



Mika Kukkonen

Muhvausprosessin digitalisointi ennaltaehkäisevän kunnossapidon osaksi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

6.5.2024

Tiivistelmä

Tekijä:	Mika Kukkonen
Otsikko:	Muhvausprosessin digitalisointi ennaltaehkäisevän kunnossapidon osaksi
Sivumäärä:	53 sivua
Aika:	6.5.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine:	Automaatiotekniikka
Ohjaajat:	Kehitysinsinööri Christer Andersson Lehtori Matti Välikylä

Opinnäytetyön tilaaja on Uponor Infra Oy, joka kuuluu Uponor-konserniin. Uponor Oyj on yksi johtavista rakennus- ja ympäristötekniikan toimittajista maailmassa.

Opinnäytetyön aiheena oli muhvausprosessin digitalisointi ennaltaehkäisevän kunnossapidon osaksi. Opinnäytetyön tarkoituksena oli tutkia muhvausprosessiin kuuluvien painejaksojen paineiden käyttäytymistä tilastollisen prosessinohjauksen menetelmiä apuna käyttäen.

Opinnäytetyössä muhvauskoneeseen asennettiin paineanturit, joiden tuottamaa dataa kerätään muhvauskoneelle asennetulle logiikalle ja lähetetään OPC UA -protokollalla tietokoneeseen tehdylle Python-ohjelmalle, jossa data analysoidaan. Analysoinnin tulokset lähetetään takaisin tuotantolinjalla olevalle käyttöliittymälle.

Opinnäytetyöstä tehdyn raportin alussa tutustutaan muhvausprosessiin ja käydään läpi opinnäytetyössä käytettyjä komponentteja sekä niiden valintaperusteita. Lopuksi tutustutaan opinnäytetyössä käytettyihin tiedonsiirtotapoihin, ohjelmointiin ja käyttöliittymään.

Testauksien ja koeajojen jälkeen voitiin todeta, että järjestelmä toimii suunnitellulla tavalla. Käyttöliittymästä pystytään helposti havaitsemaan mahdolliset ongelmat. Järjestelmän monistaminen muille tuotantolinjoille ja mahdolliset rahalliset säästöt selviävät vasta järjestelmän oltua käytössä pitempään.

Avainsanat: muhvausprosessi, tilastollinen prosessinohjaus, OPC UA

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Mika Kukkonen
Title: Digitizing the Cuffing Process as Part of Preventive Maintenance
Number of Pages: 53 pages
Date: 6 May 2024

Degree: Bachelor of Engineering
Degree Programme: Electrical and Automation Engineering
Professional Major: Automation technology
Supervisors: Christer Andersson, Development Engineer
Matti Välikylä, Senior Lecturer

The client of this thesis work was Uponor Infra Oy, which is part of the Uponor Group. Uponor Oyj is one of the world's leading suppliers of construction and environmental technology.

The topic of this thesis work was digitizing a cuffing process as part of preventive maintenance. Aim was to investigate behaviour of substances in pressure cycles of the cuffing process, using statistical process control methods as an aid.

In this thesis work, pressure sensors were installed in the cuffing machine and the data that was produced was collected by the logic installed in the cuffing machine and sent to a Python program made on a computer using the OPC UA protocol, where the data is analysed. The results of analysis are sent back to the user interface on the production line.

At the beginning of this thesis report, the cuffing process is introduced, and the components used in the thesis and their selection criteria are explained. After this, data transfer methods, programming and user interface used in this thesis are presented.

After testing and trial runs, it could be concluded that the system works as planned. Possible problems can be easily detected from the user interface. Copying of the system for other production lines and possible financial savings will only become clear after the system has been in use for a longer period.

Keywords: Cuffing process, statistical process control, OPC UA

Sisällys

Lyhenteet

1	Johdanto	1
2	Muoviputkien valmistusprosessi	2
2.1	Putkituotantolinja	3
2.2	Muhvauskone	3
2.3	Muhvausprosessi paineilmalla	4
3	Komponentit	6
3.1	Ohjelmoitavat logiikat	6
3.2	IEC 61131-standardi	7
3.3	Siemensin logiikat	7
3.4	Logiikoiden muistialueet	8
3.5	Siemensin käyttämä Profinet-kenttäväylä	9
3.6	Siemens HMI -paneelit	10
3.7	Logiikan valintakriteerit	10
3.8	Painemittaus	12
4	Tiedonsiirto	12
4.1	OPC UA -protokolla	13
4.1.1	OPC UA TIA Portal -ohjelmointiympäristössä	15
4.1.2	OPC UA Python-ohjelmassa	18
4.2	TIA Portalin PUT- ja GET-komento	19
	PUT-komennon käyttö	20
5	Koeajo	21
6	Data-analyysi	22
6.1	Aritmeettinen keskiarvo ja keskihajonta	23
6.2	Normaalijakauma	24
6.3	Lineaarinen yhteys ja korrelaatiokerroin	25
6.4	Normaaliustesti	26
6.5	Ohjausrajojen määrittely	27
6.6	SPC-kortti	28

7	Ohjelmointi	30
7.1	Siemens TIA Portal -ohjelmointiympäristö	30
7.1.1	Ohjelmointikieli Function Block Diagram	31
7.1.2	Funktiot	32
7.2	Visual Studio -ohjelmointiympäristö	32
7.3	Ohjelmistojen suunnittelu	33
7.3.1	TIA Portal -ohjelman toiminnallisuudet	34
7.3.2	Python-ohjelman vuokaavio	35
7.3.3	TIA Portal -ohjelman rakenne	35
7.3.4	Python-ohjelman rakenne	38
7.3.5	Pääohjelma	39
7.3.6	Python-ohjelman moduulit	40
7.4	Käyttöliittymän kuvat	41
7.5	Käyttöliittymän rakenne	41
7.5.1	Menuvalikko	42
7.5.2	Muhvaussivu	43
7.5.3	Muhvausasetussivu	44
7.5.4	Painejaksokartta	45
7.5.5	Hälytyssivu	46
7.5.6	SPC-korttisivu	47
8	Tulokset	48
9	Yhteenveto	48
	Lähteet	50

Lyhenteet

- A/D: *Analogia-digitaali*. Signaalin muunnos analogisesta digitaaliseen.
- COM: *Component Object Model*. Mahdollistaa ohjelmistojen tiedonvaihdon.
- CPU: *Central Processing Unit*. Mikroprosessori hoitaa laskentaa ja tietojenkäsittelyä tietokoneessa.
- CSV: *Comma-Separated Values*. Tiedostomuoto, jossa taulukkomuotoista tietoa on tallennettuna tekstimuotoon. Tiedot ovat yleensä pilkulla erotettu.
- DB: *Data Block*. TIA Portal -ohjelmistossa käytettävä tietolohko.
- FB: *Function block*. Muistin sisältävä ohjelmalohko.
- FBD: *Function Block Diagram*. Graafinen ohjelmointikieli.
- FC: *Function*. Ohjelmalohko, joka ei sisällä muistia.
- HMI: *Human Machine Interface*. Käyttöliittymä.
- IEC: *International Electrotechnical Commission*. Sähköalan standardointi-organisaatio.
- IRT: *Isochronous Real Time*. Vakioaikaerokäyrä reaaliaikainen.
- LAN: *Local Area Network*. Paikallinen verkko.
- LCL: *Lower Control Limit*. Alempi ohjausraja.
- MES: *Manufacturing Execution System*. Järjestelmä tuotannon tietojen keräämiseen.

- NS: Namespace index. OPC UA -palvelimen käyttöliittymän nimiavaruuden indeksiluku.
- OPC UA: *OPC Unified Architecture*. OPC-järjestön kehittämä avoimen lähdekoodin tiedonsiirtostandardi teollisuuden tarpeisiin.
- OSI-malli: *Open systems interconnection reference model*. OSI-malli koostuu seitsemästä kerroksesta. Kerrokset esittävät, kuinka tiedonsiirto toimii.
- PID: *Proportional-integral-derivative*. Perussäädin säätötekniikassa.
- PLC: *Programmable Logic Controller*. Ohjelmoitava logiikka.
- Python: Tietokoneen ohjelmointikieli.
- RT: *Real time*. Reaaliaikainen.
- SPC: *Statistical Process Control*. Tilastollinen prosessinohjaus.
- UCL: *Upper Control Limit*. Ylempi ohjausraja.
- URI: *Uniform Resource Identifier*. Merkkijono, joka osoittaa tietyn tiedon paikan.

1 Johdanto

Opinnäytetyön aiheena oli muhvausprosessin digitalisointi ennaltaehkäisevän kunnossapidon osaksi. Opinnäytetyön tarkoituksena oli asentaa putkituotantolinjan muhvauskoneeseen paineanturit. Paineanturien painelukemat kerätään ja käsitellään muhvauskoneelle asennetulla ohjelmoitavalla logiikalla, josta tiedot siirretään tuotantolinjan ekstruuderilla olevalle ohjelmoitavalle logiikalle, joka toimii palvelimena ohjelmoitavan logiikan ja tietokoneen välisessä OPC UA -protokollalla toteutetussa tiedonsiirrossa. Tietokoneella olevalla Python-ohjelmalla analysoidaan paineanturien lukemat. Painelukemat analysoidaan tilastollisen prosessin ohjauksen menetelmillä. Ekstruuderilla oleva ohjelmoitava logiikka operoi myös HMI:tä eli käyttöliittymää, jolle tietokoneen analyysin tulokset palautetaan.

Opinnäytetyön raportissa tutustutaan aluksi muoviputkien valmistusprosessiin ja putkiliitoksien eli muhvien valmistukseen, minkä jälkeen mietitään päättötyössä käytettävien komponenttien valintakriteereitä. Raportin loppupuolella käsitellään tiedonsiirtotapoja, tilastollista prosessinohjausta ja ohjelmointia. Lopuksi tutustutaan vielä opinnäytetyössä tehtyyn käyttöliittymään.

Opinnäytetyössä halusin erityisesti keskittyä tilastollisen prosessin ohjauksen ohella Siemensin S7-sarjan logiikoiden väliseen tiedonsiirtoon sekä logiikoiden ja tietokoneen väliseen OPC UA -protokollalla toteutettuun tiedonsiirtoon. Päättötyö voidaan katsoa onnistuneeksi, mikäli paineantureista saadaan mittausdataa, mittausdata saadaan lähetettyä tehdasverkkoon ja mittausdatan analysointiin saadaan kehitettyä ohjelma, joka palauttaa tiedon, onko mittausdata ohjausrajojen sisäpuolella. Lisäksi operaattorille saadaan reaaliaikaisesti palautettua tieto analyysin tuloksesta.

Työn onnistumisen myötä on mahdollisuus rahallisiin säästöihin säästyvinä materiaalikustannuksina, koska romuputkien määrä vähenee. Operaattoreiden turhan työn tekeminen vähenee parametrien säätämisen vähenemisenä. Työtä

voidaan mahdollisesti tulevaisuudessa myös kopioida muihin laitteisiin ja prosesseihin.

Tässä insinööriyössä ei kerrota todellisia muhvuksessa käytettyjä paineita tai muita parametrejä. Kaikki esimerkeissä käytetyt parametrit ovat kuvitteellisia eivätkä vastaa todellisuutta. Tämä ei kuitenkaan vaikuta tapaan, jolla ohjausrajat lasketaan.

Opinnäytetyön tilaaja on Uponor Infra Oy, joka kuuluu Uponor-konserniin. Uponor oyj on yksi maailman johtavista rakennus- ja ympäristötekniikan toimittajista maailmassa. Uponor toimii 26 maassa. Keskeisimmät liiketoiminta-alueet ovat Eurooppa ja Pohjois-Amerikka. Uponorin liikevaihto vuonna 2022 oli 1,3 miljardia euroa. Uponorilla työskentelee noin 4000 ihmistä. (Avainluvut.)

Uponor-konsernin juuret ajoittuvat vuoteen 1918, jolloin Aukusti Asko-Avonius perusti Lahteen puusepänverstaan, josta myöhemmin tuli Asko Oy. Uponor Oyj perustettiin vuoden 2000 alussa, jolloin vuonna 1982 yhdessä Neste Oy:n kanssa perustettu Oy Uponor Ab lakkautettiin ja fuusioitiin Askopörssi-yhtiöön. Nimeksi päätettiin ottaa tytäryhtiön nimi eli Uponor Oyj. (Better living environments – The story of Uponor 2017; Mikkonen 2008: 5.)

2 Muoviputkien valmistusprosessi

Muoviputket valmistetaan ekstruusiolla. Ekstruusiolla tarkoitetaan muovien työmenetelmää, jossa muoviraaka-aine ohjataan, useimmiten vaakojen kautta, sylinteriin sijoitetun jatkuvatoimisen ruuvin läpi. Ruuvi työntää raaka-ainetta eteenpäin kohti suulaketta, joka on muotti, jossa tuote saa muotonsa. Sylinterissä muoviraaka-aine sulatetaan vastusten tuottaman lämmön sekä materiaalin liikkeen aiheuttaman kitkalämmön avulla. (Seppälä 2008:261.)

Muovit voidaan ryhmitellä useilla erilaisilla tavoilla ja yksi näistä tavoista on jakaa muovit niiden käyttötarkoituksen mukaan valtamuoveihin, teknisiin muoveihin ja erikoismuoveihin. Valtamuovit ovat edullisuutensa vuoksi käytetyimpiä muoveja. Toinen yleinen jaottelu muoveille on jakaa ne kestumuoveihin ja

kertamuoveihin. Kestomuoveja, toisin kuin kertamuoveja, voidaan toistuvasti muovata uudelleen lämmön ja paineen avulla. Muoviputket ja kalvot valmistetaan usein valta- ja kestumuoveista. (Kurri ym. 2002: 17.)

2.1 Putkituotantolinja

Putkituotantolinja koostuu edellä mainitun ekstruuderin lisäksi jäähdytysaltaasta, vetolaitteesta, leimauslaitteesta, sahasta, liukuhihnoista, muhvauskoneesta ja mahdollisista pakkauslaitteista. Ekstruuderin jatkuvatoiminen ruuvi lämmitteää ja kuljettaa sulavan muoviraaka-aineen kohti muottia, ja edempänä linjassa oleva vetolaitte vetää sulan muoviraaka-aineen ulos työkalun suulakkeelta. Suulakkeelta eli muotista se ohjautuu kalibrointitulkin läpi jäähdytysaltaaseen. Jäähdytysaltaassa on vesisuihkut sekä säädetyn suuruinen alipaine.

Kalibrointitulkki, joka on kiinnitetty jäähdytysaltaaseen, on yleensä messinkinen putki, jonka halkaisija on jälkikutistuman verran haluttua putken halkaisijaa suurempi. Putken jäähdyttäminen aloitetaan kalibrointitulkin läpi. Kalibrointitulkki on uritettu, jolloin jäähdytysaltaan vesisuihkut jäähdyttävät tehokkaammin sekä tulkkia että sen läpi kulkevaa putken muodon saanutta jäähtyvää muoviraakaainetta. Urituksella on toinenkin tehtävä, uritus mahdollistaa putken imemisen kalibrointitulkin seinämiä vasten alipaineen avulla.

Jäähdytysaltaan jälkeen putkeen leimataan putken tiedot. Tietoja ovat esimerkiksi tuotteen nimi, mahdolliset standardit, viivakoodi ja valmistuspäivämäärä.

Lopuksi putki sahataan sahalla haluttuun mittaan. Sahalta putki kuljetetaan liukuhihnoilla muhvaukseen ja edelleen pakkaukseen.

2.2 Muhvauskone

Muhvauskoneen tehtävä on muotoilla putkenpää siten, että kaksi putkea pystytään yhdistämään toisiinsa. Kuvassa 1 näkyvä muhvi voidaan jakaa kahteen

osaan, joita ovat muhviosa sekä kumiura. Muhviosan tehtävänä on pitää putken muhvin sisälle työnnetty pistopää paikoillaan. Kumiura, johon tiiviste sijoitetaan, varmistaa, ettei liitos vuoda.



Kuva 1. Muhvi.

Muhvauskoneita on sekä mekaanisia että paineilmalla toimivia. Mekaanisissa muhvauskoneissa esilämmitetty putkenpää työnnetään muottiin, jota kutsutaan tuurnaksi. Muhvausleuat suljetaan. Tuurnassa olevat seklementit, jotka muotoilevat kumiuran, avataan hydraulisesti. Lopuksi muhvi jäähdytetään. Yleensä tämä tapahtuu tuurnassa kiertävän veden avulla. Mekaaninen muhvauskone on vähemmän häiriöherkkä johtuen tekniikan yksinkertaisuudesta.

Tässä päättötyössä tutkitaan paineilmalla toimivaa Everbell 500 H/D-muhvainta, joka on italialaisen Sica-yhtiön valmistama. Muhvauskone on valmistunut vuonna 2005.

2.3 Muhvausprosessi paineilmalla

Muhvausprosessi koostuu muhvauskoneessa neljästä paineilmajaksosta, joita ovat puhallus, sisäpuolen jäähdytys, kutistuman poisto sekä ulkopuolen jäähdytys. Kuvassa 2 näkyvässä Everbell 500 -muhvauskoneessa muhvi muotoillaan sekä lämmön että paineilman avulla. Muhvauskoneessa on kosketusnäytöllinen

käyttöliittymä, josta voidaan asetella ajettavat parametrit kyseessä olevalle tuotteelle.



Kuva 2. Muhvauskone.

Muhvaimoneella pystytään muhvaamaan yksi tai kaksi putkea kerrallaan riippuen putken halkaisijasta. 160 mm ja 200 mm halkaisijaltaan olevat putket muhvataan kaksi putkea kerrallaan ja halkaisijan 250 mm, 315 mm ja 400 mm olevat putket muhvataan yksi putki kerrallaan.

Putki tuodaan muhvaimelle muhvaimen omalla vetolaitteella. Putki pysäytetään käyttöliittymään asetetun mitan mukaiseen asemaan. Vetolaitteelta putki siirretään siirtopöydälle, joka liikuttaa joko yhtä tai kahta putkea eri työasemien välillä. Työasemia muhvaimessa on kolme kappaletta: kaksi lämmitysasemaa ja muotoiluasema.

Muhvausprosessi alkaa putken pään lämmityksellä. Putken päätä lämmitetään sekä putken sisäpuolelta että ulkopuolelta. Asetellun lämmitysajan kuluttua putki tai putket siirretään toiseen lämmitysasemaan. Ennen jälkimmäisen lämmityksen aloittamista putken päät työnnetään pääntasaajalla tarkasti haluttuun asemaan. Putken päät pitää olla tarkasti oikeassa kohdassa, jotta myöhemmin muotoiluasemassa muotin asema on oikea suhteessa putken asemaan.

Muotoiluasemassa suoritetaan muotoilusykli matala- ja korkeapaineella. Korkeapaine luodaan kompressorilla. Korkeapaine on aseteltavissa ja on riippuvainen muhvatavan putken seinämän paksuudesta, raaka-aineesta ja halkaisijasta. Matalapaine otetaan tehdasverkosta. Matalapaine ei ole aseteltavissa. Muotoiluasemassa muhvauskelkka ajetaan säädettyyn asemaan ja ulkopuoliset muotit suljetaan. Muhvi puhalletaan muotoonsa halutulla korkeapaineella. Puhallusjakson jälkeen suoritetaan sisäpuolen jäähdytys, joka tehdään matalapaineella. Sisäpuolen jäähdytysajan kesto niin kuin muutkin jaksoajat ovat aseteltavissa käyttöliittymässä. Seuraavaksi suoritetaan kutistuman poisto, joka tehdään puristamalla korkeapaineella muhvia takaisinpäin. Lopuksi muhvia jäähdytetään puhaltamalla sitä ulkopuolelta. Tätä jaksoa kutsutaan ulkopuolen jäähdytysjaksoksi.

3 Komponentit

Seuraavissa luvuissa esitellään projektissa käytettyjä komponentteja sekä syitä, miksi niitä on päätetty käyttää. Siemensillä on monta logiikkatuoteperhettä eri kokoihin projekteihin ja valintaa varten tehtiin luvussa 3.7 näkyvä taulukko, jossa vertailtiin eri logiikoiden tärkeimpiä ominaisuuksia.

3.1 Ohjelmoitavat logiikat

PLC (Programmable Logic Controller) eli ohjelmoitava logiikka on käytännössä pieni tietokone, jolla korvattiin ennen käytetyt reletaulut. Releistä loogisten piirien rakentaminen on aikaa vievää ja laitteiston koko kasvaa nopeasti. Muutoksien tekeminen reletauluun on todella hankalaa, puhumattakaan vian etsinnästä. PLC on tietokone, joten ei ole syytä rajoittaa sen toimintaa vain digitaalisiin tuloihin ja lähtöihin vaan nykyään logiikoilla käsitellään niin analogisia tuloja ja lähtösignaaleja kuin vaikkapa PID-säätimiäkin. (PLC basics Programmable Logic Controller 2019.)

Yleensä PLC:n ohjelmointi hoidetaan tietokoneella valmistajan tekemällä ohjelmistolla. Siemensin logiikoissa ohjelmointi tapahtuu TIA (Totally Integrated Automation) Portal -ohjelmistolla, johon on integroitu tarvittavat osat, kuten

WinCC-käyttöliittymien ohjelmointiin sekä Sinamics drives, jota käytetään taajuusmuuttajien ohjelmointiin.

3.2 IEC 61131-standardi

IEC 61131 on standardi ohjelmitaville logiikoille. Standardi luotiin, jotta saatiin yhtenäistettyjä käytäntöjä. Kolmas ja nyt käytössä oleva versio julkaistiin 2013. Standardi koostuu 10 osasta. Ohjelmoijille tärkein osa standardia on osa 3, joka käsittelee ohjelmistoarkkitehtuuria sekä ohjelmointiin käytettyjä kieliä, joita ovat Ladder Diagram (LD), Instruction List (IL), Structured Text (ST), Sequential Function Chart (SFC) ja Function Block Diagram (FBD). Näistä tosin Instruction list -kielen käyttöä ei enää suositella. (Devasia 2021.)

3.3 Siemensin logiikat

Siemensillä on monta logiikka tuoteperhettä, joista löytyy ratkaisu pienien laitteiden ohjauksista koko tuotantolinjan ohjaukseen. Logo!8-mikrologiikka on tarkoitettu pieniin projekteihin. Se on saatavilla useilla eri käyttöjännitteillä. Logo!8-logiikkaa on mahdollista laajentaa moduuleilla. Logo!8 ohjelmoidaan Soft Comfort-ohjelmistolla. (LOGO! Basic modules.)

S7-1200-tuoteperheen CPU-mallit 1211C-1218C on tarkoitettu pieniin ja keskisuuriin projekteihin. Mallissa 1211 ominaisuuksia on vähemmän, joten se soveltuu vähemmän vaativiin sovelluksiin, kun taas malli 1218 on vaativampiin sovelluksiin. S7-1200-tuoteperheen logiikat saa myös turvaominaisuuksilla. (Types of Siemens PLC S7-300, 400, 1200, & S7-1500 2020.)

S7-300-tuoteperhe on tarkoitettu keskisuuriin ja suuriin sovelluksiin. CPU:sta on saatavilla erilaisilla ominaisuuksilla varustettuja malleja, kuten teknologia- ja turvaominaisuuksilla varustettuja sekä kompakti malleja. (Types of Siemens PLC S7-300, 400, 1200, & S7-1500 2020.)

S7-400-tuoteperhe on tarkoitettu keskisuuriin ja suuriin sovelluksiin, kuten myös S7-300-tuoteperhe, joka on sitä edistyksempinen. CPU:sta on saatavilla

erilaisilla ominaisuuksilla varustettuja malleja, kuten teknologia- ja turvaominaisuuksilla varustettuja sekä korkeamman tavoitettavuuden malleja. Korkeammassa tavoitettavuudessa kommunikaatioväylät ovat redundanttisia eli kahdennettua. Siemens on luvannut jatkaa S7-400-tuoteperheen tukea ainakin vuoteen 2030 asti. (Types of Siemens PLC S7-300, 400, 1200, & S7-1500 2020.)

S7-1500-tuoteperhe on tarkoitettu keskisuuriin ja suuriin sovelluksiin ja on niin ominaisuuksiltaan kuin hinnaltaankin S7-300- ja S7-400-tuoteperheiden kaltainen ja tulee tulevaisuudessa korvaamaan kyseiset vanhat tuoteperheet. Eroja-kin löytyy, S7-1500-logiikat ovat nopeampia ja lisäksi niissä on näyttö, josta voidaan lukea muun muassa hälytyksiä. CPU:sta on saatavilla erilaisilla ominaisuuksilla varustettuja malleja, kuten redundanttisia, standardi, turva ominaisuuksilla varustettuja malleja, compact malleja sekä teknologia ominaisuuksilla varustettuja malleja. (Types of Siemens PLC S7-300, 400, 1200, & S7-1500 2020.)

ET 200SP -logiikka on saatavilla ilman integroitua CPU:ta, jolloin sitä voidaan käyttää hajautetuissa järjestelmissä. ET 200SP -logiikan saa myös integroidulla CPU:lla, jolloin se toimii itsenäisenä logiikkana. (Types of Siemens PLC S7-300, 400, 1200, & S7-1500 2020.)

3.4 Logiikoiden muistialueet

Ohjelmaa PLC:lle ladatessa se tallennetaan load memoryyn. Ohjelmaa ajettaessa PLC käyttää work memoryä eli työmuistia, joka jaetaan kahteen osaan, joita ovat työmuisti koodille ja työmuisti datalle. Koodityömuistiin tallennetaan kaikki toimilohkot ja datatyömuistiin kaikki tietolohkot. Sähkökatkon sattuessa työmuisti tyhjenee. (S7-1500 structure and use of the CPU memory 2013.)

Ohjelmissa on toisinaan tarve saada joitain muuttujia pysymään muistissa, vaikka sähkökatkeaisivat. Tällaisia tietoja voisivat esimerkiksi olla asetusparametrit. Tämän tyyppistä muistia kutsutaan retentive memoryksi. Sähkökatkon jälkeen muuttujat, jotka eivät ole tallennettuina retentive memory -alueelle, palautuvat oletusarvoonsa. Toisin kuin S7-1200-tuoteperheen logiikat, S7-1500-

tuoteperheen logiikat tarvitsevat erillisen muistikortin. (S7-1500 structure and use of the CPU memory 2013.)

3.5 Siemensin käyttämä Profinet-kenttäväylä

Ethernet kehitettiin LAN-verkossa eli paikallisessa verkossa tapahtuvaan laitteiden väliseen viestintään. Vuonna 1983 luotiin IEEE 802.3 Ethernet-standardi. (What is Ethernet? 2019.)

OSI-malli on viestintästandardien standardi. OSI-malli koostuu seitsemästä kerroksesta, joita ovat fyysinen-, siirto-, verkko-, kuljetus-, istunto-, esitys- ja sovelluskerros. (OSI-malli 2024.)

Profinet on teollisuuden avoin Ethernet-standardi. Normaali kotona tapahtuva tiedonsiirto ei ole samalla tavalla aikakriittinen kuin teollisuuden tarvitsema tiedonsiirron nopeus ja tarkkuus. Teollisuudessa viestien pitää saapua nopeasti ja toisaalta juuri oikeaan aikaan. Profinet-standardi kehitettiin vastaamaan tähän IEEE 802.3 -standardin ongelmaan. Profinetin nopeus perustuu siihen, että toisin kuin IEEE 802.3 -standardi, joka käyttää kaikkia OSI-mallin kerroksia, viestien toimittamisessa hypätään OSI-mallin siirtokerroksesta suoraan sovelluskerrokseen. Profinet käyttää kolmea eri viestintäkanavaa taatakseen oikea-aikaisen viestinnän. Näitä ovat TCP/IP tai UDT/IP, Profinet RT ja Profinet IRT. (Profinet introduction 2018.)

TCP/IP soveltuu käytettäväksi silloin kun ei vaadita suurinta mahdollista nopeutta. Esimerkiksi projektin lataaminen PLC:lle tai hälytykset eivät vaadi suurta nopeutta. Profinet RT soveltuu suurimpaan osaan teollisuuden tiedonsiirtotarpeista, kuten liikkeen ohjaukseen sekä prosessi- ja kappaletavara-automaatioon sykliajan ollessa 250 mikrosekunnista 512 millisekuntiin. Profinet IRT sopii vielä Profinet RT:tä nopeampaa viestintää tarvitseville sovelluksille. Profinet IRT -viesteillä on oma kaista, jolla voidaan taata niiden oikea-aikainen saapuminen. Profinet IRT:llä päästään jopa 32,25 mikrosekunnin sykliaikaan. (Profinet introduction 2018.)

3.6 Siemens HMI -paneelit

Basic HMI -paneelit on tarkoitettu yksinkertaiseen ja kustannustehokkaaseen visualisointiin. Basic-paneeleita löytyy kokoina 4”–12”, 64 000 värillä. Paneeleissa on yhdistetty kosketusnäyttö ja näppäimet, jotka voi ohjelmoida haluamallaan tavalla. Paneeleista löytyy versiot sekä Profibussille että Profinetille. (Machine-level visualization with Simatic HMI 2018.)

Comfort HMI -paneelit on tarkoitettu korkeaa suorituskykyä vaativien projektien visualisointiin. Comfort-paneeleita löytyy kokoina 4”–22”, 16 000 000 värillä. Paneeleissa on yhdistetty kosketusnäyttö ja näppäimet, jotka voi ohjelmoida haluamallaan tavalla. Paneeleista löytyy versioita sekä Profibussille että Profinetille sekä malleja näppäimistöllä ja pelkällä kosketusnäytöllä. Comfort-paneelin ohjelmointiin käytetään TIA Portal -ohjelmistoon kuuluvaa WinCC-ohjelmistoa. Comfort-paneelien ohjelmointiin vaaditaan WinCC advanced -lisenssi. (Machine-level visualization with Simatic HMI 2018.)

Päättötyössä käyttöliittymä tehdään tuotantolinjalla jo olevaan 7”:n basic paneeliin. Olemassa olevaan paneelin tehdään lisäisivut, joihin visualisoidaan muhvimen painejaksojen toimintaa, syötetään tarvittavia asetuksia sekä palautetaan data-analyysin tulokset. Mahdolliset hälytykset esitetään paneelin hälytysvalikossa.

3.7 Logiikan valintakriteerit

Logiikan valinnassa hyödynnettiin kuvassa 3 näkyvää logiikoiden vertailutaulukkoa. Taulukkoon on kerätty muutamia tärkeimpiä ominaisuuksia ET-200SP-logiikoista, kaikista S7-1200-sarjan logiikoista ja ominaisuuksiltaan pienimmistä S7-1500-sarjan logiikoista.

	SIEMENS LOGIIKKA TUOTEPERHE								
	S7-1200					S7-1500			ET 200 SP
	1211 C	1212 C	1214 C	1215 C	1217 C	1511-1 PN	1513-1 PN	1515-2 PN	1510SP-1 PN
OMINAISUUS	1211 C	1212 C	1214 C	1215 C	1217 C	1511-1 PN	1513-1 PN	1515-2 PN	1510SP-1 PN
Syötön tyyppi	DC/AC	DC/AC	DC/AC	DC/AC	DC	DC	DC	DC	DC
Input tyyppi	DC	DC	DC	DC	DC				
Output tyyppi	RLY/DC	RLY/DC	RLY/DC	RLY/DC	DC				
Load memory	1 MB	2 MB	4MB	4 MB	4 MB	M.kortti	M.kortti 32 GB	M.kortti	M.kortti 32GB
Work memory	Yht. 50 KB	Yht. 75 KB	Yht. 100 KB	Yht. 125 KB	Yht. 150 KB	300 KB/1 MB	600 KB/2.5 MB	500 KB/3 MB	200 KB/1 MB
CPU sisältää I/O	4/6	8/6	14/10	14/10	14/10	0	0	0	0
CPU sisältää AI	2	2	2	2	2	0	0	0	0
CPU sisältää AO	0	0	0	2	2	0	0	0	0
I/O moduuli [lkm]	0	2	8	8	8	32	32	32	64
Kommunikaatio moduuli	3	3	3	3	3				
OPC-UA Server	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
OPC-UA Client	Ei	Ei	Ei	Ei	Ei	Kyllä	Kyllä	Kyllä	Kyllä
Profinet IRT	Ei	Ei	Ei	Ei	Ei	Kyllä	Kyllä	Kyllä	kyllä
Profinet portteja [lkm]	1	1	1	1	1	2	2	2	1
Näyttö	Ei	Ei	Ei	Ei	Ei	Kyllä	Kyllä	Kyllä	Ei
Ohjelmointi	Basic/Prof.	Basic/Prof.	Basic/Prof.	Basic/Prof.	Basic/Prof.	Professional	Professional	Professional	Professional
Hinta	Edullinen	Edullinen	Edullinen	Edullinen	Edullinen	Kallis	Kallis	Kallis	Keskitaso

Kuva 3. Siemens logiikoiden ominaisuuksien vertailu.

Projektissa logiikan vaatimuksena on, että se pystyy keräämään kuuden painanturin analogiset lukemat ja lähettämään ne PUT-komennolla ekstruuderin logiikalle. Logiikaksi valittiin yksinkertainen ja edullinen Siemensin 1200-sarjan 1212C-logiikka. Logiikkaan lisättiin 8-kanavainen analogiasisääntulomodula

Tuotantolinjan ekstruuderilla entuudestaan oleva logiikka toimii OPC UA -palvelimena, jolloin Python-ohjelmiston OPC UA -moduuli voi hakea tiedot ekstruuderin logiikasta, joka suorittaa anturien skaalausta lukuun ottamatta muut toiminnot, kuten käyttöliittymän operoinnin.

Ekstruuderin logiikan olisi hyvä pystyä toimimaan myös OPC UA -asiakkaana eli sen pitäisi pystyä myös avaamaan OPC UA -yhteys, jolloin logiikka voisi käynnistää Python-ohjelman silloin kun linja on käynnistetty ja ajetaan muhvilista putkea. Ekstruuderin logiikka on 1200-sarjan 1214-logiikkaa, eikä se voi toimia OPC UA -asiakkaana. Ekstruuderin logiikkaa ei ainakaan tässä vaiheessa lähdetä uusimaan, joten yhteys pidetään auki aina kun ohjelma on käynnissä. Kaikista vertailuun otetuista logiikoista S7-kommunikointimahdollisuus ja Profinet-liitännät, Joten tämäkään ei muodosta eroa logiikoiden välille.

3.8 Painemittaus

Paine eli voima (F) jaettuna pinta-alalla on johdannaissuure, jonka yksikkö on Pascal (Pa). 100 000 Pascalia vastaa 1 baaria. (SI-mittayksiköt Suomessa, massa ja paine.)

Paineanturissa on usein kalvo. Kalvon puristus aiheuttaa pietsosähköisiin kiteisiin paineeseen verrannollisen jännitteen, joka pystytään muuttamaan halutuksi signaaliksi. (Keinänen & Sumujärvi 2019: 225.)

Projektiin tarvittiin kuusi paineanturia. Paineantureiksi valittiin kuvassa 4 näkyvä Sick-yhtiön valmistama PBS plus -paineanturi.

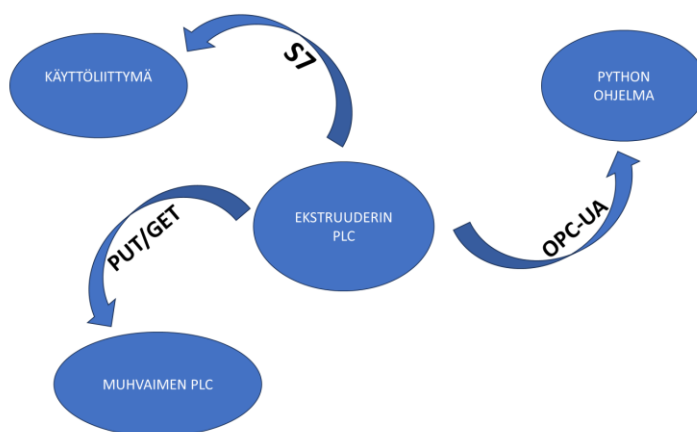


Kuva 4. PBS plus -paineanturi.

PBS plus -paineanturi valittiin, koska aikaisempien projektien takia oli tiedossa paineanturin hyvä luotettavuus ja yleensä hyvä hinta-laatusuhde. Projektiin valitulla PBS plus -paineanturilla voidaan mitata kaasun tai nesteen painetta. Paineanturin mittausalue on 0–40 baaria. Ulostulosignaaliksi voidaan valita joko 4–20 mA tai 0–10 V.

4 Tiedonsiirto

Projektissa käytettiin tiedonsiirtoon useampia eri tapoja, joita ovat OPC UA -protokola, PUT-komento ja S7-protokola. Tiedonsiirtotapoja esitellään kuvassa 5.



Kuva 5. Tiedonsiirtomenetelmät ja rajapinnat.

Muhvaimella olevalta PLC:ltä anturien data lähetettiin PUT-komennolla ekstruuderille asennetulle PLC:lle. Ekstruuderin PLC:n ja tietokoneen Python-ohjelman väliseen kommunikointiin käytettiin OPC UA -protokolaa. Python-ohjelmassa tiedot analysoitiin ja analyysin tulokset sekä hälytykset palautettiin ekstruuderin PLC:lle.

Ekstruuderin PLC hallinnoi käyttöliittymää, johon tiedot siirrettiin S7-protokollalla. S7-kommunikointi on Siemensin kehittämä protokolla. S7-kommunikointia käytetään S7-sarjan logiikoiden ohjelmointiin, logiikoiden väliseen kommunikointiin sekä tiedonsiirtoon logiikoiden ja käyttöliittymän välillä. Seuraavissa luvuissa käydään läpi käytetyt kommunikointitavat. (S7 communication with PUT/GET 2019.)

4.1 OPC UA -protokolla

OPC Unified Architecture lyhyemmin OPC UA on teollisuusautomaation tarpeisiin OPC-järjestön kehittämä laitteiden välinen kommunikointiprotokolla. OPC UA:n juuret ulottuvat 1990-luvulle, jolloin ohjelmoitavat logiikat koettiin kalliiksi halpenevien tietokoneiden takia. Microsoftin käyttöjärjestelmät olivat hallitsevassa asemassa markkinoilla, joten Microsoftin käyttöjärjestelmällä varustetut tietokoneet haluttiin toimistojen lisäksi myös tehtaiden tuotantotiloihin ohjaamaan laitteita. Käytettäessä anturia tai toimilaitetta tarvittiin jokaisen valmistajan

järjestelmään eri ajurit. Ajurien ohjelmoiminen vaati suuren määrän resursseja. Kaikki edellä mainitut syyt johtivat siihen, että vuonna 1995 useat automaatiovalmistajat halusivat luoda standardin tietojen jakamiseen eri laitteiden välillä. Päätettiin käyttää jo olemassa olevia Microsoftin teknologioita COM eli Component Object Model ja DCOM eli Distributed Component Object Model, joita käytettiin tietojen jakamiseen Windows-ohjelmien ja tietokoneiden välillä. Standardin nimeksi tuli OPC, joka alun perin on tullut sanoista OLE for Process Control. (Keiski 2016.)

OPC UA yhdisti kolme aikaisemmin julkaistua OPC-standardia, joita olivat vuonna 1997 julkaistu OPC DA, Vuonna 1999 julkaistu OPC A&E ja Vuonna 2000 julkaistu OPC HDA. OPC DA määritteli sensorin tai toimilaitteen lukeman. OPC A&E määritteli hälytyksen ja tapahtumatyypin. OPC HDA määritteli historian aikaleimatiedon eli ajan, koska hälytys tai anturin arvo on tullut. Se määrittää tiedon eli esimerkiksi sensorin tai toimilaitteen lukeman, ajan sekä laadun. (Keiski 2016.)

OPC UA sisältää paljon hyviä ominaisuuksia, joita ovat turvallisuus, monialustaisuus, integroitu tietomalli ja avoin lähdekoodi. Palvelimen kanssa kommunikoitaessa voidaan käyttää käyttäjävaltuutusta toisin kuin monessa muussa kommunikointitavassa. Tämä tekee OPC UA:sta turvallisemman vaihtoehdon. Monialustaisuus tarkoittaa sitä, että tietoja voidaan siirtää monien eri käyttöjärjestelmien välillä ja niitä ovat esimerkiksi Linux, Windows ja MAC OSX. OPC UA:n integroidulla tietomallilla on standardisoitu rakenne, josta puhutaan solmuina. Datalla on aina oma solmu-id, joka mahdollistaa tiedon vaihtamisen. Avoin lähdekoodi tarkoittaa, että tietyin ehdoin kaikki voivat tutkia lähdekoodia ja muokata sitä. (Keiski 2016.)

OPC UA -tiedonsiirto tapahtuu palvelimen ja asiakkaan välillä. Palvelimena voi toimia monenlaiset laitteet, kuten PLC tai tietokone, joka sisältää OPC UA -palvelinohjelman. OPC UA on siis yksinkertaisesti vain ohjelmakoodi. Palvelin saa tiedot sensoreilta, toimilaitteilta tai vaikka kenttäväylän välityksellä toiselta PLC:ltä. (OPC UA lesson 1- what is OPC-UA? 2020.)

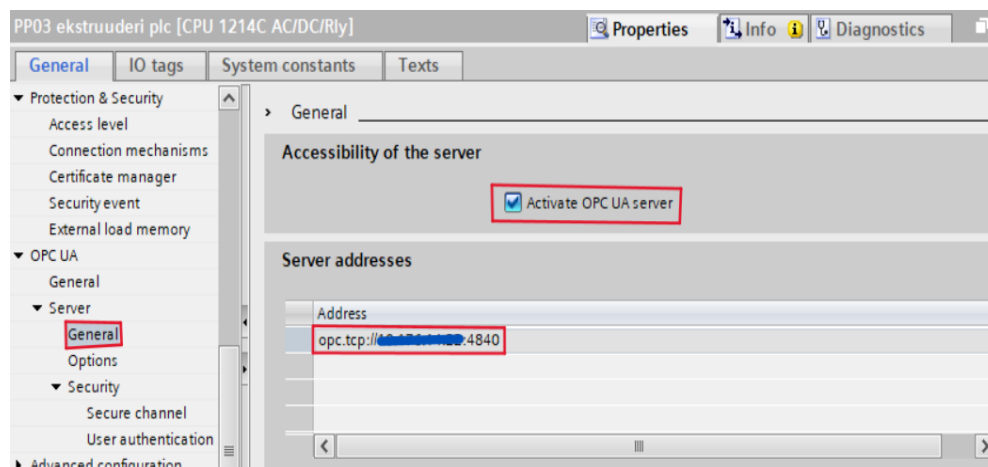
Asiakas voi olla mikä vain laite, jossa on asiakasohjelma. Asiakasohjelma voi olla esimerkiksi OPC UA expert -ohjelma tai OPC UA Python -moduuli. (Keiski 2016.)

4.1.1 OPC UA TIA Portal -ohjelmointiympäristössä

Opinnäytetyössä tiedot tietokoneella olevan Python-ohjelman sekä käyttöliittymää operoivan ekstruuderin PLC:n välillä lähetettiin OPC UA -protokollalla. Opinnäytetyössä asiakkaana toimii tietokone, jossa on Pythonin OPC UA -moduuli. Palvelimena toimii ekstruuderin PLC, joka on Siemensin S7-1200-logiikka, joka saa anturien datan sekä muhvimen tilatiedot muhvimen PLC:ltä PUT-komennon avulla.

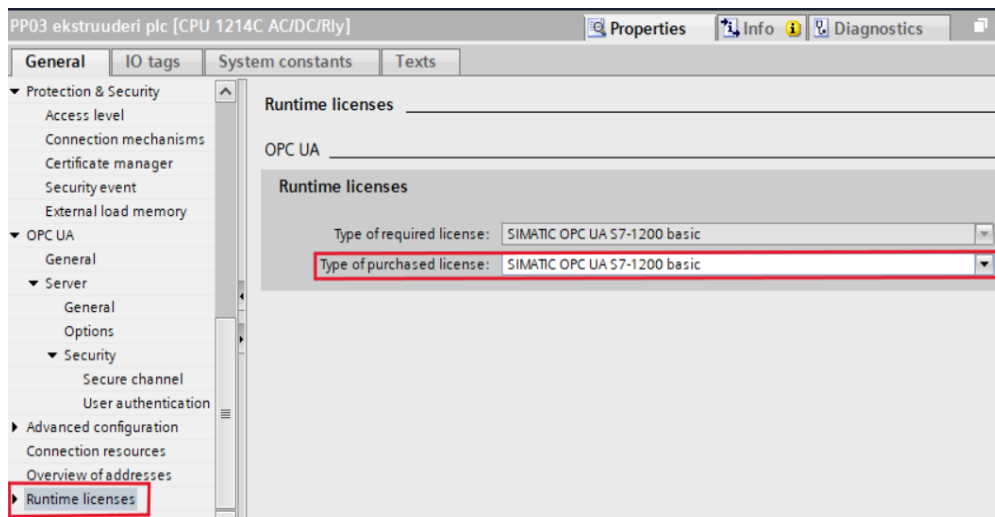
OPC UA -palvelimen käyttöönotossa on kolme huomioon otettava asiaa, jotta tiedonsiirto onnistuu. Alla on esitetty TIA Portal -projektissa tarvittavat toimenpiteet, jotta OPC UA -palvelin voidaan ottaa käyttöön.

Ensimmäiseksi kuvassa 6 näkyvä OPC UA -palvelin pitää hyväksyä käyttöön. Palvelinta käyttöönotettaessa määritetään myös palvelimen osoite sekä portti.



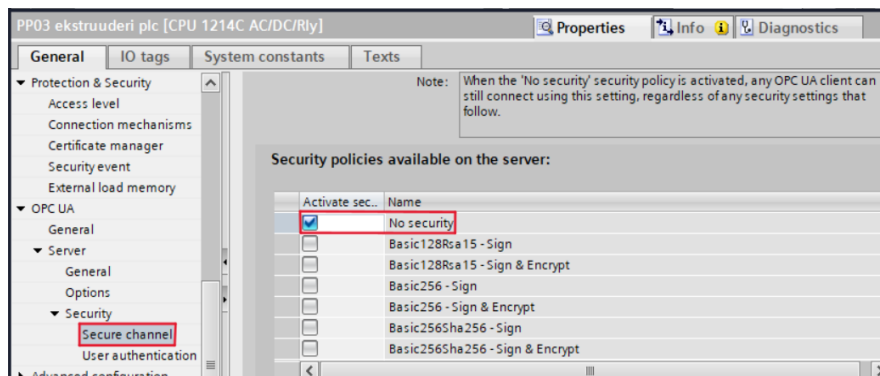
Kuva 6. OPC UA -palvelimen käyttöönotto (TIA Portal 2023).

Toisena huomioon otettavana asiana. Lisenssi tarvitsee olla ostettuna ja se valitaan käyttöön asetuksissa. Kuten kuvassa 7 on tehty.



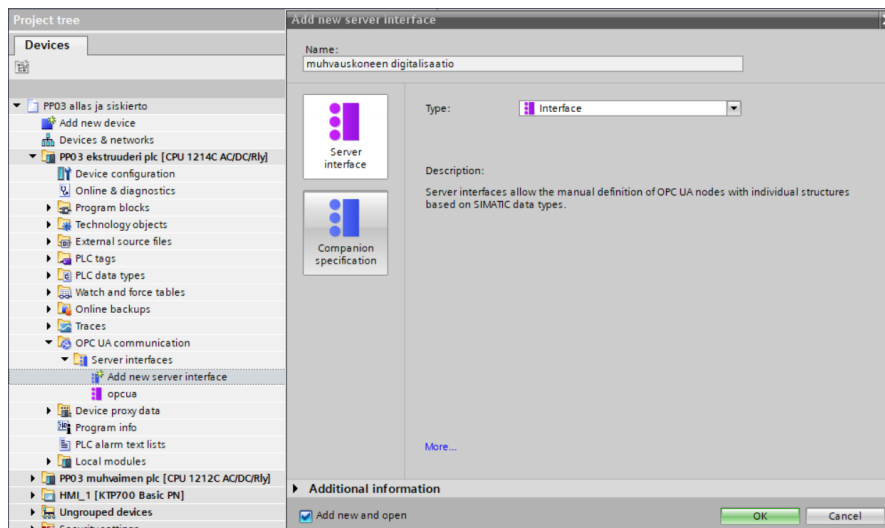
Kuva 7. OPC UA -lisenssin valinta (TIA portal 2023).

Kolmantena asiana kuvassa 8 näkyvällä turvallisuusvälilehdellä otetaan käyttöön halutut turvallisuusasetukset. Halutessaan voi myös valita, että turvallisuusasetukset eivät ole käytössä.



Kuva 8. OPC UA -turvallisuusasetukset (TIA Portal 2023).

Perusasetuksien määrittämisen jälkeen luotiin projektiin kuvassa 9 näkyvä OPC UA -palvelimen käyttöliittymä, joka löytyy OPC UA -kommunikaatiovälilehdeltä. Palvelimen käyttöliittymä nimettiin muhvauskoneen digitalisaatioksi.



Kuva 9. OPC UA -palvelimen käyttöliittymän luonti (TIA Portal 2023).

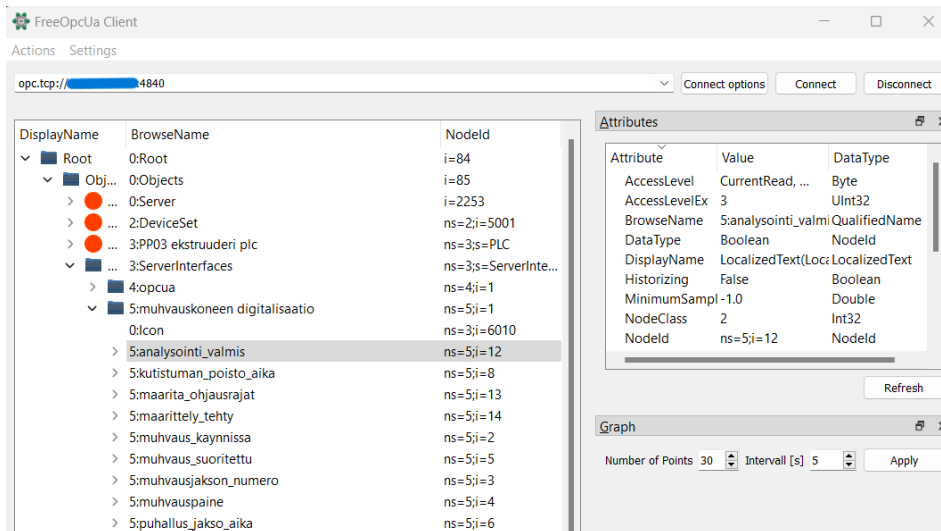
OPC UA -palvelin käyttöliittymää varten tehtiin tietolohko (DB), johon lisättiin projektissa tarvittavat muuttujat. Tehdyt muuttujat siirrettiin aikaisemmin luotuun OPC UA -palvelimen käyttöliittymään. Kuvassa 10 näkyy TIA Portal -projektissa määritelty OPC UA -palvelimen käyttöliittymä ja osa tässä projektissa siirrettävästä datasta.

OPC UA server interface	
Browse name	Node type
1	▼ muhvauskoneen digitalisaatio Interface
2	■ muhvaus_kaynnissa BOOL
3	■ muhvausjakson_numero INT
4	■ muhvauspaine REAL
5	■ muhvaus_suoritettu BOOL
6	■ puhallus_jakso_aika REAL
7	■ sis_puolen_jaahdytys_aika REAL
8	■ kutistuman_poisto_aika REAL
9	■ ulkopuolen_jaahdytys_aika REAL
10	■ puhallus_jakson_lcl REAL
11	■ puhallus_jakson_ucl REAL
12	■ analysointi_valmis BOOL
13	■ maarita_ohjausrajat BOOL
14	■ maarittely_tehty BOOL
15	■ sisajaaahdytys_lcl REAL

Kuva 10. OPC UA -palvelimen käyttöliittymä (TIA Portal 2023).

4.1.2 OPC UA Python-ohjelmassa

Tietojen vaihtoa varten tarvitsee tietää kunkin OPC UA -palvelimelle lisätyn muuttujan osoite eli Node ID. Osoitteet voidaan selvittää kuvassa 11 näkyvää OPC UA -graafista käyttöliittymää apuna käyttäen.



Kuva 11. OPC UA -graafinen käyttöliittymä (Free OPC UA-client.2024).

Osoitteiden selvittämiseksi tietokoneelle pitää ladata graafinen OPC UA Client -käyttöliittymä. Lataamisen jälkeen se voidaan käynnistää esimerkiksi Windowsin PowerShell -komentotulkissa.

PLC:hen voidaan yhdistää, kun osoite ja portti on lisätty. Yhdistämisen jälkeen Server Interface -alasvetovalikosta löytyy TIA Portal -projektiin lisätty palvelin, josta löytyy muuttujien osoitteet.

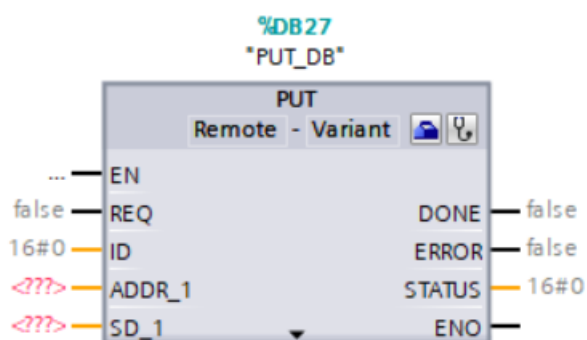
Aikaisemmin tehtiin muhvauksen digitalisaatio -palvelimen käyttöliittymä TIA Portal -ohjelmistoon. Palvelin näkyy kuvassa 11 olevassa OPC UA:n graafisessa käyttöliittymässä. Palvelimen alta voidaan katsoa esimerkiksi analysointi-valmis muuttujaa ja nähdä, että sen osoite on ns = 5; i = 12.

Node ID koostuu kolmesta osasta, joita ovat nimiavaruuden indeksi, tunniste-tyyppi ja tunniste. Tarkastellessa osoitetta ns = 5; i = 12 ns tarkoittaa

nimiavaruuden indeksiä. Seuraavana ilmoitetaan tunnistetyyppi, joka voi olla joko i eli numeerinen, s eli merkkijono, g eli GUID tai b eli opaque. Lopuksi tulee tunniste, joka tässä on numeerinen arvo 12. (OPC UA NodeID concepts.)

4.2 TIA Portalin PUT- ja GET-komento

Kuvassa 12 näkyvällä TIA Portalin PUT-komennolla pystyy lähettämään sisään-tulon, ulostulon, tietolohkon tai muistialueen. PUT-komento toimii ainoastaan S7-sarjan logiikoiden väliseen kommunikointiin. PUT-komennolla tieto lähetetään eli kirjoitetaan toiselle logiikalle. PUT-komennon sijaan voidaan käyttää GET-komentoa, jolla tieto haetaan eli luetaan toisesta logiikasta.



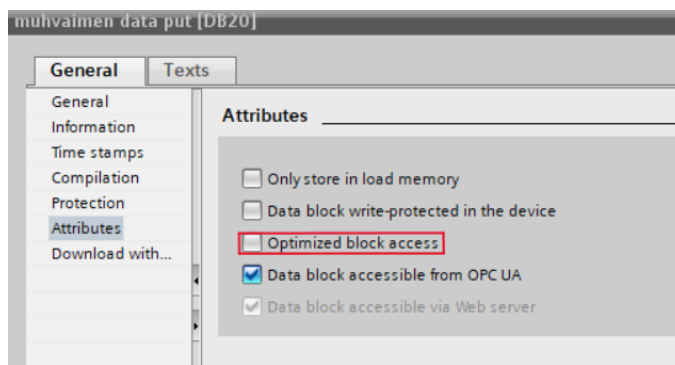
Kuva 12. PUT-komento (TIA Portal 2023).

REQ-sisääntuloon voidaan asettaa muuttuja, joka lähettää tiedon haluttujen ehtojen täytyttyä. Kellopulssia voidaan käyttää, jos halutaan syklinen lähetys, vaikka kerran sekunnissa.

PUT- ja GET-komennon käytössä on kolme huomioon otettavaa asiaa, jotta tietoa voidaan onnistuneesti siirtää:

1. IP-osoite. Kummassakin logiikassa täytyy olla samassa verkossa oleva IP-osoite käytössä. IP-osoitteen kolmas numero kertoo verkon, eli sen täytyy molemmissa logiikoissa olla sama.

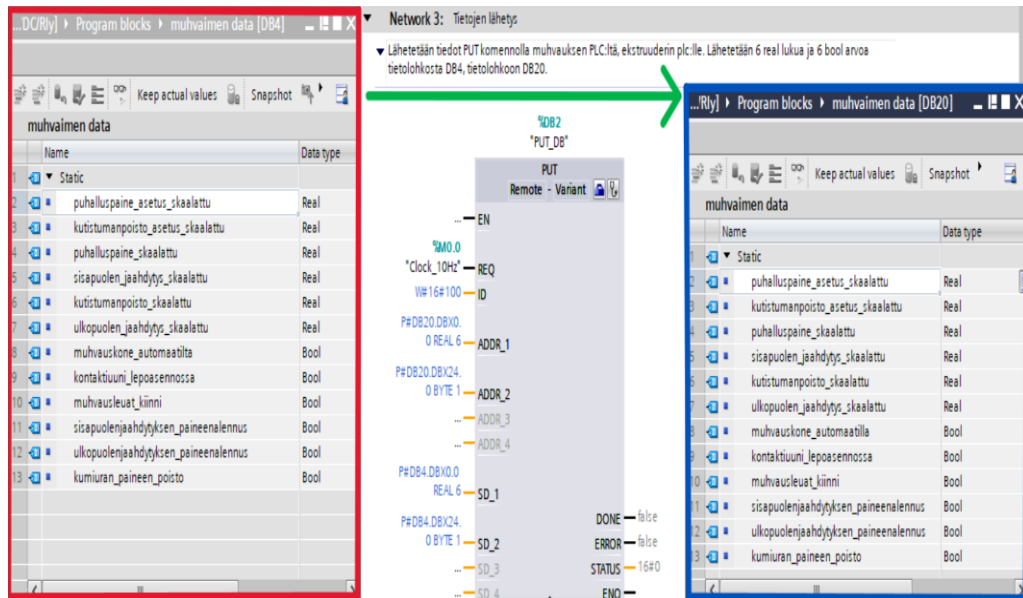
2. Turvallisuus. Turvallisuuden takia logiikassa, josta luetaan tai johon kirjoitetaan tietoa, tarvitsee olla PUT- ja GET-komennon käyttö hyväksyttynä. Lähettävässä logiikassa ei tarvita hyväksyntää.
3. Muistialue. Ohjelman selkeyden takia hyvä tapa on luoda tietolohko (DB) sekä vastaanottavaan että lähettävään logiikkaan, joiden välillä tiedot siirretään. Mikäli lähetetään tietolohkoon (DB), tarvitsee tietolohkon optimointi poistaa käytöstä, kuten kuvassa 13. (Tia portal: PUT command / PLC-PLC communication 2019.)



Kuva 13. Tietolohkon optimointi (TIA Portal 2023).

PUT-komennon käyttö

PUT-komennolla siirrettiin tietoa muhvauskoneelle lisätyn hajautetun logiikan sekä jo olemassa olevan ekstruuderin logiikan välillä. Kuvassa 14 näkyy muhvauskoneen logiikan tietolohko DB4 ja ekstruuderin tietolohko DB20 sekä niissä oleva siirrettävä data. Data sisältää 6 paineanturin skaalattua arvoa, jotka ovat real eli desimaaliukuja sekä muhvaimen laitteiden tilatiedot, jotka saavat arvoja välillä tosi tai epätosi eli ne ovat boolean-arvoja. Lisäksi kuvassa näkyy PUT-lohko, jolla tieto siirrettiin.



Kuva 14. PUT-komento ja tietolohkot (TIA Portal 2023).

PUT-komentoon tarvitsee sijoittaa lähetettävän ja vastaanottavan tietolohkon muistialue SD ja ADDR. REQ-kohtaan sijoitetaan lähetyspyyntö, joka kuvassa 14 on 10 Hz:n kellopulssi, eli tieto lähetetään 100 millisekunnin välein.

Tarkastellaan muistialueen osoitteen merkitsemistä, joka kuvassa 14 on esimerkiksi P#DB20.DBX0.0 REAL 6:

1. P# tarkoittaa osoittimen tunnistetta.
2. DB20 tarkoittaa logiikan tietolohkoa.
3. .DBX0.0 tarkoittaa aloituskohtaa tietolohkossa.
4. REAL 6 tarkoittaa kuutta desimaalilukua. (Bee 2022.)

5 Koeajo

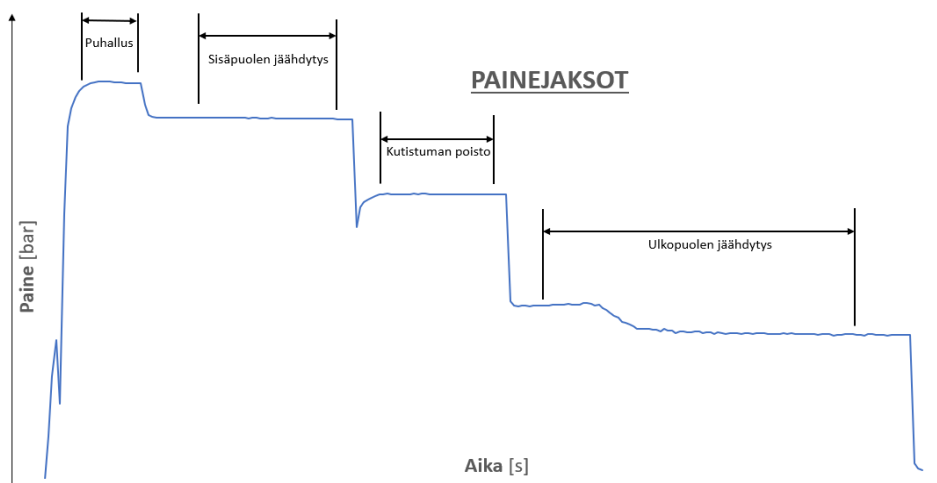
Tulevia muhvaussykliä data-analyysyä varten tarvittiin referenssiarvot muhvausjaksoista eli käytännössä tieto eri muhvausjaksojen paineista tietyillä ajanhetkillä. Muhvauksessa käytetään kahta eri paineasetusta, joita ovat

kutistumanpoistopaine sekä muhvauspaine. Paineasetukset ovat operaattorin portaattomasti säädettävissä.

Päätettiin ajaa koeajo, jolla pyrittiin selvittämään, vaikuttaako asetettu muhvaus- tai kutistumanpoistopaine keskihajontaan eli kasvaako tai laskeeko keskihajonta, mikäli asetuspaineita muutetaan. Tämä tarkoittaisi sitä, että joka kerta asetuspaineita muutettaessa myös ohjausrajat pitäisi laskea uudestaan, koska keskihajonnan muutos muuttaa myös alempaa ohjausrajaa (LCL) ja ylempää ohjausrajaa (UCL). Koeajosta saaduista tuloksista ja menetelmistä kerrotaan tarkemmin luvussa 6.3 lineaarinen yhteys ja korrelaatiokerroin.

6 Data-analyysi

Koeajoista saaduista muhvausjaksojen referenssiarvoista eroteltiin paineiden lasku- ja nousuajat vakioapaineen ajoista. Kuvan 15 painetrendiin on merkitty painejaksojennimet.



Kuva 15. Muhvausjaksot ja painetrendit.

Ohjausrajojen laskemisessa käytetään vain niitä näytteitä, joissa paine on pysynyt vakiona. Myöhemmin muhvaussykliä dataa analysoidessa lasku- ja nousu-aikoja ei oteta huomioon. Ohjausrajojen laskeminen käydään läpi askel askeleelta seuraavissa luvuissa.

Mikäli paine pysyy ohjausrajojen sisällä, prosessin oletetaan olevan kunnossa. Mikäli ohjausrajat ylitetään tai alitetaan, voidaan olettaa, että prosessi ei ole kunnossa, ja tästä varoitetaan operaattoria käyttöliittymän paneelilta löytyvässä hälytysvalikossa.

6.1 Aritmeettinen keskiarvo ja keskihajonta

Keskiarvo, jota kutsutaan myös odotusarvoksi, ja keskihajonta ovat keskeisiä termejä tilastotieteessä. Kuvassa 16 olevassa esimerkissä näytetään kuinka, keskiarvo ja keskihajonta saadaan selvitettyä näytteestä, jossa on mitattu yhden muhvausjakson puhalluspaine. Todellisia muhvauksessa käytettyjä paineita ei tässä insinööriyössä käytetä, vaan käytetyt painearvot ovat kuvitteellisia. Tämä ei kuitenkaan vaikuta ohjausrajojen laskentatapaan.

Muhvaimen puhalluspaineet mitattiin puhallusjakson aikana 10 kertaa, sekunnin välein. Lasketaan puhalluspainejakson paineiden keskiarvo ja keskihajonta.

Sekunti	Mitatut paineet [X]	perusjoukon keskiarvo \bar{X}	$(X-\bar{X})$	$(X-\bar{X})^2$	$\sum(X-\bar{X})^2$	$\frac{\sum(X-\bar{X})^2}{n-1}$	$s=\sqrt{\frac{\sum(X-\bar{X})^2}{n-1}}$
1	8,20	8,49	-0,29	0,08	0,33	0,0365	0,191
2	8,38	8,49	-0,11	0,01			
3	8,45	8,49	-0,04	0,00			
4	8,54	8,49	0,05	0,00			
5	8,65	8,49	0,16	0,03			
6	8,65	8,49	0,16	0,03			
7	8,70	8,49	0,21	0,04			
8	8,60	8,49	0,11	0,01			
9	8,56	8,49	0,07	0,01			
10	8,15	8,49	-0,34	0,12			
		Keskihajonta[σ] on 0,191 bar					

Kuva 16. Keskiarvon ja keskihajonnan laskeminen.

Keskiarvo kertoo lukujen keskimääräisen arvon. Keskiarvo saadaan jakamalla näytteen lukujen summa näytteen lukujen määrällä. Keskiarvoa tarvitaan keskihajonnan määrittämisessä.

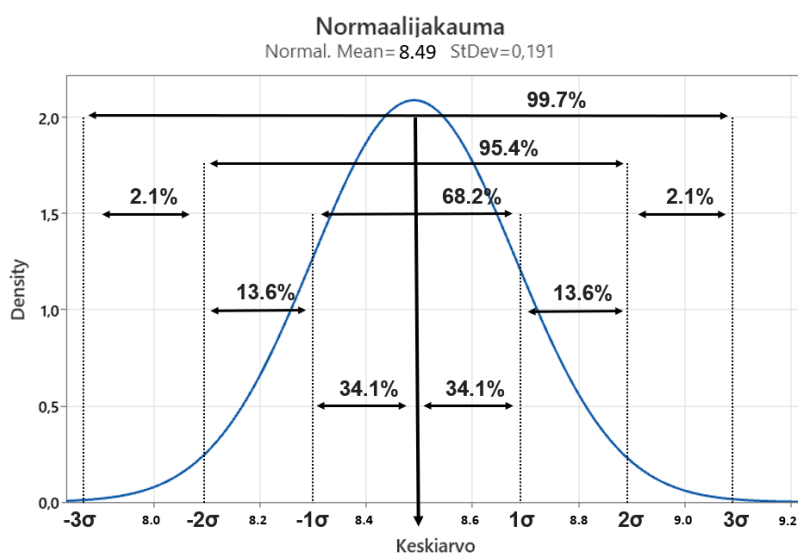
Keskihajonta kertoo, kuinka paljon näytteen lukemat keskimäärin poikkeavat keskiarvosta ylös- tai alaspäin. Keskihajonta lasketaan eri tavalla riippuen siitä,

onko kyseessä perusjoukko vai osajoukko. Perusjoukko kuvaa kaikkia näytteitä kyseisestä mittauksesta. Perusjoukko olisi kyseessä esimerkiksi silloin, jos mittaisimme kaikkien maailman ihmisten pituuden, ja osajoukko olisi kyseessä silloin, jos mittaisimme koululuokan oppilaiden pituuden. Osajoukon keskihajonnassa jakajana olevasta näytteiden määrästä vähennetään 1, muuten laskukaava on samanlainen. Keskihajontaa merkitään sigmaalla (σ), osajoukon keskihajontaa merkitään s-kirjaimella (s).

Perusjoukon keskihajonta lasketaan siten, että jokaisesta perusjoukon luvusta vähennetään perusjoukon keskiarvo ja saadun luvun summa korotetaan toiseen potenssiin. Potenssiin korottaminen poistaa lukujoukosta negatiiviset luvut. Saadut luvut lasketaan yhteen ja jaetaan näytteen lukujen määrällä. Lopuksi tuloksesta otetaan neliöjuuri. Saatu luku on yksi keskihajonta eli yksi sigma. (Keskihajonnat.)

6.2 Normaalijakauma

Havaintojen ollessa normaalisti jakaantuneita, niin kuin suurin osa luonnossa esiintyvistä prosesseista on, niistä muodostuu kuvassa 17 näkyvän normaalijakauman tiheysfunktion näköinen käyrä, jota muotonsa vuoksi kutsutaan usein myös kellokäyräksi. Normaalijakauma on symmetrinen keskikohtaansa nähden.



Kuva 17. Normaalijakauma (Minitab 2024).

Normaalijakauman keskikohta on aina havaintojen keskiarvossa. Keskihajonnan normaalijakauma jakaantuu keskiarvon ala- ja yläpuolelle. Keskihajonnan symboli on sigma, yksi sigma on yksi keskihajonta. Kuusi sigmaa eli kuusi keskihajontaa kattavat yhteensä 99,7 % havainnoista. Kaksi sigmaa keskiarvon ala- ja yläpuolelle ulottuva alue kattaa 95,4 % havainnoista ja yksi sigma keskiarvon ala- ja yläpuolelle ulottuva alue kattaa 68,2 % havainnoista.

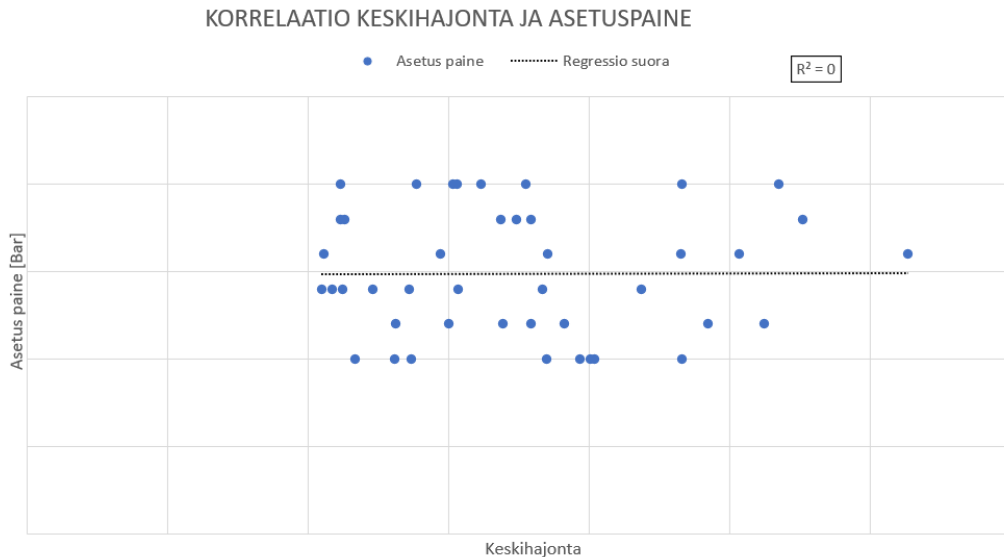
Normaalijakauman avulla voidaankin periaatteessa ennustaa tulevaa. Aiemmin todettiin, että muhvaoksen puhallusjakson havaintojen keskiarvo oli 8,49 baaria keskihajonnan eli yhden sigman ollessa 0,191 baaria. Näin ollen voidaan olettaa, että puhallusjakson aikana 997 havaintoa 1000:sta tulee osumaan välille 7,92–9,063 baaria.

6.3 Lineaarinen yhteys ja korrelaatiokerroin

Koeajodatasta haluttiin selvittää, onko muhvauspaineen asetuksen ja keskihajonnan välillä lineaarista yhteyttä. Lineaarista yhteyttä tutkitaan hajakuvaajan avulla. Lineaarinen yhteys ei ota kantaa, mikä aiheuttaa mahdollisen yhteyden ja se vain näyttää, että jokin yhteys on olemassa.

Hajakuvaajassa X-akselille sijoitetaan itsenäinen muuttuja. Itsenäiseksi muuttujaksi kutsutaan sitä muuttujaa, joka ei todennäköisesti vaikuta toiseen muuttujaan. Tässä tapauksessa voidaan olettaa, että jos muuttujien välillä on yhteyttä, niin asetuspainne vaikuttaa keskihajonnan suuruuteen, eikä niin, että keskihajonnan suuruus vaikuttaa asetuspaineeseen. (Mattila.)

Koeajossa ajettiin 6–8 muhvausjaksoa kuudella eri muhvauspaineella sekä kuitistuman poistopaineella. Kuvassa 18 on piirretty hajakuvaaja, jossa näytteistä lasketut keskihajonnan arvot on piirretty X-akselille ja muhvauspaine asetuksen arvot on piirretty Y-akselille.



Kuva 18. Hajakuvaaja, keskihajonnan ja asetuspaineen välinen suhde (Minitab 2024).

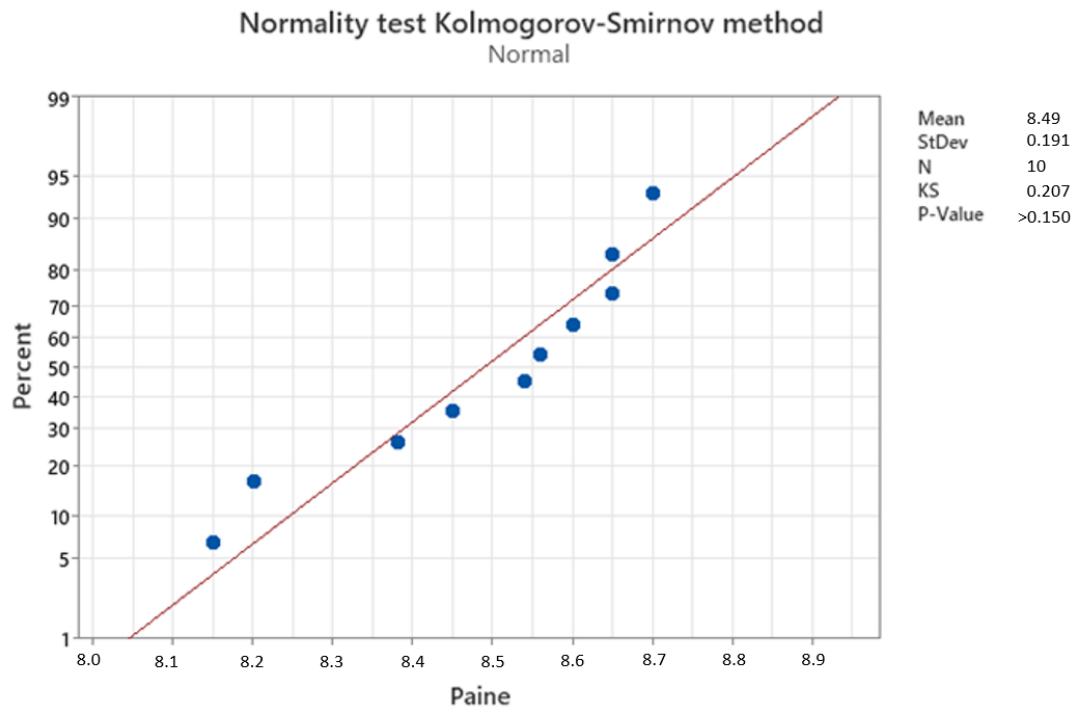
Mikäli kuvaajan arvot muodostavat nousevan suoran, sanotaan sitä positiiviseksi lineaariseksi yhteydeksi. Mikäli kuvaajan arvot muodostavat laskevan suoran, sanotaan sitä negatiiviseksi lineaariseksi yhteydeksi. (Mattila.)

Osajoukon korrelaatiokertoimen tunnus on R . R saa arvoja -1 :stä $+1$:een. Korrelaatiokertoimen ollessa 1 muuttujilla on täydellinen positiivinen yhteys, ja arvon ollessa -1 muuttujilla on täydellinen negatiivinen yhteys. Korrelaatiokertoimen saadessa arvon 0 yhteyttä ei ole. Korrelaatiokertoimen ollessa tasan -1 tai 1 kaikki havainnot osuvat regressiosuoralle. Kuvassa 18 olevasta kuvaajasta nähdään, että korrelaatiokerroin on nolla eikä havainnot jakaudu suoran ympärille. Tämä tarkoittaa, että muhvauspaineen asetuksella ja keskihajonnalla ei ole yhteyttä. (Mattila.)

6.4 Normaaliustesti

Data-analyysissä käytettävä tilastollinen prosessinohjaus edellyttää havaintojen olevan normaalisti jakautuneita. Puhalluspaineen datalle tehtiin kuvassa 19 näkyvä Kolmogorov–Smirnovin normaaliustesti Minitab-ohjelmalla. Kolmogorov–

Smirnovin testissä selvitetään, noudattavatko näytteet, tässä tapauksessa puhalluspainejakson mitatut paineet, normaalijakaumaa.



Kuva 19. Normaalisuustesti (Minitab 2024).

Minitab-ohjelmaan syötetään puhalluspainejakson paineet, minkä jälkeen ohjelma laskee P-arvon, jonka pitäisi olla suurempi kuin 0,05. Kuvasta 19 voidaan havaita, että P-arvo on suurempi kuin 0,150. Tästä voidaan päätellä, että painehavainnot ovat normaalisti jakautuneet. Kuvasta 19 voidaan myös visuaalisesti päätellä, että havainnot seuraavat melko tarkasti punaista nousevaa suoraa. Tämä tarkoittaa, että havainnot ovat normaalisti jakautuneet.

6.5 Ohjausrajojen määrittely

Tilastollisen prosessinohjauksen kehitti tohtori Walter A. Shewhart vuonna 1924. Tilastollisessa prosessin ohjauksessa erotellaan satunnaissyistä johtuva prosessin suureen vaihtelu erityissyistä johtuvasta vaihtelusta. (Kankaanranta 2015.)

Kuvassa 20 on taulukko. Taulukkoon on merkitty aikaisemmin luvussa 6.1 lasketut keskiarvo ja keskihajonta.

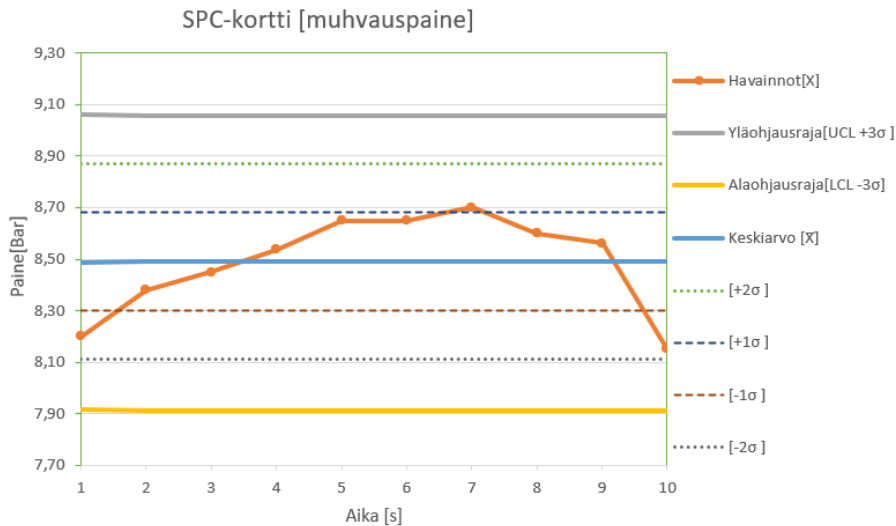
Aika [S]	Havainnot[X]	Keskiarvo [\bar{X}]	keskihajonta[σ]	Yläohjausraja[UCL +3 σ]	Alaohjausraja[LCL -3 σ]
1	8,20	8,49	0,1911	9,06	7,91
2	8,38	8,49	0,1911	9,06	7,91
3	8,45	8,49	0,1911	9,06	7,91
4	8,54	8,49	0,1911	9,06	7,91
5	8,65	8,49	0,1911	9,06	7,91
6	8,65	8,49	0,1911	9,06	7,91
7	8,70	8,49	0,1911	9,06	7,91
8	8,60	8,49	0,1911	9,06	7,91
9	8,56	8,49	0,1911	9,06	7,91
10	8,15	8,49	0,1911	9,06	7,91

Kuva 20. Ohjausrajat.

Tilastollisessa prosessinohjauksessa lasketaan myös ohjausrajat prosessille. Havaintojen osuessa näiden alemman ja ylemmän ohjausrajan välille voidaan olettaa prosessin olevan kunnossa. Kuvan 20 taulukkoon on laskettu nämä ohjausrajat. Ylempi ohjausraja saadaan selville lisäämällä havaintojen keskiarvoon kolme keskihajontaa eli sigmaa ja alempi ohjausraja vastaavasti vähentämällä keskiarvosta kolme sigmaa.

6.6 SPC-kortti

Tilastollisen prosessinohjauksen visualisointiin käytetään kuvassa 21 näkyvää SPC-korttia. SPC-kortille lisätään ainakin alemman ja ylemmän ohjausrajan apuviiva. Lisäksi kortilla voi olla keskiarvo sekä yhden ja kahden sigman eli keskihajonnan apuviivat. Kun kortille lisätään havainnot, pystytään helposti näkemään prosessin tila.



Kuva 21. SPC-kortti (Minitab 2024).

Tarkastellaan havaintoja vielä tarkemmin. Aikaisemmin todettiin, että tilastollisesti voidaan olettaa 997 havaintoa 1000:sta osuvan 6 sigman eli ± 3 sigman päähän keskiarvosta, kun kysymyksessä ovat normaalisti jakautuneet havainnot.

Kuvassa 22 on taulukko. Taulukossa vertaillaan puhalluspainejakson kymmenen havainnon tilastollisesti odotettuja etäisyyksiä keskiarvosta todellisiin etäisyyksiin verrattuna.

sigma väli	Havainnot tilastollisesti [odotettu]	Havainnot näytteessä [toteutunut]
0-1 σ	6,82	7
1-2 σ	2,72	3
2-3 σ	0,42	0

Kuva 22. Havaintojen jakautuminen.

Voidaan havaita, että havaintojen määrän ollessa 10 kappaletta tilastollisesti odotetaan 6,82 havaintoa osuvan ± 1 sigman eli 1 keskihajonnan päähän keskiarvon ala- tai yläpuolelle. Välille 1–2 sigmaa eli 1–2 keskihajontaa tulisi normaalisti jakaantuneesta datasta osua 2,42 havaintoa ja 2–3 sigman päähän 0,42 havaintoa. Näiden alueiden ulkopuolelle odotetaan osuvan 0,03 havaintoa.

Kuvasta voidaan siis nähdä, että havainnot osuivat juuri tilastollisten odotusten mukaisesti.

7 Ohjelmointi

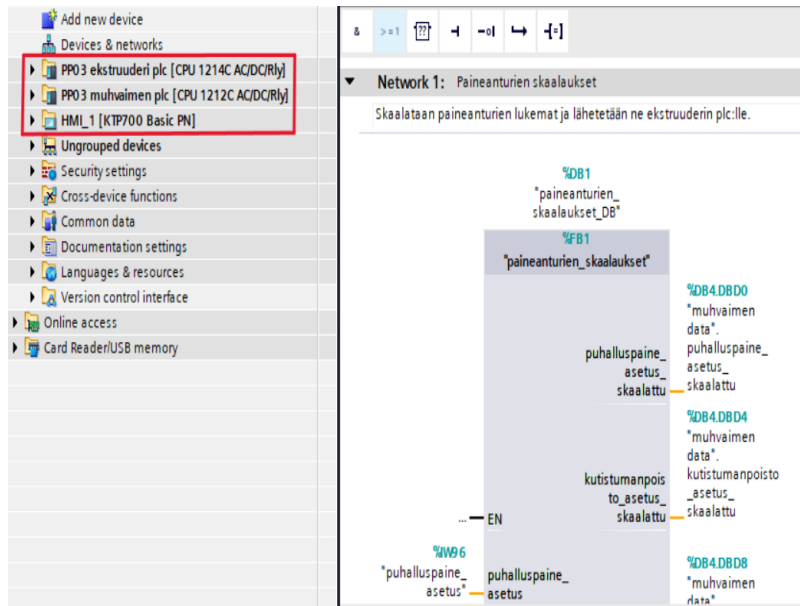
Insinööriyössä käytettiin kahta ohjelmointikieltä ja ohjelmointiympäristöä. Logiikoiden ohjelmointiin käytettiin TIA Portal -ohjelmointiympäristöä. Muhvauskooneella olevan logiikan tehtävä on kerätä paineanturien data ja lähettää se ekstruuderilla olevalle logiikalle. Ekstruuderilla oleva logiikka toimii serverinä tietokoneelle sekä lähettää tarvittavat tiedot käyttöliittymän paneelille.

Tietokoneelle tehtiin Python-ohjelma, joka on tehty Visual studio -ohjelmointiympäristössä. Python-ohjelmassa lasketaan ohjausrajat, paineanturien data analysoidaan sekä mahdolliset hälytykset lähetetään takaisin logiikalle.

7.1 Siemens TIA Portal -ohjelmointiympäristö

TIA Portal on Siemensin kehittämä ohjelmointiympäristö, joka esiteltiin marraskuussa 2010. TIA Portalissa on yhdistetty ennen erilliset ohjelmat, kuten logiikoiden ohjelmointiin tarkoitettu STEP 7, käyttöliittymien ohjelmointiin tarkoitettu WinCC, turvaratkaisujen ohjelmointi sekä käyttöjen ohjelmointi yhteen ohjelmointiympäristöön. TIA Portalin uusin versio on tällä hetkellä versio 18. (TIA Portal.)

TIA Portalissa luodaan aluksi uusi projekti, johon lisätään käytössä olevat laitteet, kuten logiikka, käyttöliittymän paneeli sekä taajuusmuuttaja. Näitä voi lisätä yhden tai useampia sen mukaan, mitä laitteita projektiin kuuluu. TIA Portal -ohjelman etusivulla kuvassa 23 näkyvät projektille lisättyinä olevat logiikat ja käyttöliittymän paneeli.

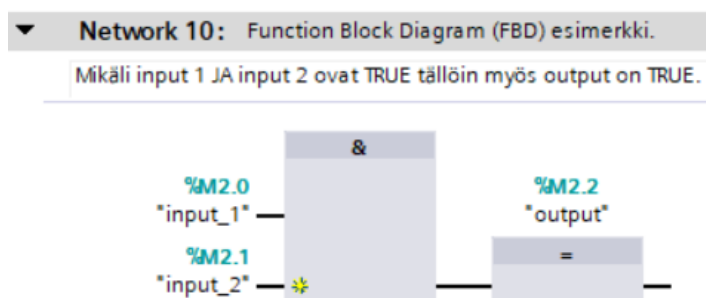


Kuva 23. TIA Portal -ohjelmaan lisätyt laitteet (TIA portal 2023).

Tässä insinööryössä käytettiin hyödyksi tuotantolinjalla valmiina olevaa logiikkaa ja siihen aiemmin luotua projektia. Aiemmin luodussa projektissa oli käytetty Siemensin 1200-sarjan 1214C-logiikkaa sekä käyttöliittymässä Siemensin 7":n paneelia. Projektiiin lisättiin muhvaimelle sijoitettu Siemensin 1200-sarjan 1212C-logiikka.

7.1.1 Ohjelmointikieli Function Block Diagram

Logiikkaan voidaan tehdä ohjelma halutulla IEC 61131-3 -standardin mukaisella ohjelmointikielellä. Tässä insinööryössä käytettiin FBD:tä eli Function Block Diagram -ohjelmointikieltä, josta on esimerkki kuvassa 24.



Kuva 24. Function Block Diagram esimerkki (TIA portal 2023).

Function Block Diagram on graafinen ohjelmointikieli. Kuvassa 24 on näytetty, miltä FBD-ohjelmointikieli näyttää. Kuvassa on yksinkertainen AND-portti. Molempien sisääntulojen ollessa tosia, myös ulostulo on tosi.

7.1.2 Funktiot

Lyhyt ohjelma voidaan kirjoittaa suoraan toimilohkoon (OB) eli pääohjelmaan, mutta ohjelman luettavuuden takia ja varsinkin ohjelman ollessa vähänkin pidempi on syytä käyttää funktioita. Niitä kutsutaan TIA Portalissa nimellä Function (FC) ja toimilohkoja, jotka ovat TIA Portalissa nimellä function block (FB). (PLC-ohjelmointi. Toimilohkon soveltaminen PLC-ohjelmassa.)

Molempiin sekä toimilohkoihin että funktioihin voidaan ohjelmoida jokin kokonaisuus kuten, moottorinohjaus tai vaikkapa liukuhihnanohjaus. Samaa toimilohkoa tai funktiota voidaan kutsua niin monta kertaa pääohjelmassa kuin tarvitaan, jos moottoreita tai liukuhihnoja on useampia kappaleita. (PLC-ohjelmointi. Toimilohkon soveltaminen PLC-ohjelmassa.)

Erona näillä kahdella ohjelmalohkolla on, että Function block (FB) sisältää pysyvän muistin toisin kuin pelkkä Function (FC). Muistia tarvitaan, mikäli tietojen halutaan säilyvän seuraavalle ohjelmakierrolle. Kuvitellaan tilanne, että halutaan lisätä johonkin lukuun yksi jokaisella ohjelmakierrolla. Mikäli ohjelma tehdään funktiolla (FC) on luvun arvo jokaisella kierroksella nolla, vaikka siihen on lisätty yksi. Tämä johtuu siitä, että lukua ei tallenneta mihinkään vaan jokaisella kierrolla, kun funktiota kutsutaan, lähdetään liikkeelle oletusarvosta. (PLC-ohjelmointi. Toimilohkon soveltaminen PLC-ohjelmassa.)

7.2 Visual Studio -ohjelmointiympäristö

Visual Studio on Microsoftin vuonna 1997 julkaisema tietokoneohjelmien tekemiseen tarkoitettu integroitu kehitysympäristö. Visual Studion uusin versio on tällä hetkellä Visual Studio 2022. (Visual Studio 2024.)

Visual Studio -ympäristössä ohjelmakoodi voidaan kirjoittaa ja se voidaan ottaa käyttöön yhdessä paikassa. Ohjelmointiin voidaan käyttää Pythonin ohella useita muitakin ohjelmointikieliä, joista mainittakoon C++, C# tai Visual Basic. (What is Visual Studio? 2023.)

Python on ohjelmointikieli, jonka Guido van Rossum kehitti 1980-luvun lopussa. Pythonilla voidaan toteuttaa ohjelma käyttämällä olio-ohjelmointia proseduraalisesti tai toiminnallisella ohjelmointi tavalla. (Python frequently asked question. What is Python? 2024.)

Python on yksi suosituimmista ohjelmointikielistä, minkä takia siihen on kehitetty suuri määrä erilaisia kirjastoja, joita voi muun muassa käyttää tieteelliseen laskentaan tai kuvaajien luontiin. Myös käyttöliittymien tekemiseen löytyy oma kirjastonsa. Nykyinen uusin käytössä oleva Python-versio on 3.0, joka ilmestyi vuonna 2008. (Python ohjelmointikieli 2024.)

7.3 Ohjelmistojen suunnittelu

Insinööriyön tarkoitus oli digitalisoida muhvauskone eli toisin sanoen asentaa siihen paineanturit, joiden tuottamaa dataa voitaisiin analysoida tilastollisen prosessinohjauksen metodeilla sitä varten tehdyllä tietokoneohjelmalla, jolloin voitaisiin varmistua muhvauskoneen painejaksojen oikeanlaisesta toiminnasta ja toisaalta hyödyntää jatkossa mahdollisesti opittavaa tietoa muissa samankaltaisissa projekteissa. Operaattorin näkökulmasta käyttöliittymän ja sen myötä projektin tärkeimmät kriteerit olivat helppokäyttöisyys ja automaattisuus eli ohjelman piti toimia niin, että operaattorin tarvitsisi tehdä mahdollisimman vähän asetuksia tai säätöjä, jotka kuluttavat työaikaa. Lisäksi hälytyksien tuli olla mahdollisimman yksiselitteisiä.

Yhtiön näkökulmasta projektin tärkeimmät kriteerit olivat luonnollisesti kustannusten pitäminen maltillisina sekä oikea-aikaisen ja oikeanlaisen datan saaminen operaattorien käyttöön, jotta mahdolliset ongelmat pystytään havaitsemaan aikaisessa vaiheessa ja poistamaan nopeasti. Tämä auttaa ongelmien poistamiseen etupainotteisesti, jolloin romuputkien määrä vähenee.

Tilaaajan kehitysinsinöörin kanssa käytyjen keskustelujen pohjalta laadittiin luvussa 7.3.2 (kuva 25) kuvattu Python-ohjelman vuokaavio, jonka pohjalta Python-ohjelmaa alettiin kehittämään. Logiikkaohjelmille laadittiin lista vaadittavista toiminnallisuuksista, jotka on esitetty luvussa 7.3.1 TIA Portal -ohjelman toiminnallisuudet.

7.3.1 TIA Portal -ohjelman toiminnallisuudet

Tilaaajan kehitysinsinöörin kanssa käytyjen keskustelujen sekä oman pohdinnan pohjalta laadittiin lista TIA Portal -projektin PLC-ohjelmien tärkeimmistä toiminnallisuuksista.

Muhvaimen PLC:n toiminnallisuudet:

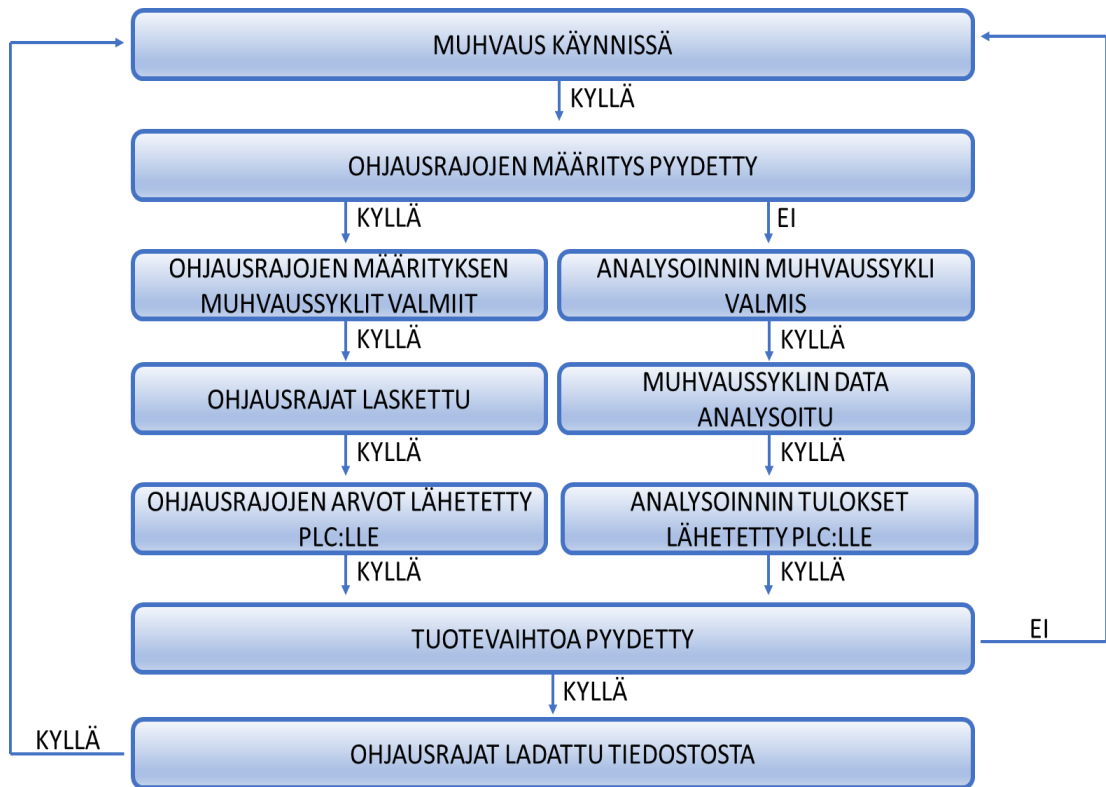
1. Muhvaimen PLC lähettää analogiatulojen lukemat sekä digitaalitulojen arvot ekstruuderin PLC:lle.

Ekstruuderin PLC:n toiminnallisuudet:

1. Vastaa vastaanottaa muhvaimen PLC:n lähettämät lukemat.
2. Toimii serverinä tietokoneen kanssa käytävässä tiedonsiirrossa.
3. Hoitaa tiedonsiirron käyttöliittymän kanssa.
4. Määrittelee muhvaimen tilan muhvaimen PLC:ltä saatujen tietojen perusteella. Muhvaimen tiloja ovat muhvaussykli on alkamassa ja muhvaussykli on loppunut.
5. Määrittelee muhvausjakson muhvaimen PLC:ltä saatujen tietojen perusteella. Muhvausjaksoja ovat puhallusjakso, sisäpuolen jäähdytysjakso, kutistuman poistojakso ja ulkopuolen jäähdytysjakso.
6. Käsittelee hälytykset, SPC-kortin tiedot ja käyttöliittymässä esitettävät animaatiot.

7.3.2 Python-ohjelman vuokaavio

Kuvassa 25 on Python-ohjelman etenemisestä laadittu vuokaavio. Vuokaaviota käytetään usein suunnittelun apuna. Vuokaaviolla havainnollistetaan prosessin etenemistä.



Kuva 25. Python-ohjelman vuokaavio.

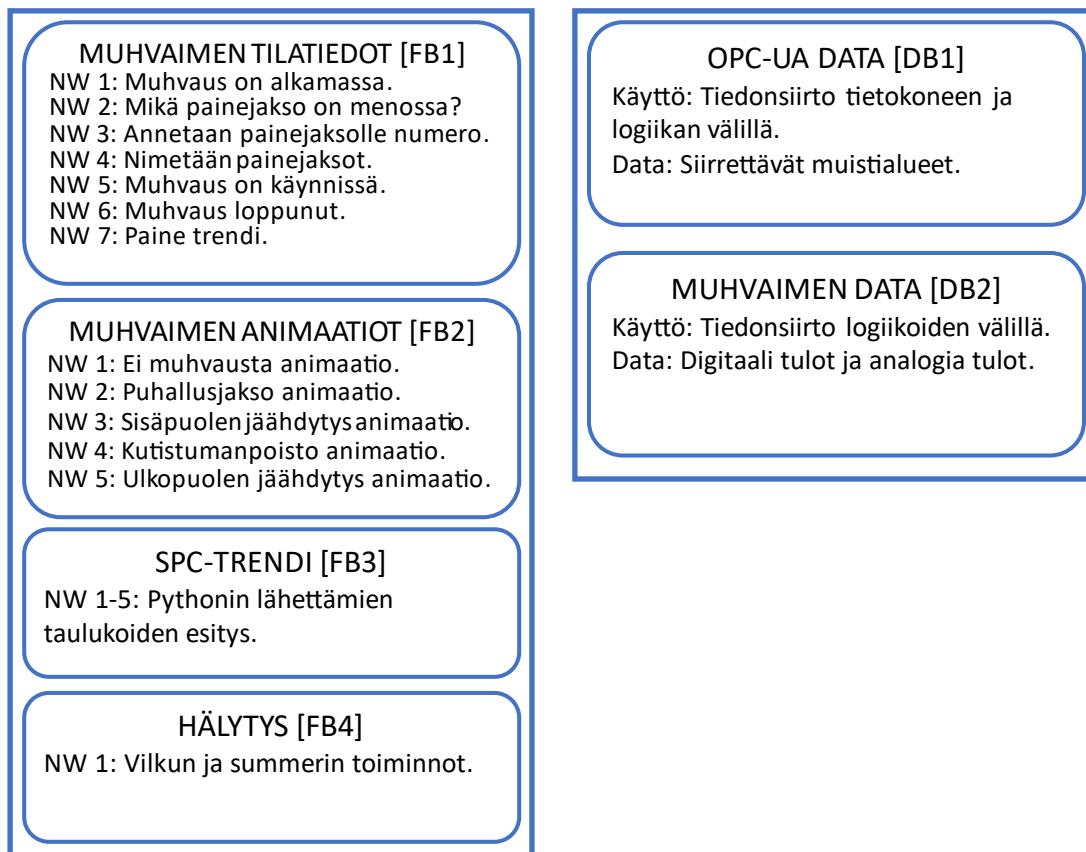
Vuokaaviossa edetään nuolen suuntaan riippuen laatikoissa olevien asioiden toteutumisesta. Vuokaaviossa voidaan myös haarautua eri poluille. Vuokaavio on hyvä apuväline ohjelman kulkua kuvatessa.

7.3.3 TIA Portal -ohjelman rakenne

TIA portal -ohjelmisto jaettiin kahteen logiikkaan, jotka molemmat, kuten myös käyttöliittymän paneeli, ovat samassa projektissa. Logiikat nimettiin ekstruderin logiikaksi ja muhvauskoneen logiikaksi.

Kuvassa 26 näytetään ekstruuderin logiikan ohjelman rakenne. Pääohjelma [OB] koostuu neljästä aliohjelmasta eli function blockista (FB) sekä kahdesta muistilohkosta (DB). Aliohjelmat on jaettu Networkeihin ohjelman luettavuuden parantamiseksi.

PLC1 [Ekstruuderin PLC] [OB1]



Kuva 26. Ekstruuderin logiikan ohjelman rakenne.

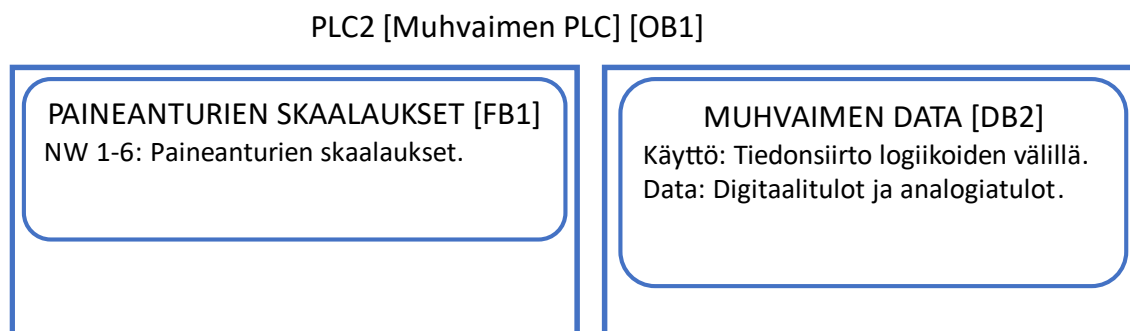
Muhvaimen tilatiedot toimilohkossa on aliohjelma, jossa päätellään muhvaimen logiikalta tulevien digitaalitulojen tilatietojen perusteella, onko muhvaus alkamassa tai käynnissä ja mikä muhvausjakso on menossa. Annetaan muhvausjaksoille numerot 1–4 sekä nimetään muhvausjaksot, jotta ne voidaan esittää käyttöliittymässä operaattorille. Muhvaussyklin jälkeen ilmoitetaan Pythonille, jotta analysointi voidaan suorittaa. Lisäksi painejaksot yhdistetään yhdeksi painetrendiksi, jotta painetrendi voidaan visualisoida käyttöliittymässä.

Muhvaimen animaatiot toimilohkossa on aliohjelma, jossa käynnistetään käyttöliittymän muhvaimen pääsivulla olevat painejaksojen animaatiot. Animaatioissa on hyödynnetty muhvaimesta tehtyjä 3D-malleja sekä TIA Portal -ohjelmiston mahdollistamia grafiikkalistoja.

SPC-trenditoimilohkossa on aliohjelma. Aliohjelmassa muutetaan Pythonin lähettämät painejaksojen painetaulukot trendikäyriksi. Trendikäyrät esitetään käyttöliittymän SPC-korttisivulla analysoinnin jälkeen.

Hälytysaliohjelmassa käsitellään Pythonin lähettämä hälytyskoodi. Hälytysvilkku ja summeri käynnistetään tarvittaessa.

Kuvassa 27 näytetään muhvaimen logiikan ohjelman rakenne. Pääohjelmassa (OB) on vain yksi aliohjelma, paineanturien skaalaukset, sekä PUT-komentoa varten luotu muhvaimen data, muistilohko (DB). Paineanturienskaalaukset-aliohjelma on jaettu Networkeihin ohjelman luettavuuden parantamiseksi.



Kuva 27. Muhvaimen logiikan ohjelman rakenne.

Paineanturienskaalaukset-aliohjelmassa paineanturien lukemat skaalataan ennen niiden siirtoa Muhvaimendata-muistilohkoon ja edelleen ekstruuderin logiikalle.

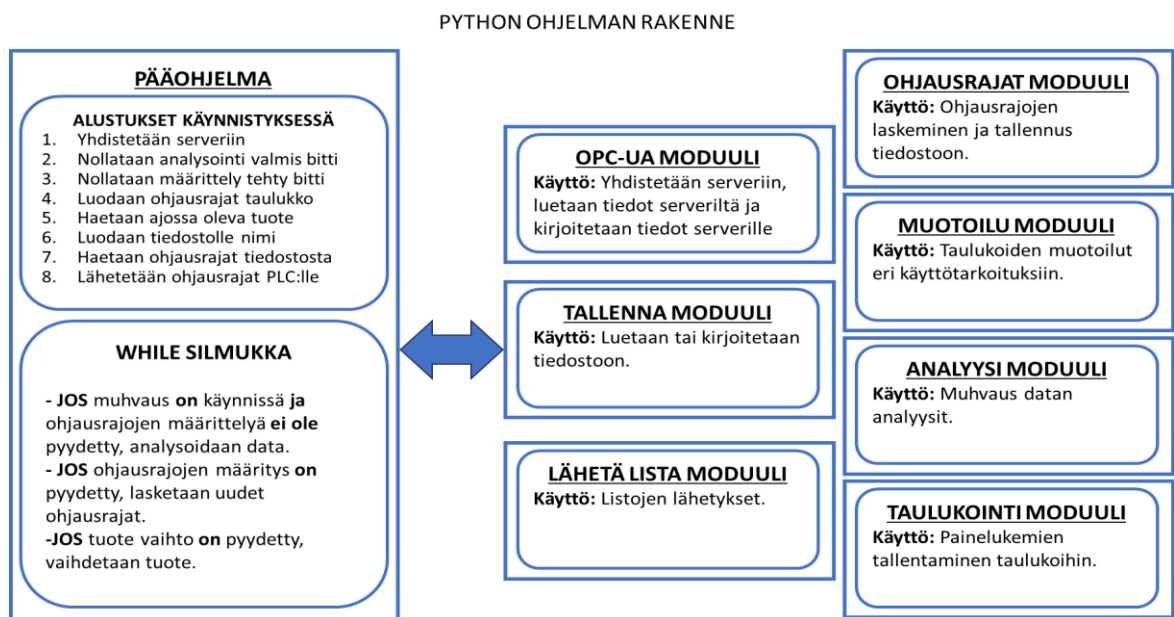
Muhvaimen logiikan analogiatulossa paineanturin signaalille tehdään A/D-muunnos, jolloin paineanturin antama 4–20 mA analoginen signaali muunnetaan digitaaliluvuksi. Skaalauksessa tämä digitaalinen luku, joka saa arvoja väliltä 0–27648, muunnetaan todelliseksi painelukemaksi. Paineanturin mitta-alue

on 0–25 bar, jolloin digitaaliluku 0 vastaa painetta 0 bar ja näin ollen digitaaliluku 27648 vastaa 25 bar painetta.

Arvot skaalataan reaailuvuiksi, jotta saadaan mahdollisimman tarkka painearvo. Reaaliluku voi olla positiivinen tai negatiivinen, ja se sisältää desimaalit toisin kuin esimerkiksi integer-datatyypin, joka on positiivinen tai negatiivinen kokonaisluku.

7.3.4 Python-ohjelman rakenne

Python-ohjelmassa on kuvassa 28 näkyvä pääohjelma ja erilaisia moduuleja. Moduuleita ovat tallenna, OPC UA, lähetä lista, ohjausrajat, muotoilu, analyysi ja taulukointi.



Kuva 28. Python-ohjelman rakenne.

Moduulissa sijaitsevat moduulin aliohjelmat ja muuttujat. Pääohjelmasta kutsutaan haluttua moduulia ja sen aliohjelmaa, mutta myös moduulien aliohjelmat kutsuvat toisiaan tarvittaessa.

7.3.5 Pääohjelma

Python-ohjelman käynnistyksessä ajetaan ensin alustus, jossa resetoitetaan analysoinnin valmistumisesta ja määrittelyn valmistumisesta kertovat muuttujat. Analysoinnin valmistumisesta kertovaa muuttujaa käytetään, jotta ekstruuderin logiikka tietää edellisen muhvin analysoinnin valmistuneen ja voidaan alkaa keräämään dataa mahdollisesta uudesta muhvaussyklistä. Ohjausrajan määrittelystä kertovaa muuttujaa käytetään, jotta ekstruuderin logiikka tietää kaikkien ohjausrajojen määrittelyssä käytettävien muhvien valmistuneen, ja mikäli uutta ohjausrajamäärittelyä ei ole pyydetty, seuraavan muhvaussyklin data analysoidaan.

Seuraavaksi alustuksessa luodaan Ohjausrajat-taulukko. Taulukkoon voidaan tallentaa ohjausrajat, on tiedettävä ajettavan tuotteen nimi, ja tämä operaattorin antama tieto haetaan ekstruuderin logiikalla olevalta OPC UA -palvelimelta. Operaattorin syöttämän tuotteen dimension perusteella luodaan tiedostolle nimi, jonka jälkeen käydään etsimässä luodun tiedoston nimen mukaista tiedostoa ja tallennetaan siellä olevat ohjausrajat Ohjausrajat-taulukkoon ja lähetetään taulukko ekstruuderin logiikalle. Mikäli tiedostoa ei löydy luodaan uusi tiedosto.

Alustuksen valmistuttua siirrytään pääohjelmaan ja tarkemmin sen while-silmukkaan. While-silmukan ehtona käytetään komentoa True, joka käytännössä aloittaa while-silmukan uudestaan alusta viimeisen koodirivin jälkeen, koska ehto on aina Tosi. While-silmukasta voitaisiin tarvittaessa poistua käyttämällä komentoa break, jos se olisi ohjelman kannalta tarpeellista.

While-silmukan alussa käydään palvelimelta lukemassa, onko muhvaus käynnissä, onko operaattori pyytänyt ohjausrajojen määrittelyä ja onko operaattori pyytänyt tuotevaihtoa. While-silmukassa on myös kolme IF- eli ehtolauseke, joiden perusteella joko ladataan uudet ohjausrajat, määritetään uudet ohjausrajat tai kerätään muhvaussyklin data ja analysoidaan se. Mikäli mikään ehto ei täyty, while-silmukka aloitetaan alusta.

7.3.6 Python-ohjelman moduulit

OPC UA -moduulissa yhdistetään ekstruuderin logiikassa olevalle OPC UA -palvelimelle sekä luetaan ja kirjoitetaan eri datatyyppejä sisältäviä muuttujia. Listan lähetykseen käytetään omaa moduulia, jossa lista puretaan yksittäisiksi muuttujiksi FOR-silmukan avulla ja lähetetään datatyyppin mukaan sopivalla OPC UA -moduulin funktiolla.

Ohjausrajat-moduuli laskee uudet ohjausrajat. Operaattori syöttää käyttöliittymään tiedon, kuinka monesta muhvista hän haluaa määrittäytyksen tehtävän, sekä käynnistää ohjausrajojen määrittäytyksen. Ohjausrajojen määrittäytyksen aikana muhvaussyklejä ei analysoida.

Muotoilu-moduuli muotoilee taulukot. Taulukoiden muotoilua tarvitaan poistamaan nousuajat ohjausrajoja määrittäessä. Mikäli niitä ei poistettaisi, ohjausrajat vääristyisivät.

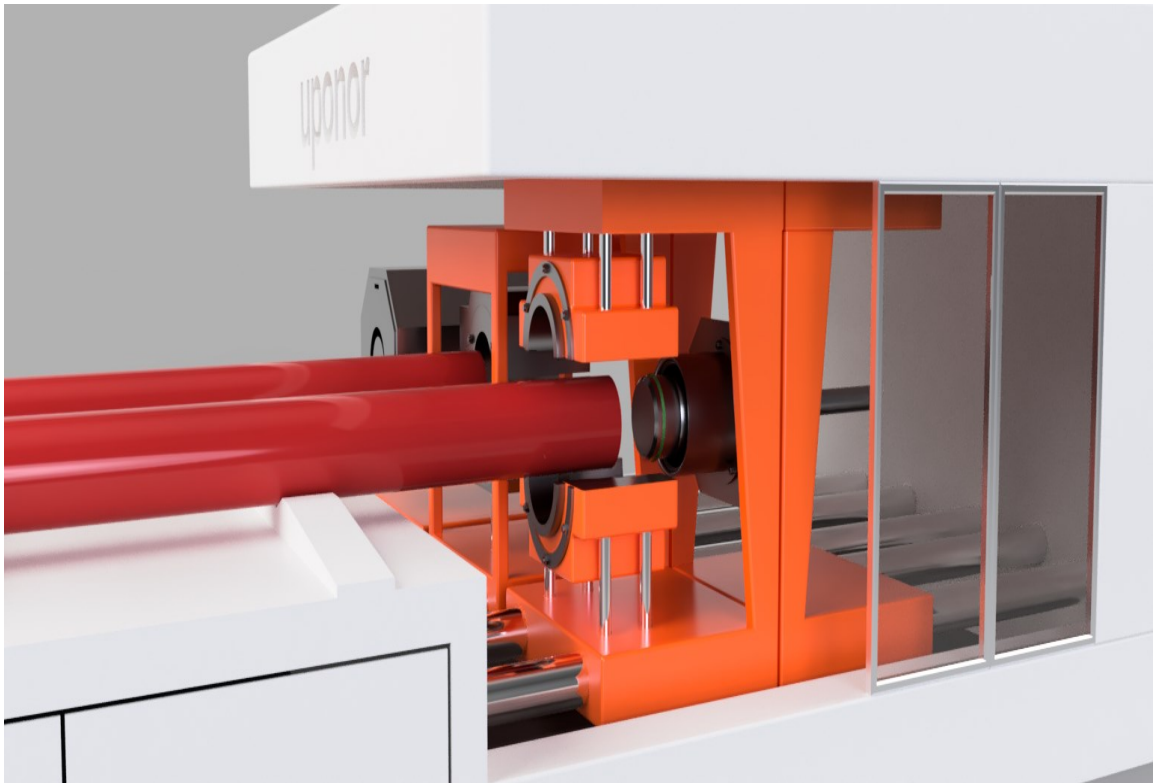
Analyysi-moduulissa selvitetään taulukkojen kokoja tutkimalla, kuinka pitkiä painejaksot ovat. Painejaksojen kestot esitetään myöhemmin käyttöliittymässä. Analyysi-moduulissa selvitetään, myös onko ohjausrajoja ylitetty tai alitettu.

Taulukointi-moduulissa otetaan yhteys palvelimeen. Palvelimelta haetaan sen hetkinen painejakson painelukema. Painelukema tallennetaan taulukkoon ja jäädään odottamaan, että tulee pyyntö hakea uusi painelukema.

Tallenna-moduulissa luetaan tiedostosta tai kirjoitetaan tiedostoon. Operaattorin pyytäessä tuotevaihtoa tai ohjelmaa käynnistettäessä Python-ohjelmassa tarkistetaan, löytyykö tuotteelle jo aikaisemmin tallennettua tiedostoa. Mikäli tiedostoa ei löydy, tiedosto luodaan ja tiedostoon tallennetaan Ohjausraja-taulukko. Tiedoston nimen luomiseen käytetään operaattorin valitsemaa tuotetta ja tarkemmin sen dimensiota. Taulukkoon tallennetaan oletusarvot, jotka ovat 0.0. Luodussa tiedostossa olevat arvot, joko oletusarvot tai määritetyt ohjausrajat, ladataan ekstruuderin logiikalle.

7.4 Käyttöliittymän kuvat

Käyttöliittymän animointiin ja visualisointiin käytetyt kuvat piirrettiin Autodesk Fusion -ohjelmistolla. Muhvauskoneesta ja muhvaustyökalusta piirrettiin karkeat 3D-mallit, joista esimerkki näkyy kuvassa 29. Kuvia käytettiin muhvausjaksojen animointiin.

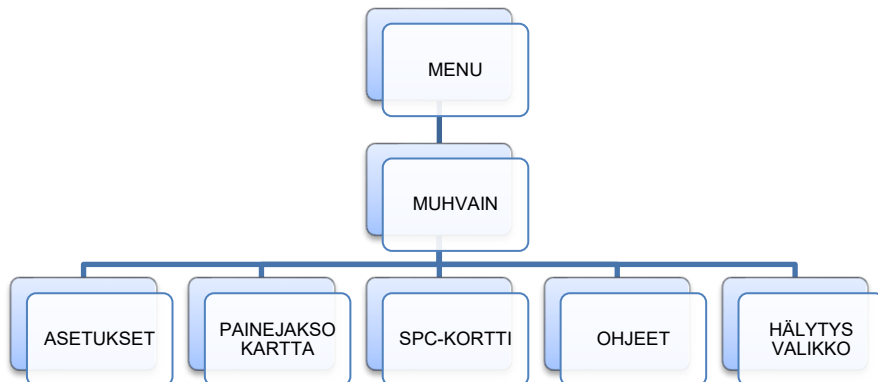


Kuva 29. Muhvauskoneen 3D-malli.

Käyttöliittymä tehtiin Siemensin paneelille. Paneeli ohjelmoitiin TIA Portal -ohjelmistoon integroidulla WinCC-ohjelmalla.

7.5 Käyttöliittymän rakenne

Olemassa olevaan käyttöliittymään tehtiin kuusi lisäsivua. Kuvassa 30 esitetyn käyttöliittymän sivuhierarkian mukaisesti.



Kuva 30. Käyttöliittymän sivuhierarkia.

Uusia sivuja ovat muhvaussivu, joka on pääsivu, johon muilta sivuilta palataan ja johon menuvalikosta saavutaan. Lisäksi tehtiin muhvaussivulle viisi alasivua, joita ovat hälytysvalikko, asetukset, painejakso kartta, SPC-kortti ja muhvain ohjeet. Käyttöliittymän sivuhierarkia näkyy taulukossa 1. Menuvalikosta siirrytään muhvainsivulle, joka on muhvaimen pääsivu. Muhvainsivulta siirrytään muhvaimen alasivuille tai takaisin menuvalikkoon

7.5.1 Menuvalikko

Käyttöliittymään oli siis jo aikaisemmassa projektissa tehty kuvassa 31 näkyvä menuvalikko. Menuvalikkoon oli tehty kuvakkeet, joista pääsee tuotantolinjan muiden laitteiden sivuille.

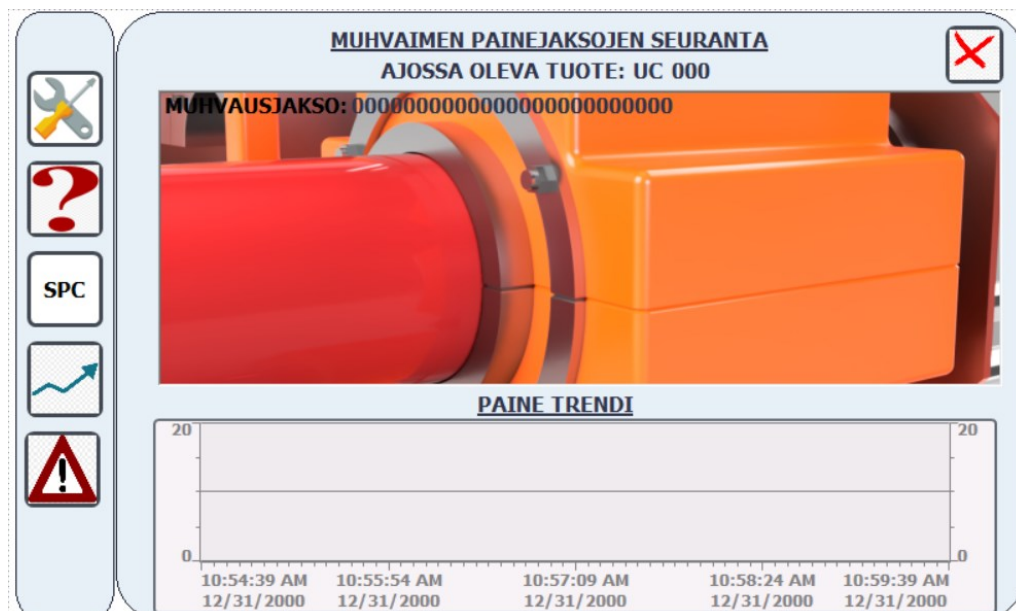


Kuva 31. Menuvalikko.

Menuvalikkoon lisättiin painonappi, jossa on muhvaimen kuva ja teksti muhvain. Painonappia painamalla siirrytään muhvainsivulle. Muhvaussivu on pääsivu, muhvaimen alisivuille.

7.5.2 Muhvaussivu

Kuvassa 32 näkyy pääsivu eli muhvaussivu. Kaikkien sivujen yläkulmasta löytyy rasti, josta pääsee palaamaan edelliselle sivulle.



Kuva 32. Muhvain sivu.

Muhvaussivulla näkyy asetussivulle määriteltynä oleva tuote ja sen dimensio. Muhvattavia putkia tuotantolinjalla ajetaan viittä eri dimensiota välillä 160–400 mm. Sivun keskellä on ruutu, jossa näytetään kunkin muhvausjakson tapahtumat animaationa ja kullakin hetkellä menossa olevan muhvausjakson nimi. Sivun alalaidassa on paineentrendikuvaaja, josta voi tarkastella sen hetkistä painetrendiä, mikäli muhvaus on käynnissä. Oikeassa laidassa on painonapit, joista pääsee siirtymään alasisivuille.

7.5.3 Muhvausasetussivu

Muhvaussivulla olevasta työkalun kuvasta siirrytään kuvassa 33 näkyvälle asetussivulle. Asetukset pyrittiin tarkoituksella minimoimaan ja niitä lisätään myöhemmin tarvittaessa.

ASETUKSET

MÄÄRITETÄÄN UUDET OHJAUSRAJAT:

VALITSE UC PUTKEN HALKAISJA:

MONTA MUHVIA OHJAUSRAJOJEN LASKEMISEEN MUHVATAAN: [kpl]

OHJAUSRAJA MÄÄRITTELYN PAINE RAJA: [Max. paine - bar]

MUHVAUSJAKSONPAINE TODELLINEN: [bar]

Kuva 33. Muhvain asetukset.

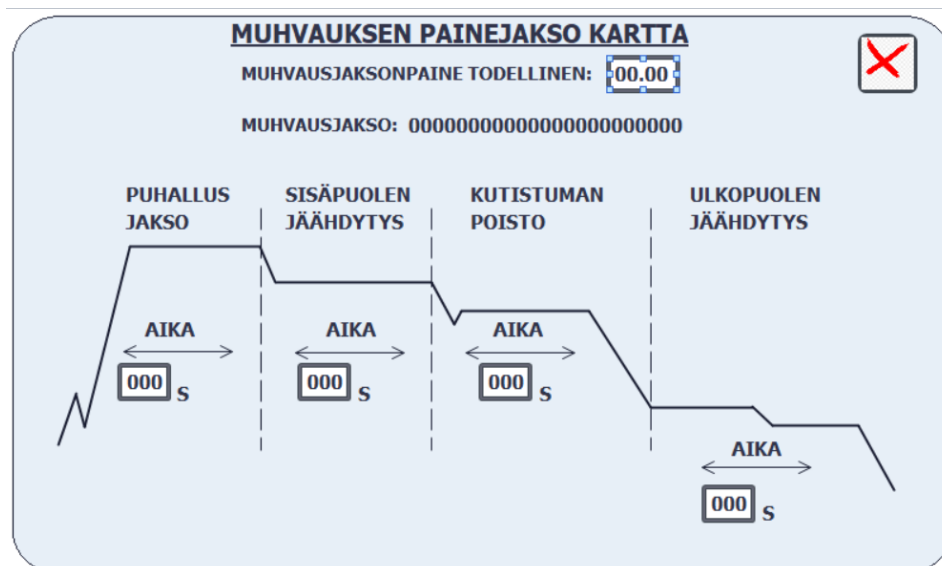
Asetussivulta löytyy kuitenkin muun muassa Määritäohjausrajat-painonappi. Kun painonappia painetaan, painonappi muuttuu vihreäksi, joka indikoi ohjausrajojen määrittämisen olevan käynnissä. Ohjausrajat määritetään, mikäli operaattori haluaa määrittää uudet ohjausrajat. Ohjausrajojen määrittämistä varten operaattorin tarvitsee myös kertoa Pythonille, kuinka monen muhvin perusteella hän haluaa ohjausrajojen määrittämisen tapahtuvan. Ohjausrajojen määrittämisen jälkeen painonappi muuttuu takaisin harmaaksi.

Tuotantolinjalla ajetaan vain yhtä muhvatavaa putkea, joten tuotteen nimen määrittämiseen tarvitaan vain putken dimensio. Python käyttää tuotteen nimeä ohjausrajojen tallentamiseen oikeaan tiedostoon ja toisaalta tuotevaihdossa tiedoston avaamiseen ohjausrajojen hakemista varten.

Asetussivulla asetetaan ohjausrajamäärittelyn paineraja. Ensimmäinen painejakso, joka on puhalluspainejakso, alkaa nollan baarin paineesta, ja viimeinen painejakso, joka on ulkopuolen jäähdytys, päättyy nollan baarin paineeseen. Paineiden nousu- ja laskuajat pitää poistaa ohjausrajojen määrittelystä. Ohjausrajamäärittelyn aikana kerätyistä painenäytteistä poimitaan suurin painelukema, josta vähennetään operaattorin antama ohjausrajamäärittelyn painerajalukema. Kerätystä datasta poistetaan kaikki tämän erotuksen alapuolella olevat luvut ennen kuin ohjausrajat lasketaan. Näin ohjausrajat eivät vääristy nousu- ja lasku-aikojen takia.

7.5.4 Painejaksokartta

Kuvassa 34 näkyy painejaksokarttasivu. Sivulta voi tarkastella viimeisimmän muhvin kunkin painejakson kestoaikaa.

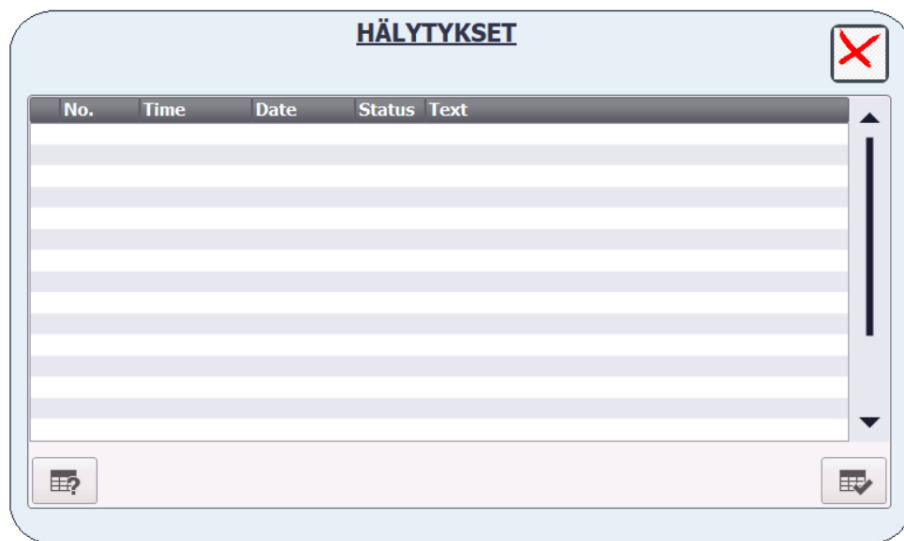


Kuva 34. Painejaksokartta.

Muhvauksenpainejaksokarttasivulta löytyy lisäksi sen hetkinen muhvausjakson nimi sekä sen hetkinen muhvausjakson painelukema. Operaattorille muistutuksena, sivulle on visualisoitu, miltä painetrendikuvaajan tulisi näyttää silloin kun ongelmia ei ole.

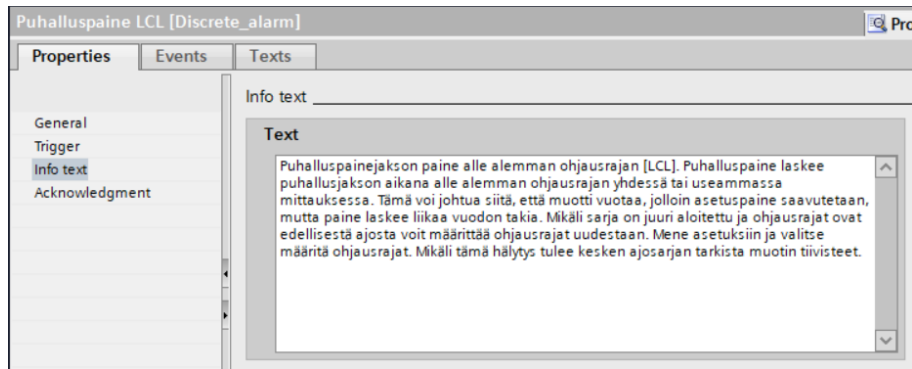
7.5.5 Hälytyssivu

Analysoinnin jälkeen Python-ohjelma lähettää hälytykset. Hälytyksiä varten luotiin OPC UA -palvelimen tietojen vaihtoa varten luotuun muistilohkoon (DB) hälytys1 -muuttuja, jonka datatyyppi on Integer (int) eli kokonaisluku. Integer-muuttuja sisältää 16 bittiä, joista jokaista käytetään yhtenä hälytyksenä, eli toisin sanoen yhteen muuttuun saadaan 16 eri hälytystä. Muhvaimelle tehtiin aluksi 8 hälytystä: jokaisen painejakson alemman ohjausrajan (LCL) alituksesta ja ylemmän ohjausrajan (UCL) ylityksestä tulee hälytys. Hälytykset näytetään käyttöliittymän hälytyssivulla, joka on esitetty kuvassa 35.



Kuva 35. Hälytyssivu.

Mikäli hälytyssivulle tulee hälytys, kysymysmerkkiä painamalla pääsee lukemaan kuvassa 36 näkyvän ohjetekstin. Hälytyssivulla on myös hälytyksen kuitauspainike.

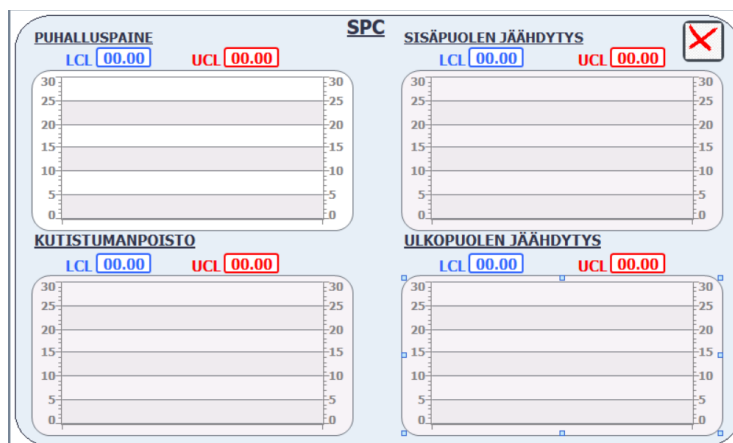


Kuva 36. Hälytyksen ohjeet.

Hälytykset eivät vaadi operaattorilta toimenpiteitä ja ne voidaan heti niiden lukemisen jälkeen kuitata. Hälytys kuitenkin uusiutuu seuraavalla muhvauksella, mikäli operaattori ei korjaa hälytyksen aiheuttajaa.

7.5.6 SPC-korttisivu

SPC-korttisivulle tehtiin neljä trendinäkömää, jokaiselle painejaksolle oma trendinäkömä. Asetussivulla määritetyn tuotteen mukaiset alemmat ohjaukset ja ylempät ohjaukset ladataan painejakson nimen alapuolelle laatikoihin, jotka näkyvät kuvassa 37.



Kuva 37. SPC-kortti.

Alemman ohjausrajan mukainen arvo piirtää trendikuvaajaan sinisen suoranviivan ja ylemmän ohjausrajan mukainen arvo piirtää trendikuvaajaan punaisen suoranviivan. Muhvauksen jälkeen Python-ohjelma lähettää neljä taulukkoa, yhden jokaisesta painejaksosta. Taulukossa on painenäytteet, jotka kyseisestä painejaksosta kerättiin pois lukien paineen nousu- ja laskuajat. Nämä poistetaan, koska muuten niistä aiheutuisi vääriä hälytyksiä.

Taulukossa olevat datapisteet esitetään mustana trendiviivana painejakson mukaisessa kuvaajassa. Tämän trendin pitäisi siis pysyä alemman ja ylemmän ohjausrajan välissä. Mikäli näin ei ole, Python antaa hälytyksen, jossa kerrotaan, missä painejaksossa ylitys tai alitus tapahtui.

8 Tulokset

Järjestelmää koeajettiin tuotantolinjalla kahtena päivänä, esiin tulleet pienet ongelmat saatiin ratkaistua ja lopulta kaikki toiminnot saatiin toimimaan suunnitellulla tavalla. Painejaksojen oikeanlainen toiminta oli helppo havaita käyttöliittymästä, ja hälytykset tulivat oikeaan aikaan myös tiedonsiirrot toimivat oikein.

Koeajon perusteella havaittiin, että Python-ohjelma oli tietyissä tilanteissa melko herkkä vioille ja vaatii muutamia vian korjauksia ennen kuin se siirretään tehdasverkossa olevalle palvelimelle. Koeajossa palvelimena käytettiin kannettavaa tietokonetta, jota ei voi pitää tuotantolinjalla pitkiä aikoja.

9 Yhteenveto

Opinnäytetyön alussa tutustuttiin muhvausprosessiin, käytiin läpi opinnäytetyössä käytettyjä komponentteja ja tutkittiin tilastollista prosessinohjausta. Tämän jälkeen tutustuttiin opinnäytetyössä käytettyihin tiedonsiirtotapoihin, tietokoneen ohjelmointiin Python-ohjelmointikielellä sekä logiikan ohjelmointiin TIA Portal -ohjelmointiympäristössä. Viimeiseksi käytiin läpi käyttöliittymän rakenne ja siihen tehdyt lisäsivut.

Opinnäytetyössä tutkittiin painejaksojen toimintaa. Muhvausprosessissa on muitakin prosessin onnistumisen kannalta kriittisiä suureita, kuten putkenpään lämpötila ennen muhvausta ja muhvauksen jälkeen. Mikäli painejaksoista saatu data osoittautuu käyttökelpoiseksi, kannattaa myös muita suureita tutkia tilastollisen prosessinohjauksen metodeilla.

Lähteet

Avainluvut. Verkkoaineisto. Uponor. <<https://www.uponorgroup.com/fi-fi/sijoittajat/taloudellinen-tieto/avainluvut>>. Luettu 10.9.2023.

Bee, Liam. 2022. TIA Portal basics - working with pointers. Verkkoaineisto. Liam Bee. <<https://liambee.me/siemens/tia-portal-basics-working-with-pointers/>>. 07.10.2022. Luettu 10.1.2024.

Better living environments-The story of Uponor. 2017. Yrityksen sisäinen aineisto. Uponor.

Devasia, Anish. 2023. PLC Programming: Overview of IEC 61131-3 in Industrial Automation Systems. Verkkoaineisto. Control Automation. <<https://control.com/technical-articles/an-overview-of-iec-61131-3-Industrial-Automation-Systems/>>. 25.03.2023. Luettu 20.10.2023.

Free OPC UA-client.2024.

Kankaanranta, Jarno. 2015. Kahden moniulotteisen SPC-kortin laskeminen Minitabilla. Verkkoaineisto. Quality Know How Oy. <<https://sixsigma.fi/spc-kortin-laskeminen-minitabilla/>>.10.06.2015. Luettu 11.8.2023.

Keinänen, Toimi & Sumujärvi, Matti. 2019. Automaatiotekniikka. Helsinki: Sanoma Pro Oy.

Keiski, Arto. 2016. OPC UA ja teollinen asioiden internet. Insinööriyö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.

Keskihajonnat. Verkkoaineisto. opinnot.net. <https://opinnot.net/kokonaisuu-det/index.php?id_kokon=209#>. Luettu 20.8.2023.

Kurri, Veijo; Malen, Timo; Sandell, Risto & Virtanen, Matti. 2002. Muovitekniikan perusteet. Helsinki: Hakapaino Oy.

LOGO! basic modules. Verkkoaineisto. Siemens. <<https://new.siemens.com/global/en/products/automation/systems/industrial/plc/logo/logo-basic-modules.html>>. Luettu 12.6.2023.

Machine-level visualization with simatic HMI. 2018. Verkkoaineisto. <<https://assets.new.siemens.com/siemens/assets/api/uuid:796d5e95-d5a9-4ddc-93d0-a3597bdaa7b5/dffa-b10135-03-br-simatic-hmi-mehrwerte-24-pages-deenus-144.pdf>>. Päivitetty 01.01.2018. Luettu 26.5.2023.

Mattila, M. Regressio-analyysi. Verkkoaineisto. Mattila M. <<https://www.mv.helsinki.fi/home/mmmattila/fsd/regressio.htm>>. Luettu 12.8.2023.

Mikkonen, Isto. 2008. Yrittäjyys, omistajuus, kansainvälisyys. Uponor Oyj 90 vuotta. Hämeenlinna: Karisto.

Minitab. Versio 21.1.0. 2024. Minitab.

OPC UA lesson 1- what is OPC-UA? 2020. Verkkoaineisto. Singh, Rajvir. <<https://www.youtube.com/watch?v=3EREV8Q5PNU>>. Julkaistu 13.10.2020. Katsottu 20.3.2023.

OPC UA NodeId Concepts. Verkkoaineisto. Unified Automation. <https://documentation.unifiedautomation.com/uasdkhp/1.4.1/html/_l2_ua_node_ids.html>. Luettu 8.4.2023.

OSI-malli. 2024. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/OSI-malli>>. Päivitetty 29.02.2024. Luettu 5.8.2023.

PLC basics. Programmable Logic Controller. 2018. Verkkoaineisto. Real Pars. <https://www.youtube.com/watch?v=PbAGI_mv5XI>. Julkaistu 6.8.2018. Katsottu 10.10.2023.

PLC-ohjelmointi. Toimilohkon soveltaminen PLC-ohjelmassa. Verkkoaineisto. Satakunnan ammattikorkeakoulu. <https://automa-tion.samk.fi/html/AuNe/AuNe_PLC_ohjelmointi_FB#FB_fi>. Luettu 11.11.2023.

PROFINET Introduction. 2018. Verkkoaineisto. MinutePROFINET. <<https://www.youtube.com/watch?v=htoCzxLyJtU>>. Julkaistu 20.02.2018. Kat-sottu 15.1.2024.

Python frequently asked question. What is Python? 2024. Verkkoaineisto. <Pyt-hon.org. <https://docs.python.org/3/faq/general.html#what-is-python>>. Päivitetty 15.04.2024. Luettu 20.1.2024.

Python (ohjelmointikieli). 2024. Verkkoaineisto. Wikipedia. <[https://fi.wikipedia.org/wiki/Python_\(ohjelmointikieli\)#Historia](https://fi.wikipedia.org/wiki/Python_(ohjelmointikieli)#Historia)>. Päivitetty 25.03.2024. Luettu 26.1.2024.

S7 communication with PUT/GET. 2019. Verkkoaineisto. Siemens. <[https://cache.industry.siemens.com/dl/fi-les/115/82212115/att_108330/v2/82212115_s7_communication_s7-1500_en.pdf](https://cache.industry.siemens.com/dl/files/115/82212115/att_108330/v2/82212115_s7_communication_s7-1500_en.pdf)>. Päivitetty 01.09.2019>. Luettu 15.1.2024.

S7-1500 structure and use of the CPU memory. 2013. Verkkoaineisto. Sie-mens. <Siemens. [https://cache.industry.siemens.com/dl/fi-les/101/59193101/att_80162/v1/s71500_structure_and_use_of_the_PLC_me-mory_function_manual_en-US_en-US.pdf](https://cache.industry.siemens.com/dl/files/101/59193101/att_80162/v1/s71500_structure_and_use_of_the_PLC_memory_function_manual_en-US_en-US.pdf)>. Päivitetty 01.01.2013. Luettu 10.06.2023.

Seppälä, Jukka. 2008. Polymeeriteknologian perusteet. Helsinki: Otatieto.

SI-mittayksiköt Suomessa, massa ja paine. Verkkoaineisto. VTT. <<https://www.vttresearch.com/fi/si-mittayksikot-suomessa>>. Luettu 1.12.2022.

Tia portal: PUT command / PLC-PLC communication. 2019. Verkkoaineisto. Hegamurl. <<https://www.youtube.com/watch?v=q85-L5u-96I>>. Julkaistu 6.10.2019. Katsottu 3.3.2023.

TIA Portal. Verkkoaineisto. Siemens. <<https://www.siemens.com/fi/fi/tuotteet/teollisuus/tia-portal.html>>. Luettu 1.2.2024.

TIA-Portal. Versio 16.0. 2023. Siemens.

Types of siemens PLC, S7-300,400,1200 & S7-1500. 2020. Verkkoaineisto. Engineering training platform. <<https://www.youtube.com/watch?v=gbRyPGCBfZI>>. Julkaistu 10.09.2020. Katsottu 10.6.2023.

Visual Studio. 2024. Verkkoaineisto. Wikipedia. <https://en.wikipedia.org/wiki/Visual_Studio#History>. Päivitetty 13.04.2024. Luettu 15.2.2024.

What is Visual Studio? 2023. Verkkoaineisto. Microsoft. <<https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>>. 24.10.2023. Luettu 1.2.2024.

What is Ethernet/IP? 2019. Verkkoaineisto. Real Pars. <<https://www.youtube.com/watch?v=mm-NHrLtRWI>>. Julkaistu 25.2.2019. Katsottu 10.12.2023.