korleis programmet er oppbygd, utan å måtte lese kodelinjer. CFC gir eit godt grunnlag for feilsøking og analyse, dette bygger vidare på filosofien med eit enkelt og fleksibelt program.

Dersom antall koplingar og linjer gjorde programmet vanskeleg å lese var det også mogleg å opprette “source” og “links” som oppretta ein trådlaus forbindelse gjennom ein unik ID.



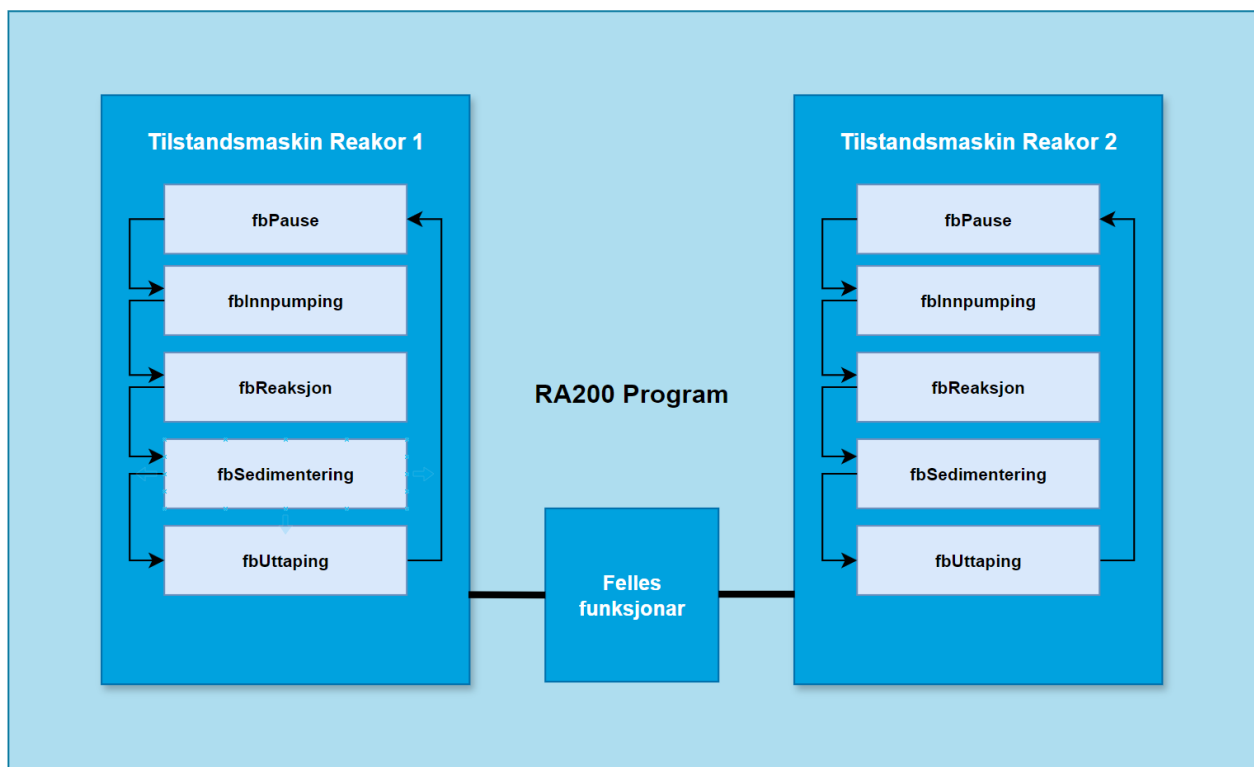
Figur 8.11: Eksempel CFC - Styring reaktor 1

8.5.2 Hovuddel

Programmet er delt opp i tre hovuddelar, ei tilstandsmaskin for kvar reaktor og ein del for samling av felles reaktorfunksjonar. Alle delane blir utført kvar PLS syklus. Tilstandsmaskina har det overordna ansvaret og passar på kva tilstandslogikkblokk som nyttast.

Fellesfunksjonar er ei samling av funksjonsblokker og utrekningar som er felles for reaktorane, og er uavhengig av tilstandsmaskinene. Driftsovervaking er ein sentral del av fellesfunksjonar, der gangtider og mengde prosessert vatn er døme utførte berekningar.

I nokre tilfelle, som ved rulling av sivbed, var vi avhengig at begge tilstandsmaskinene hadde same informasjon. Dette løyste vi ved å lage ei funksjonsblokk "fbSivbedRotation" 8.2.8, i felles funksjonar, som hentar inn og behandlar antall slamuttak for å rotere sivbed når ei gitt grense er nådd. Denne informasjonen blir deretter sendt til kvar tilstandsmaskin som sørger for at begge reaktorane har same aktive sivbed.

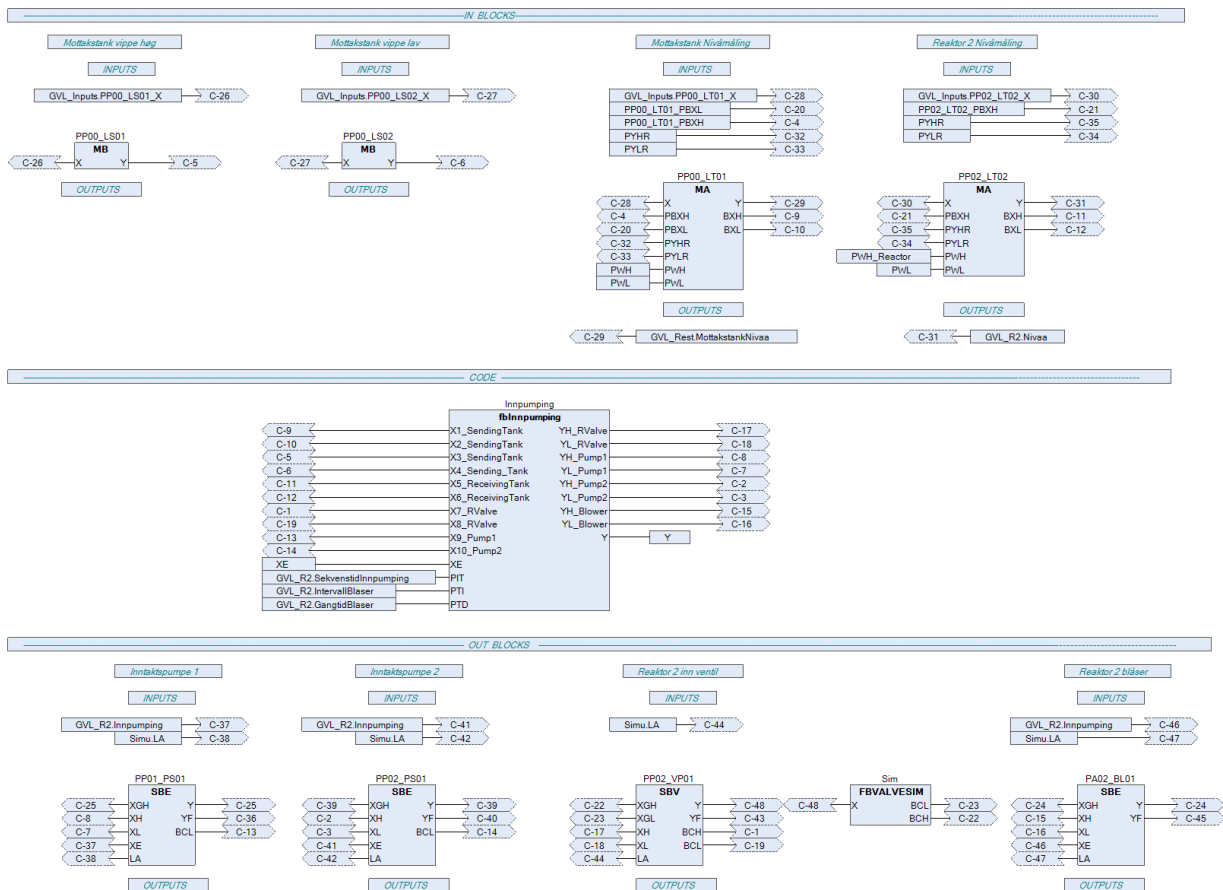


Figur 8.12: Illustrasjon oppbygging program

8.5.3 Styring tilstandslogikk

Som tidlegare skildra er det tilstandslogikken som samarbeider med IEC-blokkene. Dette samarbeidet valde vi også å gjere i eit CFC vindaue som gjorde kall og koplingar meir visuelt. CFC vindauet fikk namn etter kva sekvens i SBR-prosessen den hadde ansvar for å styre.

Oppbygginga av desse sekvensstyringane er gjort med inngangsblokker (MA og MB) øvst og utgangsblokker (SBE og SBV) i botn. Mellom desse kjem tilstandslogikkblokka som inneheld styringslogikk.



Figur 8.13: Eksempel CFC - styring innpumping

I denne figuren er ekstra inngangar, parameterinngangar og utgangar fjerna for å betre kunne visualisere koplingane og samarbeidet mellom IEC-blokkene og tilstandslogikk.

8.6 Alarm og feilhandtering

Alarm og feilhandtering utgjør ein sentral del av eit velfungerande styresystem. Det er avgjerande at anlegget effektivt handterer og varslar om uønska hendingar, slik at driftspersonell blir varsla og naudsynne tiltak kan utførast.

For å varsle om alarm og feil har vi nytta oss av Codesys sine innebygde alarmhandteringsfunksjonar [41]. Desse funksjonane gir oss høve til å kontrollere, gruppere og prioritere alarmer, samt å sende informasjon til driftspersonell.

IEC-blokkene gir oss moglegheit til å detektere ulike typar hendingar som feil, alarm og forvarsel. Dei forskjellige blokkene har ein ulik mengde hendingar som skal kunne detekterast og varslast. Alle desse moglege varslingane er samla i programmet saman med dei aktive varslingane som er i bruk på Sande reinseanlegg i dag. (Vedlegg E)

Vi har valt å dele opp alle varslingane i fire forskjellige grupper:

- **Feil** (Der system ikkje fungerer, t.d. sensorfeil og blokkfeil)
- **Alarm** (Kritiske prosessparametrar, t.d. straumbrot og veldig høge nivå)
- **Forvarsel** (Prosessparametrar nærmar seg kritisk nivå)
- **Informasjon** (Prosessinformasjon med nytteverdi)

Ved å kategorisere varslingar i fire ulike grupper med forskjellige prioriteringsnivå, vil system og driftspersonell enklare kunne forstå samanhengen og alvoret i varslingane.

Korleis dei ulike alarmane blir definert og prioriterte må evaluerast i samråd med Sunnfjord kommune.

	Timestamp	Message
0	15.05.2024 12:02:57	Feil blåser reaktor 2, mangla tilbakemelding lav
1	15.05.2024 12:02:31	Feil blåser reaktor 1, tap av aktivering
2	15.05.2024 12:01:32	Feil blåser reaktor 2, tap av aktivering
3	15.05.2024 12:01:26	Høgt nivå reaktor 2
4	15.05.2024 12:01:26	Høgt nivå reaktor 1
5	15.05.2024 12:01:26	Høgt nivå mottakstankl

Figur 8.14: Varslingar implementert i simulering

8.7 utfordringar

8.7.1 Utgangsblokker med signalkonflikt

Nokre komponentar skulle nyttast i fleire tilstandar og vi opplevde utfordringar med å ha fleire utgangsblokker for ein komponent. Orginalt var dette løyst med fleire kall av ei utgangsblokk i forskjellige CFC vindauge. Kalla av dei ulike blokkene skreiv forskjellige signal til komponenten, og sendte høg signal blei kontinuerleg overskreve.

Vi møtte denne utfordringa fleire plassar, t.d. i pumpestyringa, der kvar reaktor skulle kunne styre den same pumpa. Dette løyste vi ved å nytte ein unik utgangsvariabel for kvar blokk, og deretter skrive til ei ny blokk som samla signala til komponenten og satt utgangen til riktig verdi.

8.7.2 Mangel på sensorikk

Anlegget har begrensa mengde sensorikk. Dette har ført til at vi må berekne, estimere og programmere rundt denne mangelen.

Som døme har anlegget ein funksjon for å aktivere høgbelastningsmodus ved høg tilstrøyming, men anlegget har ingen form for strøymningsmålar.

Grunna dette har vi vore nøydd til å kalkulere ein teoretisk tilstrøyming basert på endringa av volumet i mottakstanken.

Sjølv om slike løysningar kan fungere, er det ikkje optimalt for anleggets drift. Estimering av prosessverdiar vil redusere nøyaktigheit i styresystemet og unødvendig prosessorkraft vil bli nytta på berekningar som enkelt kunne vore erstatta av ein sensor.

Det det også krevd ekstra tid å programmere desse funksjonane.

9 Dokumentasjon

I dette kapittelet tar vi for oss og samlar dokumentasjon av programmet som er utarbeidd.

9.1 Dokumentasjon av funksjonsblokker

Alle funksjonar og funksjonsblokker som er utarbeida i programmet, har sitt eige dokumentasjonsdokument. Dette dokumentet beskriv blant anna bruk, funksjonalitet, inngangar, utgangar og parametrar. Dokumentet inneheld versjonshistorikk som skal oppdaterast ved endring. Lister over omgrep, forkortingar og programmeringsbiblioteker som er naudsynt for å ta blokkene i bruk står i dokumentet. Dokumentet er utvikla etter eigen mal og er standardisert for blokkene som er inkludert i programmet.

Dokumentasjon til alle funksjonar og funksjonsblokker er tilgjengeleg via vedlegg. (Vedlegg B)

9.2 Forrigling

For å unngå utilsikta situasjonar har vi implementert forriglingar. Desse forriglingane er inkludert i tilstandslogikkblokkene.

IEC-blokkene tilbyr også funksjonalitet som gjer det mogleg å handheve forriglingar mellom komponentar.

Det ligg forrigling på styring av matepumpene til reaktorane. Det skal ikkje være mogleg at begge pumpene går samtidig og beskyttelse for dette er handtert i funksjonsblokka "fbSwap"(8.2.4).

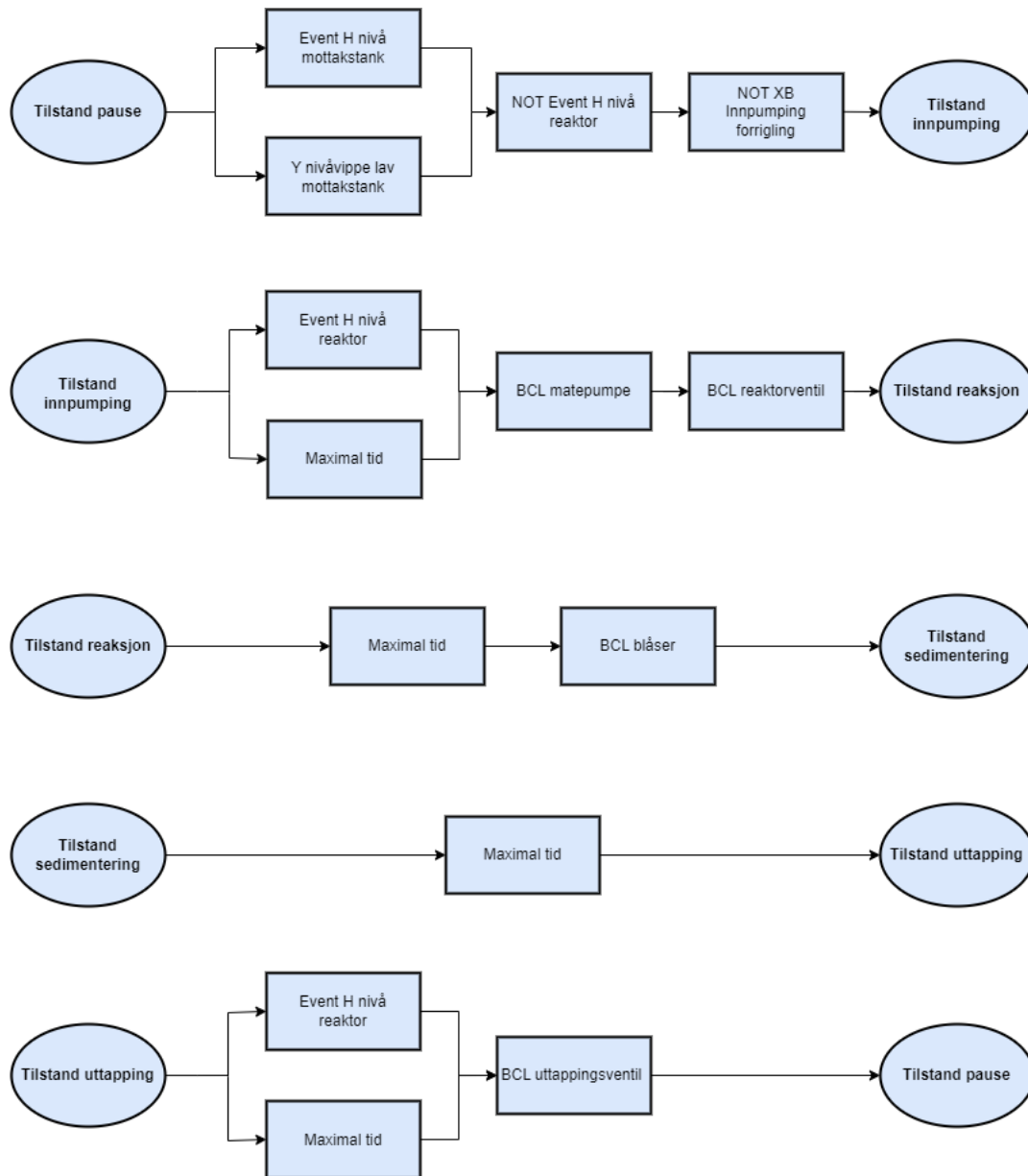
Det skal heller ikkje førekomme at begge reaktorane er i tilstand innpumping samstundes. Sidan tilstandsmaskinene som styrer kvar reaktor er uavhengige av kvarandre, måtte vi løyse dette med ein global variabel. Denne variabelen blokkerer overgangen frå pause til innpumping dersom ein reaktor allereie er i innpumping.

Programmet har overordna kontroll med at tilstandsmaskinene gir signal til XE, slik at komponentane som ikkje er i bruk ikkje er tilgjengeleg.

9.3 Tilstandsovergangsbetingelsar

For at tilstandsmaskina skal få lov til å endre tilstand, må nokre spesefikke betingelsar vere oppfylt. Fleire av betingelsane er tidsstyrte, og alle desse tidene er justerbare via parameter.

For å kunne gi ei klarare framstilling av logikken har vi valt å presentere betingelsane grafisk. Vi har utarbeidd eit skjema som dokumenterer alle overgangsbetingelsane i tilstandsmaskina.



Figur 9.1: Grafisk presentasjon av overgangsbetingelsar

9.4 IO-liste

Vi har utarbeidd IO-liste basert på tidlegare styresystem.

Vår oppgåva er teoretisk og vi har programmert med omsyn til utvidning av ny sensorikk, men har valt å ikkje leggje til noko meir sensorikk i basis programmet.

Dette medfører at I/O-liste ikkje har endringar.

I/O-liste ligger som vedlegg. (Vedlegg J)

9.5 Objektliste

Grunna arbeidet rundt utforming av P&ID, valde vi å lage ei liste over alle komponentane i anlegget som var tilkopa og styrt av styresystemet. Objektlista inneheld tagg, komponentbeskrivelse og plassering.

I samband med innhenting av informasjon til dei ulike objekta har vi etablert ein database med datablad. Databasen inneheld relevant informasjon om kvar enkelt komponent i styresystemet.

Desse datablada er tilgjengeleg i vedlegg. (vedlegg I).

I PLS-programmet har vi nytta tagg for å namngi komponentane sine blokker.

Objektlista gir høve til å knytte korrekt tagg til riktig komponent.

ID	Objekt	Plassering	Tag
1	Grovrisk	Forbehandlingsrom	Rotomat R9
2	Nivåvippe lav mottakstank	Nedsenka i mottakstank	PP00-LS02
3	Nivåvippe høg mottakstank	Nedsenka i mottakstank	PP00-LS01
4	Nivåmålar (trykk) mottakstank	Nedsenka i mottakstank	PP00-LT01
5	Overløpsmålar (trykk)	Mottakstank (utløpskasse)	PP00-LT02
6	Matepumpe 1 (vest)	Kjeller	PP01-PS01
7	Matepumpe 2 (aust)	Kjeller	PP02-PS01
8	Inntaktsventil reaktor 1 (vest)	Kjeller	PP01-VP01
9	Inntaktsventil reaktor 2 (aust)	Kjeller	PP01-VP02
10	Blåser reaktor 1	Kjeller	PA01-BA01
11	Blåser reaktor 2	Kjeller	PA02-BL01
12	Pressostat prosessluft reaktor 1	Kjeller	PA01-PR01
13	Pressostat prosessluft reaktor 2	Kjeller	PA02-PR01
14	Nivåmålar (trykk) reaktor 1	Reaktor rom, på reaktor 1	PP01-LT01
15	Nivåmålar (trykk) reaktor 2	Reaktor rom, på reaktor 2	PP02-LT02
16	Doseringspumpe reaktor 1	Reaktor rom	CH00-PH01
17	Doseringspumpe reaktor 2	Reaktor rom	CH00-PH02
18	Uttapventil reaktor 1 (vest)	Reaktor rom, på reaktor 1	TW01-VP01
19	Uttapventil reaktor 2 (aust)	Reaktor rom, på reaktor 2	TW02-VP02
20	Slamventil reaktor 1 (vest)	Kjeller	PS01-VP01
21	Slamventil reaktor 1 (aust)	Kjeller	PS02-VP01
22	Sivbedventil celle 1	Kjeller	PS00-VP01
23	Sivbedventil celle 2	Kjeller	PS00-VP02
24	Sivbedventil celle 3	Kjeller	PS00-VP03
25	Sivbedventil celle 4	Kjeller	PS00-VP04
26	Dreneringspumpe pumpehus	Pumpehus	PP03-PS01
27	Returpumpe pumpehus	Pumpehus	PP03-PS02
28	Nivåvippe drenering lav	Pumpehus	PP03-LT01
29	Nivåvippe retur høg	Pumpehus	PP03-LT02
30	Nivåvippe drenering lav	Pumpehus	PP03-LT03
31	Nivåvippe retur høg	Pumpehus	PP03-LT04
32	Vann på gulv sensor	Kjeller	PS01
33	Pressostat styreluft	Kjeller	IA01-PR01

Figur 9.2: Objektliste

9.6 SCD

SCD utgjør grunnlaget for det meste av programmeringsdokumentasjonen i oppgåva. Diagrammet er delt opp sidevis og sekvensvis og viser styring mellom program og komponentar i kvar enkelt sekvens, samt ei side for resterande fellesstyring.

Prosessen i SCDen (blå linjer) er basert på P&ID som blei laga etter gjennomgangen av anlegget i kapittel 6. SCD tar omsyn til programmerbart utstyr og har derfor ikkje med manuelle ventilar, tilbakeslagsventilar eller liknande.

For å representere eigne funksjonsblokker var vi nøydd å leggje dei inn i SCDen. SCD verktøyet hadde moglegheit for å leggje til eigendefinerte blokker med respektive inngangar og utgangar.

Funksjonsblokker og IEC funksjonstemplata nytta i SCD er dei same som vi har laga og nytta i programmeringsdelen. Dei stipla linjene viser koplingar mellom blokker og komponentar.

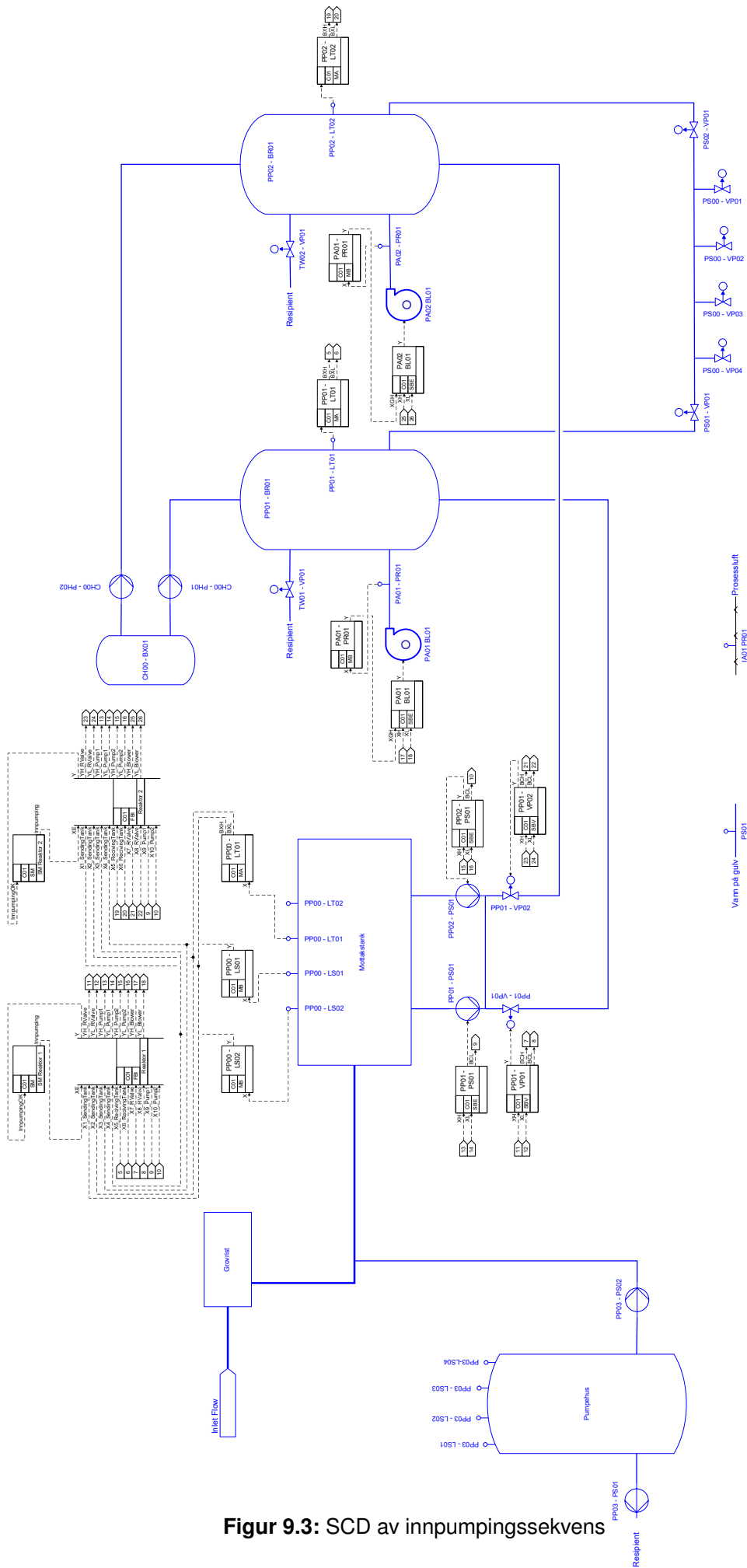
I enkelte diagram førte antal linjer til at diagrammet vart vanskeleg å lese. Vi har derfor nytta koplingar med unik ID, liknande som i CFC vindauget.

Tilstandslogikkblokker har fått eigne forkortingar i SCDen. Dette er dei relevante forkortingane for å forstå blokkene og diagrammet.

- **SM:** Tilstandsmaskin
- **FBP:** Funksjonsblokk pause
- **FBI:** Funksjonsblokk innpumping
- **FBR:** Funksjonsblokk reaksjon
- **FBS:** Funksjonsblokk sedimentering
- **FBU:** Funksjonsblokk uttapping
- **FBPH:** Funksjonsblokk pumpehus

Heile SCD er tilgjengelig i vedlegg. (Vedlegg H).

Tilstand Innpumping



Figur 9.3: SCD av innpumpingssekvens

10 Simulering og verifisering

Simulering er ein sentral del i utvikling av eit styresystem. Det gir oss moglegheita til å verifisere at programmet møter kravspesifikasjon og har ønskja effekt. Testing og simulering bidreg til å redusere risiko for feil, forbetre kvalitet og auke pålitelegheita til programmet.

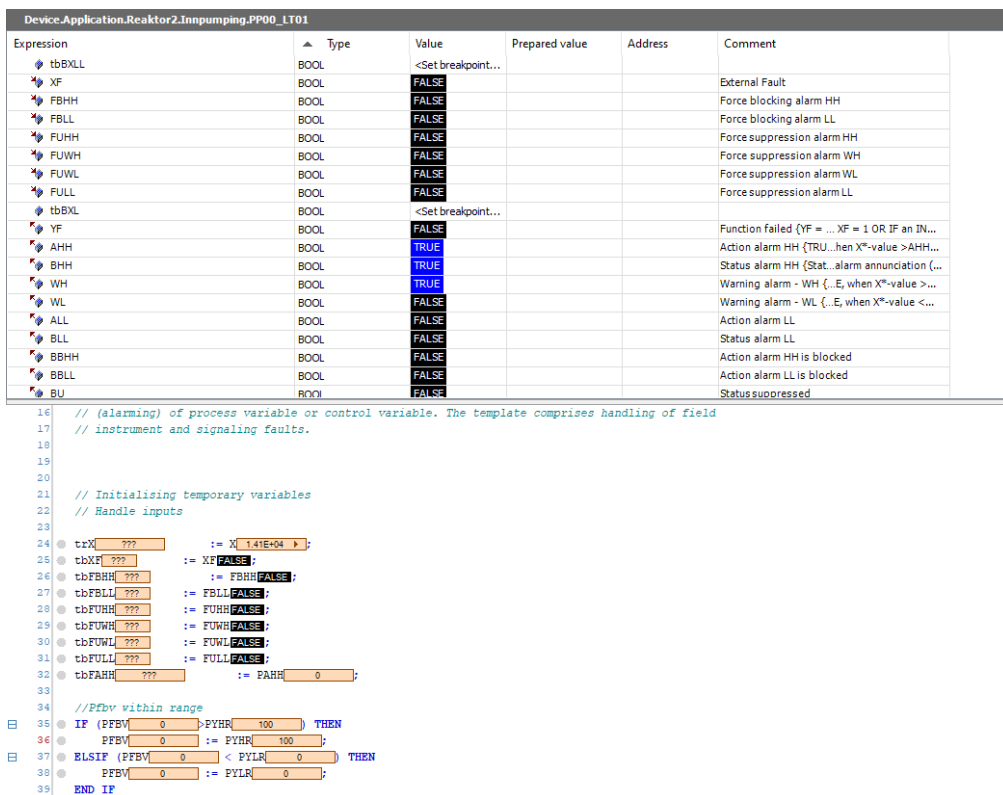
10.1 Kontinuerleg simulering

Parallelt med programmeringa har vi utført kontinuerlege testar og simuleringar av blokkene vi har utvikla. Dette har vært ein viktig del av vår arbeidsmetode.

Kontinuerleg simulering er små testar som blir utført medan ein programmerar. Det er ikkje ein isolert testfase med klare start- og stoppunkt, men heller små testar som gir ein indikasjon på om arbeidet følgjer krav.

I denne oppgåva nytta vi kontinuerleg simulering i programmering av alle blokker. IEC-blokkene hadde mange funksjonar som ikkje var avhengige av kvarandre, som gjorde det mogleg å gjennomføre små, enkle testar utan at heile blokka var ferdigstilt.

Som eit døme, skal RX resete ein utgang Y. Dette blir implementert og deretter testa ved hjelp av simulering. Kva som til slutt skal sette utgangen Y høg er ikkje relevant i dette tilfellet. Vi har implementert og testa at ein reset vil fungere.



Figur 10.1: Kontinuerleg testing ved manipulasjon av verdiar

10.2 Simuleringsblokker

For å skape eit realistisk miljø for testing, lagde vi nokre simuleringsblokker. Desse blokkene etterliknar driftssituasjonar og gjer simulering enklare og meir effektivt. Vi lagde hovudsakleg to slike blokker, ei simulerer fylling av tank og ei for tilbakemelding frå ventilar.

Blokk for tanksimulering er enkel og programmert spesifikt for reinseanlegget, som har ein mottakstank og to reaktorar. Det er lite truleg at desse simuleringsblokkene vil bli nytta vidare, men dei gav oss den realismen vi trengte.

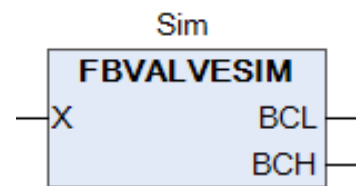
Desse blokkene forenkla også sjølve arbeidet med simuleringa. Før vi utvikla blokka for tilbakemelding på ventilar, gav vi manuelt kvar ventil XGH eller XGL basert på den tiltenkte stillinga (open/stengd).

Dette fungerer greitt dersom ein testar ei ventilblokk, men når simuleringa omfattar større delar av programmet blir denne jobben tidkrevjande.

På desse blokkene har vi tatt oss meir friheit i namngiving og feilhandtering, sidan blokkene ikkje skal nyttast når programmet er ferdigstilt.



(a) Tank



(b) Ventil

Figur 10.2: Simuleringsblokker

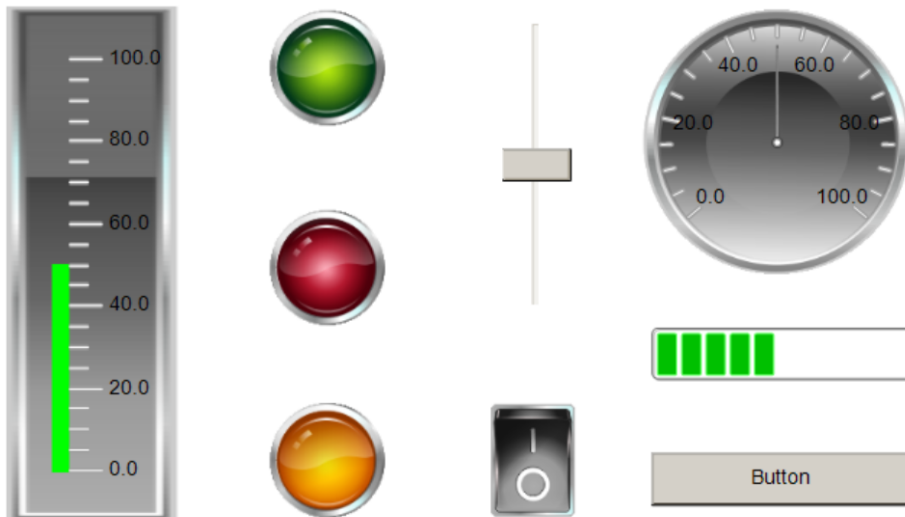
10.3 Simuleringsvindaage

Når ein simulerer eit program kan det bli mange variablar og komponentar å halde styr på. Derfor er det viktig å sortere og behalde relevant informasjon, samt å kunne presentere den på ein god måte.

Codesys Visualization [42] er eit grafisk verktøy der ein kan presentere informasjon ved hjelp av grafiske element. Dette gav oss moglegheit til å presentere tankar, ventilar og pumper ved hjelp av symbol.

Vi nytta visualiseringsverktøyet til å lage eit fullskala simuleringsvindaage, der kvar komponent vart knytt til sitt spesifikke grafiske symbol. Vi knytta vindaaget saman med programmet og kopla relevante signal. Dette gjorde at vi hadde eit grensesnitt mot programmet og kunne enkelt simulere forskjellige driftssituasjonar og konkludere med resultat.

Simuleringsvindaaget er ikkje noko HMI og tar ikkje omsyn til aktuelle normer og krav.



Figur 10.3: Eksempel Codesys visualiseringselement

10.4 Fullskala simulering

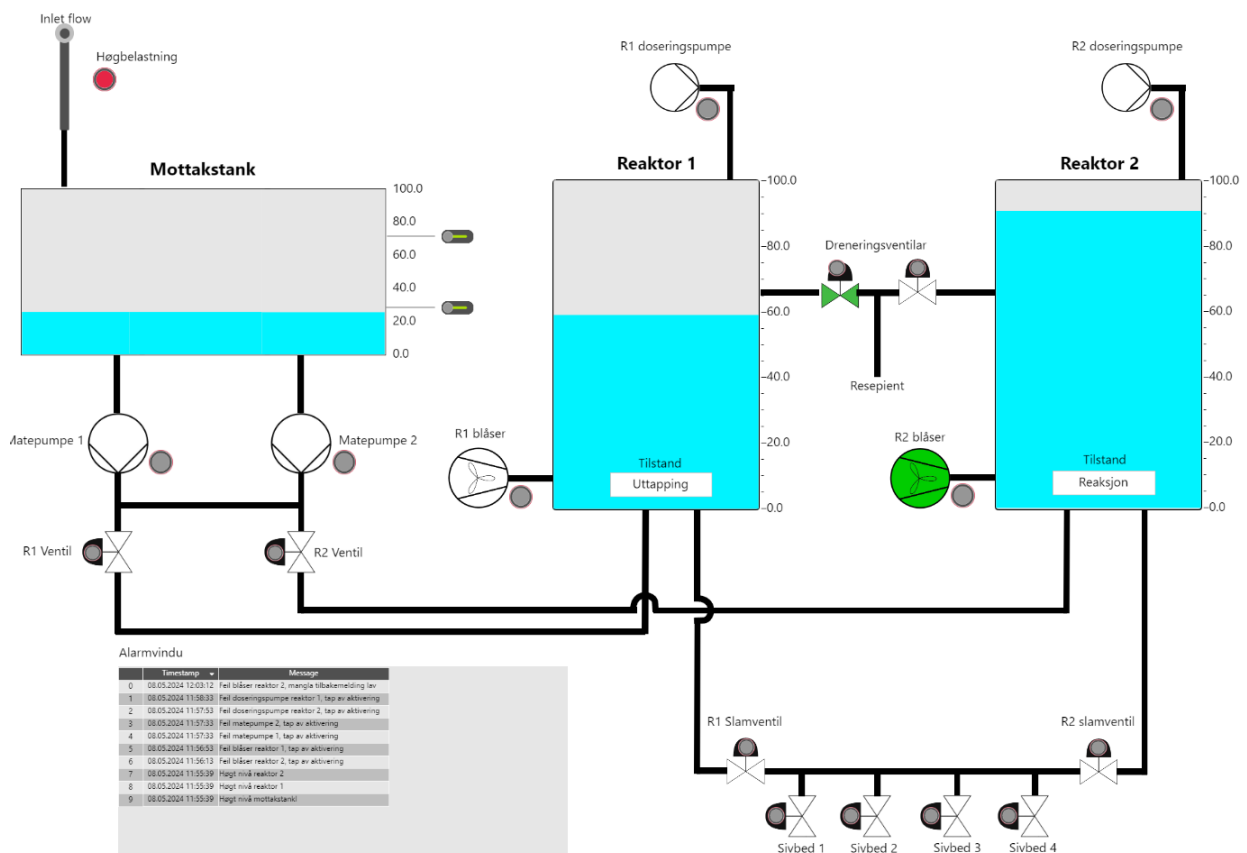
10.4.1 Oppsett

Med simuleringsblokkene klare og eit simuleringsvindaue til å vise resultat, var vi no klare for simulering av reinseanlegget. Vi implementerte ventil-simuleringsblokkja på alle ventilane i programmet og oppretta eit nytt CFC vindaue som koplta tanksimuleringa av mottakstanken og reaktorane saman med resten av programmet.

Simuleringa er sett opp slik at tilstrøyinga til anlegget kan justerast via ein glidebrytar. Dersom ein av reaktorane er i innpumpingssekvens, simulerer vi flytting av avlaupsvatn basert på pumpekurver og løftehøgde.

Reaktortilstand er presentert med ein tekstvariabel, og dei aktuelle tidsparameterane, som t.d. tid for reaksjonssekvens, er redusert for å akselerere simuleringa.

Vi har oppretta eit varslingsystem for alle moglege varslingar anlegget kan generere, uavhengig om dei er relevante for sluttprogrammet. Dette gir oss verdifull innsikt i eventuelle feil som oppstår og korleis programmet verkar.



Figur 10.4: Oppsett av simuleringsvindaue

10.4.2 Resultat

Som forventet dukka det opp avvik i simuleringa. Nokre av desse var av mindre betydning og blei utbetra undervegs, som t.d. skrivefeil, gløynde logiske operasjonar, CFC-koplingar som var feilplassert eller konfigurasjonsfeil.

Andre avvik krevde meir arbeid og endring av program. Den tiltenkte reaktorforriglinga som skulle hindre overgang til innpumpingssekvens, fungere ikkje. Her var det naudsynt med ein ny løysning.

Simuleringa av programmet var vellykka og vi fekk samla verdifull informasjon. Den visuelle representasjonen var oversiktleg, der ein kunne sjå programmet i drift og enkelt oppdage feil.

Vi konkluderte at programmet i hovudsak fungerer som planlagt. Programmet er no klar for ein meir omfattande simulering, der ein implementerer resterande funksjonar og fullfører fullskala test.

Simuleringsvideo er tilgjengeleg via Youtube.

Link: https://youtu.be/_2gtiXJDUyw

11 Diskusjon

11.1 Veggen vidare for anlegget

Sjølv om bacheloroppgåva vår avsluttast har vi fått ein ny fascinasjon for ein sjult sektor innan offentleg infrastruktur.

Uansett kva Sunnfjord kommune vel å gjere med det nye styresystemet, har vi utført eit dokumentasjonsarbeid som vil gi stor verdi for reinseanlegget. Arbeidet inkluderer nye tekniske beskrivelsar med korrekte komponenttaggar, utvikling av P&ID og informative figurar, samling av diverse informasjon til eitt dokument, samt oppretting av ein komponentliste med datablad.

Dette var ein viktig del av vår læring om korleis reinseanlegget fungerer, og dokumentasjonsarbeidet vil hjelpe andre med å oppnå same forståing med mindre innsats

Vi ønskjer å diskutere veggen vidare for anlegget og eventuelle oppgraderingar som kan gjerast for å optimalisere prosessen.

11.1.1 Ombygging

Dersom den teoretiske bacheloroppgåva skal realiserast i praksis, treng anlegget fysiske oppgraderingar. Reinseanlegget tilfredsstillar ikkje fleire relevante lover og forskrifter innanfor industri og offentleg infrastruktur. Mykje av denne delen ligg utanfor vårt fokusområde, men reinseanlegget har t.d. ingen handtering eller moglegheiter for naudstopp. Det er viktig at programmet ikkje blir sett i drift utan at anlegget går gjennom ein grundig sikkerheitsanalyse.



Figur 11.1: Første gjennomgangen av anlegget

Foto: Håvar Dankel

11.1.2 Anbefalingar sensorikk

For å forbetre anlegget og styresystemet ytterlegare, vil det vere nyttig å få inn meir instrumentering. Sensorikk vil forbetre styresystemet ved å samle meir og nøyaktig data om tilstand og ytelse. Med auka datamengder vil styresystemet kunne operere meir automatisk og vil kunne tilpasse seg varierende forhold meir effektivt. Systemet vil og ha høve til å oppdage avvik og reagere tidlegare utan manuelle inngrep, som ytterlegare aukar automasjonsevna.

Basert på vår nyerfarte kunnskap om anlegget og styresystemet, presenterer vi ei anbefaling for oppgradering av instrumentering. Sjølv om all sensorikk vil være verdifull, har vi vurdert kost-nytte og anbefaler kun sensorikk som gir tilstrekkeleg verdi.

- **Strøymningsmålar**

Ein strøymningsmålar vil gi nøyaktige mål på flyt av vatn. Dette vil forbetre kontroll på aktivering av høgbelastningsmodus, samt kontroll over, og rapportering av driftsdata. Fleire strøymningsmålarar er moglege men minimumsanbefalinga er på vatn inn og ut av anlegget.

- **Energimåling**

Energimåling vil gi moglegheit for å analysere energiforbruket for å redusere kostnader og effektivisere prosessar. Energimåling spesifikt for komponentar kan også nyttast til overvaking av utstyr for å oppdage slitasje og feil.

- **Reaktormålingar**

Målingar av oksygen, pH og temperatur er kritiske for å oppnå god biologisk reinsing i reaktorane [43].

Ved å ha kontroll på desse parameterane vil reaktor kunne finjusterast for å effektivisere den biologiske reinseprosessen.

- **Tilbakemeldingar**

Anlegget har begrensa tilbakemeldingar frå utstyr, spesielt innan ventilstyring. Tilbakemeldingar er essensielt for prosesstyring, feiloppdaging og sikkerheit. Utan tilbakemeldingar er reinseanlegget meir sårbart for feil som ikkje blir oppdaga, og slike situasjonar vil kunne resultere i nedetid, overløp og andre utilsikta hendelser.

- **Forbetra nivåmåling**

Forbetring av eksisterande nivåmålingar vil gi betre mål på slammengde og slamnivå i reaktorane. Dette vil gjere det mogleg å nøyaktig finne skilje mellom slam og reinsa vatn etter ein sedimenteringssekvens.

- **Oppløysning**

Oppløysninga på analoge målingar er idag kun 0-1000 bit på 4-20mA. (Vedlegg I.12)

Oppgradering av måleoppløysning vil gjere kvar eksisterande og nye analoge målingar meir nøyaktig, som igjen gir moglegheit for betre styring og regulering.

11.2 Vegene vidare for programmet

Programmeringsarbeidet hadde eit så stort omfang at vi ikkje klarte å ferdigstille prosjektet. Etter å ha innsett at vi ikkje ville nå målet, måtte vi gjere nokre val angående prioritering av ressursar. Vi valde å fokusere på å gjennomføre ei god samla simulering i staden for å arbeide vidare med ferdigstilling av funksjonar.

11.2.1 Programmering

Vegene vidare for programmet er ferdigstilling av funksjonane som ikkje er komplette, og implementering av funksjonar som er oppretta men ikkje tatt i bruk. Det gjenstår arbeid på feilhandtering og korleis anlegget skal reagere på spesifikke feilsituasjonar.

Her er punkter vi ønskjer å arbeide vidare med.

- Styring av pumpehuset er forberedt men ikkje programmert.
- Ferdigstilling av høgbelastningsmodus.
- Spesifikk feilhandtering

11.2.2 Simulering

Det gjenstår framleis noko simulering og testing før ein kan ansjå programmet ferdig verifisert. Til dømes er funksjonsblokkene for slamuttapping og sivbedrotasjon testa separat, men ikkje simulert saman med tilstandsmaskinene og resterande program.

Under testing av eit slikt program bør ein aktivt leite og prøve å identifisere feil og manglar for å gjere programmet best mogleg. For blokkparameterer manglar eksempelvis nokre beskyttelsar mot ulovlege parameterverdiar.

Ved å investere meir tid i vidare simuleringsfase sikrar vi at programmet oppfyller forventningane til Sunnfjord kommune og at ein eventuell praktisk innkøyringsfase går effektivt.

Programmet er no sett opp i simuleringsmodus, med simuleringsblokker og endra parameterverdiar, og det må tilbakestillaast til original tilstand før det kan nyttast.

11.3 Måloppnåing

11.3.1 Krav

Når ein tar eit tilbakeblikk på kravspesifikasjonen skildra under kapittel 5 sit vi med ein god følelse. Krava som vart godkjend saman med oppdragsgivar var relativt opne. Mykje av grunnen til dette var at vi etterlyste oppgåva frå Renasys og at det ikkje var planlagt noko kravspesifikasjon i forkant.

Oppgåva hadde dermed ingen ferdigdefinerte krav eller mållinjer, som gjorde at vi kunne definere oppgåva slik vi ville.

Det er og eit faktum at vi ikkje kom i mål med alt arbeidet og at enkelte detaljer av dokumentasjon, program og simulering kunne vore betre med meir tid. Vi føler uansett at vi har svart på dei punkt som var essensielle og levert over det som var forventa.

11.3.2 Mål

Våre personlege mål under oppgåva handla om å lære nye program og auke eigen kompetanse. Under dette prosjektet har vi lært mykje nytt. Programmeringsdelen av oppgåva har bydd på læring i nye program, samt pugging og lesing av standard og normer. Alle delane av oppgåva har bydd på utfordringar og meistring, og når vi ser tilbake personlege mål, er vi godt nøgde.

11.3.3 Forprosjekt

Ein viktig del av slutten på eit prosjekt er å sjå tilbake på starten. Klarte vi å legge ein god plan og klarte vi å følgje planen vi la? Dette er sentrale spørsmål ein er nøydd å ta stilling til dersom ein ønsker å lære mest mogeleg.

Når vi samanliknar forprosjektet og bacheloroppgåva, ser vi god gjenspegling. I løpet av arbeidsprosessen og etterkvart som vi fekk meir kunnskap, var det naturleg å gjere nokre justeringar og tilpassingar. Likevel kjenner vi at forprosjektet la ein solid plan for oss, og at vi har følgd denne planen i arbeidet på bacheloroppgåva.

Rapport for forprosjekt er tilgjengeleg via vedlegg. (Vedlegg K)

11.4 Lærdom og forbedringspotensial

Oppgåva har bydd på ei bratt læringskurve og når vi ser tilbake på arbeidet ser vi fleire moglegheiter til forbedring.

Etter læring og dokumentering av anlegget var vi ivrige på å byrje med programmering. Vi valde å starte fortløpande og så lære litt undervegs. Dette gjorde at vi på fleire plasser var nøydd til å ta ting oppatt grunna at vi hadde misforstått standardane eller kvarande. I ettertid innser vi at vi burde ha lest meir om IEC-blokkene og tatt oss betre tid før vi byrja å programmere.

I etterkant av programmeringa ser vi at vi kunne ha utnytta funksjonaliteten til IEC-blokkene betre. Ved full bruk av alle dei funksjonane vi implementerte, kunne vi potensielt ha forenkla tilstandslogikken til berre eit par logiske setningar, og kanskje fjerna den i sin heilheit.

Ved optimal bruk av IEC-blokkene kunne programmet vore betre og meir effektivt. Dersom det skulle vere naudsynt med nyprogrammering eller endringar, ville vi ha gjort ting anleis og utnytta all funksjonalitet. Blokkene kunne bidratt til betre handtering av forriglingar, ved t.d. bruk av FSL, FSH, FDH, og FDL inngangar.

Undersøkt forbedring av varslings og generell alarmreduksjon ved bruk av "blocking" og "suppression" for komponentar som ikkje er relevante i gjeldande tilstand. Vi ville også ha implementert ny bruk av inngang XE på SBE t.d. for termiske motorvern, og nytta aktivering frå tilstandsmaskin på ein anna måte.

Arbeidet med namngiving på inngangar, utgangar og parameter hadde også moglegheit for forbedring. I starten var vi flinke å sette av den naudsynte tida for å få namn korrekt, men ettersom tida blei knappare utover i oppgåva blei namngivinga dårlegare. Dette gjorde at vi var nøydd å ta alle desse namna oppatt. Vi skulle også sett oss ned i forkant å diskutert kva språk kommentarar, namn og forklaringar skulle vere i programmet.

Vi ser også at vi moglegvis sette oss litt for detaljert inn i reinseanleggets verkemåte. Sjølv om denne delen av oppgåva var vesentleg, ser vi no at noko av arbeidet vi nytta tid på kanskje ikkje var naudsynt.

Gjennom denne bacheloroppgåva har vi henta inn mykje ny kunnskap og erfaring. Vi har blandt anna lært og utdjupa oss innan:

- Offentleg infrastruktur
- Kundekommunikasjon
- Programmering av større styresystem i PLS
- Programmering av PLS i Codesys
- Standardane IEC 61131-3 og IEC PAS 63131
- GitHub intergrering
- Rapportskriving i LaTeX

12 Konklusjon

Gjennom denne bacheloroppgåva har vi oppnådd ein djupare forståing av korleis Sande avlaupsreinseanlegg fungerer og kva utfordringar dei står ovanfor. Vi har utvikla eit nytt styresystem som sikrar at reinseanlegget følgjer moderne programmeringsstandardar. Sjølv om tilnærminga vår har vore teoretisk, har vi lagt eit solid grunnlag for framtidig praktisk implementering.

Ved å velje løysingsalternativ C, har vi unngått å overføre eksisterande feil og har kunne utforske og implementere betre løysingar. Dette har ført til eit fleksibelt og teknisk moderne styresystem som vil kunne vera til god nytte for Sunnfjord kommune.

Sjølv om vi ikkje har gjennomført ein fullskala test, har simuleringane vist at systemet fungerer som forventa. Vi anbefaler at ytterlegare funksjonalitet og testing blir gjennomført før systemet eventuelt blir tatt i bruk.

Uansett kva Sunnfjord kommune skulle velje å gjere med det nye styresystemet, har dei fått ein ny dokumentasjonspakke som samlar viktig informasjon på ein strukturert måte.

Bacheloroppgåva har gitt Renasys og Sunnfjord kommune tilgang til funksjonsblokker som kan nyttast i ulike PLS-styresystem. Gjennom å nytte IEC PAS 63131-standardten har vi oppnådd ei løysing som gir oppdragsgivar fleksibilitet og gode verktøy for framtidige behov.

Til slutt har prosjektet understreka kor viktig samarbeid, kommunikasjon og nøye planlegging er. Vi er taknemlege for moglegheita til å bidra til eit så viktig område som offentleg infrastruktur, og ser fram til å sjå kva arbeidet kan føre til.

Ordliste

Term	Forklaring
batch	Ein gitt mengde av noko
CFC	Continuous Function Chart
Codesys	Ein programvareplattform for industriell automasjonsteknologi.
diffuser	Redskap for å spre strøymen av eit stoff
GitHub	Dette er ei teneste for synkronisering av filer
global variabel	Ein variabel som er tilgjengeleg i heile programmet
HMI	'Human Machine Interface' er eit grensesnitt mellom menneske og maskin.
HVL	Høgskulen på Vestlandet
hygienisering	Behandling som har som hovudmål å redusere faren for overføring av smittestoff til menneske, dyr og plantar ved disponering eller anna handtering av slam
I/O	Eit begrep som er brukt for ein felles betegnelse for inngangar og utgangar i ein pls
IEC	International Electrotechnical Commission
LaTeX	LaTeX er eit høgnivå-typsettingssystem brukt for å lage vitenskaplege og tekniske dokument
LD	Ladder Diagram
MA	Monitor Analogue
MB	Monitor Binary
NEK	Norsk Elektrotekniske Komité
P&ID	'Piping and instrumet diagram'
PAS	'Public available specification,' og har som mål av IEC å framskynde standardiseringa innan områder som har rask utviklande teknologi.
PLS	PLS står for programmerbar logisk styring og er namnet på ein type eining ein kan bruke til å styre elektriske og automatiske system
RA	Reinseanlegg
Renasys	Oppdragsgivar
SBE	Switch Binary Electrical
SBR	Sekvensiell batchreaktor
SBV	Switch Binary Value
SCD	'System controll diagram', og er ein metode for å spesifisere og dokumnetere kontrollapplikasjoner for styresystemer.
SFC	Sequential Function Chart

Term	Forklaring
ST	Strukturert tekst
Sunnfjord kommune	Oppdragsgivar
TIA	Totally Integrated Automation

Referansar

- [1] Renasys. Nettsted, 2024. Internett
Sist hentet 2024-04-19, Tilgjengeleg frå: <https://www.renasys.com/nb>.
- [2] Sunnfjord Kommune. Nettsted, 2024. Internett,
Sist hentet 2024-04-19, Tilgjengeleg frå: <https://sunnfjord.kommune.no/>.
- [3] Høgskulen på Vestlandet. Nettsted, 2024. Internett,
Sist henta 2024-04-19 Tilgjengeleg frå: <https://www.hvl.no/>.
- [4] MIDTechnology AS. Nettsted, 2024. Internett,
Sist henta 2024-05-03 Tilgjengeleg frå: <http://midt.no/>.
- [5] SCD-Toolbox. Midtechnologies, 2024. Internett,
Sist henta 2024-05-03 Tilgjengeleg frå: <http://scd-diagram.no/>.
- [6] Microsoft. Office 365, 2024. Internett,
Sist henta 2024-05-18 Tilgjengeleg frå: <https://www.office.com/>.
- [7] DrawIO. Drawio teikneprogram, 2024. Internett,
Sist henta 2024-05-18 Tilgjengeleg frå: <https://app.diagrams.net/>.
- [8] MikTeX. Latex bibliotekhandteringsprogram, 2024. Internett,
Sist henta 2024-05-18 Tilgjengeleg frå: <https://miktex.org>.
- [9] Microsoft. Visual studio code, 2024. Internett,
Sist henta 2024-05-18 Tilgjengeleg frå: <https://code.visualstudio.com/>.
- [10] Codesys. Nettsted, 2024. Internett,
Sist henta 2024-04-24 Tilgjengeleg frå: <https://www.codesys.com>.
- [11] Google Earth. Nettsted, 2024. Internett,
Sist henta 2024-05-10 Tilgjengeleg frå: <https://earth.google.com/>.
- [12] International Electrotechnical Commission. Iec pas 63131 side: 30, table a.1 terminal codes identification letters, 2024. Internett,
Sist henta 2024-05-03 Tilgjengeleg frå: <https://webstore.iec.ch/publication/60599>.
- [13] Renasys. Mission zero, 2024. Internett
Sist hentet 2024-04-19, Tilgjengeleg frå: <https://www.renasys.com/nb>.
- [14] WaterCare AS. Brønnøysundregisteret, 2024. Internett,
Sist henta 2024-04-19 Tilgjengeleg frå: <https://data.brreg.no/enhetsregisteret/oppslag/enheter/885515762>.
- [15] Regjeriga. Revisjon av avløpsdirektivet, 2023. Internett,
Sist henta 2024-05-10 Tilgjengeleg frå:

<https://www.regjeringen.no/no/sub/eos-notatbasen/notatene/2021/des/revisjon-av-avlopsdirektivet/id2966230/>.

- [16] Statsforvalter Innlandet. Vurdering av renseteknologi, 2024. Internett, Sist henta 2024-05-10 Tilgjengeleg frå: <https://www.statsforvalteren.no/siteassets/fm-innlandet/06-miljo-og-klima/forurensning/horinger/bjorli-renseanlegg/vedlegg-2-vurdering-av-renseteknologi.pdf>.
- [17] National Library of Medicine. Polyaluminium chloride (pax-18) - acute toxicity and toxicity for early development stages of common carp (cyprinus carpio), 2009. Internett, Sist henta 2024-05-10 Tilgjengeleg frå: <https://pubmed.ncbi.nlm.nih.gov/20027170/>.
- [18] Lovdata. Forskrift om gjødselvarer mv. av organisk opphav, 2003. Internett, Sist henta 2024-05-10 Tilgjengeleg frå: <https://lovdata.no/dokument/SF/forskrift/2003-07-04-951>.
- [19] MDPI. Sludge dewatering and mineralization in sludge treatment reed beds, kapittel 2, 2017. Internett, Sist henta 2024-20-05 Tilgjengeleg frå: <https://www.mdpi.com/2073-4441/9/3/160>.
- [20] Siemens. Tia-portal, 2003. Internett, Sist henta 2024-05-10 Tilgjengeleg frå: <https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>.
- [21] International Electrotechnical Commission. Nettsted, 2024. Internett, Sist henta 2024-04-24 Tilgjengeleg frå: <https://www.iec.ch/homepage>.
- [22] Store Norske Leksikon. Iec, 2022. Internett, Sist henta 2022-01-10 Tilgjengeleg frå: https://snl.no/International_Electrotechnical_Commission.
- [23] Wikipedia. Iec 91131, 2023. Internett, Sist henta 2024-04-24 Tilgjengeleg frå: https://en.wikipedia.org/wiki/IEC_61131.
- [24] International Electrotechnical Commission. Iec 61131, 2013. Internett, Sist henta 2013-02-20 Tilgjengeleg frå: <https://webstore.iec.ch/publication/4552>.
- [25] International Electrotechnical Commission. Iec pas 63131, 2022. Internett, Sist henta 2017-11-07 Tilgjengeleg frå: <https://webstore.iec.ch/publication/60599>.
- [26] NORSOK. Norsok i-005:2013. Technical report, 2015. Internett, Sist henta 2024-04-25 Tilgjengeleg frå: <https://online.standard.no/nb/norsok-i-005-2013ac-2016-2>.
- [27] Codesys. Codesys licensing, 2024. Internett, Sist henta 2024-05-10 Tilgjengeleg frå: <https://www.codesys.com/the-system/licensing.html>.

- [28] Codesys. Codesys inside, 2024. Internett,
Sist henta 2024-05-10 Tilgjengeleg frå: <https://www.codesys.com/the-system/codesys-inside.html>.
- [29] Codesys. Codesys git, 2021. Internett,
Sist henta 2024-05-10 Tilgjengeleg frå:
<https://www.codesys.com/news-events/news/article/codesys-git-connection-of-the-widely-used-version-management.html>.
- [30] Codesys. Codesys store, 2021. Internett,
Sist endra 2024-05-10 Tilgjengeleg frå: <https://store.codesys.com/en/>.
- [31] CODESYS. Building automation libraries for codesys, 2024. Internett,
Sist henta 2024-04-25 Tilgjengeleg frå: <https://store.codesys.com/en/codesys-building-automation-library.html>.
- [32] CODESYS. Date and time libraries for codesys, 2024. Internett,
Sist henta 2024-04-25 Tilgjengeleg frå: <https://store.codesys.com/en/date-and-time.html>.
- [33] Codesys. Codesys util library, 2021. Internett,
Sist endra 2024-05-10 Tilgjengeleg frå:
<https://help.codesys.com/webapp/idx-Util-lib;product=Util;version=3.5.15.0>.
- [34] Olav Sande. A1 sikkerhet-styring/regulering-overvåkning, 2020. Kompendium upublisert internt dokument.
- [35] Olav Sande. Simatic s7 and tia portal (a3), 2020. Kompendium upublisert internt dokument.
- [36] International Electrotechnical Commission. IEC pas 63131 side: 18-22, terms definitions and abbreviated terms, 2024. Internett,
Sist henta 2024-05-03 Tilgjengeleg frå: <https://webstore.iec.ch/publication/60599>.
- [37] IEC. IEC pas 63131 mb-blokk s.69, 2023. Internett,
Sist henta 2017-11-07 Tilgjengeleg frå: <https://webstore.iec.ch/publication/60599>.
- [38] IEC. IEC pas 63131 ma-blokk s.62, 2023. Internett,
Sist henta 2017-11-07 Tilgjengeleg frå: <https://webstore.iec.ch/publication/60599>.
- [39] IEC. IEC pas 63131 mb-blokk s.94, 2023. Internett,
Sist henta 2017-11-07 Tilgjengeleg frå: <https://webstore.iec.ch/publication/60599>.
- [40] IEC. IEC pas 63131 mb-blokk s.100, 2023. Internett,
Sist henta 2017-11-07 Tilgjengeleg frå: <https://webstore.iec.ch/publication/60599>.
- [41] Codesys. Codesys alarm management, 2024. Internett,
Sist henta 2024-05-12 Tilgjengeleg frå:
<https://content.helpme-codesys.com/en/CODESYS>

- [42] Codesys. Codesys visualization, 2024. Internett,
Sist henta 2024-05-13 Tilgjengeleg frå:
<https://www.codesys.com/products/codesys-visualization.html>.
- [43] Store Norske Leksikon. Bioreaktor, 2021. Internett,
Sist endra 2024-05-19 Tilgjengeleg frå: <https://snl.no/bioreaktor>.

Liste av URL

1. **RA200 programmet** <https://github.com/VegardUllebo/Bachelor-RA200/>
2. **Simuleringsvideo** https://youtu.be/_2gtiXJDUyw

Liste av vedlegg

A	Framdriftsplan og timeforbruk	4
A.1	Timeforbruk	4
A.2	Timeforbruk graf	5
A.3	Gant	6
B	Funksjons Blokker	7
B.1	FB Pause sekvens	7
B.2	FB Innpumpings sekvens	11
B.3	FB Reaksjon sekvens	16
B.4	FB Sedimentering sekvens	21
B.5	FB Uttapping sekvens	25
B.6	FB Slammuttak	29
B.7	FB Analog Alarm	35
B.8	FB Digital Alarm	40
B.9	FB Kalkuleringer	44
B.10	FB Data Prosessering	48
B.11	FB High Load	53
B.12	FB Processed Water	57
B.13	FB Pump Interlock	62
B.14	FB Sivbed Rotation	66
B.15	FB Swap	70
B.16	FB Time Meter	74
B.17	FB Timer	79
C	IEC Blokker	83
C.1	Monitor Binary	83
C.2	Monitor Analogue	88
C.3	Switch Binary Eletrical	95
C.4	Switch Binary Valve	102
D	Funksjoner	109
D.1	FC Volum rektangel	109
D.2	FC Volum Sylinder	113
E	Varslingslister	117
E.1	Alarm liste	117
E.2	Warning liste	118
E.3	Error liste	119
F	Driftsinstruks orginal	124
G	Funksjonsbeskrivelse	177
H	SCD	234
H.1	Tilstand pause	234
H.2	Tilstand innpumping	235
H.3	Tilstand reaksjon	236
H.4	Tilstand sedimentering	237
H.5	Tilstand uttapping	238
H.6	Tilstand felles styring	239
I	Manualer	240
I.1	Cat blower	240
I.2	Cat Robox	262
I.3	Doseringspumpe	276
I.4	Huber	284
I.5	Motor	286
I.6	Pumpe	288
I.7	Transmitter MJK	304
I.8	Transmitter Vegabar	308
I.9	Ventilblokk	316

I.10 Ventilator CAW	350
I.11 Ventilator CNA	367
I.12 FX-4AD SPECIAL FUNCTION BLOCK	387
J I/O liste	399
K Forprosjekt	401