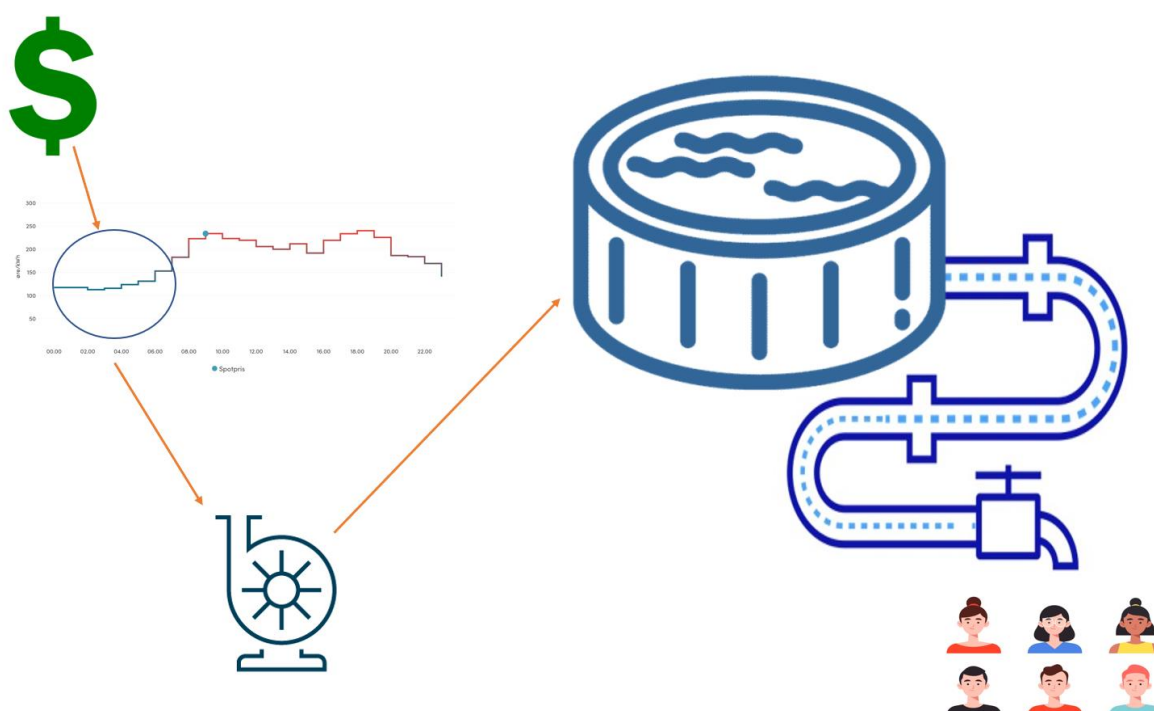


TS3000 1 Bachelor

# Strømeffektiv styring av høydebasseng



IA6-1-23

**Emne:** TS3000 1 Bachelor

**Tittel:** Strømeffektiv styring av høydebasseng

Denne rapporten utgjør en del av vurderingsgrunnlaget i emnet.

**Prosjektgruppe:** IA6-1-23

**Tilgjengelighet:** Åpen

**Gruppedeltakere:**

Marius Solli Jensen  
Odin Melås  
Rino Mikkelsen  
Fredrik Pedersen Monsen

**Veileder:**

Ole Magnus Brastein

**Prosjektpartner:**

Controlpartner AS

**Godkjent for arkivering:** \_\_\_\_\_

**Sammendrag:**

Prosjektet har som formål å optimalisere reguleringen av et høydebasseng i det kommunale vannettet ut fra strømpriser i markedet. Dette er for å spare penger på strømutgifter og å redusere belastningen på strømmettet. For å oppnå dette er det utviklet med Jython 2.7.2 en prosessimulator og en smartstyring i Ignition 8.1. Disse modulene benytter seg av strømprisene i markedet innhentet fra hvakosterstrømmen.no sitt web-API til å optimalisere styringsprofilen til systemet.

Kommunikasjonen mellom smartstyringsmodulen og prosesssimulatoren skjer via en LAN-kabel og kommunikasjonsprotokollen OPC UA. Det er brukt en datamaskin for hver av disse modulene. «Datamaskin 1» opererer som kontrollsystemet og «Datamaskin 2» som prosessen, som praktisk sett erstatter en PLS. Hver av disse datamaskinene er hverandres OPC UA server og klient. For fremvisning av prosessdata er det utviklet et web-dashbord i Perspective-modulen til Ignition 8.1. «Datamaskin 1» disponerer dette web-dashbordet og får prosessdata fra prosesssimulatoren via OPC UA. Prosessinformasjonen vises i web-dashbordet slik at operatøren har visuell kontroll over systemet.

Det er utviklet to revisjoner av smartstyringen. Ved bruk av smartstyring «revisjon 2» sammen med prosesssimulatoren ble det oppnådd en besparelse på omtrent 35% i løpet av uke 16, tilsvarende omkring 3000 NOK. Resultater oppnådd i forbindelse med forsøk gjennom prosjektperioden, viser at det kan være gunstig for kommuner å iverksette dette systemet i sine høydebasseng. Det er bevist ved hjelp av simulering at dette er kostnadsbesparende.

**Course:** TS3000 1 Bachelor

**Title:** Energy-efficient control of water reservoir

This report forms part of the basis for assessing the student's performance on the course.

**Project group:** IA6-1-23

**Availability:** Open

**Group participants:** Marius Solli Jensen  
Odin Melås  
Rino Mikkelsen  
Fredrik Pedersen Monsen

**Supervisor:** Ole Magnus Brastein

**Project partner:** Controlpartner AS

**Approved for archiving:** \_\_\_\_\_

**Summary:**

The aim of the project is to optimize the regulation of an elevated reservoir in the municipal water network based on market electricity prices, with the objective of saving money on electricity expenses and reducing the strain on the power grid. To achieve this, a process simulator and a smart control system were developed using Jython 2.7.2 in Ignition 8.1. These modules use electricity prices obtained from the hvakosterstrømmen.no web-API to optimize the system's operation.

Communication between the smart control module and the process simulator occurs via a LAN cable and the OPC UA communication protocol. One computer is used for each of these modules, with "Datamaskin 1" serving as the control system and "Datamaskin 2" representing the process, which would realistically be a PLC. Each of these computers is an OPC UA server and receiver. A web dashboard has been developed in the Perspective module of Ignition 8.1 for displaying process data, with "Datamaskin 1" controlling the dashboard and receiving process data from the process simulator via OPC UA. The process information is displayed in the dashboard, providing operators with visual control over the system.

Two revisions of the smart control system have been developed, with the use of "Revision 2" in conjunction with the process simulator resulting in a 35% savings during week 16, approximately 3000 NOK. Results obtained during the project's experimentation period indicate that it may be an advantage for municipalities to implement this system in their water reservoirs, with simulation proving it to be cost-effective.

# Forord

Bachelorrapporten utarbeides av fire elektroingeniørstudenter ved sjette semester innenfor linjen informatikk og automatisering Y-vei ved fakultetet for teknologi, naturvitenskap og maritime fag ved Universitetet i Sørøst-Norge, campus Porsgrunn.

Oppgaven er gitt i samarbeid med Controlpartner AS som er et av Norges større miljøer innen driftskontroll og automasjon samt digitale vedlikeholdssystemer.

Det er en forventning om at leseren av rapporten har en generell forståelse og kjennskap til informatikk og automatiseringsfaget.

Prosjektet benytter disse dataverktøyene:

- Microsoft Word 2021
- Microsoft PowerPoint 2021
- Microsoft Teams
- Ignition 8.1
- Microsoft Project
- UMLetino
- Python 3.9
- Jython 2.7.2

Vi vil rette en spesiell takk til Ole Magnus Brastein for hans bidrag som veileder. Hans bidrag og kunnskap har vært til meget stor hjelp for vår utvikling i dette faget. Vi vil også rette en stor takk til Controlpartner AS for alt de har bidratt med gjennom bachelor forløpet.

Porsgrunn, 19.05.2023

# Nomenklaturliste

API	-	Application Programming Interface
EUR	-	Euro
HMI	-	Human-Machine Interface
HPHMI	-	High Performance Human-Machine Interface
JVM	-	Java Virtual Machine
LAN	-	Local area network
MPC	-	Model predictive control / Modellbasert prediktiv regulering
NOK	-	Norske kroner
OPC UA	-	Open Platform Communication Unified Architecture
P&ID	-	Piping and instrumentation diagram
PLS	-	Programmerbar logisk styring

# Innholdsfortegnelse

Forord .....	4
Nomenklaturliste .....	5
Innholdsfortegnelse .....	6
<b>1 .. Innledning.....</b>	<b>9</b>
1.1 Bakgrunn for prosjektet .....	9
1.2 Problemstilling .....	9
1.3 Prosjektets mål .....	9
1.4 Avgrensninger .....	9
1.5 Metode .....	10
1.6 Leseveiledning .....	10
<b>2 .. Systemoversikt .....</b>	<b>11</b>
2.1 Systemarkitektur .....	11
2.1.1 Web-API modul .....	12
2.1.2 Prosessimulator modul .....	12
2.1.3 Smartstyringsmodulen .....	13
2.1.4 Web-dashbord .....	15
<b>3 .. Verktøy og metoder .....</b>	<b>16</b>
3.1 Ignition 8.1.....	16
3.2 OPC UA .....	16
3.3 Jython 2.7.2 .....	16
<b>4 .. Innhenting av strømpriser via web-API.....</b>	<b>17</b>
4.1 Kilde for innhenting av strømpriser.....	17
4.2 Funksjonalitet .....	17
<b>5 .. Programmering av en prosessimulator .....</b>	<b>19</b>
5.1 Innhenting av nødvendig informasjon og parametere .....	19
5.2 Vurdere metoder .....	19
5.2.1 Enkel sløyfe PI-regulator .....	19
5.2.2 Kaskaderegulering .....	20
5.2.3 Foroverkopling .....	20
5.2.4 MPC .....	21
5.3 Programmering av simulator.....	21
5.3.1 Klassediagram .....	21
5.3.2 Simulator i Python .....	22
5.3.3 Prosessforstyrrelser.....	25
5.3.4 Endring fra Python til Jython.....	26
5.3.5 Sanntidssimulering av smartstyring .....	27
5.3.6 Hurtig-simulert smartstyring.....	29
<b>6 .. Programmering av smartstyring .....</b>	<b>30</b>
6.1 Reguleringsstrategi .....	30
6.1.1 Revisjon 1.....	30
6.1.2 Revisjon 2.....	30
6.2 Samarbeid mellom smartstyring og prosessimulator .....	32

6.3	Kostnadsbesparelse med smartstyring.....	34
6.3.1	Utregning av driftspris .....	34
6.3.2	Sammenlikning av revisjon 1 og revisjon 2.....	34
7	Presentasjon av info i web-dashbord.....	35
7.1	Perspective Sessions .....	35
7.2	Design av brukergrensesnitt .....	35
7.3	Grafisk brukergrensesnitt .....	36
7.3.1	Navigering .....	36
7.3.2	«System»-visningsområde .....	37
7.3.3	«Strømpriser»-visningsområde .....	38
7.3.4	«Info»-visningsområde .....	39
7.3.5	«Parameter»-visningsområde.....	40
7.4	High performance human-machine interface.....	40
7.4.1	Knapper .....	41
7.4.2	Symboler.....	41
7.4.3	Bevegelige analoge indikatorer .....	42
7.5	Cybersikkerhet .....	43
7.6	Perspective Workstation.....	43
8	OPC UA kommunikasjon .....	44
8.1	Ignition Gateway.....	44
8.2	Kommunikasjon .....	44
8.3	Tilgjengelig OPC-tagger .....	46
8.3.1	Web-API modul.....	46
8.3.2	Smartstyringsmodul.....	46
8.3.3	Simulatormodul .....	46
8.3.4	Web-dashbord modul .....	46
9	Diskusjon .....	50
9.1	Målet med prosjektet .....	50
9.2	Krav til gruppen .....	50
9.3	Vurdering av strømpriskilde.....	50
9.4	Videreutvikling av sanntidssimulering .....	50
9.5	Revisjoner av smartstyring.....	51
9.5.1	Revisjon 1.....	51
9.5.2	Revisjon 2.....	51
9.5.3	Revisjon 3.....	52
9.6	Web-dashbord .....	52
9.6.1	Startknapp for simulatoren .....	52
9.6.2	Knapp for innhenting av strømpriser og oppretting av styringsprofil.....	53
9.6.3	Ventilstyring .....	53
9.6.4	Alarmsystem.....	53
9.6.5	Responsivt brukergrensesnitt.....	53
9.6.6	Fryste verdier .....	54
9.7	OPC UA kommunikasjon .....	54
9.7.1	LAN benyttes som erstatning for Wi-Fi.....	54
9.7.2	Datalagring.....	54
9.8	Utfordringer.....	54
9.8.1	Windup .....	54
9.8.2	Håndtering av Ignition-krasj .....	55
9.8.3	For-løkker .....	55
9.8.4	Power chart .....	55

<i>9.8.5 Etablere forbindelse mellom «StartStrom» og «StartSim»</i> .....	<b>55</b>
<b>10 Konklusjon</b> .....	<b>56</b>
<b>Referanser</b> .....	<b>57</b>
<b>Vedlegg</b> .....	<b>62</b>



# 1 Innledning

Dette kapitlet gir en beskrivelse av bakgrunn for prosjektet, hvorfor akkurat strømeffektiv styring av et høydebasseng er relevant for dagens strømsituasjon og hvordan dette er svært attraktivt for kommuner. Kapitlet gir en oversikt over hva gruppen ønsker å oppnå med prosjektet.

## 1.1 Bakgrunn for prosjektet

Med den nåværende situasjonen i strømmarkedet, har flere kommuner satt strømsparing høyt på prioriteringslisten. Påfylling av rent drikkevann i høydebassenger er en energikrevende prosess i det kommunale vannettet, og det vil derfor ha stor verdi for kommunene å optimalisere fyllingsprosessen i forhold til gjeldende strømpriser i markedet. Dette kan inkludere å justere fyllingsprosessen til tider på døgnet med lavere strømpriser, eller å investere i mer energieffektive metoder for fylling. På denne måten kan kommunene redusere sine strømutgifter og bidra til å redusere belastningen på strømmettet.

Ettersom strømmarkedet står overfor utfordringene knyttet til økte CO<sub>2</sub>-priser, vil fossile kraftproduksjonskilder fases ut og erstattes med bærekraftige kilder som vind, vann og sol [1]. Disse kildene kan imidlertid være uforutsigbare og væravhengige, og dermed vil prisene variere med været. Ved lite nedbør øker verdien på vannet som gjenstår i vannmagasinene, og strømprisene går opp [1] [2]. Når det er lite vind, vil ikke vindmøller kunne produsere nok strøm for å forsyne forbruket [3]. Solkraft er avhengig av klarvær og er kun i drift på dagtid, noe som gjør at denne type kraftproduksjon ikke egner seg for områder med lite sol [4].

En annen faktor som bidrar til den ustabile strømsituasjonen, er krigen som foregår i Ukraina [1] [2]. Europa har i mange år vært avhengig av gassen som Russland leverer, og siden Russland erklærte krig mot Ukraina, har europeiske land nektet å importere gass fra Russland [2]. Dette fikk ringvirkninger, og strømprisene har økt i takt med sanksjonene mot Russland [1] [2].

## 1.2 Problemstilling

Prosjektets problemstilling er å optimalisere reguleringen av et høydebasseng i det kommunale vannettet ut fra strømpriser i markedet. Målet med prosjektet er å styre fyllingsprosessen etter når strømmen er billig. Dette vil resultere i besparelser på strømutgifter og redusert belastning på strømmettet.

## 1.3 Prosjektets mål

Formålet med prosjektet er å besvare og løse problemstillingen på best mulig vis, presentere resultater relatert til temaet «strømeffektiv styring av høydebasseng», samt å formidle kunnskap om det bestemte området. Oppgaven har som formål å demonstrere studentenes evne til å gjennomføre selvstendig forskning og å anvende teori og metoder på en praktisk måte.

## 1.4 Avgrensninger

Prosjektet vil konsentrere seg om å utforske og vurdere ulike metoder for optimalisering av fylling og regulering av et høydebasseng i det kommunale vannettet. Det vil ikke omfatte andre aspekter av vannettet, slikt som vannkvalitet, sikkerhet, eller infrastruktur. Prosjektet vil kun

sette søkelys på å redusere strømutfgifter og bidra til å redusere belastningen på strømmettet, men vil ikke inkludere en vurdering av sosiale eller økologiske effekter. I tillegg er det spesifisert av Controlpartner at prosjektet skal ta i bruk programvaren Ignition og det innbygde utviklingsmiljøet Jython og kommunikasjonsprotokollen OPC UA. Tidsrammen for gjennomføringen av oppgaven vil være fem måneder.

## 1.5 Metode

Prosjektet går ut på å optimalisere fylling av et høydebasseng ved bruk av strømpriser, informasjon om prosessen, pumpekurver og historisk forbruk av vann. Dette er for å beregne den mest kostnadseffektive kjøreprofilen for pumpestasjonen tilknyttet høydebassenget.

Det skal programmeres en klient i Ignition sitt skript system som henter strømpriser fra et web-API. Det skal også programmeres en prosessimulator i Ignition sitt skript system Jython som skal brukes for å teste/validere beregninger som kommer fra smartstyringen.

For å få til dette må det hentes inn strømpriser fra en egnet web-API kilde sammen med informasjon om prosessen, pumpekurver og historisk forbruk av vann. Denne informasjonen brukes i beregninger hvor resultatet blir den mest kostnadseffektive kjøreprofilen for pumpestasjonen tilknyttet høydebassenget. Disse resultatene tilgjengeliggjøres for andre moduler via Ignition sin innbygde OPC UA server.

Det skal lages et brukergrensesnitt som bruker informasjonen fra smartstyringen og simulatoren. Dette skal presenteres i et web-dashbord ved hjelp av Perspective-modulen til Ignition. Systemet skal kunne betjenes fra dashbordet, samt at relevant nøkkelinformasjon vises.

## 1.6 Leseveiledning

Kapittel 2 handler om systemoversikt

Kapittel 3 handler om hvilke verktøy og metoder som er benyttet under gjennomførelsen av prosjektet

Kapittel 4 handler om innhenting av strømpriser via et web-API

Kapittel 5 handler om programmeringen av en prosessimulator

Kapittel 6 handler om programmering av smartstyring

Kapittel 7 handler om presentering av relevant informasjon i et web-dashbord

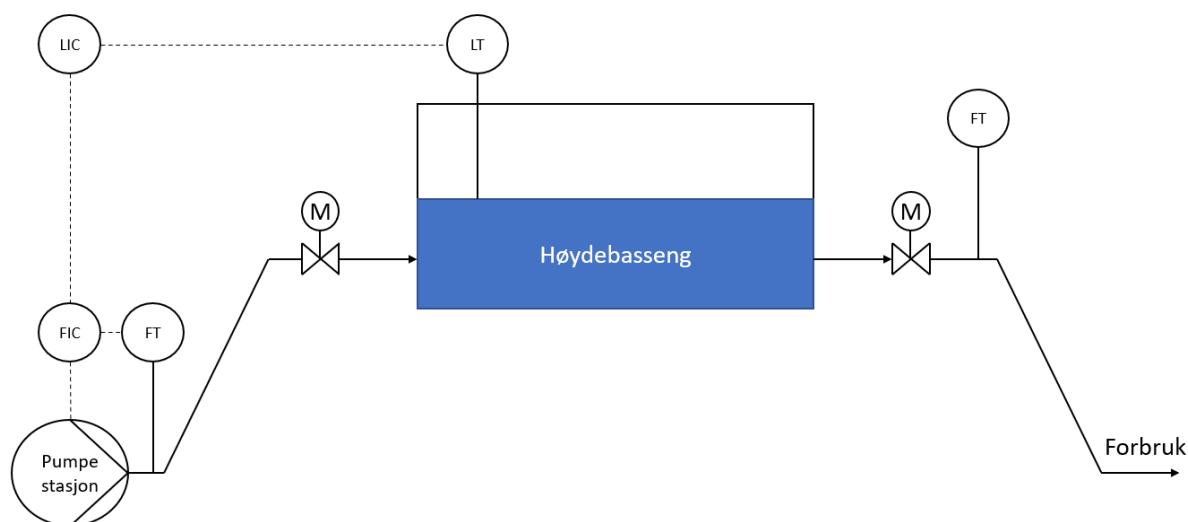
Kapittel 8 handler om OPC UA kommunikasjon

Kapittel 9 handler om diskusjon

Kapittel 10 handler om konklusjon

## 2 Systemoversikt

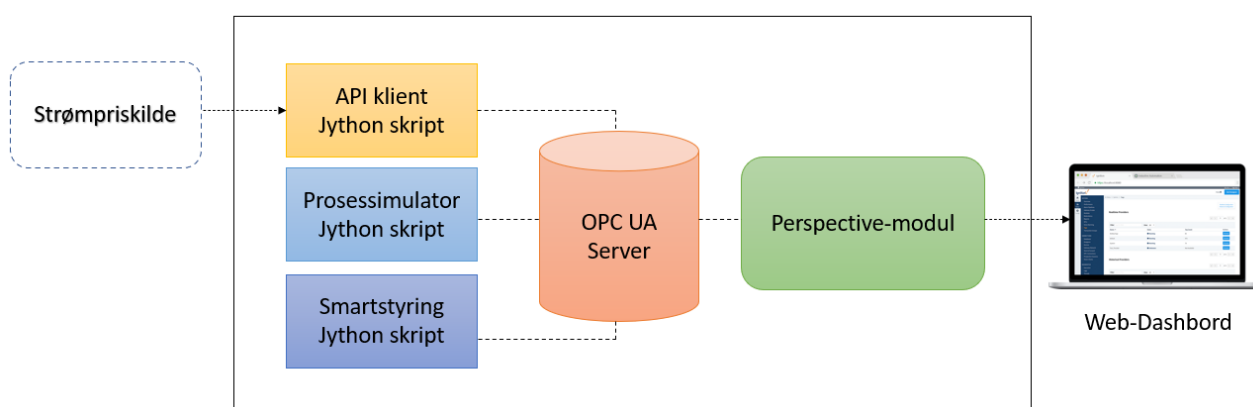
I dette kapitlet gis det en overordnet beskrivelse av prosjektet som er utviklet med det formål å fylle høydebassenget når strømprisene er på sitt laveste. Systemet består av flere moduler som er koblet sammen, og som samarbeider for å oppnå dette formålet. Det vil også bli forklart hvordan systemarkitekturen er satt sammen, og hvordan systemet samler inn nødvendige data for å få en strømeffektiv styring av høydebasseng. Den fysiske prosessen illustreres i Figur 2-1, som viser at prosessen benytter en kaskaderegulering for regulering av nivået i bassenget hvor utløpet er koblet til forbrukerne av vannet.



Figur 2-1 P&ID for pumpe stasjon og høydebasseng

### 2.1 Systemarkitektur

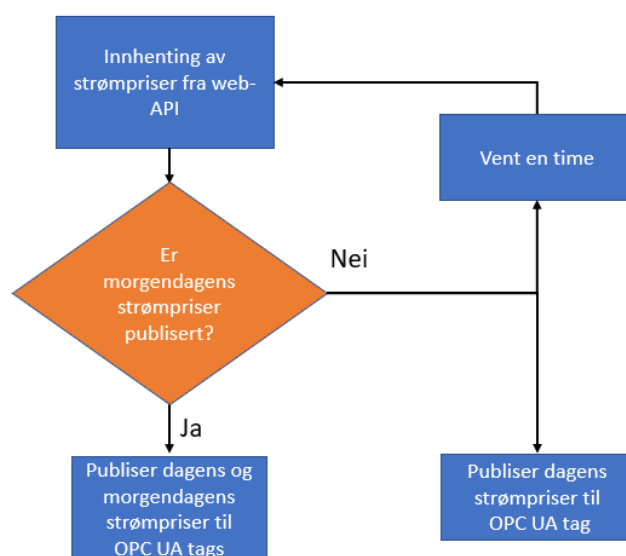
Systemet er delt opp i fire moduler. Disse modulene er et web-API, prosessimulator, smartstyring og fremstilling av et web-dashbord. OPC UA er brukt for å utveksle data og informasjon på tvers av modulene. Figur 2-2 viser oversikt over systemet. Kildekode for hver av disse modulene er tilgjengelig i Vedlegg F Kildekode.



Figur 2-2 Oversikt over systemet

### 2.1.1 Web-API modul

Web-API modulens oppgave er å hente strømpriser fra et eksternt API og lagre denne informasjonen i variabler som er tilgjengelig for andre moduler via OPC UA. Modulen er konfigurert til å hente inn dagens og morgendagens strømpriser ved kjøring av skriptet. Selv om morgendagens strømpriser ikke alltid er tilgjengelige når skriptet kjører, vil modulen sjekke regelmessig hver time inntil prisene er offentliggjort. Flytskjema for innhenting av strømpriser vises i Figur 2-3.

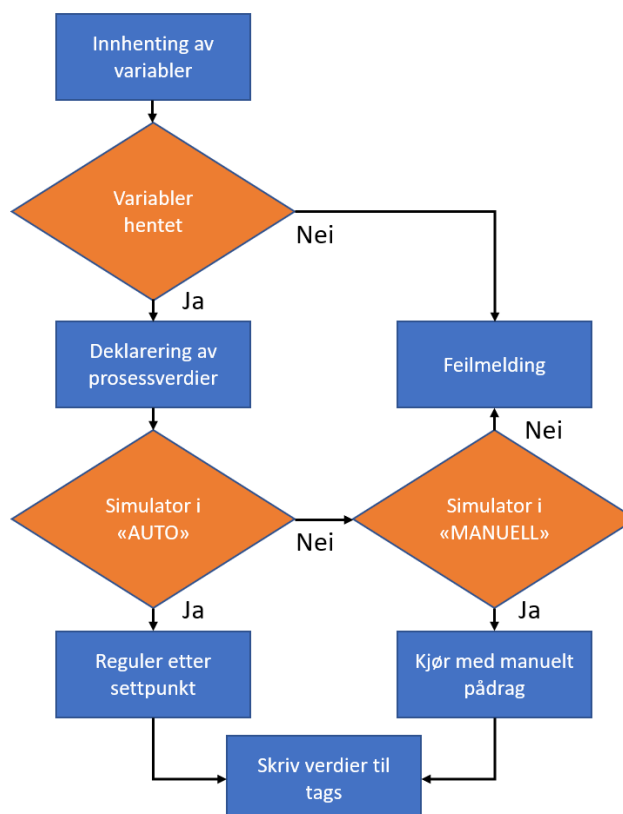


Figur 2-3 Flytskjema for innhenting av strømpriser

### 2.1.2 Prosesssimulator modul

Prosesssimuleringsmodulen har som oppgave å simulere driften av høydebassenget og de tilhørende pumper og ventiler. Simulatormodulen blir brukt som en erstatning av et virkelig høydebasseng med tilhørende reguleringsutstyr. Modulen simulerer prosessen som blir styrt av en smartstyring som regner ut styringsprofilen til pumpen ut fra dagens strømpriser. Prosesssimulatoren bruker massebalanse og Eulers metode som matematiske modeller for å beregne nivået for tanken i simulatoren. Modulen er tilkoblet en OPC UA server for å lese og skrive til OPC UA variabler. Disse variablene representerer tilstanden til høydebassenget og pumpepådraget.

Prosesssimuleringsmodulen kan konfigureres og justeres etter behov for å simulere ulike scenarier og situasjoner. Modulen kan derfor brukes som et realistisk simuleringsverktøy, og kan brukes til å validere og teste funksjonaliteter til systemet under ulike forhold. Flytskjema for prosesssimulator vises i Figur 2-4.



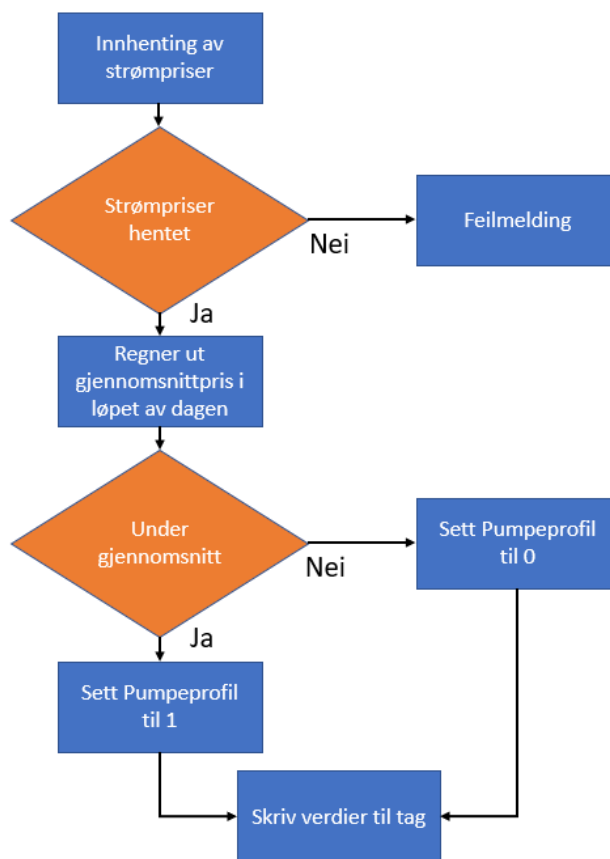
Figur 2-4 Flytskjema for prosessimulator

### 2.1.3 Smartstyringsmodulen

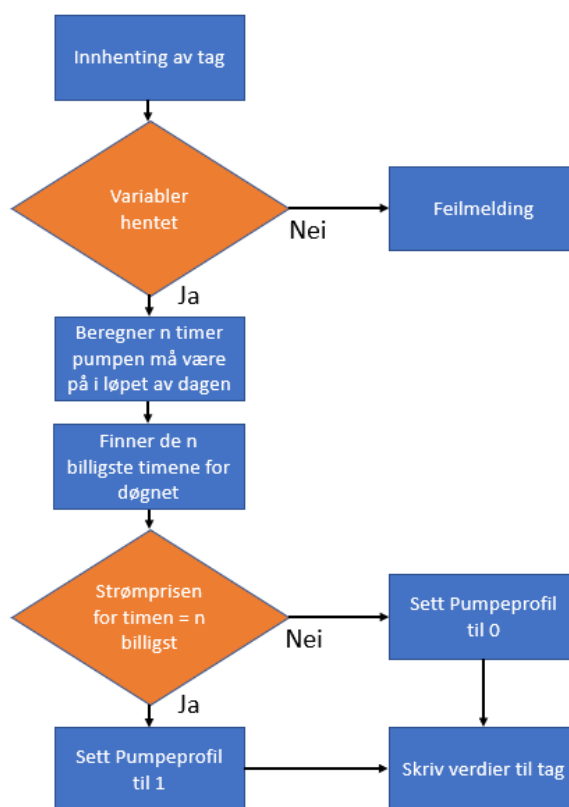
Smartstyringsmodulen har som oppgave å ta beslutning om når det er mest gunstig å fylle høydebassenget basert på innhenting av informasjon om strømpriser og hvilken tilstand høydebassenget har. Modulen henter strømpriser fra web-API-modulen og beregner optimal styringsprofil til pumpen for fylling av høydebassenget basert på hvilken revisjon som er i bruk. Smartstyringsmodulen bruker informasjon om fyllingsgraden til høydebassenget fra prosesssimuleringsmodulen slik at ikke bassenget overfylles eller tappes for mye.

Smartstyringsmodulen justerer fyllingsgraden kontinuerlig basert på strømprisene og tilstanden til bassenget og sørger for at bassenget alltid har tilstrekkelig med vann tilgjengelig, se kapittel 6.1 for mer informasjon. Modulen kommuniserer med prosesssimuleringsmodulen og web-dashbordet via OPC UA for å motta og sende parametere og justeringer. Det er opprettet OPC UA variabler i Ignition for å få til dette. Disse variablene brukes for å overvåke og logge tilstanden til høydebassenget og pumpestasjonen, samt for å motta eventuelle alarmer eller varslinger.

Smartstyringsmodulen kan konfigureres og justeres etter behov for å tilpasse ulike scenarier og situasjoner. Det er i prosjektet utviklet to revisjoner av smartstyringsmodulen, med ambisjoner om en tredje revisjon. Flytskjema for smartstyringsmodul revisjon 1 i Figur 2-5 og flytskjema for smartstyringsmodul revisjon 2 i Figur 2-6.



Figur 2-5 Flytskjema for smartstyringsmodul revisjon 1

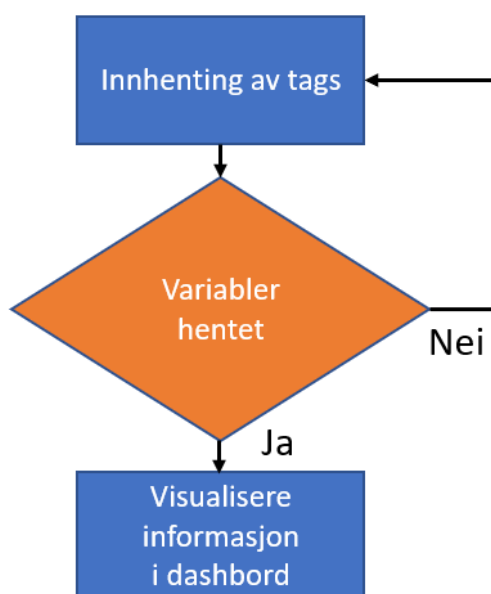


Figur 2-6 Flytskjema for smartstyring revisjon 2

### 2.1.4 Web-dashbord

Web-dashbordet har til oppgave å fremstille aktuelle data om høydebassenget og pumpestasjonen på en brukervennlig måte. Dashbordet gir en oversikt over informasjon om strømpriser og prosessdata som vannivå og strømforbruk.

Dashbordet er bygget på Ignition-plattformen ved bruk av Perspective-modulen, og gir en intuitivt og brukervennlig opplevelse for brukeren. Brukere kan tilpasse dashbordet ved å legge til eller fjerne sanntidsdata og grafer. Et flytskjema er vist i Figur 2-7. Web-dashbordet gir en komplett oversikt over høydebassenget, og gir brukeren mulighet til å overvåke, justere data og driftsparametere som er hentet via OPC UA og presenterer denne informasjonen på dashbordet. Dashbordet gir brukeren muligheten til å evaluere og optimalisere driften av anlegget med en oversikt over dagens strømpriser.



Figur 2-7 Flytskjema for Perspective-modul

## 3 Verktøy og metoder

I dette kapitlet gis det en beskrivelse om ulike verktøy og metoder som benyttes for utvikling av systemet.

### 3.1 Ignition 8.1

Ignition er en programvareplattform som er utviklet av Inductive Automation. Plattformen gir mulighet for å designe og utvikle løsninger for automatisering og overvåkning av prosesser i industrielle miljøer [5]. Ignition kan betraktes som alt-i-ett programvareplattform, fordi Ignition gir muligheten til å integrere en rekke tredjeparts programvare, med ERP-løsninger og SCADA løsninger [5]. Dette gir brukeren mulighet til å tilpasse og skalere systemet til brukerens spesifikke behov. Ved bruk av Ignition kan man samle data fra forskjellige kilder. Eksempler på dette er sensorer, PLS-er og databaser. Dette gjør det mulig å få en mer helhetlig oversikt over produksjonsprosessen og dataene som generes i sanntid.

Perspective-modulen i Ignition lar brukeren presentere innholdet enkelt på et web-basert HMI-løsning som kan vises på PC, nettbrett og smarttelefoner [5]. På denne måten får man en enkel oversikt over anlegget man vil overvåke.

### 3.2 OPC UA

OPC UA er en kommunikasjonsprotokoll som blir benyttet til å utveksle data og informasjon mellom ulike industrielle automasjonssystemer på en sikker, pålitelig og effektiv måte [6]. Fordelen med OPC UA er at det er en åpen standard og er plattformuavhengig protokoll som gir mulighet til samarbeid på tvers av forskjellige operativsystemer og enheter [6].

I prosjektet er OPC UA brukt for å utveksle data og informasjon fra modulene som befinner seg på forskjellige datamaskiner. Kommunikasjonen mellom to datamaskiner via OPC UA kan foregå på forskjellige måter avhengig av applikasjonen og systemene som brukes. Det vil bli vurdert hvilken måte det best egner seg å kommunisere via, for eksempel via LAN eller Wi-Fi.

### 3.3 Jython 2.7.2

Jython er en implementasjon av Python, og er basert på en tidligere versjon av Python [7]. Jython kjører på Java Virtual Machine (JVM) og kan derfor bli sett på som en hybrid mellom Java og Python [7]. Selv om Jython er basert på Python er det noen funksjoner som finnes i Python som ikke støttes i Jython. Python er skrevet i C-språket og har kjente biblioteker som for eksempel NumPy [8]. Disse støttes ikke av Jython fordi bibliotekene ikke er kompatible med JVM [9]. Prosjektet bruker Jython 2.7.2 da dette er siste versjon av Jython i Ignition 8.1.



## 4 Innhenting av strømpriser via web-API

Kapittelet tar for seg innhenting av strømpriser fra et web-API, hvordan programmet er oppbygd, kildene som er vurdert opp mot løsningen og implementasjonen av funksjonaliteten.

### 4.1 Kilde for innhenting av strømpriser

For innhenting av strømpriser via et web-API er det valgt å bruke et gratis API fra [hvakosterstrømmen.no](http://hvakosterstrømmen.no). Dette API-et gir tilgang til dagens og morgendagens norske strømpriser, samt historisk data datert tilbake til 1 desember 2021 med ulike tidspunkter og områder [10]. [Hvakosterstrømmen.no](http://hvakosterstrømmen.no) benytter [Entsoe.eu](http://Entsoe.eu) som kilde for henting av strømpriser. Entsoe er en europeisk forening som samler inn og distribuerer data om energi og strømmnettverk over hele Europa. Strømdataen som blir hentet fra [Entsue.eu](http://Entsue.eu) av [hvakosterstrømmen.no](http://hvakosterstrømmen.no), blir representert i en statisk JSON-fil. Det gjør det enkelt å behandle dataene og bruke det videre i prosjektet [11].

[Hvakosterstrømmen.no](http://Hvakosterstrømmen.no) henter strømpriser fra [entsoe.eu](http://entsoe.eu) i euro valuta, denne strømprisen blir konvertert til norske kroner etter eurokurs gitt av Norges Bank [11]. Dette kan medføre et usikkerhetsmoment ved at strømprisen kan ha marginale avvik fra [nordpool.no](http://nordpool.no) sine strømpriser [11].

En bedre løsning er å ta i bruk Nord Pool som strømpriskilde, siden de aller fleste strømleverandører i Norge kjøper strøm fra Nord Pool [12]. Strømprisene vil derfor være mer representative fra Nord Pool. Utfordringen med å hente strømpriser fra Nord Pool er at man må være en betalende kunde for å få tilgang til Nord Pool sitt API for å hente dagens og morgendagens strømpriser, samt prishistorikk [13] [14] [15]. Det kan derfor være behov for å finne alternative løsninger som å bruke en annen tredjepartsleverandør som benytter seg av Nord Pool sitt API. Det er mulig å implementere Nord Pool sitt API i en senere utgivelse av programmet.

### 4.2 Funksjonalitet

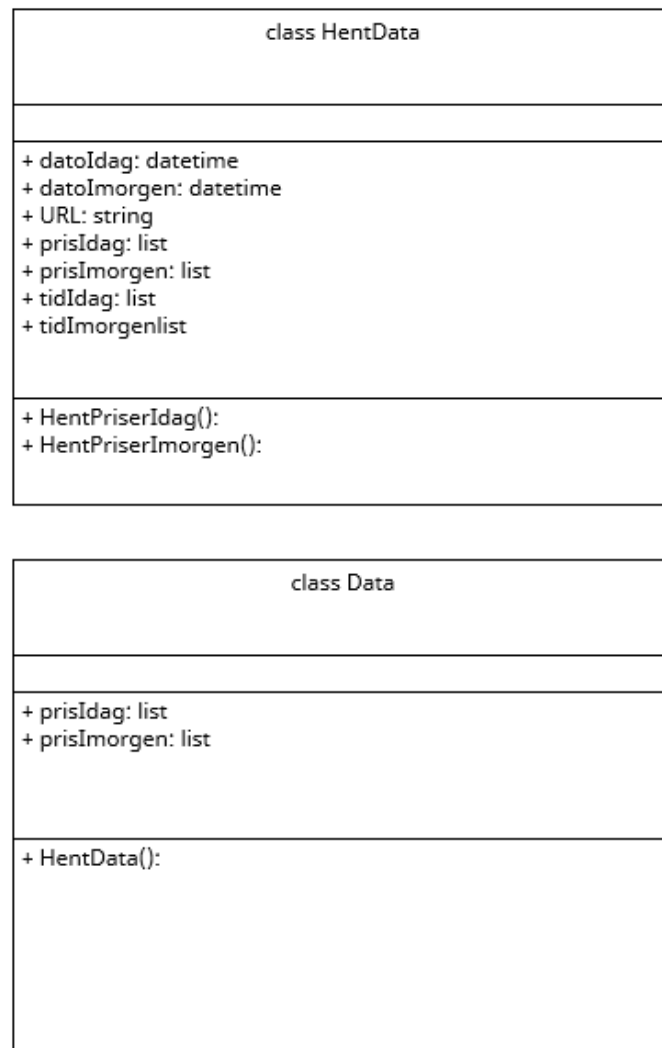
Web-API modulens oppgave er å hente strømpriser fra et eksternt API, og samle denne dataen og lagre dette i variabler i Ignition. Det eksterne API-et bruker JSON-format [11], som modulen konverterer til numeriske verdier før disse blir lagret i OPC UA variabler i Ignition. Disse variablene er tilgjengelig for andre moduler i systemet.

Modulen er konfigurert til å hente strømpriser for dagens og morgendagens strømpriser ved kjøring av skriptet. Morgendagens strømpriser er vanligvis offentligjort kl. 13.00 hver dag [11]. Hvis morgendagens strømpriser ikke er tilgjengelig når skriptet kjøres, vil den sjekke regelmessig hver time inntil strømprisene er publisert. Selv om skriptet er utviklet til å innhente morgendagens strømpriser, er dette en funksjonalitet som ikke er tatt i bruk av smartstyringen. Dette er mulig å implementere i en senere utgave for å potensielt kunne planlegge en billigere styringsprofil, samt gi en mer effektiv strømsparing. Dette er mer beskrevet i kapittel 9.5.3.

Konfigurasjonen av modulen kan justeres etter behov. Det er for eksempel mulig å endre hyppigheten av innhentinger, bytte til et annet API for innhenting av strømpriser eller endre prisområde for spotpris. Prosjektet bruker NO1, som er prisområdet for Oslo/Øst-Norge [11].

I koden er det opprettet klasser for å få en bedre oversikt, og mulighet til å endre deler av koden på en enkel og ryddig måte. To klasser er opprettet «HentData» og «Data». I «HentData»-klassen blir strømprisene for dagens og morgendagens strømpriser hentet fra kilden, og blir

behandlet videre i to funksjoner i klassen. I «Data»-klassen blir lagret data hentet fra «HentData»-klassen. Figur 4-1 viser et klassediagram over klassene.



Figur 4-1 Klassediagram for API-modulen

Dagens strømpriser blir lagret i et strømpris-array. Arrayet er et endimensjonalt array av typen float og inneholder 24 elementer med priser i form av NOK/KWh. Disse elementene representerer priser for en gitt time i døgnet. Videre blir arrayet publisert i en OPC UA variabel som hentes og brukes av de andre modulene.

## 5 Programmering av en prosessimulator

Dette kapittelet beskriver utviklingen til simulatoren. Simulatoren er utviklet ved bruk av en matematisk modell som representerer systemet. Kapittelet beskriver ulike reguleringsmetoder som er brukt for å regulere høydebassenget.

### 5.1 Innhenting av nødvendig informasjon og parametere

For å lage en matematisk modell som representerer et system, vil det være nødvendig å velge parametere så nært det ekte systemet som mulig, gjerne helt lik prosessen man modellerer etter [16]. For å oppnå dette er det tatt i bruk verdier fra et eksisterende høydebasseng, tilgjengeliggjort av Controlpartner. Men merk at disse variablene kan endres dersom det skal tas i bruk i et annet anlegg. Følgende parameter er valgt for simulatoren:

Sylinderformet reservoar:

- Høyde = 7m
- Radius = 11m

Pumpe:

- Maks kapasitet =  $120 \text{ m}^3/\text{t} = 33 \text{ l/s}$
- Karakteristikk = Lineær

Utløpsventil:

- Strømningskoeffisient =  $0.005 \text{ m}^3/\text{s}$

Simulering av pumpe har blitt grovt forenklet, eksempelvis antas det at den har tilstrekkelig sughøyde, at hodetrykket ikke har noen innvirkning på strømmingen og at ytelse er et lineært forhold mellom pådragssignal og strømming/mA.

Det forutsettes at pumpekarakteristikken er lineær. Programvaren forventer 50 prosent strømming ved 50 prosent pådragssignal. Dersom pumpen ikke skulle være lineær, er det nødvendig med tilstrekkelig linearisering. Tilstrekkelig linearisering kan eksempelvis være stykkevis linearisering av pumpekurven [17].

### 5.2 Vurdere metoder

I dette delkapitlet er det beskrevet og vurdert de ulike metodene for regulering av høydebassenget. Det er vurdert enkel sløyfe PI-regulator, kaskaderegulering, foroverkobling og MPC. Formålet med å vurdere de ulike metodene er for å kunne stabilisere prosessen på best mulig måte.

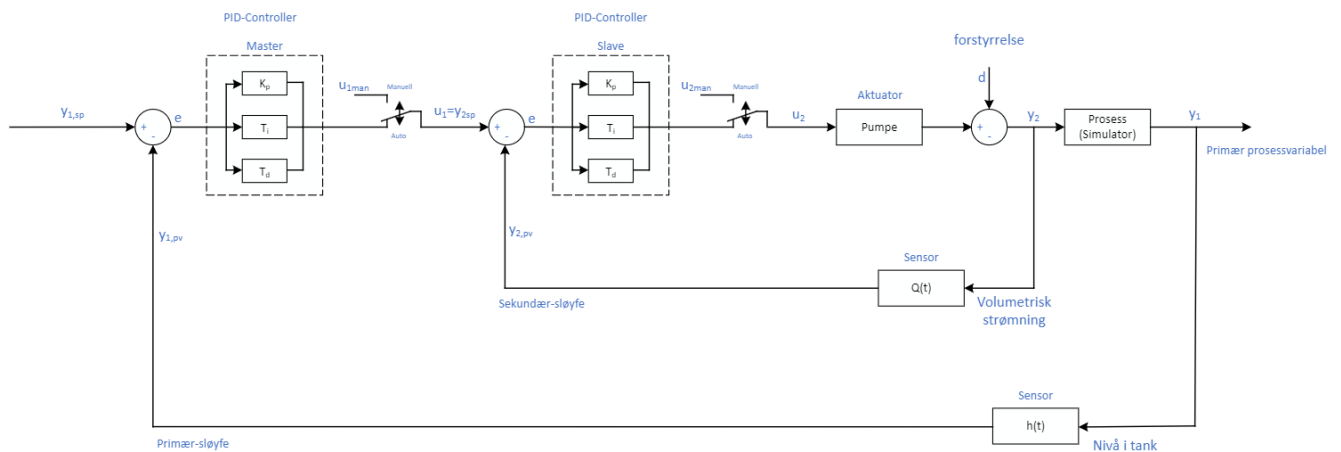
#### 5.2.1 Enkel sløyfe PI-regulator

En PI-regulator kan benyttes for å regulere nivået i tanken. P- og I-leddet vil jobbe prosessverdien til settpunktet som er gitt. P-leddet vil alltid være proporsjonal med avviket «e». Det benyttes derfor I-ledd for å få reguleringsavviket lik null [18] slik at prosessverdien kan bli likt settpunkt. PI-regulatoren er en enkel løsning for regulering som anvendes mye i industrien [18]. Regulatoren benytter ikke D-leddet, det er fordi det ikke er nødvendig. Dersom D-leddet hadde blitt implementert, forventes det at den kun hadde forsterket målestøy [18].

Det er laget et Python-program som inneholder en enkel sløyfe med PI-regulator. Det ble brukt for å teste kodingen av regulatoren opp mot prosessen, men også for å se hvordan prosessen reagerte med bare en PI-regulator. Figur 5-3 viser hvordan en PI-regulator kan fungere i den prosessen den er anvendt i.

### 5.2.2 Kaskaderegulering

Kaskaderegulering er en reguleringsteknikk som anvendes når man ønsker å oppnå bedre kompensering for prosessforstyrrelser i prosessen som skal reguleres [19]. Dette bidrar til at reguleringsavviket blir mindre [18]. I kaskade benyttes det to regulatorer, én i den primærsløyfen som kalles primær-regulator, og én i sekundærsløyfen som kalles sekundær-regulator [19]. Figur 5-1 er et blokkskjema av kaskadereguleringen for nivået i høydebassenget. Dette viser for eksempel at styringssignalet til primær-regulatoren er settpunktet for sekundær-regulatoren.



Figur 5-1 Blokkskjema for kaskaderegulering

### 5.2.3 Foroverkobling

En annen reguleringsstrategi som kan brukes for nivåregulering av høydebasseng, er foroverkobling. Med foroverkoblet regulering baserer pådraget seg på at prosessforstyrrelser blir målt, settpunktene til de ulike prosessvariablene er kjent og en matematisk modell av prosessen er utviklet [20]. Slik vil foroverkoblingen justere pådraget til den ideelle verdien for at reguleringsavviket skal bli null [20]. Dette skjer bare hvis man har perfekt kunnskap rundt prosessen, matematiske modellen og at målingen av forstyrrelsen er korrekt [20]. Dette er en metode som kan brukes for prosessen. Men siden programmet skal kunne benyttes av andre vannverk, er det bedre med kaskade. Dette fordi foroverkobling må ha en bra forståelse av prosessen som benyttes og forstyrrelser som må måles korrekt [20].

### 5.2.4 MPC

Modellbasert prediktiv regulering bruker modellen av prosessen som skal reguleres for å kunne beregne den optimale pådragssekvensen for en viss tid inn i fremtiden [21] [22]. Det gjør den ved å beregne for et bestemt fremtidig tidsintervall fra nå-tidspunktet [21] [22]. Ved å gjøre dette tar man i bruk pådraget for nåværende tidsskritt, før den gjentar det samme ved å hoppe et tidsskritt frem. MPC er derfor basert på et prinsipp som heter «moving horizon» [21] [22]. MPC forsøker å finne den optimale pådragssekvensen som vil produsere ønsket oppførsel av prosessen i forhold til tidsintervallet [21] [22]. For å få dette til, må MPC ta i bruk og vurdere begrensinger som påvirker prosessen [21] [22].

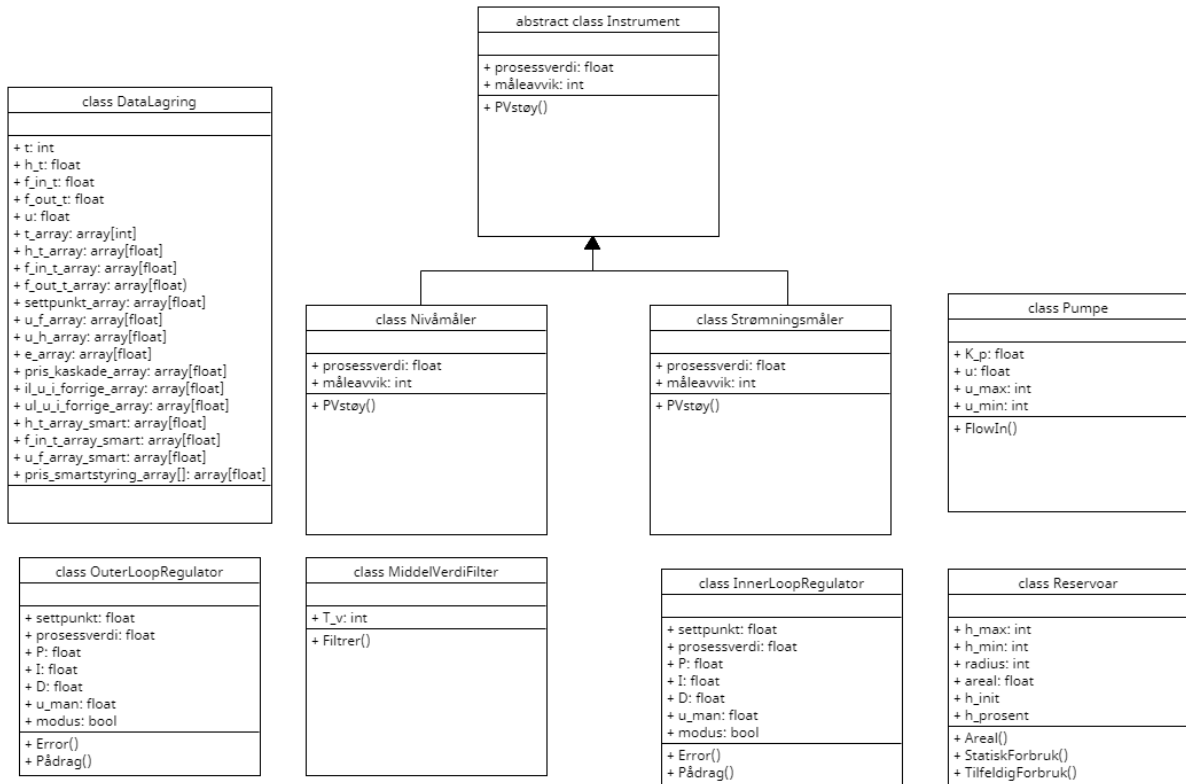
En implementasjon av dette vil bidra til å finne riktig fylling av høydebassenget målt opp mot vannforbruk og strømpriser. Her ville MPC vært en «supervisory» regulator som finner det optimale settpunktet ut fra forbruket, strømpriser og det nåværende nivået i tanken [23]. Etter å ha funnet det optimale settpunktet vil MPC gi kaskadereguleringen et settpunkt å styre etter. Regulatorene vil dermed lese av verdiene fra transmitterne og videreformidler informasjonen til MPC som utfører samme prosess om igjen i neste tidsskritt [23]. MPC kan korrigere forutsigbare forstyrrelser for systemet før de inntreffer, ved å vite om ulike forstyrrelser som kan oppstå i prosessen. Derimot vil en PID regulator kun reagere på feil eller avvik fra settpunkt i øyeblikket [18] [23].

## 5.3 Programmering av simulator

Formålet med simulatoren er å undersøke ulike reguleringsstrategier samt regulatorparameter for å oppnå en mest mulig strømeffektiv prosess. Ved å simulere dette systemet kan man få en bedre forståelse av hvordan reguleringen kan implementeres og optimaliseres i liknende systemer. I dette prosjektet er simulatoren brukt som en erstatning av en ekte prosess som beskrevet i delkapittelet 2.1.3.

### 5.3.1 Klassediagram

Klasser tas i bruk i koden for å opprettholde struktur, og sørger for at dersom en variabel endres, endres den i alle instanser den er tatt i bruk. Figur 5-2 viser en oversikt over klassene som er brukt. Det er brukt en abstrakt klasse for arving til to instrumenter, det samme kunne blitt gjort om det var flere pumper av ulik størrelse til stede. Klassen «DataLagring» er brukt som et verktøy til å holde på informasjon, samt plotting i grafer. Denne klassen blir redundant og unødvendig ved programmering i Ignition, ettersom informasjonen lagres til tags.



Figur 5-2 Klassediagram for simulatormodulen

### 5.3.2 Simulator i Python

Førsteutkast er programmert i utviklingsmiljøet «Spyder» med programmeringsspråket Python. Følgende parameter inngår i denne revisjonen av simulatoren:

PI-regulator:

- $u_{man} = 4.056$
- $P = 10.00875$
- $I = 603.75$

Disse verdiene er funnet ved å bruke Relaxed Ziegler Nichols svingemetode [24]. For å se hvordan nivået i tanken utvikler seg er det brukt Eulers metode og følgende massebalanse, se formel 5.1.

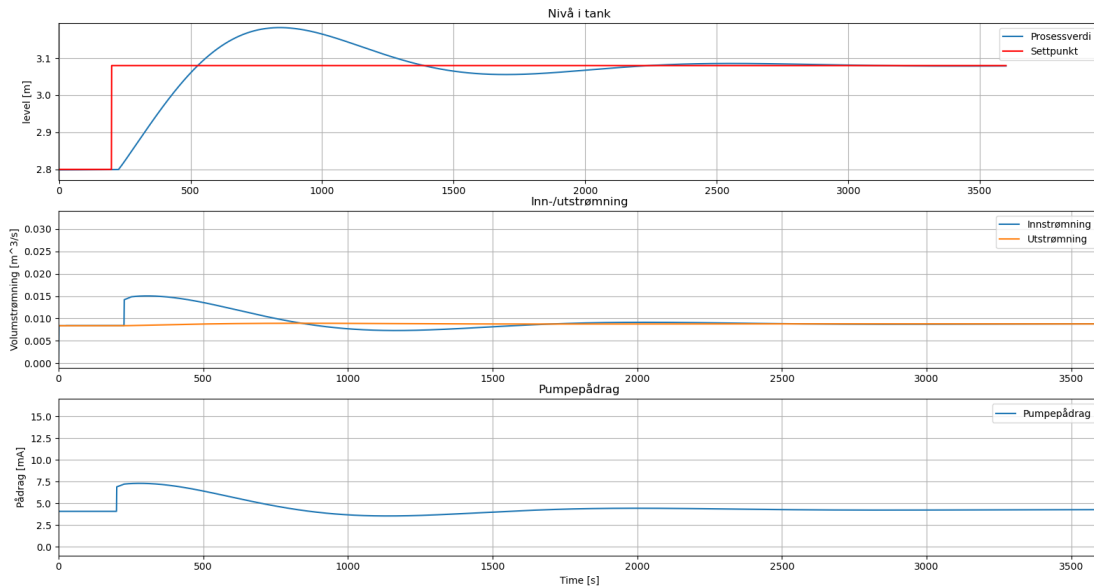
$$\frac{dh}{dt} = \frac{1}{A}(q_{in} - q_{out}) \tag{5.1}$$

hvor:

- $\frac{dh}{dt}$  [m] representerer endringen i nivået per tidsenhet
- $q_{in}$  [ $m^3/t$ ] representerer strømmingen inn i tanken per tidsenhet, dette er et resultat av pådragssignalet fra regulatoren som fungerer på pumpen(e) i prosessen.
- $q_{out}$  [ $m^3/t$ ] representerer strømmingen ut av tanken per tidsenhet, i denne fasen av simulator-utviklingen vil dette være en funksjon av høyden i tanken.

Figur 5-3 viser hvordan høyden, inn- og utstrømning samt pumpepådrag utvikler seg ved et sprang i settpunkt på 0.28 meter ved bruk av enkel sløyfe regulator. Harmoniske svingninger

brukes for å finne kritisk forsterkning og kritisk periode og for å tune regulatoren. For å oppnå harmoniske svingninger er det introdusert en tidsforsinkelse på 120 iterasjoner. Her kan det tenkes at det tar en viss tid før strømmingen fra pumpe har en effekt på nivået i tanken.



Figur 5-3 Lukket sløyfe enkel regulator

Det skal nevnes at denne reguleringen er usedvanlig treg. Dette kommer av at tankens radius er 11m – endring i nivå utgjør et enormt volum. X-aksen i denne figuren er i minutter og man kan se at reguleringen stabiliserer seg på settpunkt ved omkring  $t = 2800$ , det er ca. 2 dager. Dette er for lang tid. Derfor vil det nå bli introdusert kaskaderegulering med hensikt til å eliminere overdreven tidsbruk samt avvik.

Figur 5-4 viser den samme prosessen med implementert kaskaderegulering. Her er x-aksen i sekunder og man kan se at tiden det tar for stabilt nivå har drastisk gått ned. Ved  $t = 12000$  er det nokså stabilt, det tilsvarer omkring 3 timer og 20 minutter – mye bedre enn før. Merk at det er utført samme sprang i settpunkt for å gjøre sammenlikningen så reell som mulig. For regulatorene er følgende parametere valgt:

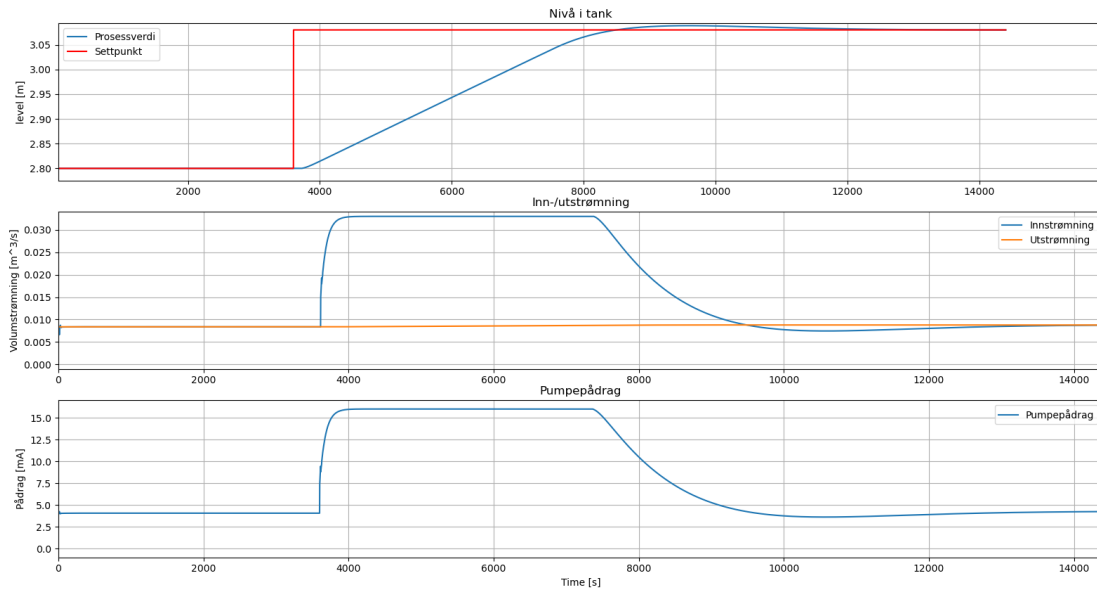
Primærregulator:

- $u_{\text{man}} = 0.008366$
- $P = 0.5$
- $I = 3750$

Sekundærregulator:

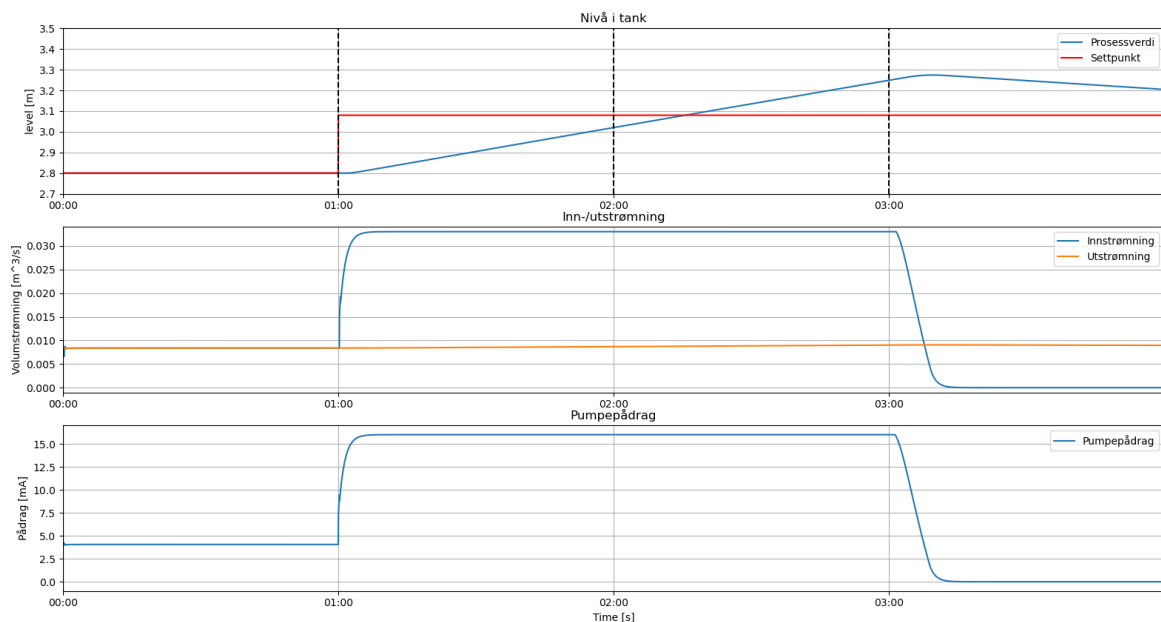
- $u_{\text{man}} = 4.0566$
- $P = 484.81$
- $I = 28$

Parameterne for sekundærsløyfen er funnet ved å modellere strømmingen fra pumpen isolert og tunet etter Relaxed Ziegler Nichols svingemetode. Deretter kan man eksperimentere med regulatorparameter for primærsløyfen til et akseptabelt pådrag blir nådd.



Figur 5-4 Kaskaderegulering med sprang i settpunkt

Ved metning av pådragssignal oppstår det en utfordring med «windup». Problemstillingen med windup er kort fortalt at pådraget fra integralleddet bygger seg opp i evigheten (ved lang nok metningstid) [18]. Figur 5-5 viser samme systemet som i Figur 5-4, med samme sprang i settpunkt, men uten «anti-windup». Her kan man se at intergratorleddet bruker lang tid på å integrere pådraget til stabil verdi, ettersom pådragssignalet har vært mettet i omkring to timer.



Figur 5-5 Eksempel på problemstilling ved «windup»

For å unngå dette ble det iverksatt følgende logikk, se Figur 5-6. Denne logikken gjør at integratorleddet begrenses dersom pådragssignalet mettes. Men dersom pådraget er mellom minimum- og maksimumsverdi, vil integratorbidraget utvikle seg som det skal. Merk at dersom



metningsgrensene på pådragssignalet aldri blir nådd, vil en aldri møte på problemer med «windup». Dette kommer igjen an på hvordan regulatoren tunes.

```
#Oppdater u_i_forrige + antiwindup
if u > u_max:
    self.u_i_forrige = self.u_i_forrige
elif u < u_min:
    self.u_i_forrige = self.u_i_forrige
else:
    self.u_i_forrige = u_i
```

Figur 5-6 Anti-windup

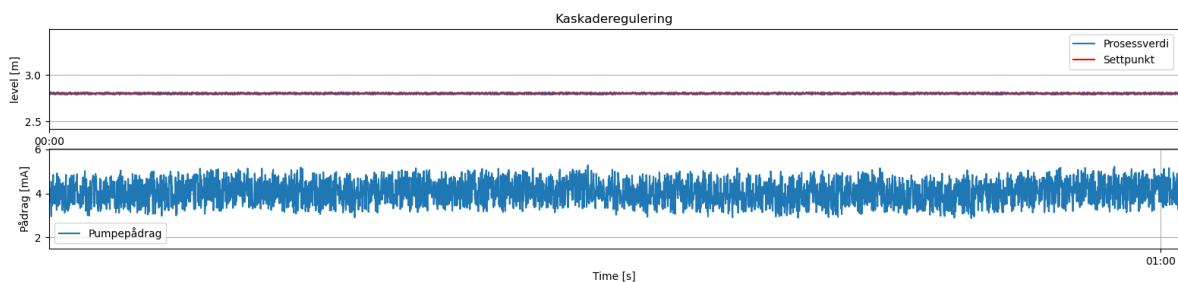
### 5.3.3 Prosessforstyrrelser

En kjent problemstilling i måleteknikken er målestøy. Det kan forekomme eksempelvis av elektromagnetisk støy eller mekanisk slitasje. For å simulere støy er følgende logikk brukt, se Figur 5-7.

```
PVavvik = (self.prosessverdi * self.måleavvik) / 100
PVstøy = random.uniform(self.prosessverdi - PVavvik, self.prosessverdi + PVavvik)
```

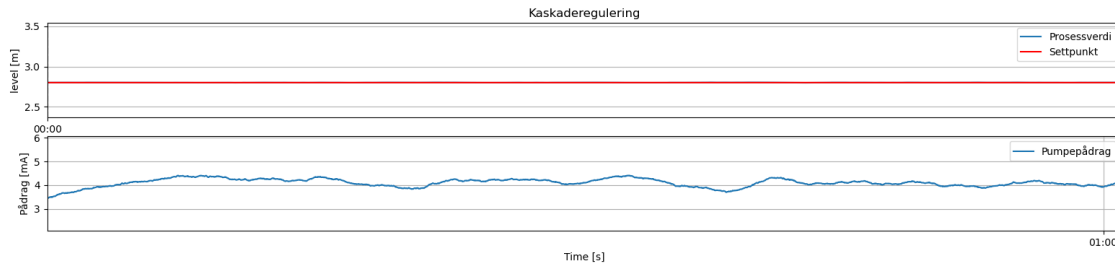
Figur 5-7 Måleavvik logikk

hvor *prosessverdi* kan være nivå eller gjennomstrømning, og *måleavvik* er forventet unøyaktighet i det aktuelle instrumentet. Dette danner grunnlag for metoden *PVstøy*, og det blir generert en tilfeldig verdi som finnes innenfor et intervall lik  $PV \pm PV_{avvik}$ . Denne variabelen er direkte knyttet til regulatorens pådragsfunksjon. Derfor er det hensiktsmessig å implementere et målefilter. Figur 5-8 viser kaskadereguleringen kjørt i en time med 0.5% måleavvik for både nivåmålingen og strømningsmålingen. Ut fra denne figuren kan man konkludere med at pumpen vil kjøre ujevnt. Dette vil antagelig i lengden være skadelig for en pumpe.



Figur 5-8 Simulert måleavvik

For å dempe støyen er det implementert et middelværdifilter [25]. Middelværdifilteret tar inn et «vindu» som inngang (et array med verdier), finner gjennomsnittsverdien til «vinduet» ved å dele summen av verdiene på antall verdier, for så å gi ut middelværdien som utgang [25]. Se Figur 5-9 for effekten filteret har på både nivåmålingen og pådraget. Her er det valgt størrelse på tidsvindu lik 100. Merk at større tidsvindu gir en mer stabil måling, men øker samtidig tidsforsinkelsen mellom filtrert verdi og den reelle prosessverdien (ved endring i prosessen) [25]. Dette vil forplante seg som et treghetsmoment på pådragssignalet til regulatorene. Det er verdt å nevne at tidsvindu til middelværdifilteret er et tuningsparameter. Parameteret bør finnes eksperimentelt for en gitt prosess [25].



Figur 5-9 Middelvefilter

### 5.3.4 Endring fra Python til Jython

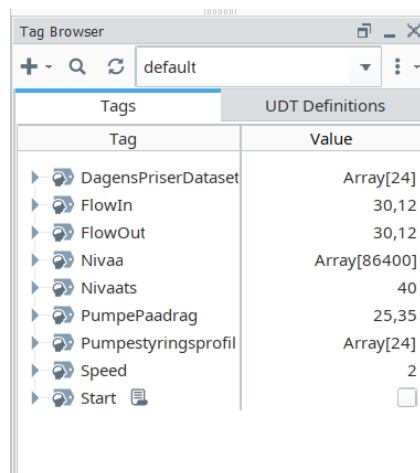
Etter testing av simulatoren i Python ble den konvertert til Jython, slik at det kan brukes i Ignition. I Jython er ikke alle Python-bibliotekene tilgjengelige. Det var derfor nødvendig å gjøre endringer i koden. NumPy er et av de mest populære bibliotekene [26], og det ble brukt i simulator-programmet utviklet i Python. I Jython er NumPy-biblioteket erstattet med array- og math-bibliotekene. Se Figur 5-10 for importering av biblioteket samt oppretting av array.

```
#Bibliotek
import math
import time
import random
import array

# Oppretter Pumpestyringsprofil array
timer = 24
styringsProfilArray = array.array('d', [0] * timer)
stromPrisArrayDagensSekundforSekund = []
```

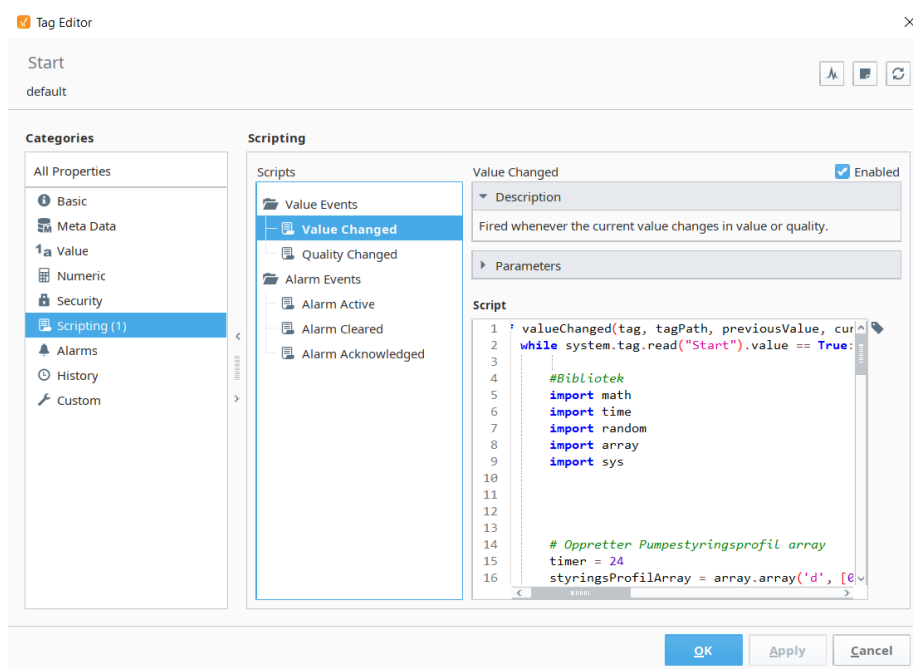
Figur 5-10 Import av bibliotek og oppretting av array i Jython

Når man skal visualisere verdiene fra simulatoren, er det i Python brukt biblioteket matplotlib.pyplot for å plote verdiene i en graf. Men i Ignition blir verdiene i simulatoren skrevet til tags, og tilgjengeliggjøres via OPC UA. Figur 5-11 viser hvordan «tag browser» ser ut. Disse verdiene anvendes blant annet i Perspective-modulen for fremvisning i dashbordet.



Figur 5-11 «Tag Browser» i Ignition

For å kunne starte skript via Perspective-modulen, er det brukt en tag for boolske verdier. Denne tagen er linket opp med en bryter i Perspective-modulen og vil endre seg fra «False» til «True» når man trykker på den. Skriptet er lagret i et «value changed»-event slik at skriptet kjører når verdien til tagen endrer seg. Det er ønskelig at skriptet kun skal kjøre når taggen er «True». For å oppnå dette, er det lagt til et boolsk flagg sammen med while-løkken, se Figur 5-12 for hvordan dette er gjort. Ved å benytte tag på denne måten, er det mulig å kjøre simulatoren fra en annen OPC UA klient.



Figur 5-12 «Value changed» skript i tag

### 5.3.5 Sanntidssimulering av smartstyring

Sanntidssimuleringen kjøres i et skript på samme måte som strømprismodule og smartstyringsmodule, i en tag som bruker kan endre fra «false» til «true». Når taggen endres til «true», starter simulatoren og kjører i den hastigheten brukeren velger i web-dashbordet. Verdiene som simulatoren trenger, blir hentet fra ulike tags som f.eks. radius for utregning av areal og styringsprofilen til pumpen. Simulatoren produserer verdier som blir skrevet til tags og brukes av web-dashbordet til fremvisning. Se Figur 7-4 for å se sanntidssimuleringen.

Det blir benyttet initialbetingelser som vises når simulatoren ikke kjøres for å ha samme startposisjon hver gang. Initialverdiene som er skrevet inn er strømning for inn- og utløp satt til null, pumpepådraget satt til null og nivået i tanken satt til 40 prosent. Nivået i tanken er satt til 40 prosent fordi simulatorens initiale verdi er 40 prosent (2.8m).

Opprinnelig ble skriptet utviklet i Python-miljøet, og for-løkker ble brukt for å iterere gjennom prosessutviklingen ved bruk av Eulers metode [27]. Dette viste seg å være en utfordring i Ignition fordi skriptkonsoll i Ignition ikke har noen god måte å terminere for-løkker på etter den har begynt. For å løse dette problemet ble for-løkker erstattet med while-løkker, sammen med et boolsk flagg som vist i Figur 5-13.

```

%% Program-Loop

# Denne løkken simulerer fullt innløp
i = 0
#for i in range ( f_meas_delay):
while i < f_meas_delay and system.tag.read("Start").value == True:
    f_meas_delay_array[i] = init
    i += 1

t = t_start
#for t in range (t_start, t_stop):
while t < t_stop and system.tag.read("Start").value == True:

    speed = system.tag.read("Speed").value
    # forsinket innstrømning i tanken med 120 iterasjoner | bare valgt et tall her
    f_in_delayed = delay_array[-1]
    delay_array[1:] = delay_array[0:-1]
    delay_array[0] = minPumpe.FlowIn()

```

Figur 5-13 Erstatter for-løkke med while-løkke

### 5.3.6 Hurtig-simulert smartstyring

I vinduet strømpriser i web-dashbord blir det benyttet en hurtig simulator som kjøres ferdig på kort tid. Dette er for å se hvordan nivået utvikler seg i tanken utover dagen med dagens strømpriser og et fast settpunkt. Programmet har en funksjon som viser hvor dyrt det er å kjøre kaskaderegulering uten, og med smartstyringen et helt døgn. For å få dette til, er det laget en for-løkke som kjører to ganger. Når første iterasjon kjøres, er det kun kaskaderegulering som simuleres, og under andre, er det kaskade med smartstyring. På denne måten kan det bevises om det er mer økonomisk å kjøre med smartstyringen i forhold til kaskaderegulering uten smartstyring. Dette vises i Figur 5-14.

```

# Simulerer kaskaderegulering uten smartstyring
if sim == 0:

    # Generer nytt pådrag til pumpe
    minPumpe.u = innerLoopRegulator.Paadrag()

    # Lagring av verdier fra kaskaderegulering
    minDataLagring.h_t_array[t] = minNivaamaaler.PVstoy()
    minDataLagring.f_in_t_array[t] = minStromningsmaaler.PVstoy()
    minDataLagring.u_f_array[t] = minPumpe.u
    minDataLagring.pris_kaskade_array[t] = minPumpe.u * pumpe_energi_forbruk_per_s_per_mA * StromPrisSekundforSekund[t]

    # Lagrer pris
    pris_kaskade += minDataLagring.pris_kaskade_array[t]

# Simulerer kaskaderegulering med smartstyring
elif sim == 1:

    # Generer nytt pådrag til pumpe dersom smartstyringen tillater det
    if StyringsProfilSekundforSekund[t] == 1:
        minPumpe.u = innerLoopRegulator.Paadrag()

    elif innerLoopRegulator == False:
        minPumpe.u = innerLoopRegulator.Paadrag()
    else:
        minPumpe.u = 0

    # Lagring av verdier fra smartstyring
    minDataLagring.h_t_array_smart[t] = minNivaamaaler.PVstoy()
    minDataLagring.f_in_t_array_smart[t] = minStromningsmaaler.PVstoy()
    minDataLagring.u_f_array_smart[t] = minPumpe.u
    minDataLagring.pris_smartstyring_array[t] = minPumpe.u * pumpe_energi_forbruk_per_s_per_mA * StromPrisSekundforSekund[t]
    #pris_smartstyring += minDataLagring.pris_smartstyring_array[t]

```

Figur 5-14 Kaskaderegulering med og uten smartstyring

Nivåverdiene fra simulatoren blir lagret til et array-tag for å kunne vise endringene som skjer i løpet av en dag. Dersom man vil endre settpunktet, er det nødvendig å kjøre simulatoren på nytt. Hurtigsimulatoren viser ett døgns forløp med brukervalgte parameter og systemverdier. Hurtigsimulatoren kan derfor brukes som verktøy til å teste og validere nye parametere og systemverdier før de settes i drift. Se Figur 7-6 for hurtig-simulert smartstyring.

## 6 Programmering av smartstyring

Dette kapitlet beskriver hvordan smartstyringen skal påvirke prosessen og i hvilken grad smartstyringen er «smart». Det er diskutert hvilke reguleringsstrategier som brukes i smartstyringen, og det blir en sammenlikning mellom kaskade uten og med smartstyring for å undersøke hvor mye penger som kan spares. Utviklingen av smartstyringen er diskutert i form av revisjoner hvor det er begrunnet hvilke som egner seg best for formålet, samt fordeler og ulemper.

### 6.1 Reguleringsstrategi

Dette underkapittelet beskriver hvordan de ulike revisjonene av smartstyring er utviklet, samt ambisjoner om videreutvikling.

Smartstyringen vil fungere som en forrigling på pådragssignalet til pumpen. Det dannes en variabel med navn «styringsprofil» som har verdi «0» eller «1». Dette fungerer i praksis som en boolsk «true» eller «false». Med parametere for strømpriser vil smartstyringen kunne regulere i et tidsrom hvor prisen er lav. Styringsprofilen tilgjengeliggjøres via OPC UA kommunikasjon slik at andre moduler kan bruke den.

Smartstyringen er utstyrt med høy- og lavkutt verdier som vil overstyre smartstyringen dersom en viss terskel blir brutt. Eksempelvis i tilfellet hvor vannstanden i reservoaret blir kritisk lav samtidig som strømprisen er høy, iverksettes tiltak uavhengig av om det er ineffektivt i forhold til strømpris i øyeblikket eller ei.

Dersom minimums-terskelen blir brutt, vil smartstyringen fylle styringsprofil-arrayet med «1» fra gjeldende tidspunkt til en halvtime frem i tid. Det antas at ved å fylle bassenget i bare en halvtime, finner smartstyringen en nokså god balanse mellom strømutfgifter og for lavt nivå.

Minimums-terskel i et høydebasseng skal til vanlig planlegges nøye. Denne terskelen dikteres av en rekke faktorer, blant annet av sikkerhetsreserve og brannvannreserve [28]. Videre kan terskelen beregnes dersom forventet vannforbruk er tilgjengelig. Dette er ikke aktuelt i denne sammenhengen. Derfor er lavkutt midlertidig satt til 1.4m (20%), og høykutt til 5.6m (80%) ettersom bassenget er 7m høyt. Det er ikke er noe pådragsorgan for å redusere nivået i tanken i dette systemet. Høy-kutt verdien vil derfor ikke ha noen praktisk virkning, men det legges til rette for det.

#### 6.1.1 Revisjon 1

Første revisjon av smartstyringen henter inn dagens strømpriser via OPC UA som er publisert av strømpris API-et, beskrevet i kapittel 4. Deretter finner den gjennomsnittsprisen for døgnet, for så å skrive til styringsprofil-arrayet for hver time avhengig av om prisen i en gitt time er over eller under gjennomsnittsprisen. Dersom strømprisen i en gitt time er under gjennomsnitt, blir det skrevet «1» til styringsprofil-arrayet. Dersom strømprisen er over gjennomsnitt, blir det skrevet «0». Denne løsningen er enkel, men samtidig effektiv, og er videre diskutert i kapittel 9.5.1.

#### 6.1.2 Revisjon 2

Andre revisjon av smartstyringen henter inn dagens strømpriser på samme måte som i revisjon 1. Men i stedet for å finne gjennomsnittsprisen, finner den antall timer pumpen trenger å kjøre

på maks pådrag for å motvirke forventet vannforbruk for døgnet. Forventet vannforbruk i første omgang vil være en funksjon av nivået i tanken med en idé om å kunne predikere vannforbruk basert på tidligere historikk. Hensikten er å kun fylle det som er nødvendig i det tidsintervallet det er billigst.

Smartstyringen bruker følgende likning for å finne best mulig styringsprofil for døgnet, se formel 6.1

6.1

$$n = \frac{\Delta V}{q_{inn_{max}}}$$

hvor  $q_{inn_{max}}$  er pumpekapasitet ved maks pådrag,  $\Delta V$  er forventet volumendring i tanken i løpet av døgnet i forhold til settpunkt og  $n$  er antall timer avrundet til nærmeste heltall oppover. Tallet  $n$  blir brukt til å finne de  $n$  billigste timene i døgnet.

Det skal nevnes at denne revisjonen kun vil fungere på bakgrunn av at forventet volumendring stemmer overens med faktisk volumendring. Formel 6.2 viser uttrykket for utstrømning brukt i simulatoren. Dette er kun en funksjon av det hydrostatiske trykket i tanken og vil neppe oppføre seg likt som virkelig vannforbruk.

6.2

$$q_{ut} = \sqrt{h} * C_{v2}$$

hvor  $\sqrt{h}$  er kvadratroten av høyden i tanken i gjeldende tidskritt og  $C_{v2} = 0.005$  er strømningskoeffisienten til utløpsventilen. Grunnet manglende mulighet til å regne ut strømningskoeffisient, ble 0.005 valgt. Dette gir muligheten til å finne  $\Delta V$  som nevnt i Formel 6.1 slik, se formel 6.3.

6.3

$$\Delta V = q_{ut} \left[ \frac{m^3}{s} \right] * 60 * 60 * 24$$

hvor  $60 * 60 * 24 = 86400s$  lik ett døgn. Videre er det mulig å endre formel 6.1 slik, se formel 6.4.

6.4

$$n = \frac{\Delta(V + V_0)}{q_{inn_{max}}}$$

hvor  $\Delta V_0$  er volumforskjellen fra PV til SP og kan finnes slik, se formel 6.5

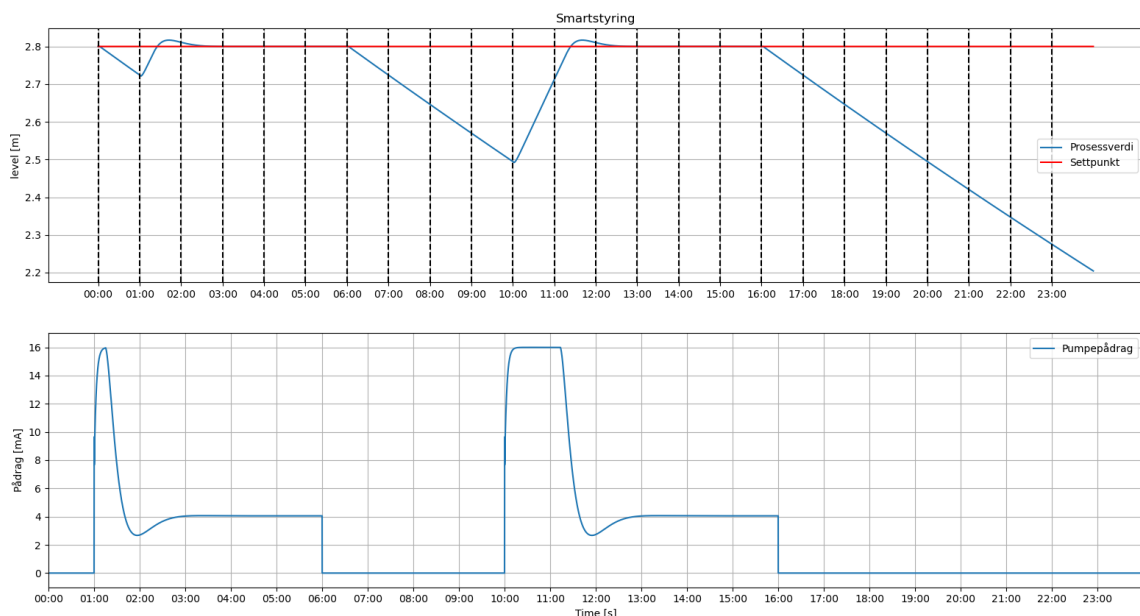
$$\Delta V_0 = (SP - PV_{init}) * r^2 * \pi$$

Denne løsningen tar med avviket i systemet i et døgnskifte, sammen med forventet volumutstrømning for dagen til å regne ut total mengde som må fylles for å nå settpunkt.

Den praktiske forskjellen mellom formel 6.1 og formel 6.4 er deres evne til å holde settpunkt mot deres evne til å spare penger. Det finnes argumenter for begge, men til dette formålet har evnen til å spare penger blir vektet strengere enn evnen til å holde settpunkt. Dette kan ha utslag på opptil 0.56 bar på det hydrostatiske trykket ut på vannettet. Det antas at dette ikke vil ha noen konsekvenser av betydning for verken utstyr eller forbruker.

## 6.2 Samarbeid mellom smartstyring og prosessimulator

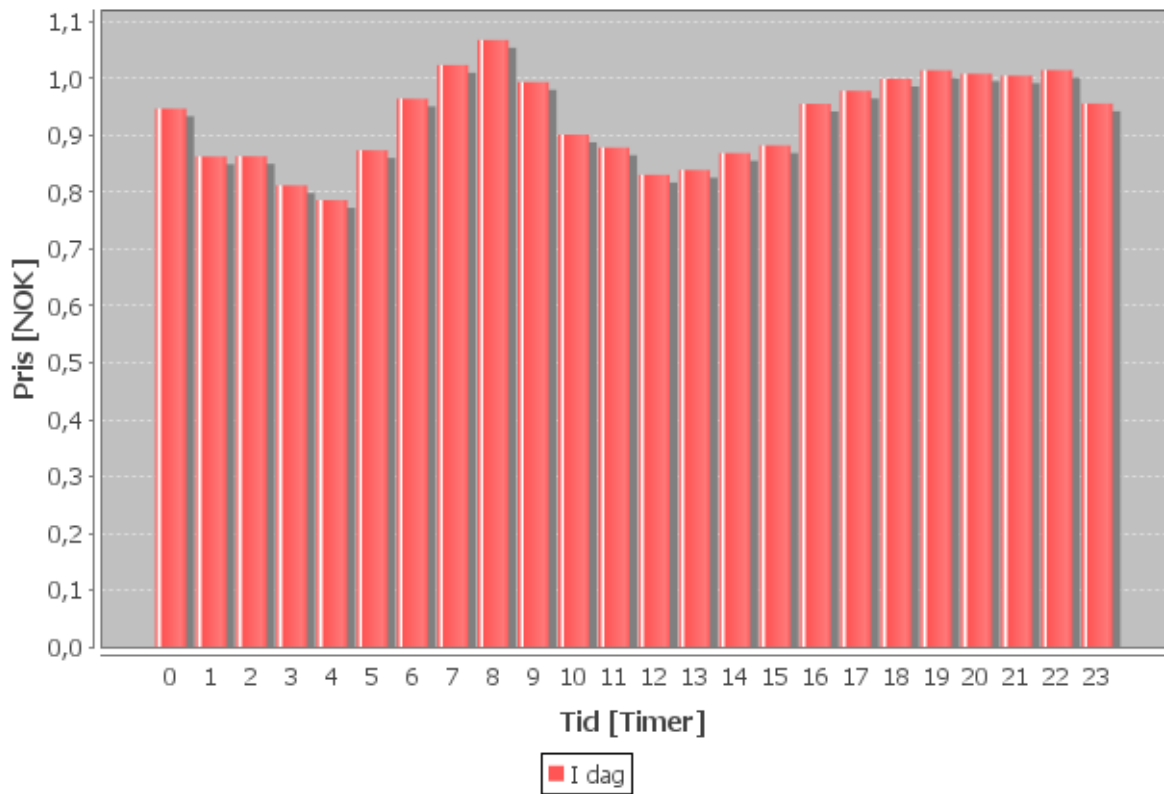
Prosessimulatoren bruker styringsprofilen laget i smartstyringen sammen med kaskaderegulering for å regulere nivået mot settpunkt. Figur 6-1 viser simulatoren kjørt sammen med første revisjon av smartstyringen. Se også Figur 6-2 for å se sammenhengen mellom pumpepådraget og strømprisene for døgnet, og legg merke til at pumpepådraget er omvendt proporsjonalt med strømprisen. Smartstyringen vil til tider fungere nokså likt som en av-og-på-regulator. Men i praksis er det fremdeles en kaskaderegulering som forklart i kapittel 5. Det antas at frekvensomformerer tilhørende pumpestasjonen har opprampningstid. Fordi det kan resultere i uheldige slag på vannettet hvis omformereren går fra null til maks umiddelbart.



Figur 6-1 Første revisjon av smartstyring

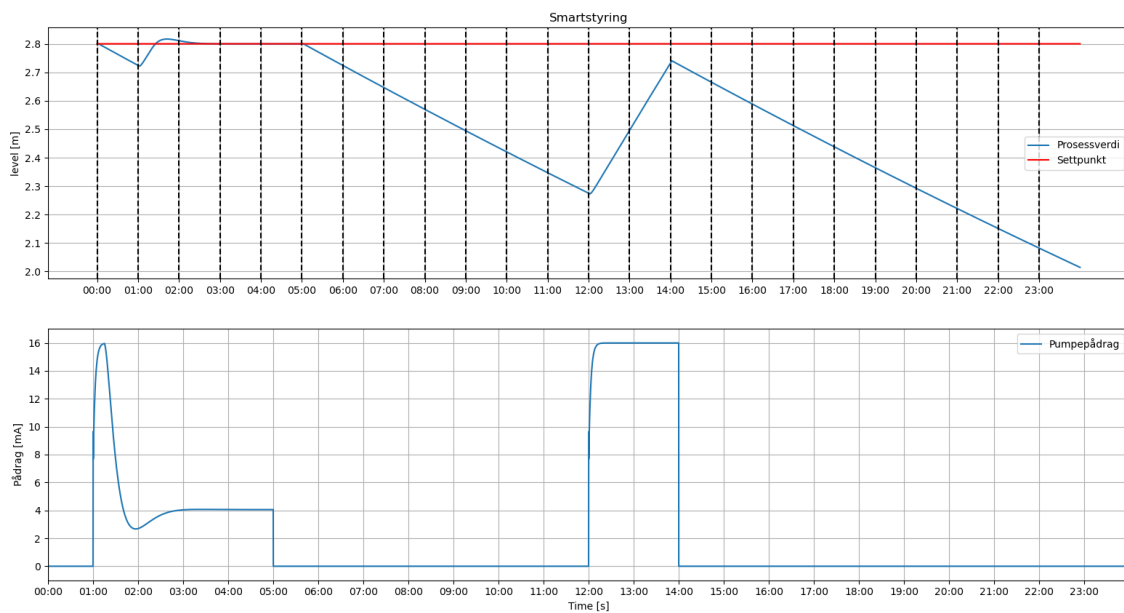


### Strompriser



Figur 6-2 Strømpriser fra 24.03.23

Figur 6-3 viser andre revisjon av smartstyringen med samme strømpriser, initialhøyde og settpunkt i tanken. Denne løsningen er nokså lik som revisjon 1, men strengere vektet når det kommer til strømforbruk.



Figur 6-3 Andre revisjon av smartstyring

### 6.3 Kostnadsbesparelse med smartstyring

Dette underkapittelet beskriver hvordan utregning av driftspris er gjort, og gir en sammenlikning av revisjonene som er utviklet.

Det kan være interessant å vite hvor mye en kan spare ved å ta i bruk smartstyringen. Det kan eksempelvis være et mål på hvorvidt implementering av denne løsningen er lønnsomt, og hvor lang tid det vil ta før det blir lønnsomt.

#### 6.3.1 Utregning av driftspris

Det er benyttet strømpris-arrayet som er beskrevet i kapittel 4. Arrayet er i utgangspunktet 1x24 som inneholder priser i form av NOK/kWh. Dette er gjort om til NOK/kWs slik at det samsvarer med tidskrittet i simuleringen. Total energikost blir beregnet på følgende måte, se formel 6.6.

$$Kostnad = \sum_{t_{start}}^{t_{stopp}} P\ddot{a}dragssignal * Pumpe_{energiforbruk} * Str\ddot{o}mpris \tag{6.6}$$

Denne utregningen blir utført på kaskaderegulering både med og uten smartstyring, med de samme strømprisene. Hvor kostnaden til smartstyringen bruker tidligere kaskaderegulering uten smartstyring som referansepris.

#### 6.3.2 Sammenlikning av revisjon 1 og revisjon 2

Figur 6-4 viser en sammenlikning mellom smartstyring revisjon 1 og revisjon 2. Her er sammenlikningen gjort med tanke på hvilken av de to som er mest strømeffektiv. Analysen ble gjort ved å la styringsprofilen generert av smartstyringen være eneste varierende faktor. Figuren viser at revisjon 1 resulterte i en total besparelse på omtrent 2000 NOK, mens revisjon 2 oppnådde en besparelse på omtrent 3000 NOK. Basert på disse tallene kan det konkluderes med at revisjon 2 er mer kostnadsbesparende.

For å få sammenhengende resultater for en ukes simulering, ble prosessverdien i slutten av ett døgn simulering brukt som start-verdi for neste døgn.

Revisjon 1				Revisjon 2			
dato	h_init [m]	besparelse [NOK]	besparelse [%]	dato	h_init [m]	besparelse [NOK]	besparelse [%]
17.04.2023	2,8000	139,1544	9,5387	17.04.2023	2,8000	539,2241	36,9524
18.04.2023	2,7995	88,0665	6,5930	18.04.2023	2,3500	345,4127	25,8721
19.04.2023	2,7997	409,6238	32,6484	19.04.2023	2,3517	403,8168	32,1921
20.04.2023	2,4216	191,2278	14,9105	20.04.2023	2,2773	222,5522	17,3498
21.04.2023	2,2757	79,1455	6,3472	21.04.2023	2,2805	385,5150	30,9171
22.04.2023	2,3482	504,2290	44,4826	22.04.2023	2,2786	524,9099	46,3073
23.04.2023	2,2765	546,6368	48,0314	23.04.2023	2,2791	593,1160	52,1179
Gjennomsnittelig [%]			23,2217	Gjennomsnittelig [%]			34,5298
Total besparelse [NOK]			1958,0838	Total besparelse [NOK]			3014,5467

Figur 6-4 Sammenlikning av revisjon 1 og 2 i uke 16, 2023

Legg merke til at når nivået i reservoaret vandrer mellom lavkutt-terskel og settpunkt i løpet av et døgn, vil strømeffektiviseringen være best mulig. Samtidig som den gjør settpunktet ubetydelig.

## 7 Presentasjon av info i web-dashbord

Dette kapitlet tar for seg bruken av «root container» for utvikling av det grafiske brukergrensesnitt. Kapitlet forklarer også hvordan grensesnittet er bygd opp og hvordan det samarbeider med simulator, smartstyringen og de innhentede strømprisene. Det blir også drøftet fordeler og ulemper, og hva som er optimalt for skjermssystemet som er tiltenkt å opereres via en HMI-skjerm.

### 7.1 Perspective Sessions

Perspective Sessions [29] kan kjøres på enhver enhet som støtter en nettleser, og det er også mulig å kjøre økter i Ignition Perspective-appen. Dette gir en unik mulighet for støtte til mobile enheter og nettbrett [29]. Perspective-funksjonene er designet etter et responsivt design paradigme, som betyr at de vil tilpasse seg enheten og skjermstørrelsen på en smidig og brukervennlig måte [30]. Forutsetningen er at det benyttes de riktige «root containers» til riktig formål [31] [30].

### 7.2 Design av brukergrensesnitt

Ignition Perspective-modul brukes til å designe et grafisk brukergrensesnitt som har som formål å visualisere og kontrollere systemet. Dette brukergrensesnittet fungerer sammen med skript som henter inn dagens strømpriser, samt med en simulator og en smartstyring for systemet. For å strukturere brukergrensesnittet brukes ulike «views» som definerer forskjellige nivåer av grensesnittet. Hvordan de ulike brukergrensesnitt-sidene er deklarerert er vist i Figur 7-1.

For utviklingen av brukergrensesnittet er «coordinate» benyttet som «root container» [32]. Denne containeren gir ikke et like responsivt grensesnitt ved endring av skjermoppløsning [30]. Men siden dette er et skjermssystem som er tiltenkt å opereres via en HMI, så er «coordinate container» passende [30].

«Coordinate container» er en type «container» som gir utviklere mulighet til å plassere elementer på et skjermområde ved hjelp av koordinater, i stedet for å benytte seg av layout-komponenter som strekker seg og justerer seg til skjermen [32]. «Root container» refererer til den øverste "containeren" i grensesnitt-hierarkiet [30].

En ulempe ved å benytte seg av «coordinate container» som «root container», er at det kan føre til at grensesnittet er mindre responsivt. Dette skyldes at elementene ikke vil endre størrelse og posisjon automatisk når skjermstørrelsen endres. Imidlertid er «coordinate container» optimalt for skjermssystemer som er tiltenkt å opereres via HMI [32]. HMI-systemer opererer typisk på faste skjermer som har en bestemt størrelse og oppløsning. Dette betyr at en fast skjerm ikke trenger å være like responsivt som et grensesnitt som skal vises på forskjellige enheter med forskjellige størrelser og oppløsninger [30].

Figur 7-1 Sidekonfigurasjon av «views» i brukergrensesnittet

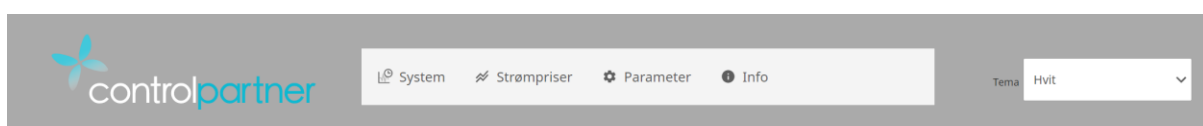
## 7.3 Grafisk brukergrensesnitt

Dette underkapittelet beskriver hvordan navigering av det grafiske brukergrensesnittet er bygget opp og hvordan de utviklede visningsområdene «System», «Strømpriser», «Info» og «Parameter» er utformet. En detaljert beskrivelse av systemet og brukergrensesnittet er tilgjengelig i Vedlegg G Brukerveiledning.

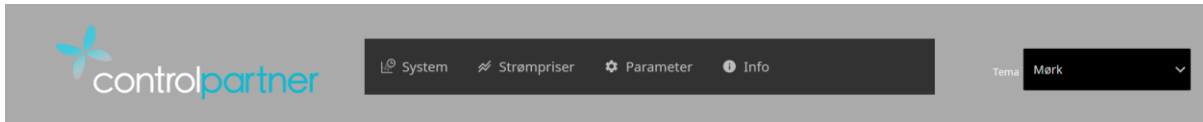
### 7.3.1 Navigering

Det er utviklet en navigasjonsbar øverst i brukergrensesnittet som fungerer som en «header» i alle visningsområdene. Dette fordi det skal være enkelt for brukeren å navigere seg mellom disse fire visningsområdene. Det er også mulig for brukeren å velge ulike temaer for brukergrensesnittet som for eksempel tema «mørk». Navigasjonsbaren er vist med hvitt tema i Figur 7-2, og med mørkt tema i Figur 7-3.

Ved å trykke på det enkelte visningsområde, utføres en aksjon som gjør at brukeren blir navigert til det tilhørende visningsområde som er deklart i knappen sin «target» funksjon. I denne funksjonen deklarerer URL for hvert view.



Figur 7-2 Navigasjonsbar med hvitt tema

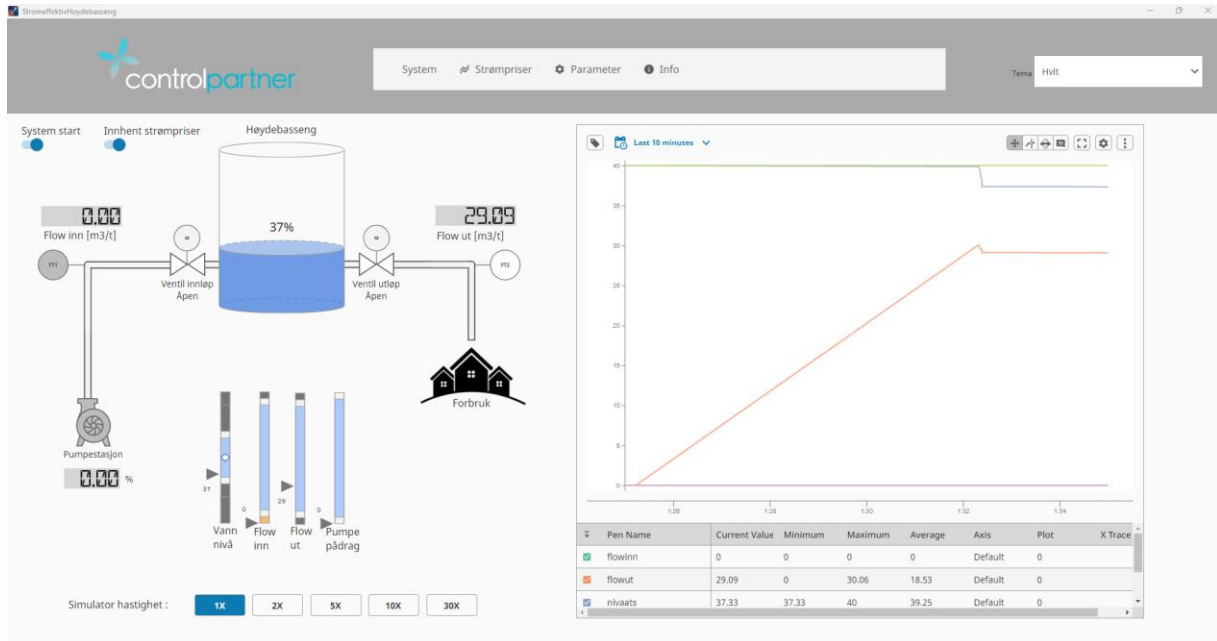


Figur 7-3 Navigasjonsbar med mørkt tema

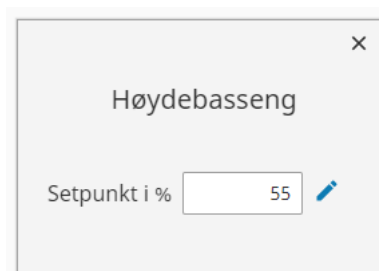
### 7.3.2 «System»-visningsområde

I system-visningsområde har brukeren full oversikt over prosessen. Grensesnittet er bygd opp slik at det er en enkel systemoversikt i form av visuelle komponenter, rør og basseng. Dette gir en overordnet oversikt over hvordan systemet er satt sammen fysisk. En grafisk fremvisning av hvordan dette visningsområde ser ut, er vist i Figur 7-4. Prosessverdier blir fremvist grafisk i form av grafer og enkle numeriske felt. Det er også benyttet bevegelige analoge indikatorer for å gi en bedre oversikt over prosessverdiene i forhold til tilhørende prosessgrenser. Dette gjør det lettere å tolke prosessverdiene for operatører. For illustrasjon av vannsøylenivået er det i tillegg benyttet en dynamisk illustrativ tank som representerer høydebassenget. Høydebasseng illustrasjonen er programmert med et «click event» som gir brukeren et «popup»-vindu hvor det er mulig for operatør å endre systemets settpunkt. En grafisk visning av hvordan dette popup-vinduet ser ut er vist i Figur 7-5.

Et «power chart» [33] brukes for fremvisning av prosessinfo. Denne grafen er dynamisk og vil vise prosessens verdier i sanntid. For visning av verdier er det brukt «historian tags» for å visualisere historiske data, noe som er nødvendig for «power chart» [33]. For å få dette til er det opprettet en «Internal Historian» under «Historian Tag Provider» i Ignition [34]. Denne har begrenset lagring av historiske data [34], men fungerer gunstig etter formålet for dette prosjektet. Tagger som skal være historiske og kunne vises i en graf, må defineres. En utfordring med visningen i grafen som vist i Figur 7-4 er at grafen plottet hvert sekund i sanntid. Simulatoren kjører fortere og oppdaterer verdien i forhold til hastigheten brukeren har valgt. Grafen sin x-akse er ikke linket opp mot simulatorens hastighetsøkning. Dette resulterer i at grafen ikke klarer å fremvise simuleringens forløp etter den faktiske tiden prosessen vil bruke, men viser den økte proseshastigheten etter tiden i sanntid. På denne måten vil prosessen kunne simuleres fortere, men plottet vil derfor se veldig sammenslått ut om simulatorens kjører for fort.



Figur 7-4 Grafisk brukergrensesnitt av systemet med eksempelverdier

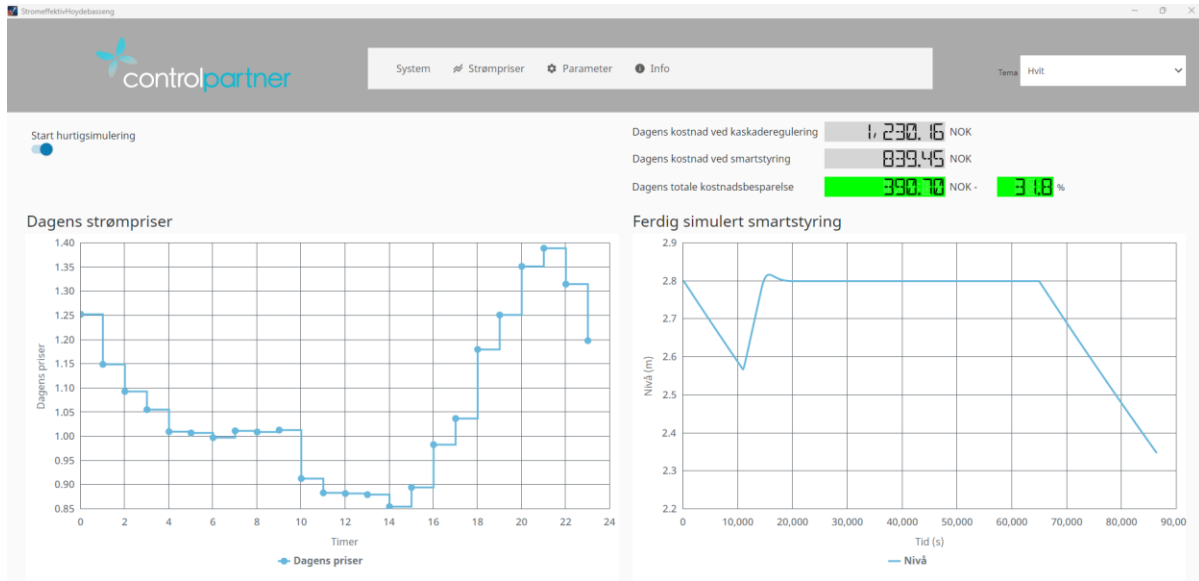


Figur 7-5 Popup-vindu for endring av settpunkt

### 7.3.3 «Strømpriser»-visningsområde

Det er utviklet et visningsområde for strømprisene som er hentet fra web-API modulen. Visningsområdet viser dagens strømpriser i en graf for å gi en visuell oversikt over strømknudene nåværende døgn. Dette gjør det enklere å forstå grunnlaget for styringsprofilen som blir gitt av smartstyringen.

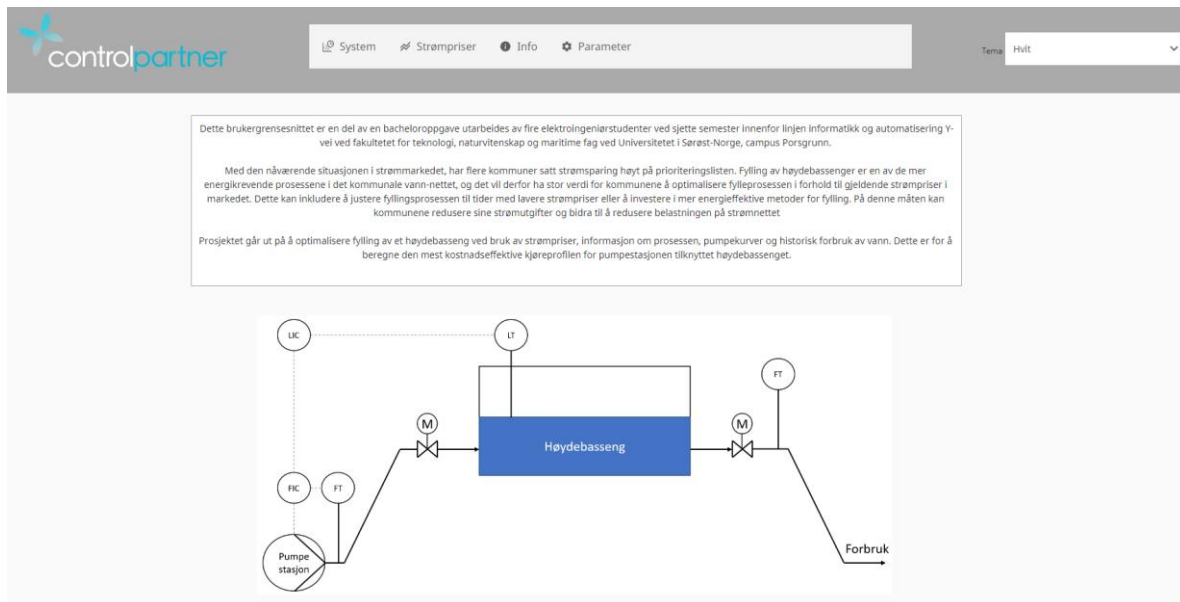
Det vurderes praktisk at operatøren har tilgang til å overvåke mer enn kun den økonomiske besparelsen, derfor vil døgnetts totale kostand bli vist i to separate numeriske felt. Et felt for døgnetts totale kostnad ved en ren kaskaderegulering, og et annet felt for døgnetts totale kostnad ved bruk av smartstyringen. Dette gjør at operatøren til enhver tid har full oversikt over hvor økonomisk effektiv smartstyringen av høydebassenget faktisk er. Det analoge feltet som viser den økonomiske besparelsen, har en grønn bakgrunnsfarge som gir et tydelig blikkfang og indikerer en positiv vekst i form av penger spart. «Strømpriser»-visningsområde er grafisk vist i Figur 7-6.



Figur 7-6 Grafisk brukergrensesnitt av «strømpriser»-vindu med data fra 01.05.2023

### 7.3.4 «Info»-visningsområde

For fremvisning av prosjektets bakgrunn og hensikt er det opprettet et eget visningsområde kalt «Info». Dette visningsområde benyttes til å gi brukeren en overordnet oversikt. I visningsområde er det et tekstfelt hvor prosjektinfo fremvises, samt en overordnet skisse over prosessen. Visningsområde har som formål å formidle oppgaven grunnleggende til utenforstående. Visningsområde «Info» er vist i Figur 7-7.



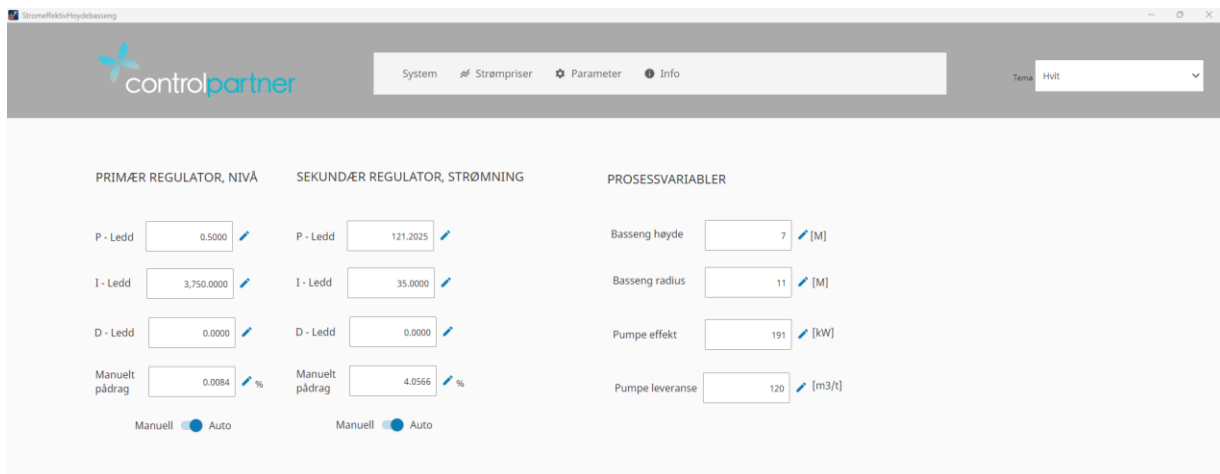
Figur 7-7 Grafisk brukergrensesnitt av «Info»-vindu

### 7.3.5 «Parameter»-visningsområde

Dette visningsområde er designet slik at brukeren av systemet skal kunne tilpasse parametere til et hvert system. Dette gjør systemet mer tilpasningsdyktig til å kunne implementeres i ulike prosesser. Basseng-info slik som høyde og radius, er meget relevant for å kunne fastslå volum og vil variere fra system til system. Ved en verdiendring vil den tilknyttede taggen bli overskrevet med den nye verdien.

Det samme gjelder for pumpeeffekt og pumpeleveranse. Effekt vil variere fra pumpe til pumpe og vil gi utslag på strømforbruk. For å riktig kunne kalkulere kostnadsbesparelsen blir pumpeeffekten endret i visningsområde slik at den stemmer overens med den faktiske effekten til pumpen. Pumpeleveranse har betydning for hvor mye pumpen kan levere ved gitt effekt, og vil gi et estimat på hvor lang tid det vil ta å fylle opp bassenget til ønsket nivå.

Brukergrensesnittet gir operatøren mulighet til å endre regulatorparametere. Ved implementasjon av smartstyringen i en allerede eksisterende prosess, vil denne prosessens regulatorparametere enkelt kunne deklarerer i brukergrensesnittet. Disse parametere blir deretter skrevet til tagger, som igjen benyttes i simulatoren. En grafisk fremvisning av «Parameter»-visningsområdet er vist i Figur 7-8.



Figur 7-8 Grafisk brukergrensesnitt av «Parameter»-vindu

## 7.4 High performance human-machine interface

Et high performance human-machine interface (HPHMI) [35] refererer til et grensesnitt mellom mennesker og maskiner som er spesielt utviklet for å forbedre ytelsen og effektiviteten til systemet. HPHMI fokuserer på å optimalisere interaksjonen mellom mennesker og maskiner [35], slik at systemet fungerer bedre, og brukeren kan utføre oppgavene sine mer effektivt

NORVAR (Norsk Vann) er en norsk interesseorganisasjon for vann og avløpsbransjen [36]. De har utviklet standarder og retningslinjer for design av brukergrensesnitt [37]. Dette er for å sikre en brukeropplevelse som er enhetlig, effektiv og enkel.

Ut fra NORVAR-rapport 153 [37] for design av brukergrensesnitt kan det tolkes følgende fokusområder: Konsistent design, brukervennlighet og symbolbruk.



### 7.4.1 Knapper

For endring av hastigheten til simulatoren er det benyttet fem individuelle knapper med faste verdier. Disse knappene endrer verdien til taggen «SimSpeed». Denne taggen er anvendt i skriptet til simulatoren som deklarerer hastigheten. For å enkelt gi operatøren av HMI-en en tilbakemelding over hvilken hastighet simatoren blir kjørt i, er taggen linket «Bidirectional» til knappen. Dette gjør at knappen har en funksjon som kan lese av verdien i taggen, samt at den har muligheten til å overskrive den nåværende verdien. Knappene er vist i Figur 7-9. Merk at ved å simulere i «1X» tilsvarer prosessen i sanntid.



Figur 7-9 Knapper for simulator hastighet

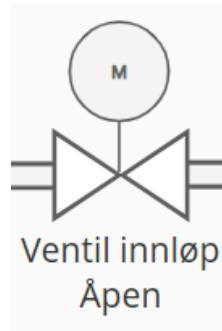
### 7.4.2 Symboler

Det er benyttet Ignition sine innebygde instrumentsymboler. I henhold til NORVAR-rapport 153 [37] er de brukte symbolene egnet etter NORVAR sine normer for symboler i driftskontroll-systemer for VA-sektoren. Symbol for pumpestasjon avviker noe fra hva NORVAR-rapport 153 [37] tilsier, men siden det er brukt Ignition sine standardsymboler vurderes symbolet til å være egnet [38] [39] [40]. Symbolene brukt er vist i Figur 7-10. Symbolet har en grafisk illustrasjon som indikerer at pumpen er i drift ved at sentrifugaldelen av pumpe symbolet beveger seg i en sirkulær bevegelse og symbolet endrer farge fra lys grå til hvit.

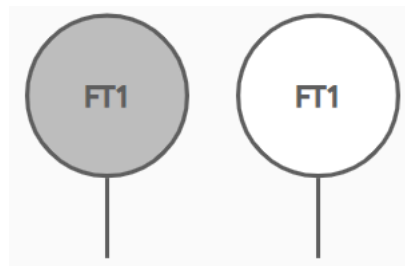


Figur 7-10 Symbol for pumpestasjon i inaktiv (grå) og aktiv (hvit) tilstand

Andre symboler brukt avviker ikke fra NORVAR rapport 153 [37], og vurderes til å oppfylle NORVAR sine normer for symbolbruk knyttet til driftskontrollsystemer for VA-sektoren. Dette gjelder symboler brukt for ventiler og mengdemålere. Det er en indikasjon på instrumentene for mengdemåling som indikerer når instrumentet er aktivt og utgir en måling. Denne indikeringen viser at symbolet endrer farge fra lys grå til hvit. Ventilsymbolene brukt er hvite, dette er fordi at ventilene alltid er åpne med mindre annet er bestemt. Motorsymbolet til ventilen er grå, siden motoren er inaktiv og ikke aktiv i drift. Symbolene for ventil og mengdemåling er vist i Figur 7-11 og Figur 7-12.



Figur 7-11 Symbol for motorstyrt ventil

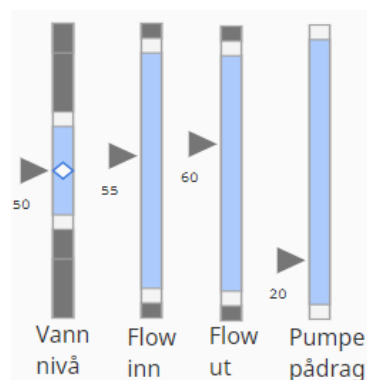


Figur 7-12 Symbol for mengdemåler i inaktiv (grå) og aktiv (hvit) tilstand

### 7.4.3 Bevegelige analoge indikatorer

Det å legge til en rekke tall til på en skjerm, vil ha en negativ effekt på situasjons-bevisstheten til operatøren [35]. Derimot bevegelige analoge indikatorer, som vist i Figur 7-13 representerer disse verdiene på en mer effektiv og ryddig måte. Analoge fremvisninger er et kraftig verktøy for operatørene fordi mennesker intuitivt forstår analoge avbildninger bedre enn en rekke tall på en skjerm [35] [41].

De analoge indikatorene gir operatøren en forståelse av prosessverdiene og ser dem i forhold til prosessen sine satte grenseverdier. Eksempelvis gir den analoge indikatoren operatøren en indikasjon på hvordan vannivået er i forhold til settpunkt, lavkutt- og høykuttverdien.



Figur 7-13 Analoge indikatorer for visning av prosessverdier

## 7.5 Cybersikkerhet

Ved å gi brukeren muligheten til å endre systemets parametere i web-dashbordet er det viktig å ta cybersikkerhet i betraktning [42]. Dette er for å forhindre uautorisert tilgang, datainnbrudd og andre ulike cybertrusler. Noen enkle grep som kan øke sikkerheten er:

- Sikre nettverket
  - Før benyttelse av Perspective anbefales det å sørge for at nettverket er sikkert. Benytt sterke gode passord, aktiver brannmur og pass på at alle tilkoblede enheter er oppdatert med de nyeste sikkerhetsoppdateringene [43].
- Benytt rollebasert tilgangskontroll
  - Perspective gir brukeren muligheten til å kontrollere tilgangen til ulike deler av systemet basert på brukerroller. Det kan derfor benyttes rollebasert tilgangskontroll for å kunne sikre at kun autorisert personell får endre på parametere [43].

Det er ikke tatt i bruk rollebasert adgangskontroll, men dette er noe man kan utvide systemet med i en senere utgave. Det er tatt hensyn til sikkert nettverk i den grad at det kun er benyttet enheter med aktiv brannmur og de nyeste sikkerhetsoppdateringene. Men det er ikke tatt i bruk sterke gode passord. Disse tiltakene bør tas i betraktning i en senere utgave av systemet slik at sikkerheten blir ivaretatt. Videre har Inductive Automation en veiledning for å bevare og øke sikkerheten til systemet [44].

## 7.6 Perspective Workstation

Prosjektet tar i bruk Perspective Workstation for å presentere, overvåke og kontrollere systemet. Perspective Workstation er en applikasjon som fungerer som en oppstarter av en Perspective økt [45]. Programmet må installeres for å kunne brukes. Workstation gir brukeren muligheten til å kjøre den utviklede applikasjonen som en skrivebordsapplikasjon. Applikasjon er designet for HMI-er, panel-PC-er og dedikerte arbeidsstasjoner økt [45].

Perspective Workstation kan kjøre i Windowed-modus og fullskjerm Kiosk-modus ved hjelp av Workstation sine innebygde funksjoner [45]. Disse funksjonene eliminerer distraksjoner fra det underliggende operativsystemet.

## 8 OPC UA kommunikasjon

Dette kapitlet beskriver hva Ignition Gateway er, hvordan OPC UA er brukt og hvordan de ulike modulene kommuniserer med hverandre.

### 8.1 Ignition Gateway

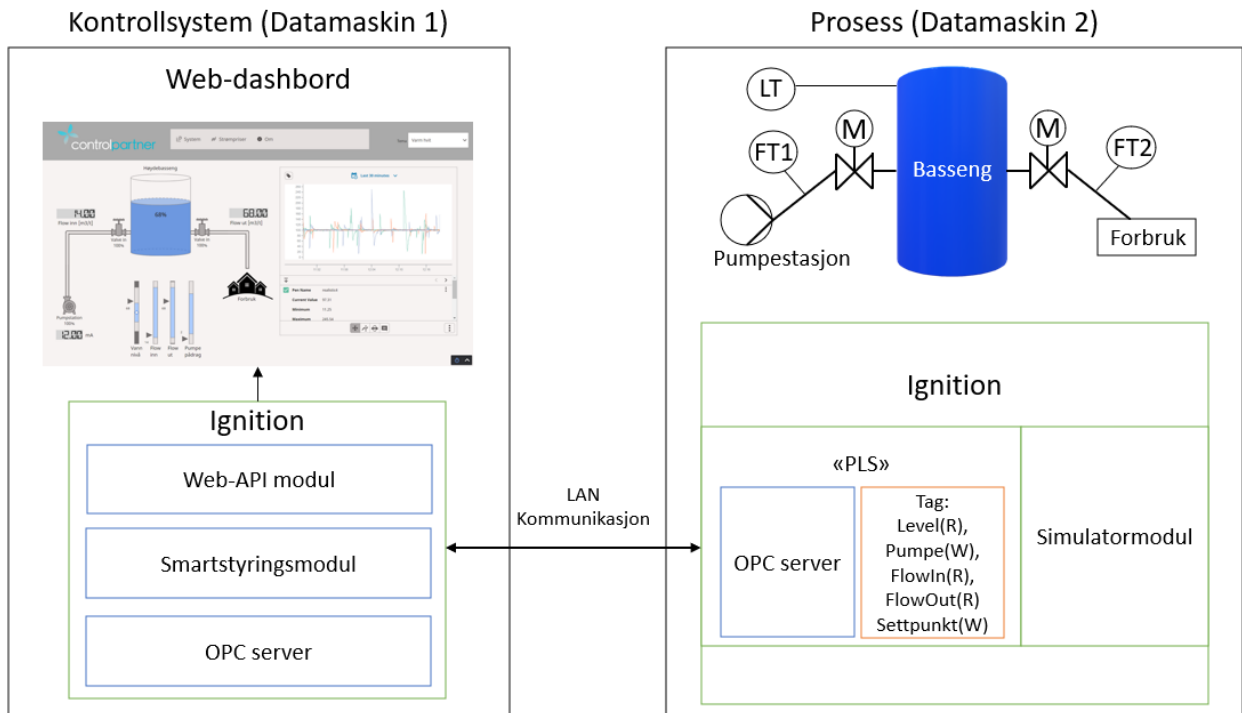
Ignition Gateway er hovedprogramvaretjenesten som kjører alt i Ignition [46]. Den kobler sammen og koordinerer systemer og enheter [46]. Eksempelvis kan dette være PLS-er og databaser. En av de viktigste modulene som benyttes i prosjektet fra Ignition Gateway, er den innebygde OPC UA-modulen. Ved bruk av denne modulen kan forskjellige enheter og systemer kommunisere med hverandre.

### 8.2 Kommunikasjon

I prosjektet er det benyttet OPC UA for å kommunisere mellom de ulike modulene. For å få dette til, er det satt opp to maskiner med to OPC UA-servere ved bruk av LAN-tilkobling. Det er viktig å nevne at det må aktiveres fil- og skriverdeling under innkommende regler på brannmur på begge maskinene for at datamaskinene skal kunne kommunisere med hverandre [47]. Det er verdt å merke seg at det kan være nødvendig å starte Gateway på nytt for å få serverne til å være tilkoblet hverandre. En tilkobling mellom de to maskinen vises i Figur 8-1. Hver maskin har sin egen server, en som heter «Kontrollsystem OPC UA Server» og en som heter «Prosess OPC UA Server». Førstnevnte tar hånd om smartstyringsmodulen og Web-API-modulen. «Prosess OPC UA server» tar hånd om simulatormodulen. Figur 8-2 viser en illustrasjon av kommunikasjonen mellom de to serverne.

Name	Type	Description	Read Only	Status	
Kontrollsystem OPC UA Server	OPC UA	A "loopback" connection to the Ignition OPC UA server running on this gateway.	false	Connected	More ▾ edit
Prosess OPC UA Server	OPC UA		false	Connected	More ▾ edit

Figur 8-1 Kommunikasjon mellom serverne



Figur 8-2 Illustrasjonsbilde av kommunikasjonen mellom de to systemene

Det ble valgt å bruke LAN-tilkobling fremfor WIFI-tilkobling fordi nettverket til USN har strenge restriksjoner, og det må åpnes porter for IP-adressene som brukes i OPC UA kommunikasjonen. Selv om WIFI-tilkobling hadde vært mer praktisk, er LAN-tilkobling mer stabilt og pålitelig og mindre tidkrevende for prosjektet.

For å få til kommunikasjon mellom datamaskinene, blir en tag satt til «OPC»-tag for kontrollsystem og «Memory»-tag for Prosess, men med forbehold om at begge taggene har samme datatype, eksempelvis «float». Maskinene kan skrive og lese til hverandre når «OPC»-taggen linkes opp mot «Memory»-taggen. Tagger som ønskes å trendes, hukes av som «historisk» for å trende data i grafer. Det kan velges å bruke andre metoder for lagring av historiske data, eksempelvis databaselagring. Det er valgt å bruke «Historian»-tags fordi Ignition har en intern lagring av historiske data kalt «Internal Historian» [34].

Det er viktig å påpeke at en prøvelisens av Ignition brukes, og må fornyes hver andre time for å opprettholde kommunikasjon mellom de to datamaskinene [48]. En betalt lisens av Ignition kan kjøpes for å unngå behovet for å reaktivere prøveperioden for Ignition [48].

Ignition benytter seg i utgangspunktet av tags knyttet til en PLS og dens fysiske innganger og utganger. Men i dette prosjektet vil fysiske I/O erstattes med prosesssimulatoren. Eksempelvis ville nivåmåleren i anlegget vært koblet til en PLS, og PLS-en koblet til Ignition via OPC UA. Ettersom en simulator har erstattet anlegget, vil det heller bli skrevet direkte til en tag med «writeBlocking»-syntaksen.

## 8.3 Tilgjengelig OPC-tagger

Dette underkapitlet gir en oversikt over de tilgjengelige OPC-taggene. Videre er det beskrevet formålet og funksjonene til taggene. Taggene brukes til å overføre data mellom de forskjellige modulene.

### 8.3.1 Web-API modul

I web-API modulen innhentes dagens og morgendagens strømpriser. Disse er lagret som float array tagger med navn «DagensPriser» og «MorgendagensPriser». Smartstyringsmodulen tar kun i bruk dagens strømpriser, ettersom det ikke er laget en funksjonalitet for å anvende morgendagens strømpriser. Smartstyringen vil med morgendagens strømpriser kunne planlegge en billigere styringsprofil som kan gi en mer effektiv strømsparing. Dette er tidligere beskrevet i underkapittel 4.2 og senere diskutert i underkapittel 9.5.3.

Taggen «DagensPriser» blir publisert slik at simulatormodulen har tilgang til prisene og kan implementere dem i simulator-skriptet. Taggen publiseres som en OPC-tag, dette gjør den tilgjengelig for de andre modulene.

### 8.3.2 Smartstyringsmodul

Smartstyringsmodulen tar i bruk dagens strømpriser som skrives til tag via Web-API modulen. Den tar i bruk prisene og skriver en styringsprofil til tag «Pumpestyringsprofil» som simulatoren benytter for å kjøre pumpen. Dette skjer i kontrollsystemet som har ansvar for å hente inn priser og lage styring for prosessen.

### 8.3.3 Simulatormodul

Simulatormodulen henter reguleringsparametere fra parametervinduet, som tidligere vist i Figur 7-8. Disse parameterne brukes for å definere og sette ønskede regulatorverdier til primær- og sekundærregulatoren, finne settpunkt og sjekke om regulatorene er satt i auto eller manuell. Det er i tillegg mulighet for å endre bassengets høyde- og radiusparameter til å passe anlegget. Tabell 8-3 viser tagger skrevet fra regulatorvinduet i web-dashbordet. Simulatorene bruker disse verdiene for å kunne simulere systemet som er valgt, og skriver ulike verdier som strømning inn og ut av tanken, nivået som endrer seg og hva slags pådrag pumpen har. Disse verdiene blir skrevet til taggene i Tabell 8-1

### 8.3.4 Web-dashbord modul

Web-dashbord modulen benytter seg av ulike tagger hvor den leser av verdiene som nevnt i Tabell 8-1, slik at den kan vise hvordan prosessen oppfører seg. For å kunne styre simulatorene, må verdiene fra web-dashbord skrives til den tilhørende taggen som er oppført i Tabell 8-3.

Tabell 8-1 Tagger for visning i web-dashbord for «System»-vindu

Tag navn	Eier	Data type	Beskrivelse
FlowInn	Prosess	Float	Viser simulert strømning på innløp til høydebasseng
FlowUt	Prosess	Float	Viser simulert strømning på utløp til forbruk
Nivaats	Prosess	Float	Viser simulert nivå i tanken per tidsskritt i prosent
PumpePaadrag	Prosess	Float	Viser simulert pådragssignalet til pumpen i prosent

Utrekning av pris for reguleringen ville normalt skje i kontrollsystemet, men er blitt gjort i simulatoren fra prosess. Det blir derfor feil å si at kontrollsystemet eier taggene når det er simulatoren som gjør utregningene og skriver til tag. Prosess er sett på som eier av taggene i Tabell 8-2.

Tabell 8-2 Tagger for visning i web-dashbord for «Strømpriser»-vindu

Tag navn	Eier	Data type	Beskrivelse
DagensPriser	Kontrollsystem	Float array	Dagens strømpriser time for time
Besparelse	Prosess	Double	Definerer differansen mellom KaskadePris og SmartPris
SmartPris	Prosess	Float	Total dagskostnad ved smartstyring.
BesparelseProsent	Prosess	Float	Definerer differansen mellom KaskadePris og SmartPris i prosent
KaskadePris	Prosess	Float	Viser den totale dagskostnaden ved kaskaderegulering
Nivaa	Prosess	Float array	Viser simulert nivå i tanken for en hel dag med et bestemt settpunkt

Tabell 8-3 Tagger skrevet fra web-dashbord

Tag navn	Eier	Data type	Beskrivelse
ProporsjonalVerdiPrimaer	Kontrollsystem	Float	Definerer primær regulatorens forsterkning
ProporsjonalVerdiSekundaer	Kontrollsystem	Float	Definerer sekundær regulatorens forsterkning
IntegralVerdiPrimaer	Kontrollsystem	Float	Definerer primær regulatorens integraltid
IntegralVerdiSekundaer	Kontrollsystem	Float	Definerer sekundær regulatorens integraltid
DerivatVerdiPrimaer	Kontrollsystem	Float	Definerer primær regulatorens derivattid
DerivatVerdiSekundaer	Kontrollsystem	Float	Definerer sekundær regulatorens derivattid
ManueltPaadragPrimaer	Kontrollsystem	Float	Definerer primær regulatorens manuelle pådrag
ManueltPaadragSekundaer	Kontrollsystem	Float	Definerer sekundær regulatorens manuelle pådrag
BassengHoyde	Kontrollsystem	Float	Definerer bassengets høyde
BassengRadius	Kontrollsystem	Float	Definerer bassengets radius
PumpeEffekt	Kontrollsystem	Float	Definerer pumpen sin maks effekt.
PumpeMaksStrømning	Kontrollsystem	Float	Brukes til å definere hva pumpestasjonen leverer i m <sup>3</sup> /t ved maks effekt
Settpunkt	Kontrollsystem	Float	Definer settpunkt for simulatoren



SimSpeed	Kontrollsystem	Integer	Definer simulatorhastigheten
StartSim	Kontrollsystem	Boolean	Brukes til å starte simulatoren
PrisOgProfil	Kontrollsystem	Boolean	Brukes til å innhente dagens strømpriser og beregne pumpeprofil

## 9 Diskusjon

Dette kapitlet tar opp hva som er blitt gjort, hva som fungerte, hva som kunne blitt gjort annerledes og hva som kan implementeres i en senere utgave.

### 9.1 Målet med prosjektet

Prosjektet har som oppgave å optimalisere fylling av et høydebasseng basert på strømpriser og informasjon om prosessen. Med dette beregnes den mest kostnadseffektive styringsprofilen for pumpestasjonen tilknyttet høydebassenget.

### 9.2 Krav til gruppen

Gruppen har samarbeidet for å løse den aktuelle problemstillingen. Dette involverte å ta i bruk kunnskaper og erfaringer som gruppen har tilegnet seg gjennom studieløpet og fra tidligere arbeidserfaringer. Resultatet av arbeidet er en grundig og strukturert prosjektering som er faglig innholdsrikt og relevant for en strømeffektiv styring av høydebasseng.

### 9.3 Vurdering av strømpriskilde

I prosjektet er det benyttet strømpriskilde fra hvakosterstrømmen.no [11]. Som tidligere nevnt i underkapittel 4.1, er det verdt å nevne at hvakosterstrømmen.no henter strømpriser i euro direkte fra entsoe.eu [10] [11]. Disse strømprisene blir deretter konvertert til norske kroner med kurspriser fra Norges Bank [11]. Strømprisene kan derfor avvike noe fra Nord Pool [11].

Det er verdt å merke seg at det kun er brukt én strømpriskilde i prosjektet. Det kan være risikabelt å være avhengig av at én kilde alltid er pålitelig og tilgjengelig. Strømpriskilden kan bli utsatt for dataangrep og manipulasjon. Hvis en uautorisert person eller organisasjon får tilgang til systemene eller dataene til kilden og deretter endrer til feilaktig informasjon, vil dette påvirke påliteligheten til kilden. Det bør derfor vurderes å bruke flere kilder for å få en mer helhetlig og pålitelig oversikt over strømprisene i markedet. Det er tidligere nevnt i underkapittel 4.1 at det bør vurderes å bruke Nord Pool som strømpriskilde, siden flere strømkjøper i Norge benytter seg av Nord Pool [12].

Som tidligere nevnt i underkapittel 4.2 kjøres skriptet en gang i timen. Dette kunne vært brukt til å diagnostisere web-APIet. En enkel tilbakemelding til operatøren, hvor web-API-status oppdateres hver time, ville vært tilstrekkelig. I en senere utgave av programmet er dette ønskelig.

### 9.4 Videreutvikling av sanntidssimulering

Tidlig i prosjektet valgte gruppen å bruke signal 0-16mA, med mål om å endre det til 4-20mA etter hvert som større utfordringer ble taklet. Dette ble nedprioritert, men kan gjøres slik [49], se formel 9.1.

$$\text{Signalverdi} = \frac{\text{Signalomfang}}{\text{Måleomfang}} (\text{Måleverdi} - \text{Nedre målegrense}) + \text{Nedre signalverdi}$$

Det er videre diskutert at dersom programvaren blir implementert i et ekte system, ville sanntidssimulering og «System»-vinduet i web-dashbordet bli erstattet med ekte verdier fra en PLS via OPC UA kommunikasjon. Sanntidssimulatoren kunne da blitt omprogrammert til en operatørtreningsmodul [50]. Operatørtrening kunne eksempelvis inneholdt forhåndsprogrammerte relevante scenarioer som ellers ikke lar seg testes på et ekte system.

Det ble også diskutert noen idéer rundt en prosessimulator i form av en digital tvilling [51]. Hvor den digitale tvillingen utfører ulike feilsøkingsskript. Et eksempel på en feil den digitale tvillingen kunne luket frem, er lekkasje på innløp. Ved å sammenlikne forventet volumendring i den ekte prosessen mot den matematiske volumendring i den modellerte prosessen, kan en mulig konkludere med at det er en lekkasje på innløpet til reservoaret. Ved en god nok modell av prosessen finnes det også muligheter for å midlertidig erstatte prosessverdier målt av instrumenter, med modellerte prosessverdier.

## 9.5 Revisjoner av smartstyring

Dette underkapitlet vil se nærmere på de ulike revisjonene av smartstyringen. Målet med revisjonene har vært å optimalisere energibruk til systemet. Det vil diskuteres styrker og svakheter ved hver revisjon, og hvordan de kan forbedres, med noen idéer om hvordan de videre kan utvikles.

### 9.5.1 Revisjon 1

Revisjon 1 av smartstyringen har en nokså enkel oppbygning, men fungerer etter formålet den ble utviklet for. Formålet med revisjon 1 var å undersøke hvordan strømprisene påvirket reguleringsprosessen uten å nødvendigvis gi optimalisering noe oppmerksomhet. Det er en gjenganger at denne revisjonen har flere timer den tillater å kjøre pumpen, så den vil være sterkere til å holde settpunkt enn revisjon 2.

Denne revisjonen krever færre parametere enn revisjon 2, og vil da være noe enklere å implementere, men med stort rom for forbedring.

### 9.5.2 Revisjon 2

Revisjon 2 bygger på samme idé som revisjon 1, men med noen tydelige forskjeller spesielt med tanke på optimalisering av energibruk. Revisjon 2 forsøker å fylle nøyaktig samme volum som forventet vannforbruk i de billigste tidsintervallene på døgnet. Forsøk viser at evnen til å holde settpunkt mot evnen til å spare penger er omvendt-proporsjonale faktorer. Dermed vil væskesøylen i tanken ved bruk av denne revisjonen variere mer, noe som muligens kan merkes på trykket hos forbruker.

Dersom revisjon 2 har tilgang til predikert vannforbruk som stemmer overens med faktisk vannforbruk, vil denne fremgangsmåten være best mulig økonomisk sett. Vannforbruket for et døgn er foreløpig satt til  $608.2 \text{ m}^3$ . Volumet ble funnet ved å tvinge pådragssignalet til pumpen

til null og simulere ett døgn forløp. Som nevnt i kapittel 6.1.2 vil dette neppe fungere som ekte vannforbruk. Gruppen forsøkte å finne relevant historikk på vannforbruk, men det viste seg å være utfordrende å oppnå.

### 9.5.3 Revisjon 3

Revisjon 3 bygger på samme idé som tidligere revisjoner, men kunne tatt i bruk morgendagens strømpriser og MPC for optimalisering av styringsprofil. Et eksempel på bruk av morgendagens strømpriser kunne tenkelig forekommet dersom dagens strømpriser var dyrere enn morgendagens, samtidig som nivået i høydebassenget ikke går under kritisk lav verdi.

Det er ønskelig at revisjon 3 tar høyde for endring i gjennomsnittlig vannforbruk på høytider uten fast dato. Det forventes at det brukes mer vann på høytider enn på en vanlig hverdag. Dette ser gruppen på som overkommelig dersom historikk av vannforbruk i lang tid tilbake er tilgjengelig. Dette vil gjøre at systemet ikke lener seg på lavkutt-logikken. Som beskrevet i kapittel 6.1, vil det fylles uavhengig av pris dersom lavkutt-terskelen brytes.

Tidlig i prosjektet ble det diskutert ulike styringsprofiler for et døgn. Det ble nevnt at modellbasert prediktiv regulering ville finne best mulig styringsprofil, basert på hvordan ønskelige egenskaper blir vektet. Gruppen hadde ambisjoner om å ta i bruk MPC, men ble fort frarådet dette før et forsøk på å lage en smartstyring med egendefinerte «smarte» løsninger var til stede. Dette sammen med andre målsetninger gruppen hadde, viste seg å være tidskrevende. Det skal også nevnes at MPC er masterstudie-pensum, og derfor er utviklingen av denne revisjonen nedprioritert.

## 9.6 Web-dashbord

Dette underkapitlet gir en oversikt over noen funksjonaliteter og løsninger som kunne vært implementert på en annen måte eller utviklet videre i prosjektet. Det diskuteres videre ideer til fremtidig arbeid som kan være gunstig for systemet. For videre utvikling av systemet er det ønskelig å importere et bibliotek med relevante symboler i henhold til NORVAR standarden [37].

### 9.6.1 Startknapp for simulatoren

I brukergrensesnittet til visningsområde «System» er det utviklet en startknapp. Denne knappen starter simulatoren når man endrer dens boolske verdi fra «false» til «true». Å starte simulatoren på denne måten er ikke feil, men det er ikke en optimal løsning. I utgangspunktet var det ønskelig å benytte et «gateway event» for å starte simulatoren ved endring av verdi i taggen. Dette «eventet» bruker «legacy» Python «scope» hvor man kun har lokal eller global «scope». Det oppstår problemer når man prøver å modifisere en global variabel i en metode eller klasse med denne typen «scope» [52] [53]. Det oppsto også en utfordring med importering av biblioteker, hvor funksjoner eller klasser ikke har tilgang til bibliotekene. Løsningen er å importere gjeldende bibliotek direkte inn i metoder og klasser som gjør det ganske uryddig.

Det er derfor valgt å sette skriptet for simulatoren inn i et «Tag changed event» til taggen «StartSim». Skriptet vil kun kjøre når den boolske verdien på taggen blir «True». Denne taggen er knyttet opp mot «System Start» som kan finnes i web-dashbordet. Det var planlagt å sette start av sanntidssimulering og hurtigsimulatoren sammen ved å etablere forbindelse mellom «StartStrom»-taggen og «StartSim»-taggen slik at begge starter samtidig [54]. «StartStrom»-knappen har hurtigsimulatoren og «StartSim» har sanntidssimulering. Etter et par simuleringer oppsto det utfordringer med at sanntidssimuleringen stoppet å kjøre. Det er derfor brukt en

egen startknapp for hurtigsimulatoren i vinduet «strømpriser» for å kunne kjøre begge simulatorene.

### 9.6.2 Knapp for innhenting av strømpriser og oppretting av styringsprofil

I prosjektet var det først planlagt å holde modulene adskilt ved å legge de i hvert sitt «gateway event», men støttet på samme utfordring som beskrevet i underkapittel 9.6.1. For å fortsatt holde modulene adskilt, men samtidig unngå to separate knapper, én for innhenting av strømpriser, og én for styringsprofil, ble det valgt å kombinere begge skript. Innhenting av strømpriser kjøres først og deretter blir smartstyringen kjørt og skriver styringsprofilen til tag som blir benyttet av simulatorene. Ved å trykke på knappen én gang i løpet av hvert døgn, oppdateres dagens strømpriser og styringsprofil. I en fremtidig versjon av Ignition, vil det ikke lenger være en utfordring [52]. Det vil da bli mulig å legge skriptene i «gateway events».

### 9.6.3 Ventilstyring

I en videre utvikling av systemet er det ønskelig å implementere funksjonalitet for ventilstyring. Dette er for å lettere ha kontroll på tilstanden til ventilene i form av tilbakemelding. Det å kunne fjernstyre ventilene er en funksjonalitet som er ønskelig, eksempelvis i forbindelse med sikkerhet eller vedlikehold.

### 9.6.4 Alarmsystem

For å forbedre brukergrensesnittet bør det legges til et alarmsystem som brukes som tilbakemelding om farlige tilstander eller tilløp til farlige tilstander [55]. I Ignition finnes det en modul for alarmer som kan brukes til implementasjonen av et alarmsystem [56]. For å opprette alarmer i modulen kan man gå inn i de ulike taggene og sette alarmer i forhold til de øvre og nedre grensene som er bestemt. Det er også mulig å legge til prioritering som «Low», «Medium», «High» eller «Critical» for å indikere alvorlighetsgraden til alarmene for brukeren [56].

Det er mulig å undertrykke alarmer. Det betyr at alarmsystemet vil ignorere de alarmene som undertrykkes slik at de ikke vises i alarmtabellen. Dette kan være nyttig når man trenger å skifte eller utføre vedlikehold på transmittere i prosessen. Da unngås det at alarmer for den spesifikke transmitteren utløses hele tiden. Det kan være tilfeller der funksjonsfeil på utstyr skaper forstyrrende alarmer som går av kontinuerlig. Alarmundertrykkelse har vanligvis en tidsperiode før de undertrykte alarmene reaktiveres automatisk, eller brukeren kan manuelt reaktivere alarmene.

For å vise alarmer kan man bruke Perspective alarmstatus-tabellen, som viser aktive, ikke-merkerte eller merkerte alarmer [56]. I tabellen kan man se datoen og klokkeslettet for når alarmen ble aktivert, visningsveien for å se hvor alarmen kommer fra, hvilke prioritering alarmen har og hva statusen på alarmen er. Dette gir brukeren en tydelig oversikt over hvilke alarmer som er aktive og når de ble utløst [56].

### 9.6.5 Responsivt brukergrensesnitt

Perspective gir utvikleren muligheten til å lage et responsivt brukergrensesnitt som fungerer på ulike enheter [57]. For å oppnå dette må brukergrensesnittet utvikles med ulike "containere" som passer til forskjellige skjermoppløsninger [57]. Når man bruker en mobil enhet med en liten skjerm, vil Perspective-modulen automatisk velge det brukergrensesnittet som er utviklet

spesifikt for den skjermopløsningen [57]. Ved å ha et responsivt design vil brukergrensesnittet være fleksibelt og tilpasse seg enhver enhetstype [57]. Dette er en betydelig forbedring da brukergrensesnittet blir mer allsidig.

Det er verdt å merke seg at denne funksjonaliteten ikke er implementert i systemet, da det betraktes som en «ekstra» funksjon som ikke påvirker systemets grunnleggende funksjonalitet. Systemet er ment å være et kontrollrom med en fastmontert skjerm for styring og overvåking av systemet.

### 9.6.6 Fryste verdier

Fryste verdier er en funksjonalitet som ikke er implementert i systemet. Formålet med å fryse verdier er å kunne utføre vedlikehold eller bytte ut defekt utstyr. På denne måten kunne man enklere gjenopprette systemet til sin opprinnelige tilstand. Selv om fryste verdier ikke ble implementert, kan det være en gunstig funksjonalitet å ta med i en fremtidig versjon av systemet.

## 9.7 OPC UA kommunikasjon

Dette underkapitlet omhandler kommunikasjon og datalagring i forbindelse med OPC UA i prosjektet. Det er videre beskrevet hvorfor LAN benyttes i stedet for Wi-Fi, hvordan data som sendes mellom serverne blir lagret og hvordan dette kan forbedres.

### 9.7.1 LAN benyttes som erstatning for Wi-Fi

Det benyttes LAN kommunikasjon mellom to OPC UA serverne hvor den ene tilsvarende kontrollrom og den andre prosessen. LAN er valgt i stedet for Wi-Fi fordi nettet som benyttes på USN er restriktivt for kommunikasjon mellom serverne. Dette er løst ved å bruke LAN og Ethernet-kabel mellom to datamaskiner. En forbedring av dette er å bruke Wi-Fi slik at rekkevidden ikke begrenses av kabellengden.

### 9.7.2 Datalagring

Lagring av data via OPC UA foregår ved at noen få utvalgte verdier fra ulike tagger blir lagret i «Internal Historian» [34]. Et forbedringspotensial hadde vært å bruke database for lagring av historiske verdier og lagring av nødvendige tagger som strømpriser, styringsprofil og kostnader. Disse verdiene kan over tid gi innsikt i hvor mye som spares og hvordan systemet ble styrt ut fra styringsprofilen. I Ignition er det en modul for å koble en SQL-database til OPC UA-serveren, noe som gir muligheter for å utnytte lagring av data til database [58].

## 9.8 utfordringer

Dette kapitlet tar for seg utfordringer gruppen støtte på ved utviklingen av systemet og ved benyttelse av programvaren Ignition 8.1.

### 9.8.1 Windup

Ved utvikling av prosesssimulator og tilhørende regulatorer oppsto det en utfordring med ukjent opphav, det viste seg å skyldes «windup» i integrator-leddet hos begge regulatorene [18].

Løsningen til dette er å implementere integrator-begrensning. Dette er tidligere beskrevet i underkapittel 5.3.2.

### 9.8.2 Håndtering av Ignition-krasj

Under prosjektet opplevde gruppen utfordringer med gjentatte krasjer i Ignition. Dette oppsto under testing av de ulike skriptene. Det ble raskt oppdaget at krasjene oppsto når skriptkonsollvinduet i Ignition ble åpnet, ettersom alle skriptene i prosjektet ble kjørt samtidig. Dette kunne føre til en evig løkke som aldri stoppet. En løsning på denne utfordringen var å merke av for «read only» i Ignition.

Ignition krasjet også ved overføring av store datamengder, spesielt når man forsøkte å redigere grafen for ferdig simulert smartstyring etter at den var fylt. En midlertidig løsning var å slette verdiene i arrayet før man redigerte grafen.

### 9.8.3 For-løkker

Ved oversetting av Python til Jython og overføring fra Spyder til Ignition støtte gruppen på utfordringer rundt terminering av for-løkker. Dersom ett «event» kaller opp et skript, vil løkken iterere til den er ferdig uavhengig av om «eventet» som kalte det opp i første omgang, er aktivt lenger eller ei. Løsningen for dette ble å erstatte gjeldende for-løkker med while-løkker kombinert med et boolsk flagg. Dette er tidligere beskrevet i underkapittel 5.3.5.

### 9.8.4 Power chart

Det viste seg utfordrende å visualisere prosessen i et «power chart» når simulatoren arbeidet i økt hastighet. Prosessverdiene ble skrevet til tagger i en økt hastighet, samtidig som x-aksen til grafen fortsatte å visualisere verdiene i sanntid. Dette medførte at grafen sin x-akse ikke stemte overens med simulatorens x-akse. Prosessverdiene plottes, men innenfor feil x-akse verdier. Denne løsningen vurderes som egnet, da det i fremtidig implementasjon av systemet ikke vil være aktuelt å kjøre den fysiske prosessen med økt hastighet. Dette er tidligere beskrevet i underkapittel 7.3.2.

### 9.8.5 Etablere forbindelse mellom «StartStrom» og «StartSim»

I forsøket på å ha en startknapp for begge simulatorene ble det forsøkt å etablere en forbindelse mellom taggen «StartStrom» og «StartSim». Til å begynne med fungerte dette som forventet. Men etter hvert oppstod det utfordringer med å kjøre begge skriptene samtidig. Sanntidssimuleringen stoppet uventet, noe som førte til at simuleringen i «System»-vinduet ikke fungerte som forventet. For å unngå denne situasjonen ble det benyttet en startknapp for sanntidssimuleringen i «System»-vinduet og en for hurtigsimulatorene i «Strømpris»-vinduet.

## 10 Konklusjon

Prosjektet går ut på å optimalisere reguleringen av et høydebasseng i det kommunale vannettet ut fra strømpriser i markedet og styre fyllingsprosessen etter når strømmen er billig. Ved å implementere en smartstyring, simulator og bruke informasjonen om strømpriser fra et web-API, har gruppen teoretisk klart å oppnå et resultat som vil gi besparelser på strømutfgifter og redusert belastningen på strømmettet.

Det er utviklet en web-API modul for innhenting av strømpriser som fungerer etter formålet. Imidlertid bør det vurderes å videreutvikle løsningen ved å inkludere flere strømpriskilder. Dette vil bidra til pålitelighet og stabilitet i tilfelle en av strømpriskildene blir utilgjengelig.

Gjennom utviklingen av to revisjoner av smartstyringen og evalueringen av resultatene, har det blitt konkludert med at revisjon 2 er den mest egnete løsningen inntil revisjon 3 blir utviklet. I løpet av forsøk utført i uke 16, 2023, oppnådde gruppen ved simulering en gjennomsnittlig besparelse på omtrent 3000 NOK. Dette beløpet tilsvarer 156 000 NOK per år for det aktuelle høydebassenget. Dersom vannettet tilhørende høydebassenget tåler variasjoner i væskesøylen, vil dette være svært lønnsomt. Videre forskning og utvikling innen revisjon 3 av smartstyringen kan ytterligere forbedre besparelsene og optimaliseringen av reguleringen av høydebassenget.

Simulatoren kan tilpasses ulike høydebasseng ved å anvende parameterverdiene til det aktuelle bassenget som skal testes. Slik vil det være mulig å validere hvor mye som kan spares ved hjelp av testing. Dette kan gi kommunene et innblikk i hvor lønnsom investeringen kan være. Spesielt med tanke på at det kun er en programvareoppgradering uten behov for innkjøp av maskinvare. Det tas forbehold om at det er et allerede eksisterende anlegg.

Videre har prosjektet utviklet et web-dashbord ved hjelp av Perspective-modulen til Ignition 8.1. Dette dashbordet gir operatøren en visuell kontroll over systemet slik at prosessdata og styringsprofil kan overvåkes etter behov. Dette bidrar til å sikre at reguleringen av høydebassenget opprettholder ønsket kvalitet og effektivitet. I en senere utgave vil det være mulig å integrere eksisterende alarmsystem til web-dashbordet.

Gjennom prosjektet er OPC UA brukt som kommunikasjon mellom modulene for å gjøre prosjektet mest mulig realistisk. Det benyttes to datamaskiner som servere, en som heter «Kontrollsystem OPC UA server», og en som heter «Prosess OPC UA Server». Dette fungerer etter formålet gruppen hadde. Ved implementering av programvaren i et ekte system, er det mulig å erstatte «Prosess OPC UA Server» med en PLS.

Strømprisene har de siste årene steget kraftig og er stadig varierende fra time til time. Dette gjør det aktuelt å utnytte variasjonene til strømsparing når strømprisen er lav. Prosjektet tilbyr en slik løsning. Dersom prisen på energi forblir høy og strømprisene fortsetter å variere, er det absolutt verdt å investere i denne løsningen.



# Referanser

- [1] Statnett, «Om strømpriser,» Statnett, [Internett]. Available: <https://www.statnett.no/om-statnett/bli-bedre-kjent-med-statnett/om-strompriser/>. [Funnet 04 mai 2023].
- [2] O.-. o. e. T. Aasland, «Hvorfor er strømmen så dyr?,» Regjeringen, 08 desember 2022. [Internett]. Available: <https://www.regjeringen.no/no/aktuelt/hvorfor-er-strommen-sa-dyr/id2950280/>. [Funnet 04 mai 2023].
- [3] K. Hofstad, «Vindenergi,» Store Norske Leksikon, 19 August 2022. [Internett]. Available: <https://snl.no/vindenergi>. [Funnet 10 Mai 2023].
- [4] K. Hofstad, «Solenergi,» Store Norske Leksikon, 4 januar 2023. [Internett]. Available: <https://snl.no/solenergi>. [Funnet 10 Mai 2023].
- [5] Inductive Automation, «The Platform For Unlimited Digital Transformation,» Inductive Automation, [Internett]. Available: <https://inductiveautomation.com/ignition/>. [Funnet 16 Februar 2023].
- [6] T. S. Flemming Graversen, «Er industrien klar for OPC UA?,» 30 Mars 2022. [Internett]. Available: <https://blogg.triple-s.no/er-industrien-klar-for-opc-ua>. [Funnet 15 Februar 2023].
- [7] A. Gamela, «Jython vs. Python: Main differences and when to use them,» 10 August 2022. [Internett]. Available: <https://www.imaginarycloud.com/blog/jython-vs-python-main-differences-and-when-to-use-them/>. [Funnet 30 Mars 2023].
- [8] «Jython - Overview,» tutorialspoint, [Internett]. Available: [https://www.tutorialspoint.com/jython/jython\\_overview.htm](https://www.tutorialspoint.com/jython/jython_overview.htm). [Funnet 30 Mars 2023].
- [9] C. R. Ben Fry, «Python, Jython and Java,» [Internett]. Available: <https://py.processing.org/tutorials/python-jython-java/>. [Funnet 10 April 2023].
- [10] Entsoe, «Entsoe,» [Internett]. Available: <https://www.entsoe.eu/about/market/#what-we-do>. [Funnet 10 Januar 2023].
- [11] hvakosterstrømmen, «Strømpris API,» hvakosterstrømmen, [Internett]. Available: <https://www.hvakosterstrommen.no/strompris-api>. [Funnet 10 Januar 2023].
- [12] «Hva består strømregningen av?,» norgesenergi, 3 Mars 2022. [Internett]. Available: <https://norgesenergi.no/stromsmart/hva-bestar-stromregningen-av/>. [Funnet 23 Februar 2023].
- [13] «API,» Nord Pool, [Internett]. Available: <https://www.nordpoolgroup.com/en/trading/api/>. [Funnet 23 Februar 2023].
- [14] Nordpool, «Power Data Services Products and Prices,» 2020.

- [15] Ø. Z. Lie, «Rødt kritisk til at Nord Pool legger data bak betalingsmur,» *energiteknikk*, 18 Mars 2022. [Internett]. Available: <https://energiteknikk.net/2022/03/rodt-kritisk-til-at-nord-pool-legger-data-bak-betalingsmur/>. [Funnet 23 Februar 2023].
- [16] F. A. Haugen, «Matematisk prosessmodellering,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget Vigmostad & Bjørke AS, 2014, pp. 205-228.
- [17] G. Liu, «Piecewise linearization of a quadratic function,» ResearchGate, Juli 2016. [Internett]. Available: [https://www.researchgate.net/figure/Piecewise-linearization-of-a-quadratic-function\\_fig1\\_310453509](https://www.researchgate.net/figure/Piecewise-linearization-of-a-quadratic-function_fig1_310453509). [Funnet 10 Mai 2023].
- [18] F. A. Haugen, «PID-regulatoren,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget Vigmostad & Bjørke AS, 2014, pp. 279-316.
- [19] F. A. Haugen, «Kaskaderegulering,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget Vigmostad & Bjørke AS, 2014, pp. 171-178.
- [20] F. A. Haugen, «Regulering med foroverkobling,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget Vigmostad & Bjørke AS, 2014, pp. 229-250.
- [21] F. A. Haugen, «Reguleringsteknikkens historie,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget Vigmostad & Bjørke AS, 2014, pp. 459-462.
- [22] F. Haugen, «Model-based predictivecontrol (MPC),» i *Advanced DYNAMICS and CONTROL*, TechTeach, 2010, pp. 273-284.
- [23] R. Sharma, Lecture notes for the course IIA 4117: Model Predictive Control, Porsgrunn: USN, 2019.
- [24] F. A. Haugen, «Innstilling av PID-regulatorer,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget Vigmostad & Bjørke AS, 2014, pp. 317-388.
- [25] F. A. Haugen, «Middelverdifilter,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget, 2024, pp. 99-101.
- [26] DataScientest, «NumPy : the most used Python library in Data Science,» 15 Mars 2023. [Internett]. Available: <https://datascientest.com/en/numpy-the-python-library-in-data-science>. [Funnet 10 Mai 2023].
- [27] S. Hervik, «Eulers metode,» Store norske leksikon, 31 August 2021. [Internett]. Available: [https://snl.no/Eulers\\_metode](https://snl.no/Eulers_metode). [Funnet 10 Juli 2023].
- [28] G. M. o. M. T. John Sirum, «Veiledning i bygging og drift av drikkevannsbasseng,» Norsk Vann BA, Hamar, 2011.
- [29] Inductive automation, «Perspective Sessions,» Ignition, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+Sessions>. [Funnet 05 mai 2023].
- [30] Ignition, «Views and Containers in Perspective,» Inductive Automation, [Internett]. Available:

- <https://docs.inductiveautomation.com/display/DOC80/Views+and+Containers+in+Perspective>. [Funnet 05 mai 2023].
- [31] I. automation, «A Vision-Oriented Guide to Perspective,» [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC80/A+Vision-Oriented+Guide+to+Perspective>. [Funnet 23 februar 2022].
- [32] Ignition, «Perspective - Coordinate Container,» Inductive automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+-+Coordinate+Container>. [Funnet 02 Mai 2023].
- [33] Ignition, «Perspective - Power Chart,» Inductive automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+-+Power+Chart>. [Funnet 02 mai 2023].
- [34] Inductive Automation, «Tag History Providers,» [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Tag+History+Providers>. [Funnet 10 Mai 2023].
- [35] H. P. Bill Hollifield, «Maximize operator effectiveness, High Performance HMI Principles and Best Practices,» PAS The human Reliability Company, 2015.
- [36] Norsk vann, «Om norsk vann,» Norsk vann, [Internett]. Available: <https://norsk vann.no/om-norsk-vann/>. [Funnet 08 mai 2023].
- [37] NORVAR, «Norm for symboler i driftskontrollsystemer for VA-sektoren,» NORVAR BA, Hamar, 2007.
- [38] Ignition, «Perspective - Pump,» Ignition by inductive automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+-+Pump>. [Funnet 08 mai 2023].
- [39] Ignition, «Perspective - Sensor,» Ignition by Inductive automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+-+Sensor>. [Funnet 08 mai 2023].
- [40] Ignition, «Perspective - Valve,» Ignition by Inductive automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+-+Valve>. [Funnet 08 mai 2023].
- [41] P. Gruhn, «Human Machine Interface (HMI) Design: The Good, The Bad, and The Ugly (and what makes them si),» ICS triplex | Rockwell Automation, Houston, Texas, 2011.
- [42] T. H. Nätt, «Datasikkerhet,» Store norske leksikon, 02 august 2022. [Internett]. Available: <https://snl.no/datasikkerhet>. [Funnet 02 mai 2023].
- [43] Inductive Automation, «Ignition Security Hardening Guide,» [Internett]. Available: <https://www.inductiveautomation.com/resources/article/ignition-security-hardening-guide#step--hardening-the-environment>. [Funnet 08 Mai 2023].

- [44] Inductive Automation, «Ignition Security Hardening Guide,» 25 Mai 2022. [Internett]. Available: <https://www.inductiveautomation.com/resources/article/ignition-security-hardening-guide#step--hardening-the-environment>. [Funnet 08 Mai 2023].
- [45] I. Automation, «Perspective Workstation,» Ignition 8.1 by Inductive Automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+Workstation>. [Funnet 25 April 2023].
- [46] Inductive Automation, «Gateway,» [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC80/Gateway>. [Funnet 24 April 2023].
- [47] R. Allen, Active Directory Pro, 23 Mai 2022. [Internett]. Available: <https://activedirectorypro.com/allow-ping-windows-firewall/>. [Funnet 11 Mai 2023].
- [48] Inductive Automation, «Licensing and Activation,» [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Licensing+and+Activation>. [Funnet 08 Mai 2023].
- [49] M. P. & P. K. Fylkesnes, «Instrumenteringsteknikk - Måleomformere,» [Internett]. Available: <https://usn.instructure.com/courses/26008/pages/maleomformere>. [Funnet 11 Mai 2023].
- [50] F. A. Haugen, «Simulering av reguleringsystemer,» i *Reguleringsteknikk*, Bergen, Fagbokforlaget Vigmostad & Bjørke AS, 2014, pp. 411-419.
- [51] K. Martinsen, «digital tvilling,» Store Norske Leksikon, 12 Desember 2022. [Internett]. Available: [https://snl.no/digital\\_tvilling](https://snl.no/digital_tvilling). [Funnet 11 Mai 2023].
- [52] PGriffith, «Using Sin,Cos in Gateway Event scripts,» Inductive automation, November 2022. [Internett]. Available: <https://forum.inductiveautomation.com/t/using-sin-cos-in-gateway-event-scripts/66258>. [Funnet 1 Mai 2023].
- [53] L. P. Ramos, «Python Scope & the LEGB Rule: Resolving Names in Your Code,» Real Python, [Internett]. Available: <https://realpython.com/python-scope-legb-rule/>. [Funnet 1 05 2023].
- [54] Inductive Automation, «Types of Tags,» [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Types+of+Tags#TypesofTags-ReferenceTags>. [Funnet 09 Mai 2023].
- [55] N.-O. Skeie, «Alarm systems,» i *Industrial control systems*, Porsgrunn, USN, 2022, pp. 66-88.
- [56] I. automation, «Alarming,» Inductive automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Alarming>. [Funnet 1 05 2023].
- [57] i. a. Ignition, «Perspective Design Principles,» Ignition, Inductive Automation, [Internett]. Available:

<https://docs.inductiveautomation.com/display/DOC80/Perspective+Design+Principles>.  
[Funnet 03 mai 2023].

- [58] Inductive Automation, «SQL Bridge Module,» Inductive Automation, [Internett]. Available: <https://inductiveautomation.com/ignition/modules/sql-bridge>. [Funnet 09 05 2023].

# Vedlegg

Vedlegg A Prosjektbeskrivelse

Vedlegg B Gruppeavtale

Vedlegg C Fremdriftsplan

Vedlegg D Medforfattererklæring

Vedlegg E Veiledningsavtale

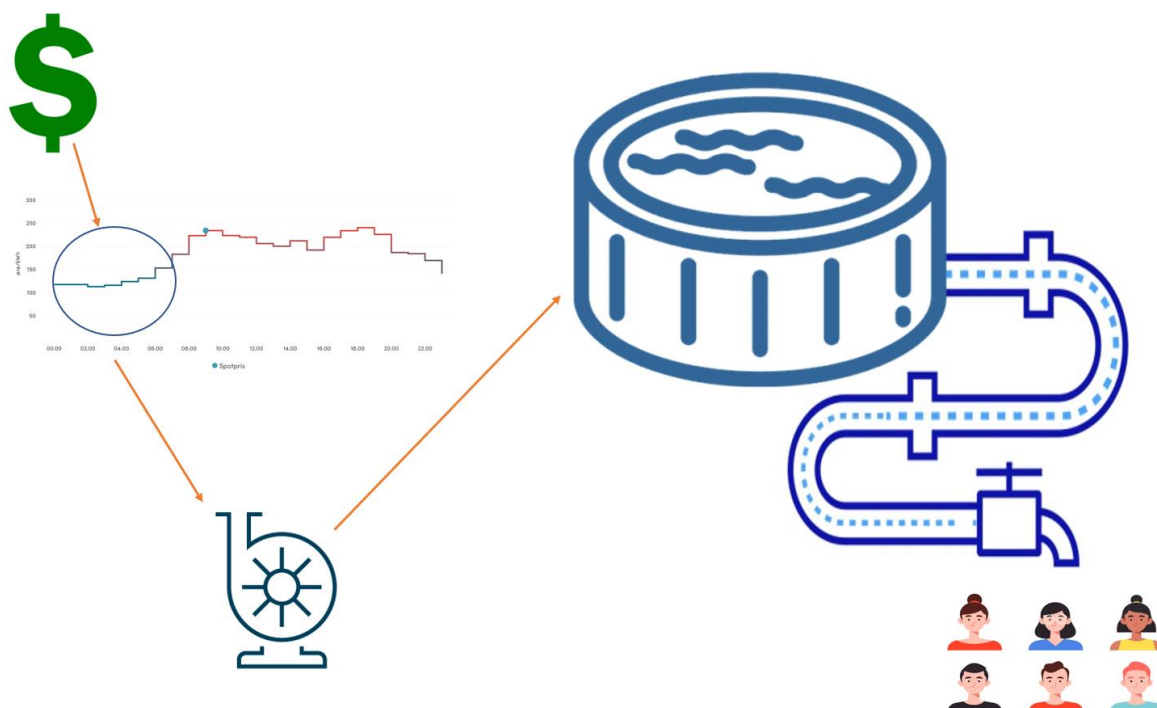
Vedlegg F Kildekode

Vedlegg G Brukerveiledning

TS3000 1 Bachelor

# Vedlegg A

## Prosjektbeskrivelse



IA6-1-23

## **TS3000 1 Bacheloroppgave**

**Studieretning:** Informatikk og Automatisering

**Prosjektgruppe:** IA6-1-23

**Tittel:** Strømeffektiv styring av høydebasseng

**Veileder:** Ole Magnus Brastein

**Samarbeidspartner:** Controlpartner AS

### **Prosjektets bakgrunn:**

Med den nåværende situasjonen i strømmarkedet, har flere kommuner satt strømsparing høyt på prioriteringslisten. Fylling av høydebassenger er en av de mer energikrevende prosessene i det kommunale vann-nettet, og det vil derfor ha stor verdi for kommunene å optimalisere fyllingsprosessen i forhold til gjeldende strømpriser i markedet. Dette kan inkludere å justere fyllingsprosessen til tider med lavere strømpriser eller å investere i mer energieffektive metoder for fylling. På denne måten kan kommunene redusere sine strømutfgifter og bidra til å redusere belastningen på strømnettet.

### **Målbeskrivelse for prosjektet:**

Prosjektet går ut på å optimalisere fylling av et høydebasseng ved bruk av strømpriser, informasjon om prosessen, pumpekurver og historisk forbruk av vann. Dette er for å beregne den mest kostnadseffektive kjøreprofilen for pumpestasjonen tilknyttet høydebassenget. Det skal programmeres en klient i Ignition sitt script system (Python/Jython) som henter strømpriser fra et web-API. Det skal også programmeres en prosesssimulator i Ignition sitt script system (Python/Jython) som skal brukes for å teste/validere beregninger som kommer fra smartstyringen.

For å få til dette må det hentes inn strømpriser fra en egnet kilde (web-API) som sammen med informasjon om prosessen, pumpekurver og historisk forbruk av vann,



**Fakultet for Teknologi, Naturvitenskap og Maritime fag**

brukes i beregninger hvor resultatet blir den mest kostnadseffektive kjøreprofilen for pumpestasjonen tilknyttet høydebassenget Disse resultatene tilgjengeliggjøres for andre moduler via Ignition sin innbygde OPCUA server.

Det skal lages et brukergrensesnitt som bruker informasjonen fra smartstyringen og simulatoren. Dette skal presenteres i et web-dashbord ved hjelp av «Perspective» modulen til Ignition. Man skal kunne betjene systemet fra dashbordet, samt at relevant nøkkelinformasjon vises.

**Oppgaver:**

Oppgavene kan brytes ned i følgende del-oppgaver:

**1. Innhenting av strømpriser via et web-API**

- a. Vurder pålitelige kilder for innhenting av strømpriser.
- b. Designe logikken til funksjonaliteten.
- c. Implementere funksjonaliteten.
- d. Bruke egnet filformat ved lagring av informasjon.
- e. Publisert innhentede strømpriser i OPCUA.

**2. Programmering av en prosess simulator i Ignition sitt script system (Python/Jython)**

- a. Hent inn nødvendig(e) parameter og informasjon tilhørende modellen fra industri partner.
- b. Vurder ulike metoder som skal benyttes til å teste/validere beregningene fra smart styringen.
- c. Programmere en simulator som likner anlegget det er modellert etter, på best mulig måte.
  - i. Optimalisere parametere.
  - ii. Teste nødvendige funksjoner beskrevet i funksjonsbeskrivelse.
- d. Lage en brukerveiledning.

Fakultet for Teknologi, Naturvitenskap og Maritime fag

### 3. Programmering av smart styring

- a. Vurder best reguleringsstrategi.
- b. Design funksjonaliteten for reguleringen.
- c. Vurder hvilken informasjon som skal ligge tilgjengelig i OPCUA.
- d. Publisert relevant informasjon i OPCUA.
- e. Iverksett samarbeid mellom smartstyring og simulator.

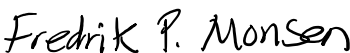
### 4. Presentasjon av relevant informasjon i et web-dashbord


- a. Benytt relevant informasjon tilgjengelig i OPCUA.
- b. Lage et brukergrensesnitt for systemet i «Perspective» modulen i Ignition hvor diverse informasjon fremstilles.
- c. Benytt NORVAR standard for å opprettholde HPHMI (High performance human-machine interface).

#### Eventuelt:

- Ta høyde for endring i gjennomsnittlig vannforbruk på høytider uten fast dato i den matematiske modellen.
- Benytte MPC (Model Predictive Control) som overordnet regulator (supervisory MPC).
- Lage et alarmsystem.
- Opprett restriksjoner på hvilken informasjon som er tilgjengelig for hvem, avhengig av bruker. Dette gjelder OPCUA og brukergrensesnitt.

  
Marius Solli Jensen

  
Fredrik Pedersen Monsen

  
Rino Mikkelsen

  
Odin Melås

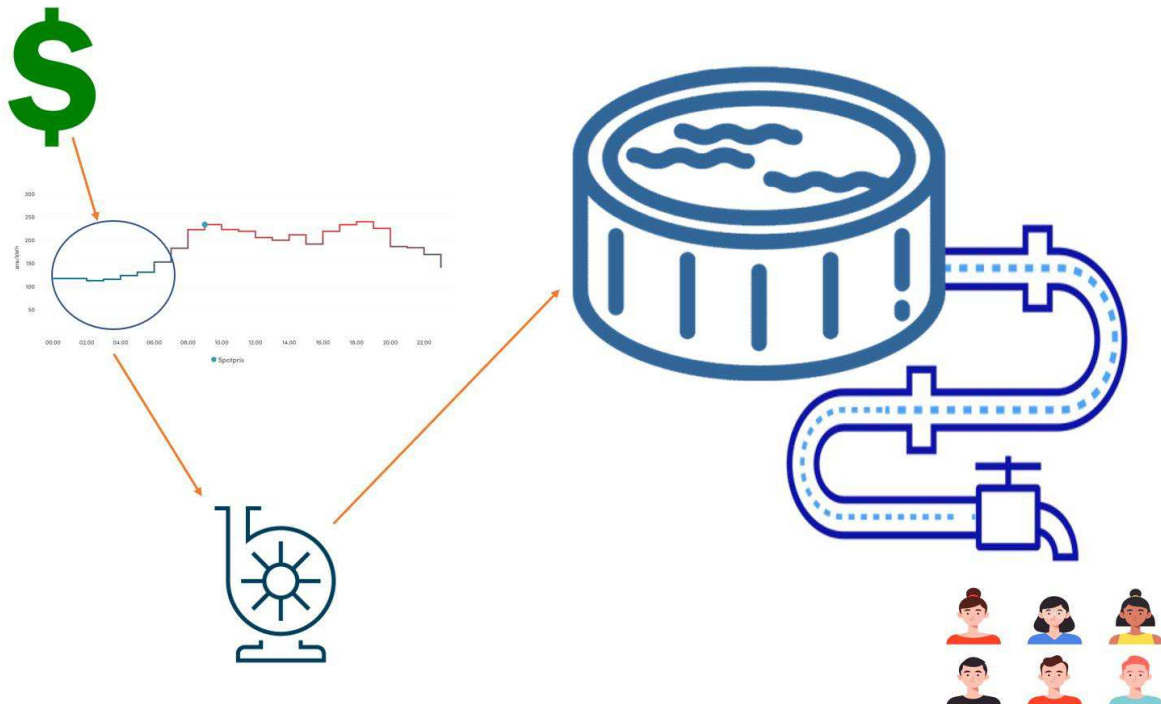
  
Ole Magnus Brastein (Veileder)

Porsgrunn, 03.02.2023

TS3000 1 Bachelor

# Vedlegg B

## Gruppeavtale



IA6-1-23

# Gruppeavtale

Denne gruppeavtalen gjelder for gruppe IA6-1-23

## Prosjektvarighet

Denne avtalen gjelder fra 11.01.2023 til 20.06.2023

## Gruppemedlemmer

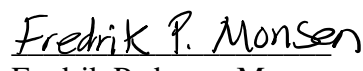
Navn	Telefonnummer	E-post
Marius Solli Jensen	984 26 567	238065@usn.no
Odin Melås	905 45 050	238057@usn.no
Rino Mikkelsen	948 90 612	230008@usn.no
Fredrik Pedersen Monsen	955 27 850	237051@usn.no

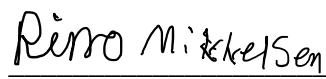
## Krav til prosjektdeltagere

1. Alle møter til avtalt tid og sted
2. Gruppemedlemmene skal levere avtalte arbeidsoppgaver til avtalt tid
3. Alt fravær skal varsles i god tid
4. Problemer med å følge framdriften i prosjektet skal meldes så fort som mulig.
5. Respektere hverandres synspunkter og ideer
6. Mobiltelefon skal ikke brukes i møtesammenheng
7. Agenda for gruppemøte planlegges ved innkalling

Porsgrunn 20.01.2023

  
Marius Solli Jensen

  
Fredrik Pedersen Monsen

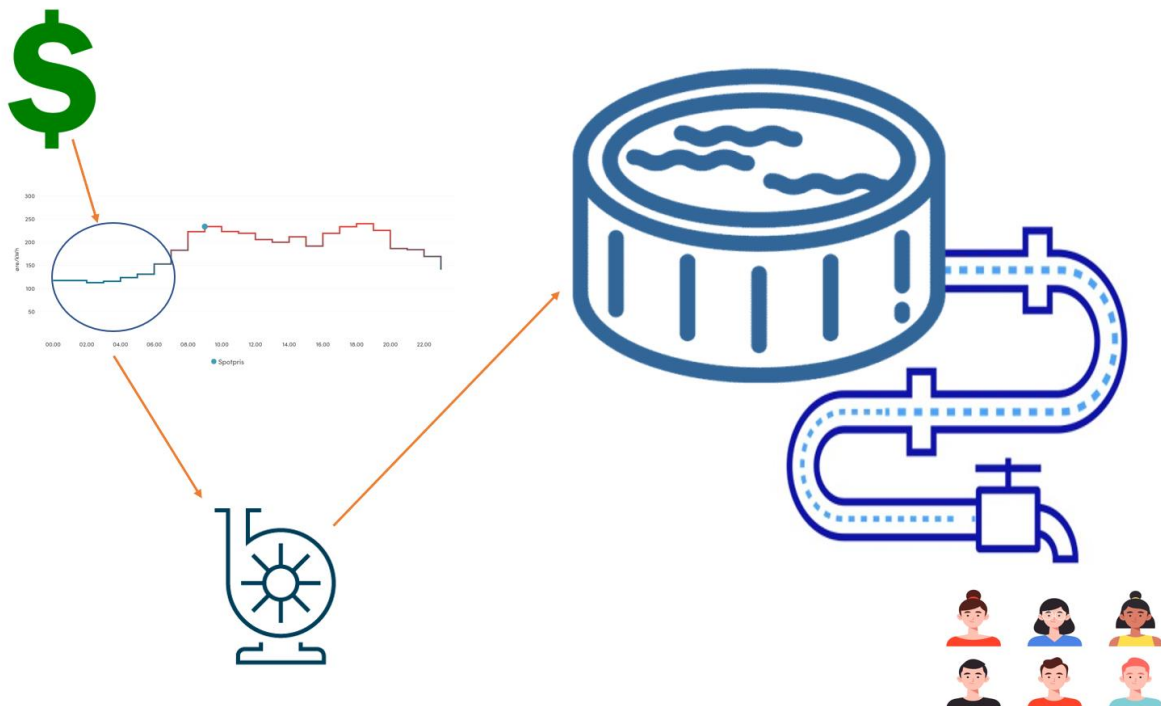
  
Rino Mikkelsen

  
Odin Melås
































TS3000 1 Bachelor

# Vedlegg C




















## Fremdriftsplan






















IA6-1-23




















ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	% Work Complete
1			<b>Startfase</b>	<b>26 days</b>	<b>Wed 11.01.23</b>	<b>Wed 15.02.23</b>			<b>100%</b>
2			Utarbeide et Gantt diagram	26 days	Wed 11.01.23	Wed 15.02.23			100%
3			Gruppeavtale	2 days	Wed 11.01.23	Thu 12.01.23			100%
4			Veilederavtale	1 day	Wed 11.01.23	Wed 11.01.23			100%
5			Utarbeide prosjektbeskrivels	26 days	Wed 11.01.23	Wed 15.02.23			100%
6			Lage mappestruktur	2 days	Wed 11.01.23	Thu 12.01.23			100%
7			Oppstartsmøte med Controlpartner	1 day	Wed 25.01.23	Wed 25.01.23			100%
8			<b>Midtfase</b>	<b>48 days</b>	<b>Wed 15.02.23</b>	<b>Fri 21.04.23</b>			<b>100%</b>
9			<b>Innhenting av strømpriser gjennom et web-API</b>	<b>25 days</b>	<b>Wed 15.02.23</b>	<b>Tue 21.03.23</b>			<b>100%</b>
10			Vurder pålitelige kilder for innhenting av	3 days	Wed 15.02.23	Fri 17.02.23			100%
11			Designe logikken til funksjonaliteten	10 days	Mon 20.02.23	Fri 03.03.23	10		100%
12			Implementere funksjonalit	5 days	Mon 06.03.23	Fri 10.03.23	11		100%
13			Bruke egnet filformat ved lagring av informasjon	1 day	Fri 10.03.23	Fri 10.03.23			100%
14			Publiser innhentende strømpriser i OPCUA	7 days	Mon 13.03.23	Tue 21.03.23	12		100%
15			<b>Programmering av en prosessimulator</b>	<b>25 days</b>	<b>Wed 15.02.23</b>	<b>Tue 21.03.23</b>			<b>100%</b>






















Project: Gantt diagram  
Date: Fri 12.05.23

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

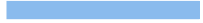


















ID	 Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	% Work Complete
16			Hent inn nødvendig(e) parameter og informasjon tilhørende modellen fra industri	2 days	Wed 15.02.23	Thu 16.02.23		100%
17			Vurder ulike metoder som skal benyttes til å teste/validere beregningene fra smart	25 days	Wed 15.02.23	Tue 21.03.23		100%
18			Programmere en simulator som likner anlegget det er modellert etter, på best mulig måte	25 days	Wed 15.02.23	Tue 21.03.23		100%
19			Optimalisere parameter	25 days	Wed 15.02.23	Tue 21.03.23		100%
20			Test nødvendige funksjoner beskrevet i funksjonsbeskrivelse	25 days	Wed 15.02.23	Tue 21.03.23		100%
21			Lage en brukerveiledning	2 days	Mon 20.03.23	Tue 21.03.23		100%
22			<b>Programmering av smartstyring</b>	<b>23 days</b>	<b>Wed 22.03.23</b>	<b>Fri 21.04.23</b>	<b>15</b>	<b>100%</b>
23			Vurder best reguleringsstrategi	2 days	Wed 22.03.23	Thu 23.03.23		100%
24			Design funksjonaliteten for reguleringen	17 days	Fri 24.03.23	Mon 17.04.23	23	100%

Project: Gantt diagram  
Date: Fri 12.05.23
















Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	% Work Complete
25			Vurder hvilken informasjon som skal ligge tilgjengelig I OPCUA	23 days	Wed 22.03.23	Fri 21.04.23			100%
26			Publiser relevant informasjon i OPCUA	23 days	Wed 22.03.23	Fri 21.04.23			100%
27			Iverksette samarbeid mellom smartstyring og	5 days	Mon 17.04.23	Fri 21.04.23			100%
28			<b>Presentasjon av relevant informasjon i et web-dashboard</b>	<b>23 days</b>	<b>Wed 22.03.23</b>	<b>Fri 21.04.23</b>	<b>9;15</b>		<b>100%</b>
29			Benytt relevant informasjon tilgjengelig I	23 days	Wed 22.03.23	Fri 21.04.23			100%
30			Lage et brukergrensesnitt for systemet i «Perspective» modulen i Ignition hvor diverse informasjon fremstilles	23 days	Wed 22.03.23	Fri 21.04.23			100%
31			Bruke NORVAR sin standard for å opprettholde HPHMI (High performance	3 days	Wed 22.03.23	Fri 24.03.23			100%
32			<b>Slutfase</b>	<b>42 days?</b>	<b>Mon 24.04.23</b>	<b>Tue 20.06.23</b>			<b>100%</b>
33			Teknisk Rapport	20 days	Mon 24.04.23	Fri 19.05.23			100%
34			Forberedelse av muntlig presentasjon	3 days	Thu 15.06.23	Mon 19.06.23			100%

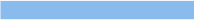


















Project: Gantt diagram  
Date: Fri 12.05.23

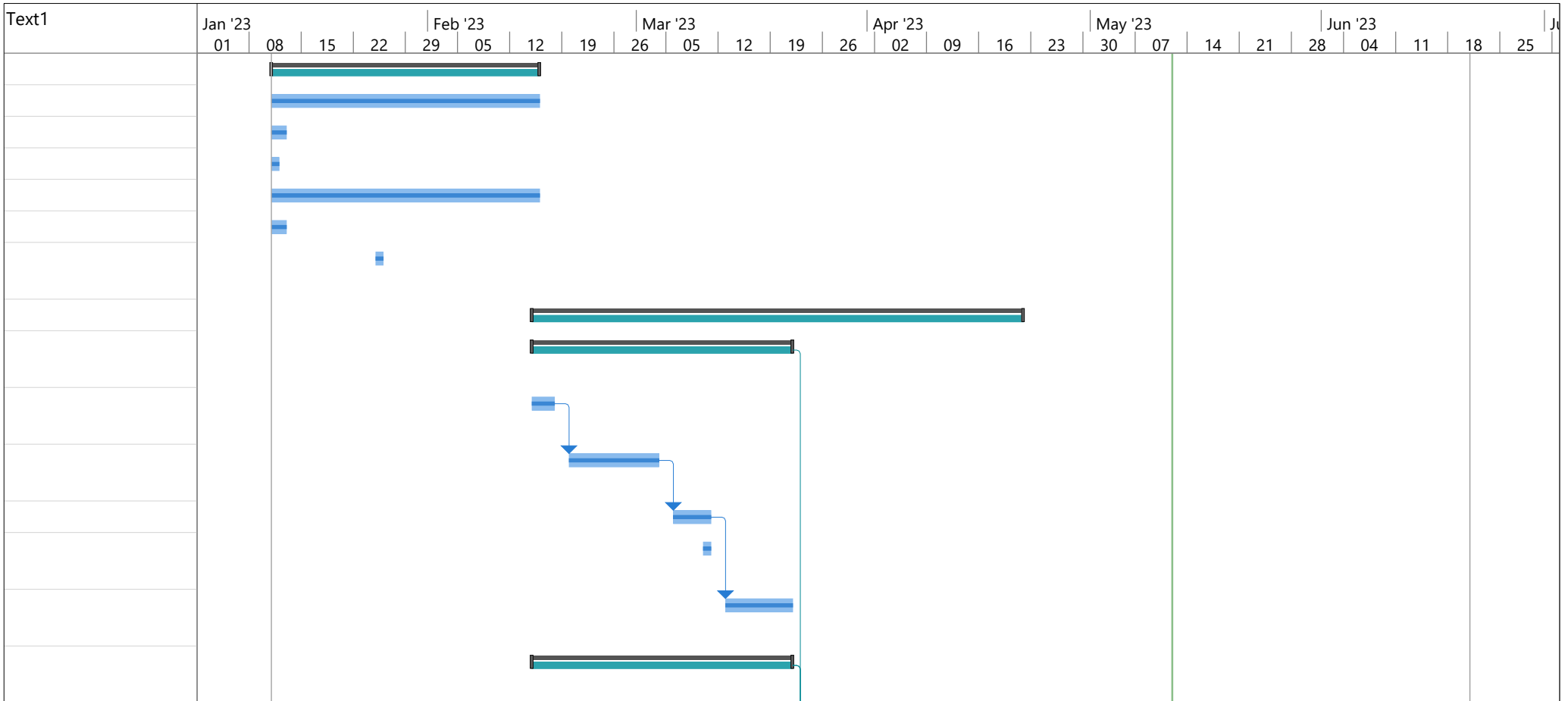
Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			



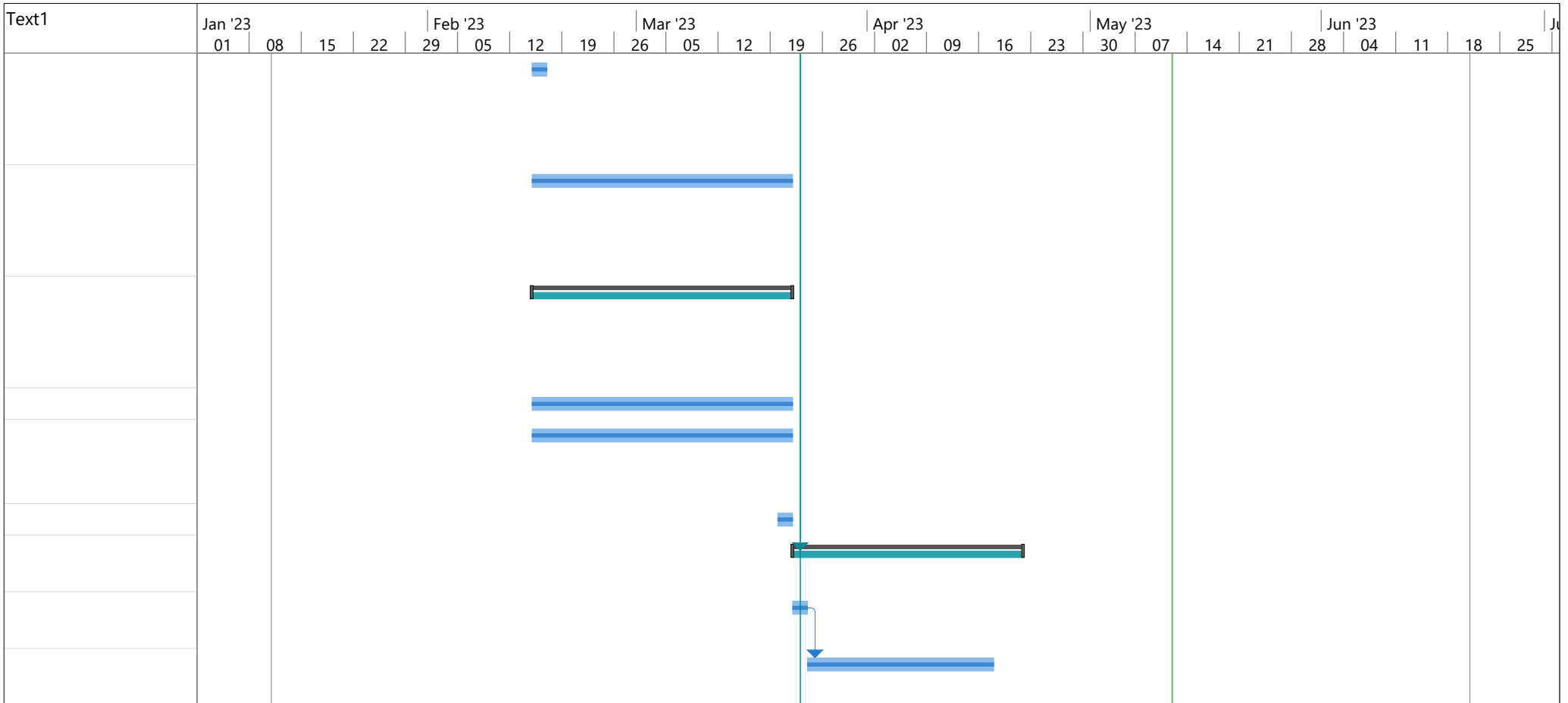
ID		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	% Work Complete
35			<b>Diverse tidsfrister</b>	<b>33 days</b>	<b>Mon 01.05.23</b>	<b>Wed 14.06.23</b>			<b>100%</b>
36			Prosjektbeskrivelse til EXPO katalog	11 days	Mon 01.05.23	Mon 15.05.23			100%
37			Innlevering av Bacheloroppgave i Wiseflow	1 day	Fri 19.05.23	Fri 19.05.23			100%
38			USN EXPO	1 day	Wed 24.05.23	Wed 24.05.23			100%
39			Avslutningsermoni	1 day	Thu 25.05.23	Thu 25.05.23			100%
40			Muntlig høring med sensorene	1 day	Mon 05.06.23	Mon 05.06.23			100%
41			Karakterfastsettelse	1 day	Wed 14.06.23	Wed 14.06.23			100%

Project: Gantt diagram  
Date: Fri 12.05.23

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

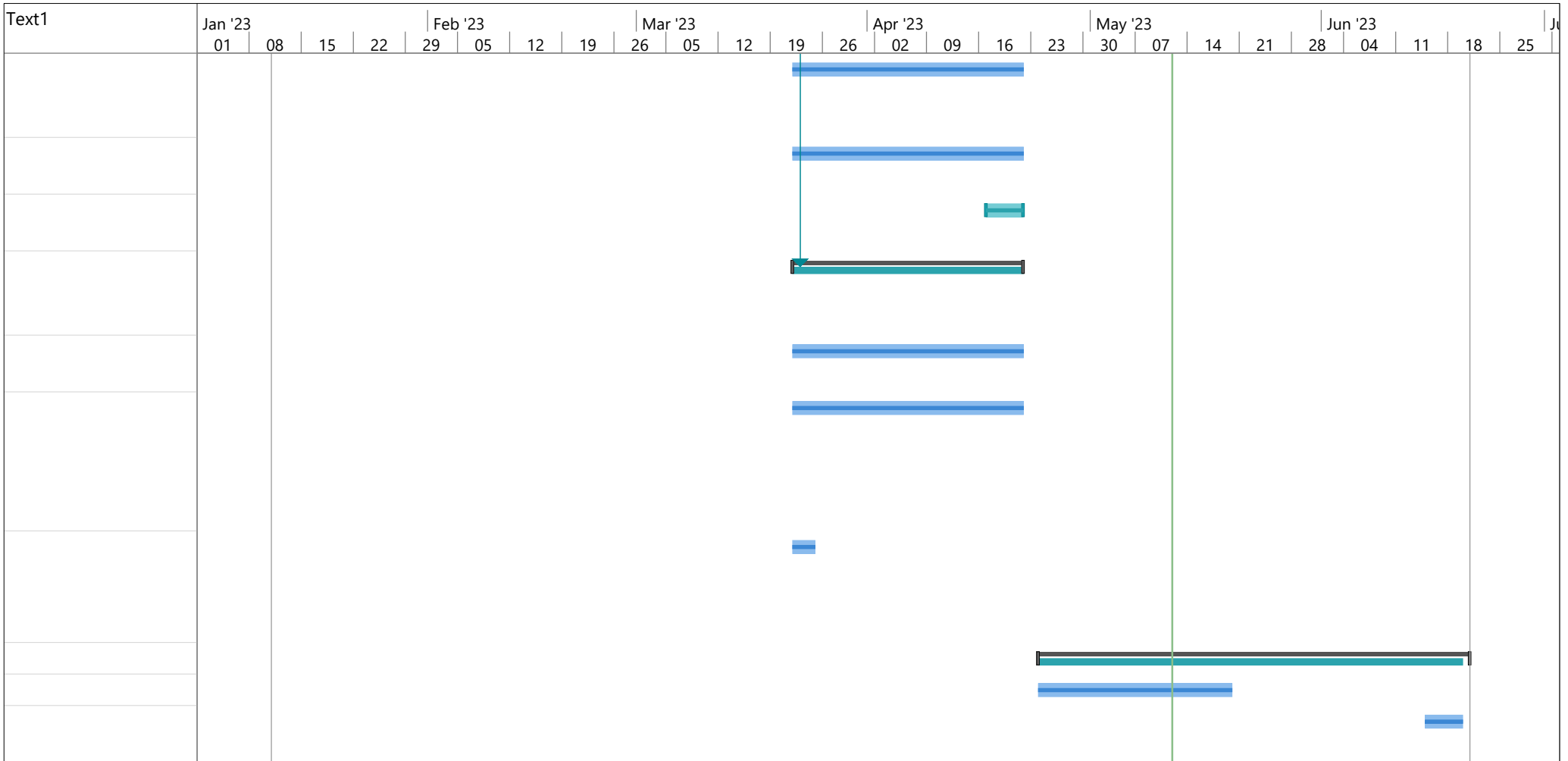


Project: Gantt diagram Date: Fri 12.05.23	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

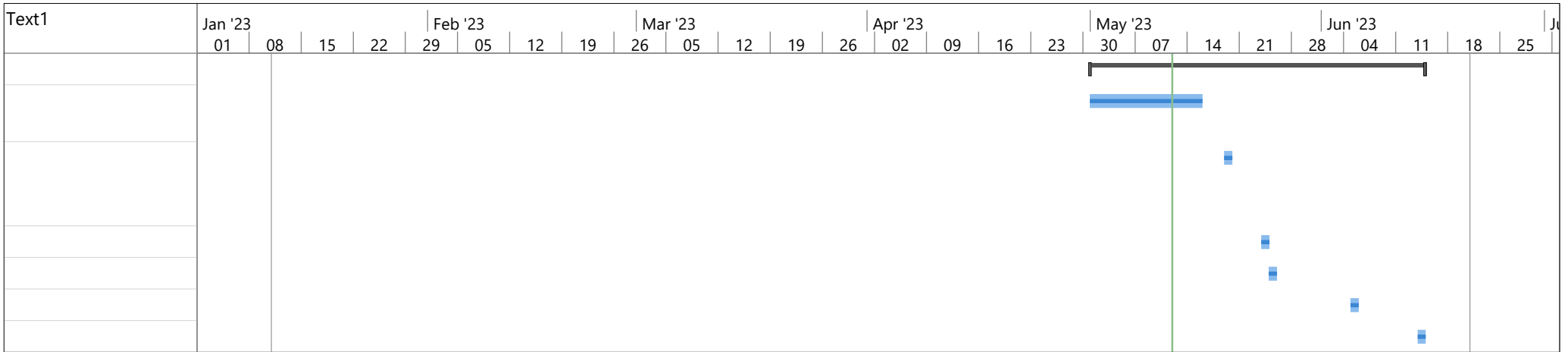


Project: Gantt diagram  
Date: Fri 12.05.23

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			



Project: Gantt diagram Date: Fri 12.05.23	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

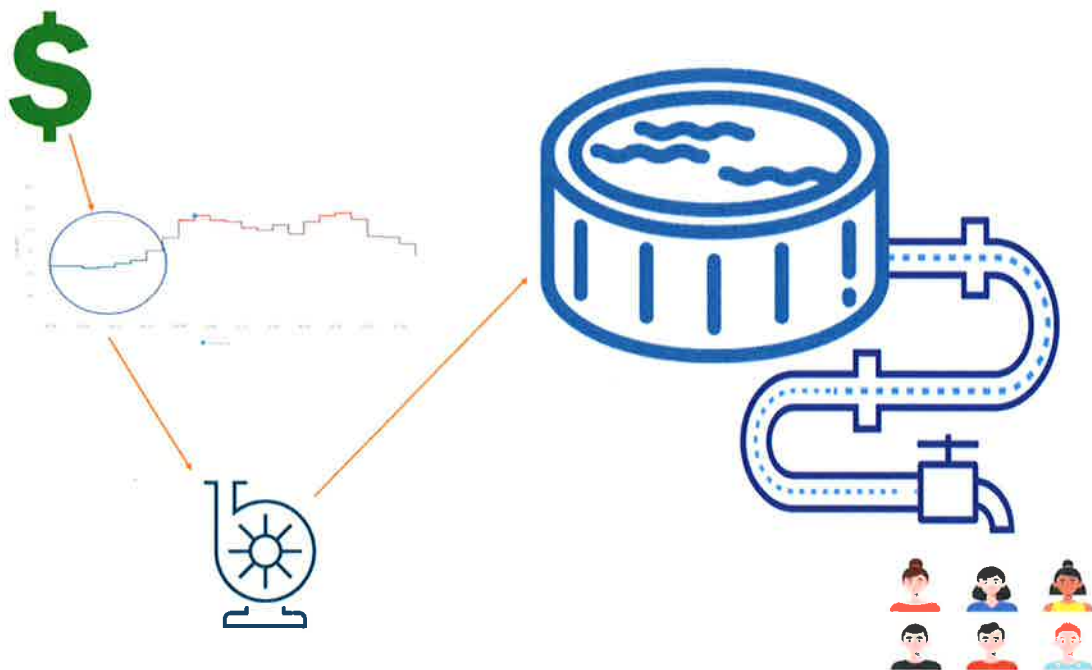


Project: Gantt diagram Date: Fri 12.05.23	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			

TS3000 1 Bachelor

# Vedlegg D

## Medforfattererklæring



IA6-1-23

## Medforfattererklæring - bacheloroppgave

Dette skjemaet skal fylles ut og signeres av alle studentene i prosjektgruppen. Ferdig utfylt og signert skjema skal ligge som et vedlegg i rapporten.

<b>Tittel på oppgaven</b>	Strømeffektiv styring av høydebasseng
<b>Veileder fra USN</b>	Ole Magnus Brastein

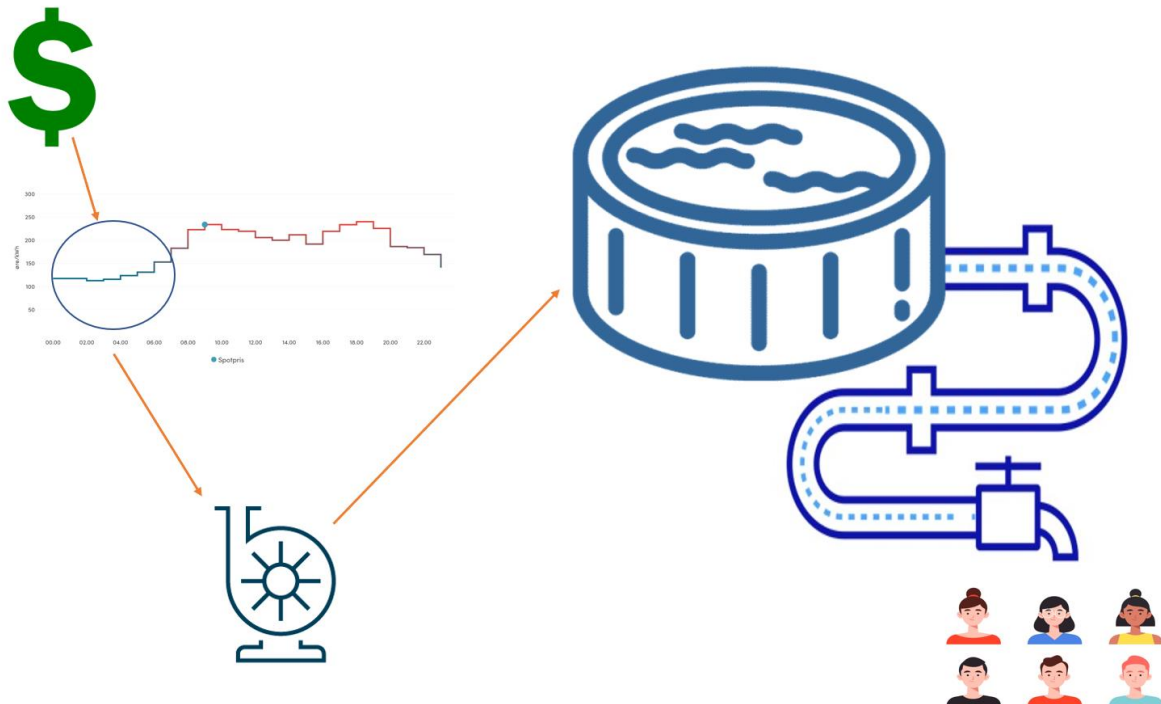
<p><b>Beskriv hva hver student i prosjektgruppen har bidratt med i bacheloroppgaven.</b>  <i>Eksempelvis i forhold til problemformulering, litteratursøk, planlegging av forsøk/valg av metoder, datainnsamling/bygging av prototype, analyse/tolking av data/uttesting, skriving osv.</i>  <i>Husk at alle studentene er ansvarlige for helheten av den innleverte oppgaven.</i></p>
<p>[Jensen, Marius Solli] har bidratt med:  Hovedansvarlig for utvikling av web-dashbord i Perspectivemodulen.  Bidratt med innhenting av strømpriser, rapportskriving.</p>
<p>[Melås, Odin] har bidratt med:  Hovedansvarlig for utvikling/funksjonstesting av smartstyring.  Bidratt med prototype for prosessimulator, rapportskriving.</p>
<p>[Mikkelsen, Rino] har bidratt med:  Hovedansvarlig for utvikling av prosessimulator.  Bidratt med OPC UA kommunikasjon, rapportskriving.</p>
<p>[Monsen, Fredrik Pedersen] har bidratt med:  Hovedansvarlig for web-API (innhenting av strømpriser), OPC UA kommunikasjon.  Bidratt med utvikling av webdashbord, rapportskriving.</p>

Dato	Signatur
19/05-23	Odin Melås
19/05-23	Marius Solli Jensen
19/05-23	Fredrik P. Monsen
19/05-23	Rino Mikkelsen

TS3000 1 Bachelor

# Vedlegg E

## Veiledningsavtale



IA6-1-23



## Veiledningsavtale for bacheloroppgave

Studieprogram	Informatikk og Automatisering Porsgrunn Y-vei
Emne	TS3000 1 Bachelor
Studenten(e)s navn	Marius Solli Jensen, Odin Melås, Rino Mikkelsen, Fredrik Pedersen Monsen
Navn på evt. gruppeleder	Gruppelederervrevet rulleres
Oppgavens tittel/arbeidstitel	Strømeffektiv styring av høydebasseng

*Målet med veiledningen er blant annet å gi studentene en dypere forståelse av fagområdet for bacheloroppgaven, erfaring i kritisk problemformulering og trening i å bruke relevant kildestoff og forskningsmetodiske verktøy.*

1. Det gis inntil 15 timer veiledning, inkludert veileders for- og etterarbeid.
2. Veiledningen og oppgaveskrivingen skal basere seg på de kriterier og rammer som gjelder for bacheloroppgaven, herunder krav i studie- og emneplan, krav til kildehenvisninger (jf. [Kildekompasset.no](http://Kildekompasset.no)) og [USNs forskrift om studier og eksamen](#).
3. Det settes krav til obligatorisk oppmøte til veiledning jf. emneplanen. Ikke godkjent fravær fra obligatorisk veiledning innebærer tap av retten til å gå opp til eksamen/få oppgaven sensurert, jf. forskrift om studier og eksamen § 2-2 jf. § 7-1.
4. Veiledningen skal foregå i et aktivt samspill mellom veileder og student(er), og tidspunkt for gjennomføringen skal skje ved at:

**Alt. A** – studenten(e) kontakter veileder for å sette opp timer.

**Alt. B** - veileder setter opp timer.

**Alt. C** – veileder og student(er) løpende avtaler veiledning basert på behov og fremdrift i oppgaveskrivingen.

5. Student(er) og veileder skal møte forberedt til veiledningstimene for å sikre best mulig utbytte av veiledningsprosessen. Senest 2 virkedager før avtalt veiledning skal studenten(e) sende inn et veiledningsgrunnlag som skisserer spørsmål og problemområder studenten(e) ønsker å drøfte.
6. Veileders hovedansvar er å gi råd om formulering og avgrensning av tema og problemstilling, faglitteratur, drøfte og vurdere hypoteser, metoder, fremstillingsform og resultater, samt holde seg orientert om progresjonen i studentens arbeid.
7. Den løpende veiledningen innebærer ikke en forhåndsvurdering av oppgaven. Endelig vurdering av prestasjonen foretas av formelt oppnevnte sensorer i tråd med premisene i universitets- og høyskoleloven § 3-9.
8. Studenten(e) er selv ansvarlig for at den innleverte oppgaven inneholder korrekte kildehenvisninger. Manglende og/eller uriktige henvisninger kan føre til mistanke om fusk. Ved behov for oppfriskning/supplerende opplæring oppfordres studenten(e) til å delta på Universitetsbibliotekets kurstilbud om referanser og kildebruk m.m., se [Undervisningstilbud - Biblioteket USN](#).
9. Avmeldingsfristen for innlevering av bacheloroppgaver følger av vilkårene i USNs forskrift om studier og eksamen kap. 7.

## Del II - særskilt om gruppeoppgaver

*Et akademisk arbeid i gruppe stiller høye krav til studentene både når det gjelder prosessen og det endelige resultatet. For å oppnå et sluttresultat som dokumenterer samtlige deltageres faglige kunnskap og innsikt, kreves det en solidarisk innsats både på det faglig plan og ansvarlighet i forhold til hvordan gruppen skal fungere rent mellommenneskelig.*

10. Sammensetningen av gruppen og vilkårene for oppnevning av gruppemedlemmer følger av emneplan og medfører at:  
  
Gruppen har selv valgt gruppemedlemmene. Gruppeleverervet rullerer mellom gruppemedlemmene.
11. Endringer i gruppesammensetningen kan kun foretas etter avtale med USN og avgjøres av Olav Dæhli. Dokumentasjon på ny gruppesammensetning skal utarbeides som vedlegg til denne avtalen.
12. Dersom én av gruppemedlemmene sier opp studieplassen, mister studieretten, blir midlertidig utestengt, har et langvarig fravær eller tilsvarende, skal veileder og emneansvarlig/-koordinator vurdere konsekvensen for gruppen og omfanget av gruppeoppgaven. Dette for å sikre at kunnskap og ferdigheter blir vurdert på en faglig forsvarlig måte ved sensur.

13. Gruppen er solidarisk ansvarlig for fremdrift og resultat, herunder kravene til kildehenvisninger, og alle gruppemedlemmene skal aktivt og lojalt bidra under hele prosessen med bacheloroppgaven.
14. Eventuelle konflikter i gruppen skal umiddelbart tas opp av den som opplever at det har oppstått konflikt. Gruppeleder er ansvarlig for å løfte saken til diskusjon i gruppen, og ved behov oppsummere resultatet av dette skriftlig. Redegjørelsen kan eventuelt sendes til veileder som underlag for anmodning om bistand til å løse konflikten.
15. Gyldig fravær fra enkeltstående gruppeaktiviteter meldes til gruppeleder, og gjentakende/lengre fravær - f.eks. på grunn av sykdom - meldes til veileder. Ved langvarig fravær hos ett eller flere av gruppemedlemmene vurderer veileder og emneansvarlig/-koordinator om gruppesammensetningen skal endres, jf. pkt. 11.
16. Gruppen har ikke anledning til selv å ekskludere medlemmer fra gruppesamarbeidet etter at det er inngått avtale. Dersom det oppstår konflikt underveis i arbeidet skal gruppen først gjøre en dokumentert innsats for å løse denne. Som deltakere i et akademisk arbeid innenfor høyere utdanning skal alle gruppemedlemmene bidra aktivt i arbeidet med å løse konflikten. Dersom dette likevel ikke lykkes, kan veileder kontaktes for videre bistand se pkt. 14.
17. Gruppeleder er ansvarlig for å innlevere oppgaven på vegne av hele gruppen innen fastsatt innleveringsfrist. Avmeldingsfristen er fire uker, og fakultetet skal informeres skriftlig om avmeldingen, jf. forskrift om studier og eksamen ved USN kap. 7.
18. Vurdering av oppgaven skjer samlet for gruppen, selv om karakteren gis individuelt. Det framgår av emneplanen om det skal gjennomføres justerende muntlig eksamen. Ved gruppegjennomføring kan den individuelle sluttkarakteren justeres med mer enn én karakter jf. USNs forskrift om studier og eksamen kap. 8. Sensur kunngjøres i Studentweb.
19. Det er individuell klagerett ved gruppeeksamen. Karakteren som oppnås ved klagesensur blir tellende resultat, og en eventuell endring av karakter etter klage vil kun gjelde for den som har klaget.
20. Gruppen kan selv supplere med egne regler for gruppearbeidet i separat avtale, f.eks. regler om samarbeidsform, arbeidsfordeling og gjennomføring av møter, men innholdet må ikke være i motstrid til denne avtalens bestemmelser.

Denne avtalen gjelder fra dato for avtaleinngåelse, og til oppgaven er levert innen ordinær eller utsatt innleveringsfrist, jf. USNs forskrift om studier og eksamen.

Kontaktinformasjon:

Navn	Telefonnummer	Epost	Rolle
Ole Magnus Brastein	47450569	ole@bedko.com	Veileder
Marius Solli Jensen	984 26 567	238065@usn.no	Student
Odin Melås	905 45 050	238057@usn.no	Student
Rino Mikkelsen	948 90 612	230008@usn.no	Student
Fredrik Pedersen Monsen	955 27 850	237051@usn.no	Student

Underskrifter:

Sted/ dato: Porsgrunn, 23.01.2023

Marius Solli Jensen

Marius Solli Jensen (Student)

Fredrik P. Monsen

Fredrik Pedersen Monsen (Student)

Rino Mikkelsen

Rino Mikkelsen (Student)

Odin Melås

Odin Melås (Student)

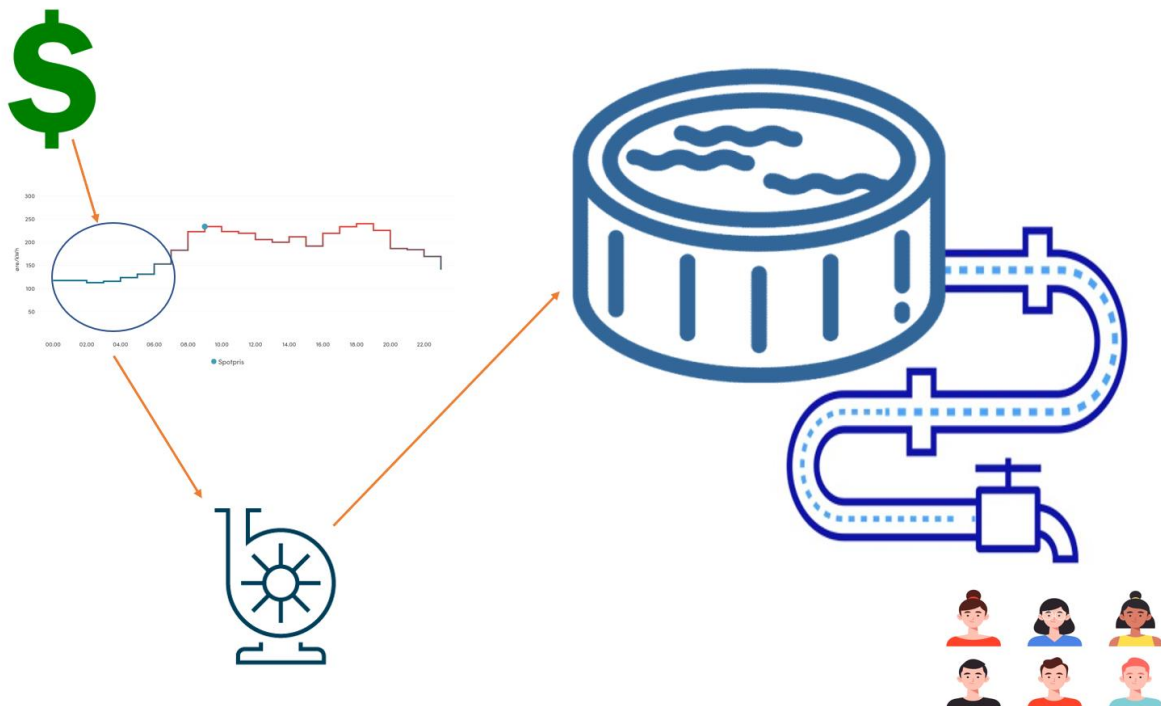
Ole M. Brastein

Ole Magnus Brastein (Veileder)

TS3000 1 Bachelor

# Vedlegg F

## Kildekode



IA6-1-23

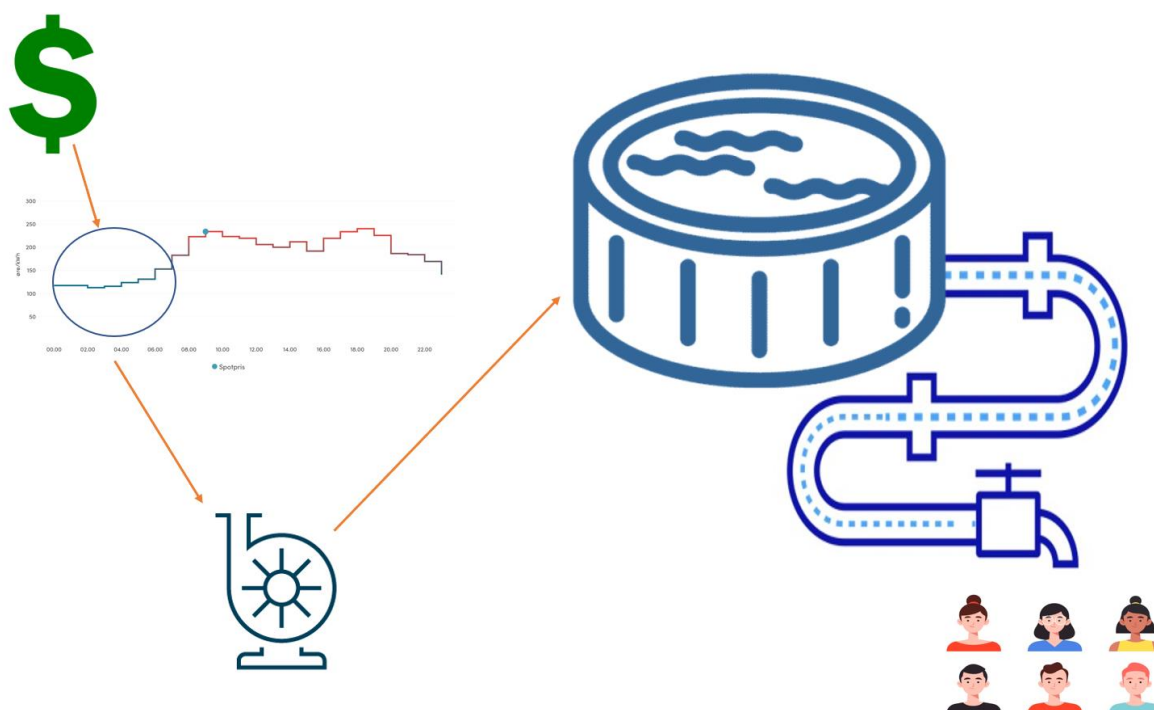
URL til kildekode: <https://github.com/Storeodin/Bachelor/>

Her finner man kildekoden til prosjektet i form av zip-filer til importering i Ignition som ligger i hver sin mappe. Eller kildekoden som tekst som kan åpnes i eksempelvis Visual Studio Code. Se også «README.md» for oppsett av systemet.

TS3000 1 Bachelor

## Vedlegg G

# Brukerveiledning for smartstyring av høydebasseng



IA6-1-23

# Nomenklaturliste

URL	-	Uniform Resource Locator
GB	-	Gigabyte
RAM	-	Random Access Memory
OPC UA	-	Open Platform Communication Unified Architecture
HTTP	-	Hypertext Transfer Protocol
HTTPS	-	Hypertext Transfer Protocol Secure
m <sup>3</sup> /t	-	Kubikmeter i timen



# Innholdsfortegnelse

Nomenklaturliste .....	2
Innholdsfortegnelse .....	3
<b>1 .. Innledning.....</b>	<b>4</b>
1.1 Introduksjon .....	4
1.2 Om brukermanualen.....	4
1.3 Om strømstyrt smartstyring.....	4
<b>2 .. Installasjon og konfigurasjon.....</b>	<b>5</b>
2.1 Krav til systemet .....	5
2.1.1 <i>Operativsystem</i> .....	5
2.1.2 <i>Database</i> .....	5
2.1.3 <i>Systemkrav</i> .....	5
2.2 Installasjon av programvare .....	6
2.2.1 <i>Hvordan installere Ignition</i> .....	6
2.2.2 <i>Importere prosjekt</i> .....	9
2.2.3 <i>Installasjon av Ignition Perspective Workstation og kjøring av systemet</i> .....	10
<b>3 .. Brukergrensesnitt .....</b>	<b>12</b>
3.1 Oversikt over brukergrensesnittet .....	12
3.2 Navigasjon og funksjonalitet.....	12
3.2.1 <i>Hvordan bruke systemet</i> .....	12
<b>Referanser.....</b>	<b>16</b>

# 1 Innledning

Dette kapitlet gir en introduserende beskrivelse av brukermanualen, systemet, og gir leseren informasjon om hvorfor det er gunstig med strømstyrt smartstyring av høydebasseng for kommuner.

## 1.1 Introduksjon

Systemet er et smartstyringsystem som arbeider etter dagens strømpriser for fylling av rent drikkevann til et høydebasseng.

## 1.2 Om brukermanualen

Denne brukermanualen har som hensikt å formidle tilstrekkelig med kunnskap og informasjon om systemet til teknisk kyndige brukere, samt gi brukeren god nok forståelse til å kunne forstå brukergrensesnittet og dets struktur. Brukermanualen kan brukes som en referanseguide til å oppnå tilstrekkelig med kunnskap i hvordan man bruker systemet.

## 1.3 Om strømstyrt smartstyring

Med den nåværende situasjonen i strømmarkedet, har flere kommuner satt strømsparing høyt på prioriteringslisten. Påfylling av rent drikkevann i høydebassenger er en av de mer energikrevende prosessene i det kommunale vann-nettet, og det vil derfor ha stor verdi for kommunene å optimalisere fyllingprosessen i forhold til gjeldende strømpriser i markedet. Dette kan inkludere å justere fyllingsprosessen til tider på døgnet med lavere strømpriser eller å investere i mer energieffektive metoder for fylling. På denne måten kan kommunene redusere sine strømutgifter og bidra til å redusere belastningen på strømmettet.

## 2 Installasjon og konfigurasjon

Dette kapitlet beskriver hvilke krav som må oppfylles for å kunne installere og kjøre programvaren Ignition 8.1. Det er gitt en detaljert guide på hvordan man installerer programvaren på datamaskinen.

### 2.1 Krav til systemet

#### 2.1.1 Operativsystem

Ignition 8.1 krever at datamaskinen oppfyller de nødvendige systemkravene for å kjøre programvaren [1]. Operativsystemene som støttes av Ignition 8.1 er [1]:

- Windows Server 2016/2019/2022
- Windows 10/11
- MacOS (10.16+)
- Linux (med støtte for populære distribusjoner, testet med Ubuntu 20.04)

#### 2.1.2 Database

Ignition 8.1 støtter flere typer databaser [1]. Dette inkluderer Microsoft SQL server, MySQL, MariaDB, PostgreSQL, og enhver database som støtter JDBC driver [1].

#### 2.1.3 Systemkrav

Det kreves å ha følgende systemkrav for å kjøre Ignition 8.1 [1]:

- Dual-core prosessor
- 4 GB RAM
- 10 GB ledig lagringsplass på harddisken

Ignition 8.1 er kompatibel med følgende nettlesere [1]:

- Google Chrome
- Microsoft Edge
- Mozilla Firefox
- Safari

## 2.2 Installering av programvare

I dette delkapitlet vil det bli gått igjennom trinnene for installasjon av programvaren Ignition 8.1. Merk at det er anbefalt å sørge for at datamaskinen oppfyller de nødvendige systemkravene som beskrevet i delkapittel 2.1.3 før installering.

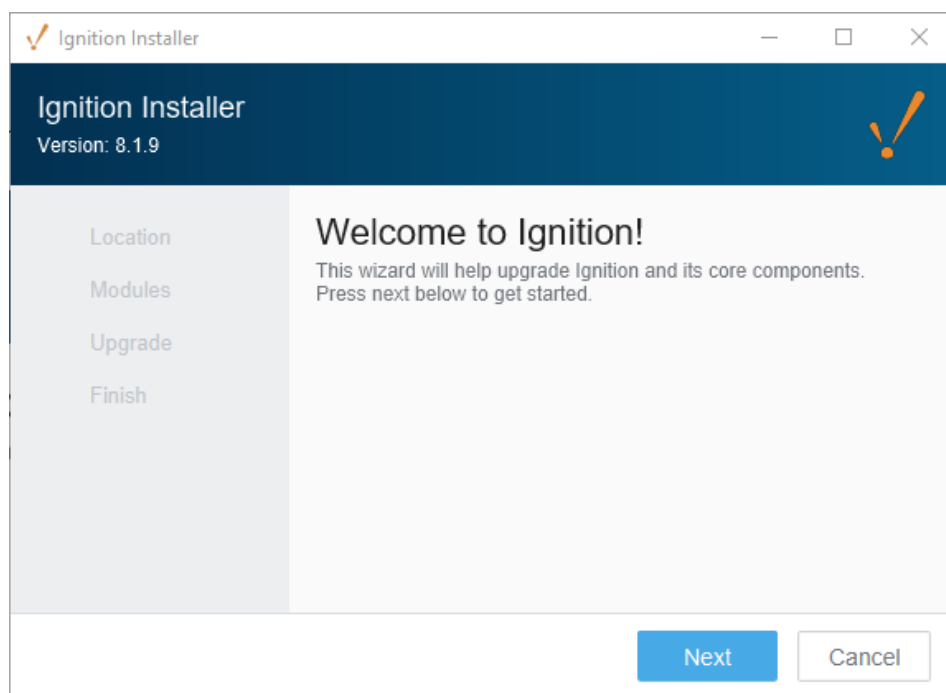
### 2.2.1 Hvordan installere Ignition

Bruk følgende oppskrift for installering av Ignition [2]. Merk at dette oppsettet kun er for Microsoft Windows. For installering av Ignition på andre operativsystem se Ignition sin installeringsside.

<https://docs.inductiveautomation.com/display/DOC81/Installing+and+Upgrading+Ignition>

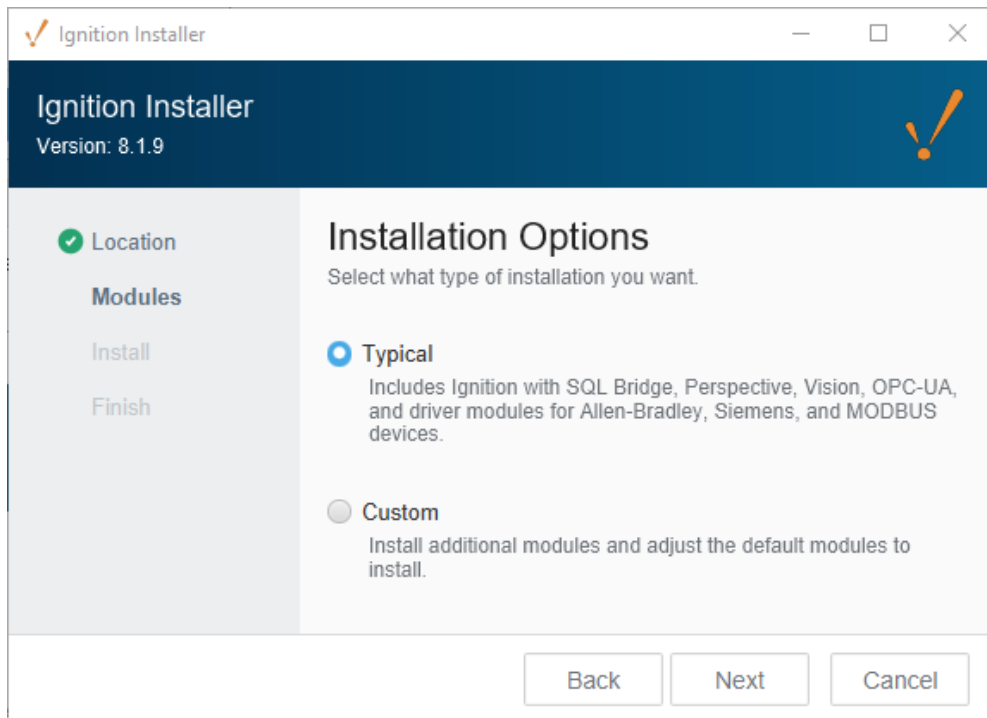
#### Installering av Ignition for Windows:

1. Last ned fra Ignition sin nedlastningsside:  
<https://inductiveautomation.com/downloads/ignition/>
2. Etter nedlastning kjør installasjonsfilen. Et installeringsvindu vil dukke opp på skjermen som vist i Figur 2-1.



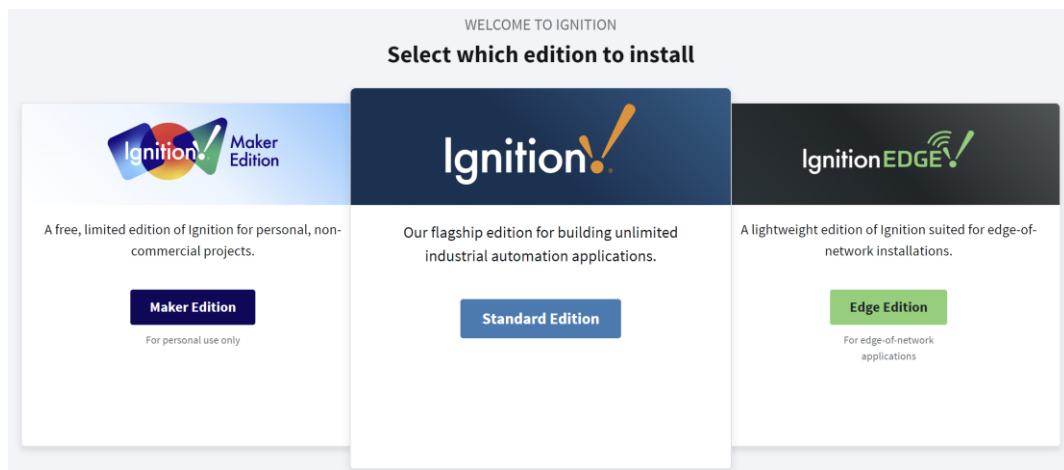
Figur 2-1 Installeringsvindu for Ignition [2]

3. I neste vindu kan man velge to forskjellige installeringstyper vist i Figur 2-2. Typisk installering eller definert installering. Det anbefales for prosjektet å velge typisk installering for å få de ønskede modulene som for eksempel OPC-UA. Trykk «neste» for å starte installeringen av Ignition.



Figur 2-2 Installeringsvalg [2]

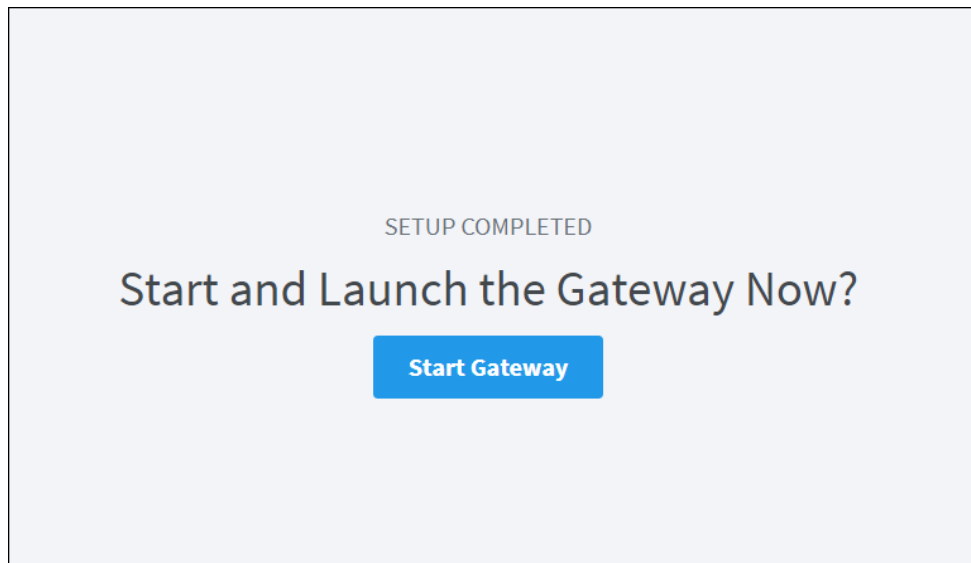
4. Etter at installeringen er ferdig, vil en nettleser åpne et nettleservindu som vist i Figur 2-3. Her må det velges hvilken versjon av Ignition som ønskes å installere. Det anbefales å velge Ignition sin standard utgave for dette prosjektet. Trykk på «Standard Edition» for å komme til neste installeringstrinn.



Figur 2-3 Velg utgave av Ignition [2]

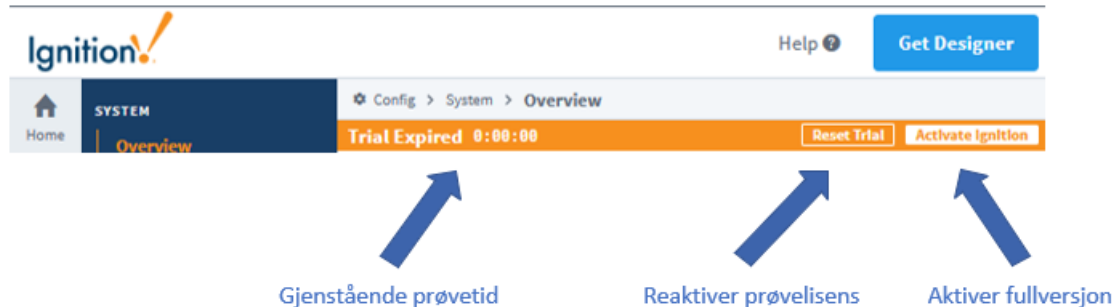
5. I neste steg blir man bedt om å opprette en bruker. Denne brukeren vil kunne ha full tilgang til Ignition. Brukeropplysningene kan modifiseres senere.
6. HTTP konfigurasjon. Her blir man bedt om å velge porten man ønsker Ignition skal benytte. Det anbefales å velge standardporten for HTTP som er 8088, og HTTPS sin standardport som er 8043.

- Oppsettet er fullført. Man kan nå starte Ignition Gateway, vist i Figur 2-4. Benytt <http://localhost:8088> for å være i Ignition sin Gateway.



Figur 2-4 Oppsett ferdig. Start Gateway [2]

- Før Ignition kan benyttes må Ignition aktiveres. Aktiver for prøveperiode lisens eller aktiver for fullversjon, se Figur 2-5.



Figur 2-5 Aktivering av Ignition

Merk at hvis det ønskes flere detaljer om installeringen av Ignition. Anbefales det å sjekke Ignition sin installeringsside for Ignition.

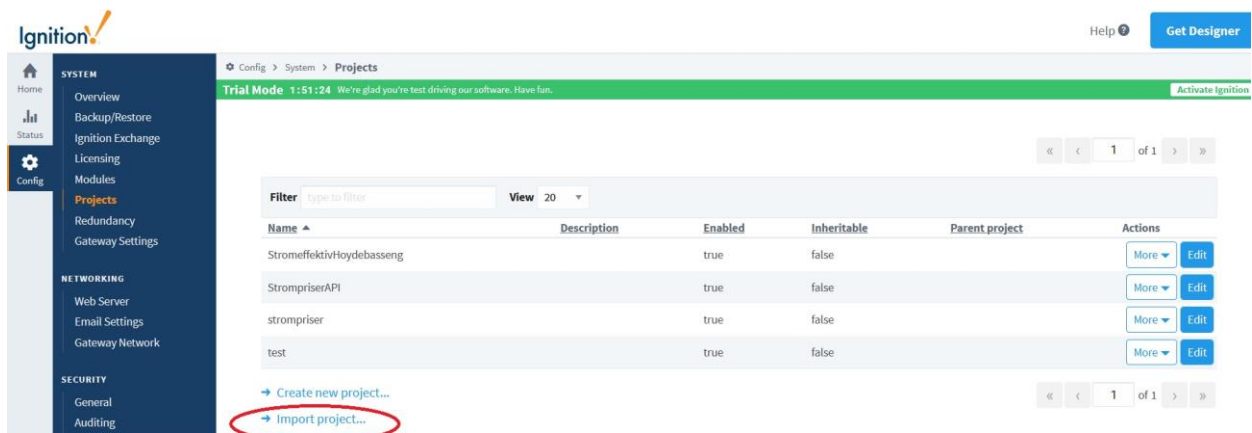
<https://docs.inductiveautomation.com/display/DOC81/Installing+and+Upgrading+Ignition>

## 2.2.2 Importere prosjekt

For å kunne importere et prosjekt anbefales det å gjøre dette i Igniton Gateway. Bruk <http://localhost:8088>. Det kan brukes Igniton Designer Launcher, men dette blir ikke vist i denne brukerveiledningen.

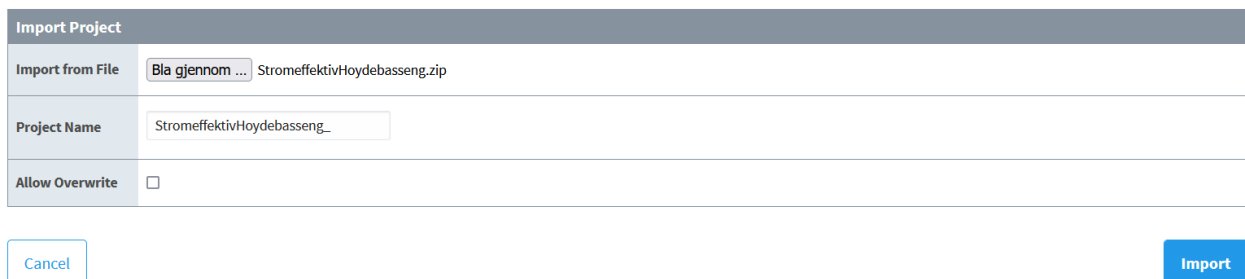
### Hvordan importere prosjekter:

1. Naviger til meny «System», naviger så til «Projects». Her vil man kunne importere eller lage nye prosjekter. Velg her importer prosjekt vist i Figur 2-6.



Figur 2-6 Oversikt over prosjekter og importering av prosjekter

2. Velg «StromeffektivHoydebasseng.zip» og importer prosjekt, vist i Figur 2-7.



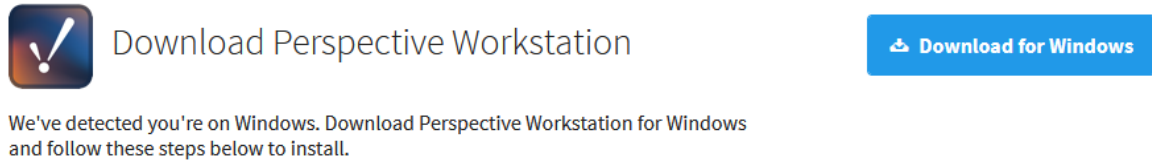
Figur 2-7 Importer valgt prosjekt

### 2.2.3 Installasjon av Ignition Perspective Workstation og kjøring av systemet

For å kunne benytte systemet. Må det installeres Perspective Workstation.

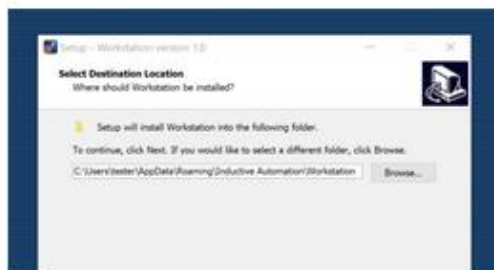
#### Hvordan installere:

1. Naviger til Home i <http://localhost:8088>. Bla ned til Perspective Workstation. Trykk på «Download Perspective Workstation» [3]. Vist i Figur 2-8.



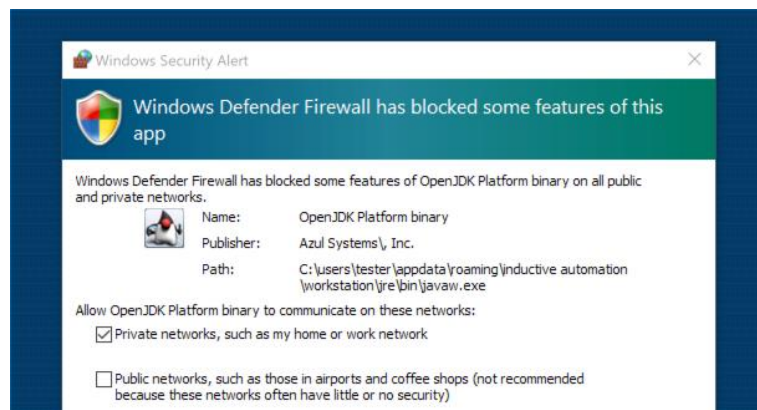
Figur 2-8 Nedlastning av Perspective Workstation [3]

2. Åpne installeringen, og følg instruksene i installeringen. Vist i Figur 2-9



Figur 2-9 Installeringsoppsett [3]

3. Huk av for privat og offentlig nettverk for Windows Defender brannmur. Vist i Figur 2-10.

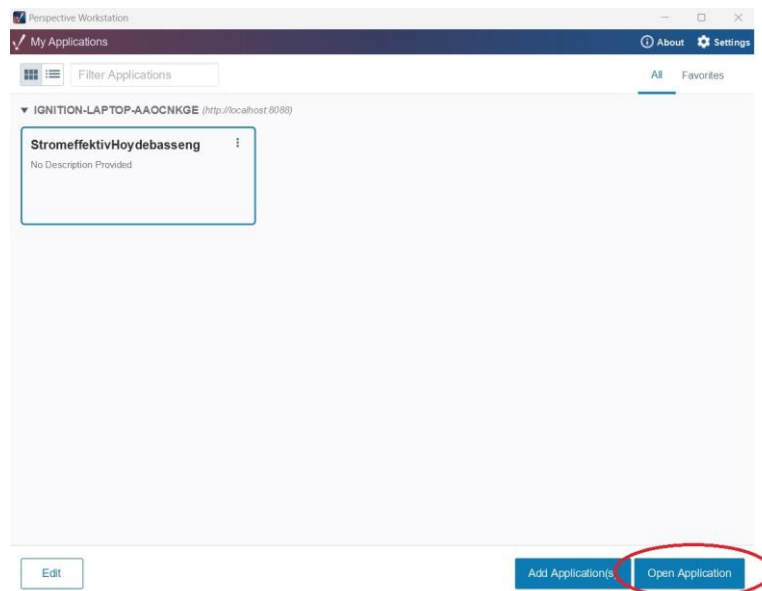


Figur 2-10 Brannmur [3]



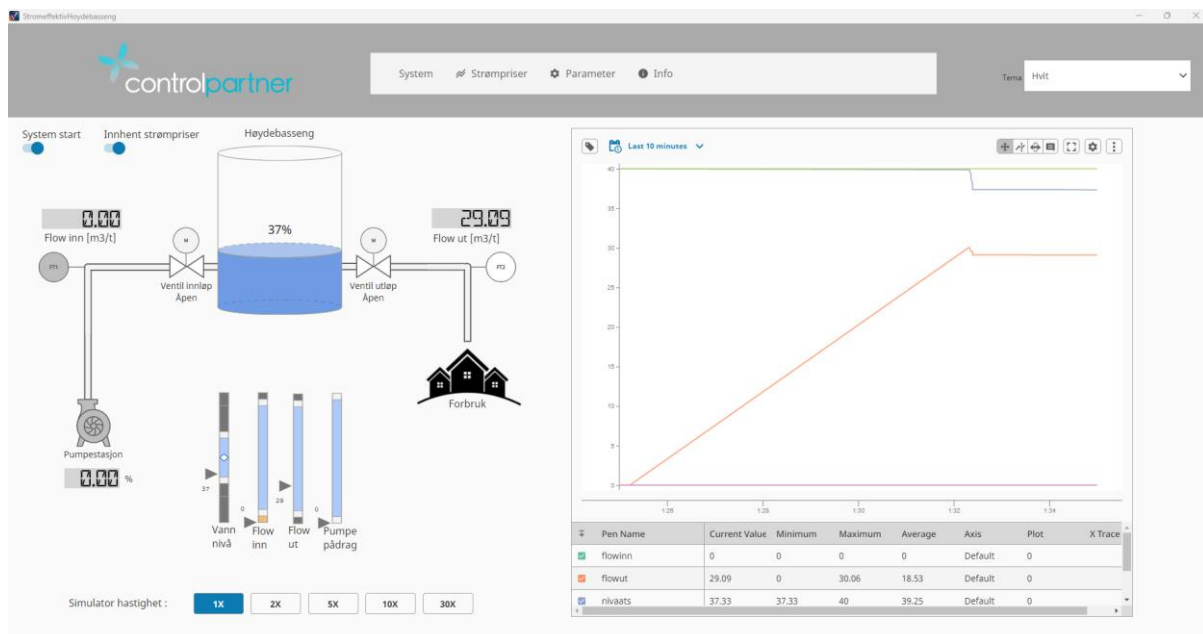
## Hvordan starte prosjektet:

1. For å starte prosjektet. Velg «StromeffektivHoydebasseng», og trykk på «open application» som vist i Figur 2-11.



Figur 2-11 Start applikasjonen

2. Visning av systemet, som vist Figur 2-12.



Figur 2-12 Visning av systemet

## 3 Brukergrensesnitt

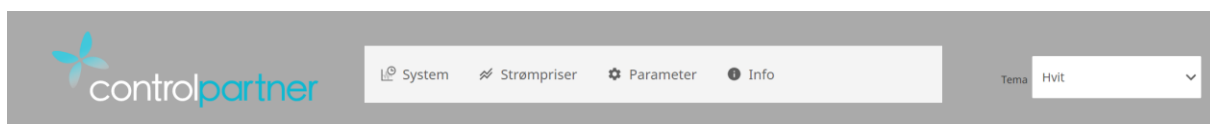
Dette kapitlet tar for seg det grafiske brukergrensesnittet av systemet, hvordan det er bygd opp, hvordan det fungerer, samt en guide på hvordan det kan opereres av den kyndige brukeren.

### 3.1 Oversikt over brukergrensesnittet

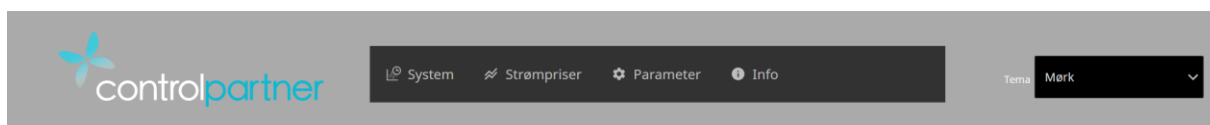
Brukergrensesnittet er bygd opp av fire ulike visningsområder. Ett for «System», ett for «Strømpriser», ett for «Parameter» og ett for «Info». Alle disse fire visningsområdene

### 3.2 Navigasjon og funksjonalitet

Det er utviklet en navigasjonsbar øverst i brukergrensesnittet som fungerer som en «header» i alle visningsområdene. Dette er for at det skal være enkelt for brukeren å navigere seg mellom disse fire visningsområdene. Det er også mulighet for brukeren å velge ulike temaer for brukergrensesnittet, slikt som for eksempel tema «mørk». Navigasjonsbaren er vist med hvitt tema i Figur 3-1, og med mørkt tema i Figur 3-2. Det er videre forklart hvordan man bruker systemet, beskrevet i underkapittel 3.2.1.



Figur 3-1 Navigasjonsbar med tema: hvit

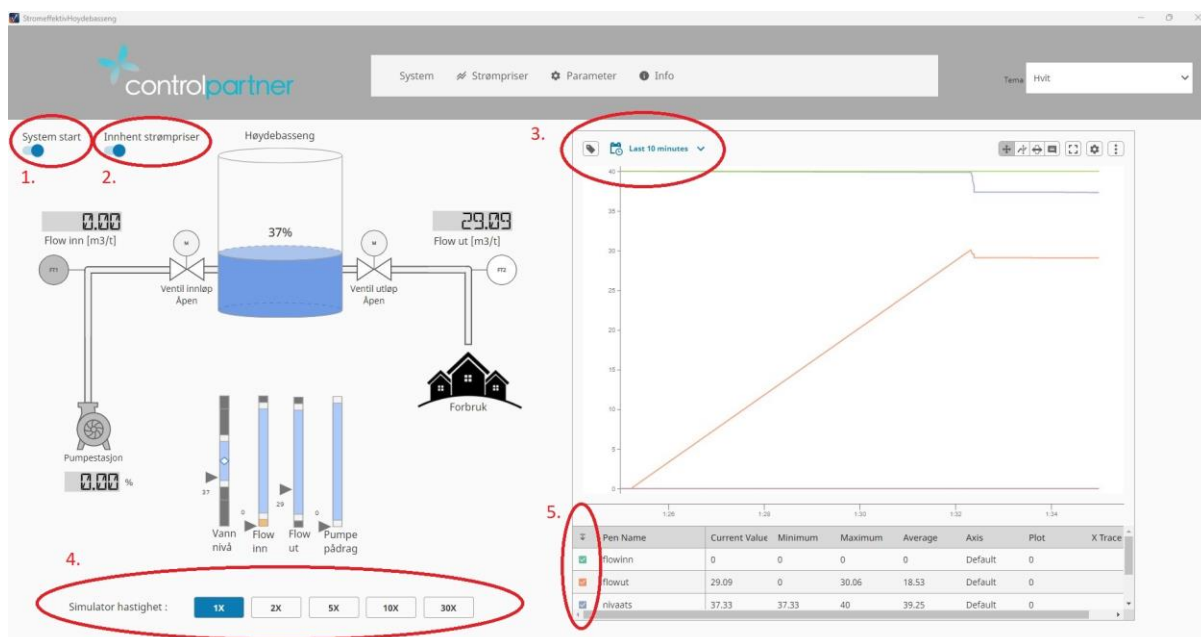


Figur 3-2 Navigasjonsbar med tema: mørk

#### 3.2.1 Hvordan bruke systemet

I brukergrensesnittet til visningsområde «System» er det utviklet en rekke funksjonaliteter som vist i Figur 3-3.

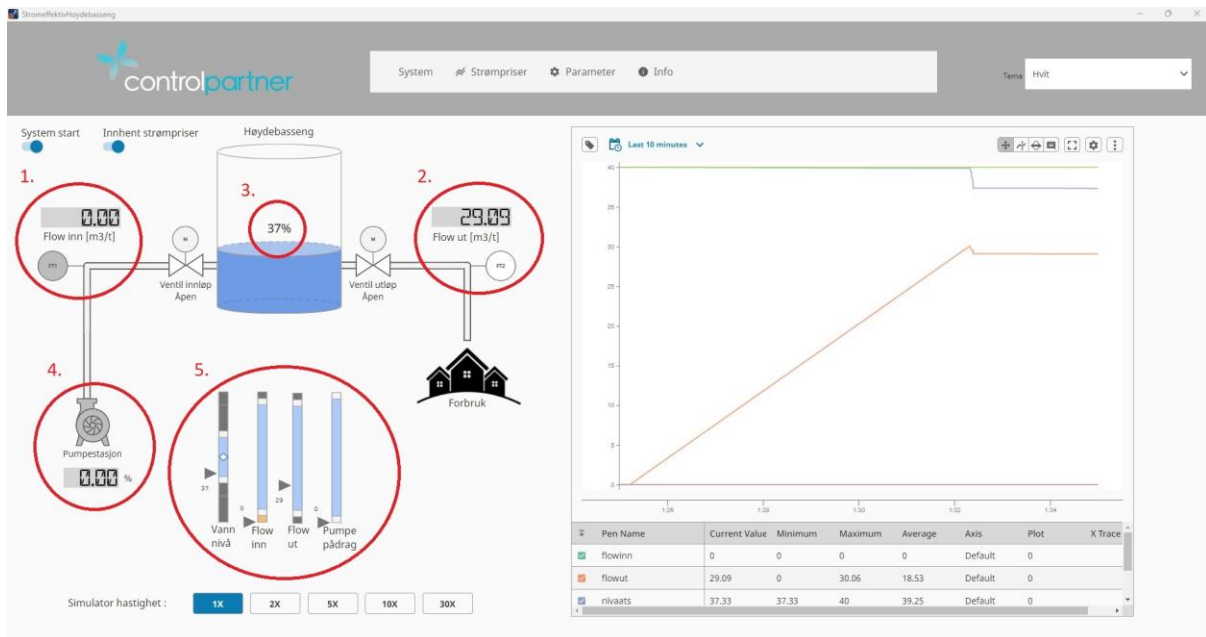
1. System start – En knapp som starter simuleringen av systemet.
2. Innhent strømpriser – En knapp som starter programmet som innhenter strømpriser fra Web-API.
3. Tag browser og historisk fremvisning – Her kan brukeren velge tagger som kan vises i grafen, samt definere hvilket tidsrom som er ønskelig å se for.
4. Simulator hastighet – her kan brukeren definere hastigheten prosessen skal simuleres i. Hvor 1X er prosessen i sanntid. Det er også en tilbakemelding i form av farge slik at brukeren vet hvilken hastighet som benyttes.
5. Pen name – Her kan operatøren legge til og fjerne valgte tagger, slik at grafen viser nøyaktig de tag verdiene som er ønskelig å vise.



Figur 3-3 Systembilde sine funksjonaliteter

I brukergrensesnittet til visningsområde «System» er det utviklet en rekke grafiske fremvisninger av prosessverdier som vist i Figur 3-4.

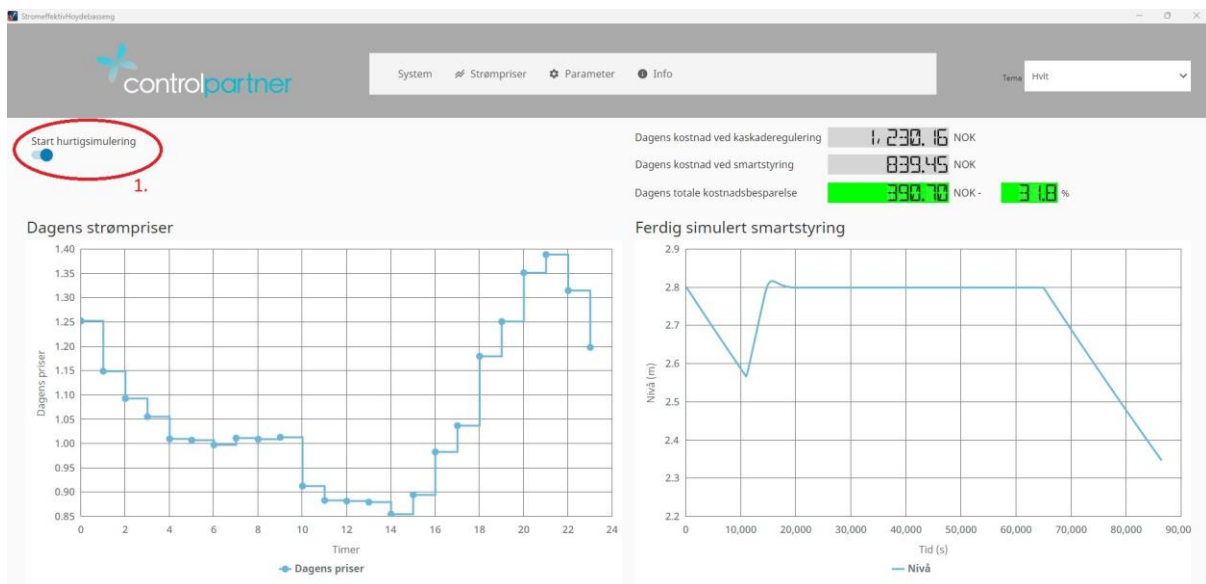
1. Flow inn – Her er det en grafisk fremvisning av strømmingen som kommer inn i høydebassenget samt en grafisk status fremvisning av instrumentet. Når instrumentet er grått, er det ikke aktivt. Verdien er vist i  $m^3/t$ .
2. Flow ut - Her er det en grafisk fremvisning av strømmingen som går ut av høydebassenget samt en grafisk status fremvisning av instrumentet. Når instrumentet er hvitt, er det aktivt. Verdien er vist i  $m^3/t$ .
3. En numerisk fremvisning av nivået i høydebassenget vist i prosent.
4. En grafisk fremvisning av pådragssignalet til pumpestasjonen vist i prosent. Pumpestasjonen er også grafisk fremvist i form av et symbol i grå tilstand (ikke aktiv). Ved aktiv tilstand så illustreres dette grafisk og symbolet endrer farge til hvit.
5. Bevegelige analoge indikatorer – Disse fremviser prosessen sine verdier, og gir operatøren en bedre oversikt over prosessverdiene i forhold til optimale driftstilstander.



Figur 3-4 Systembilde sine prosessverdier

I brukergrensesnittet til visningsområde «Strømpriser» er det utviklet en funksjonalitet som vist i Figur 3-5.

1. Start hurtigsimulator – En knapp som starter hurtig simulatoren

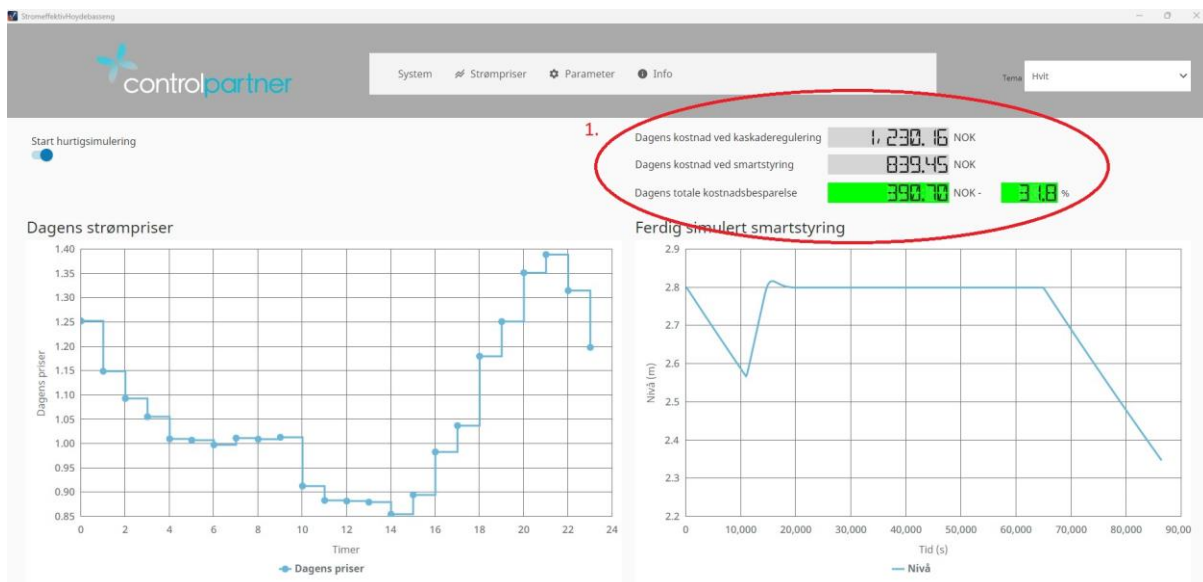


Figur 3-5 Strømprisbilde sin funksjonalitet

I brukergrensesnittet til visningsområde «Strømpriser» er det utviklet en rekke grafiske fremvisninger av prosessverdier som vist i Figur 3-6.

1. Grafisk fremvisning av prosessverdier
  - a. Total dagskostnad ved kaskaderegulering – Her vises dagens totale kostnad ved å kjøre systemet med en ren kaskaderegulering

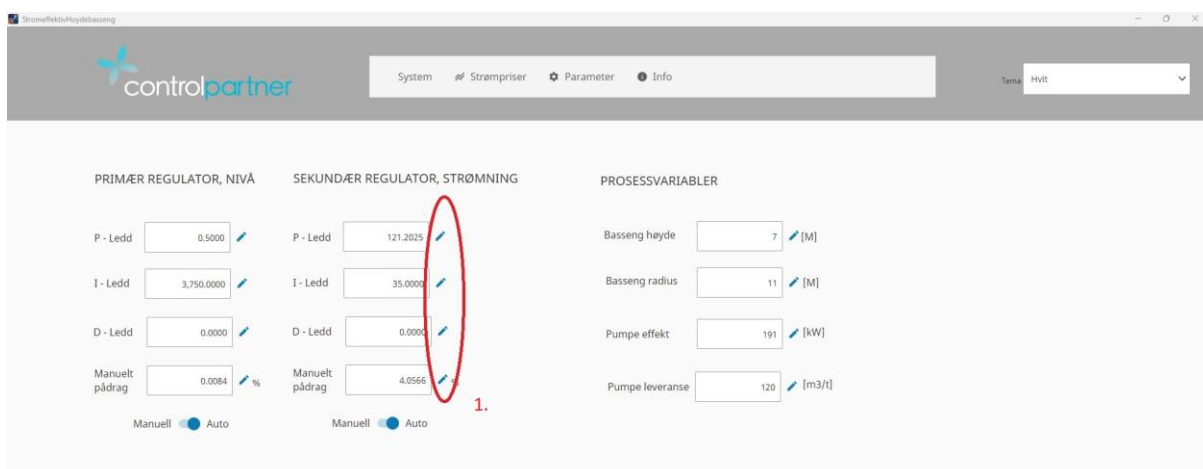
- Total dagskostnad ved smartstyring – Her vises dagens totale kostand ved å kjøre systemet med strømstyrt smartstyring
- Dagens totale kostnadsbesparelse – Her vises differansen mellom ren kaskade regulering og strømstyrt smartstyring, som igjen er den totale besparelsen for døgnet ved å benytte smartstyring.
- Dagens totale kostnadsbesparelse - Her vises differansen mellom ren kaskade regulering og strømstyrt smartstyring i prosent.



Figur 3-6 Strømprisbilde sine prosessverdier

I brukergrensesnittet til visningsområde «Strømpriser» er det utviklet en rekke grafiske fremvisninger av prosessvariabler samt funksjonalitet for å endre disse verdiene. Brukergrensesnittet for dette er vist i Figur 3-7.

- Funksjonalitet for endring av regulatorparametere og prosessvariabler. Det er også en grafisk fremvisning av disse verdiene.



Figur 3-7 Funksjonalitet i Parameterbilde

# Referanser

- [1] «Download Ignition,» Inductive Automation, [Internett]. Available: <https://inductiveautomation.com/downloads/ignition/>. [Funnet 1 Mai 2023].
- [2] I. Automation, «Perspective Workstation,» Ignition 8.1 by Inductive Automation, [Internett]. Available: <https://docs.inductiveautomation.com/display/DOC81/Perspective+Workstation>. [Funnet 25 April 2023].
- [3] Inductive Automation, «Download Perspective Workstation,» [Internett]. Available: <http://localhost:8088/web/perspective-workstation?35>. [Funnet 1 Mai 2023].