TAMPEREEN
AMMATTIKORKEAKOULU

AUTOMAATIOLINJOJEN TIEDONKERÄYS LOGIIKASTA

Järjestelmän kokonaistehokkuuslaskenta

Soile Pihlajaviita

Opinnäytetyö
Maaliskuu 2016
Sähkötekniikan koulutusohjelma
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkötekniikan koulutusohjelma
Automaatiotekniikka

PIHLAJAVIITA, SOILE:
Automaatiolinjojen tiedonkeräys logiikasta
Järjestelmän kokonaistehokkuuslaskenta

Opinnäytetyö 20 sivua
Maaliskuu 2016

Massadatan kerääminen on teollisuudessa yleistyvä ilmiö. Tiedon avulla voidaan luoda merkittäviä säästöjä huollossa tai parempaa kuvaa hallittavuudesta ja siten luotettavuudesta. Tässä opinnäytetyössä rakennettiin robottijärjestelmiä toimittavalle Orfer Oy:lle pohja tiedonkeräysjärjestelmälle. Tiedonkeräyksen päätarkoituksena oli antaa materiaalia järjestelmän kokonaistehokkuuden (Overall Equipment Efficiency) laskentaan tilanseurannalla, prosessien tehostamiseen häiriöiden aiheuttajien ja pullonkaulojen löytämisellä ja ennakoivaan huoltoon kunnonseurannalla. Työn aikana keskityttiin Siemens 300/400-sarjan ohjelmoitaviin logiikoihin ja niistä saatavilla olevaan tietoon. Tehtävänä oli luoda logiikkaan luettava tietorakenne ja tehdä sen uudelleentoteuttamisesta mahdollisimman yksinkertaista.

Työn tuloksena järjestelmä jaettiin toiminnan perusteella alueisiin. Näistä alueista kerättiin tilatietoja niiden senhetkisestä valmius- ja toimintatilasta sekä suoritetuista työvaiheista ja niiden kestoista. Luotiin kolme erilaista logiikan tietolohkoa (Data Block): logiikan ja tiedonkeräyksen yhteyden tilasta kertova DB, logiikan ohjaamien järjestelmäkokonaisuuksien DB:t, jotka sisältävät järjestelmän alueiden tiedot, ja komponenttien DB, joka sisältää yksittäisten osien toimintatietoja kunnonvalvontaa varten. Tiedon tallentamiseksi luotiin ohjelmoitavaan logiikkaan ohjelmointikirjasto, jonka avulla keräyksen uudelleen toistaminen on helppoa nopeaa. Lisäksi työn aikana luotiin erilaisia dokumentteja, kuten kirjaston käyttöohje, uudelleentoteutuksen helpottamiseksi. Työn aikana tiedonkeräys toteutettiin kahteen robottihitsaussoluun ja yhteen asiakkaan tilaamaan lavaajaan.

Toteutetun tiedonkeräyksen avulla kokonaistehokkuuden laskennasta kyetään laskemaan järjestelmän käytettävyys ja nopeus. Laadunhallinnan suunnittelu vaatii prosessikohtaista suunnittelua ja sen ratkaiseminen yleisellä tasolla on erittäin haastavaa. Tiedonkeräyksen tuottaman tiedon avulla voidaan myös analysoida, mitkä työvaiheet tuottavat eniten virheitä. Komponenttien suoritustietojen keräyksen tallennusmoduuleja tulee luoda ohjelmointikirjastoon jatkossa aina, kun uudentyyppinen laite otetaan käyttöön järjestelmissä. Tiedonkeräystä pystytään laajentamaan keräämällä tietoa roboteista ja ympäristön tilasta. Pyrkimällä päivittämään työssä tuotettuja dokumentteja pysyy osaaminen, ja siten uudelleentoteutuksen tehokkuus, yhtiössä.

Asiasanat: tiedonkeräys, kokonaistehokkuus, KNL-laskenta, logiikkaohjelmointi, Siemens S7-sarja, teollinen Internet

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical Engineering
Automation Engineering

PIHLAJAVIITA, SOILE:
Data Collection of Automated Systems from a Programmable Logic Controller
Overall Equipment Efficiency Calculation

Bachelor's thesis 20 pages
March 2016

Internet of things and era of mass data are concepts growing in popularity. By collecting data you are able to create substantial savings by increasing the accuracy of preventive maintenance to predictive or boost your sales with intuitive monitoring systems and therefore making the entire product more believable. The purpose of this thesis was to create a basis of a data collecting system for Orfer Oy which provides automation systems built around robots. The data collection should offer information for Overall Equipment Efficiency Calculation by state monitoring, potential process performance increase by finding out common causes of faults and hindrances and for predictive maintenance by monitoring component performance. In this thesis the focus was on the data offered by Siemens 300/400 series Programmable Logic Controllers. The task was to develop a framework for the external data reading and make rebuilding the frame as simple and fast as possible.

In the course of the work it was decided to divide each system to areas by function. From these areas data of their state and stage durations would be collected. Three different kinds of Data Blocks were created in the logic: a DB for the state of the PLC and connection between the PLC and its external data collector, DBs for each system controlled by the PLC and a DB for component performance data for predictive maintenance. For easy rebuilding of the data collection system a programming library was created for Simatic Step 7. In addition documentation, such as manual for the programming library, was created to further hasten the process. During this thesis work the data collection was applied to two existing robot welding cells and a palletizing system delivered to a customer.

With the framework availability and performance values can be calculated from Overall Equipment Efficiency. To calculate quality value much planning and knowledge of the process in question is required and is extremely difficult to implement on a general level. The stages on which most faults occur can also be deduced from the data provided by this work. Data saving modules for new component types should always be added in the progress of applying one in a system. The data collection system can be expanded by collecting information from robots and the state of the system’s surroundings. The efficiency of implementing the framework can be sustained by keeping the provided documentation up to date.

Key words: data collection, overall equipment efficiency, OEE calculation, PLC programming, Siemens S7-series, Internet of things

SISÄLLYS

1 JOHDANTO

Teollisuudessa suuren tiedon määrän kerääminen on kasvava ilmiö. Tietokoneiden tallennustila on nykyään merkityksettömän halpaa ja niihin voidaan tallentaa jatkuvaa mittaustietoa lukuisista laitteista vuosien ajan. Myös erilaisia analysointialgoritmeja on kehitteillä esimerkiksi komponenttien kuntoanalyysiin. Massadatan avulla voidaan saada paljon arvokasta tietoa muun muassa ennakoivaan huoltoon, jolla voidaan minimoida tehtaan käyttämättömyysaikaa.

Tässä opinnäytetyössä luotiin robottijärjestelmien tuottaja Orfer Oy:lle tiedonkeräysmalli ohjelmoitavaan logiikkaan. Aluksi tavoitteena oli saada materiaalia järjestelmän kokonaistehokkuuslaskennan käytettävyyteen ja häiriötilanteiden ehkäisemiseen. Myöhemmin siitä voidaan laajentaa laadunvalvonnan ja suorituskyvyn seurantaan, ennakoivan huollon kuntoanalyysiin ja jopa prosessin pullonkaulojen parantamiseen. Työn tavoitteena oli luoda kestävä ja helposti toistettavissa oleva tiedonkeräysmalli. Tämä toteutettiin suunnittelemalla tietorakenne, joka on koottavissa ja laajennettavissa pienemmistä muuttumattomista osista, ja tallennuskirjasto, jolla tiedonkeräyksen lisääminen järjestelmään on helppoa ohjelmointisuunnittelun yhteydessä.

Tiedon tallennuskirjasto luotiin Siemensin Step 7 -ohjelmistolla 300/400-sarjan ohjelmoitavaan logiikkaan. Logiikassa tieto kerätään tietolohkoihin, joista analyysin palveluntarjoajan tiedonkeräin käy säännöllisin väliajoin lukemassa tiedot ja lähettää ne eteenpäin etäpalvelimelle laskentaa varten. Tiedonkeräys toteutettiin suunnittelun tukena ja toiminnan varmistamiseksi kahdessa erilaisessa hitsaussolussa: ihmisen työpanostusta vaativa solu, jossa logiikan ja hitsausrobotin välillä on minimaalinen kommunikaatio, ja täysin automaattisesti hitsausrobotin kanssa toimiva solu.

2 LÄHTÖKOHTIEN ESITTELY

Työn tavoitteena oli luoda pohja Orfer Oy:n tuotteiden kokonaistehokkuuden laskennalle ja ennakoivan huollon mallille. Tiedonkeräyksen määrittely tehtiin tiiviissä yhteistyössä Orfer Oy:n automaatiopäällikön Jesse Luhdin ja tiedonkäsittelyn palveluntarjoajan kanssa.

2.1 Orfer Oy

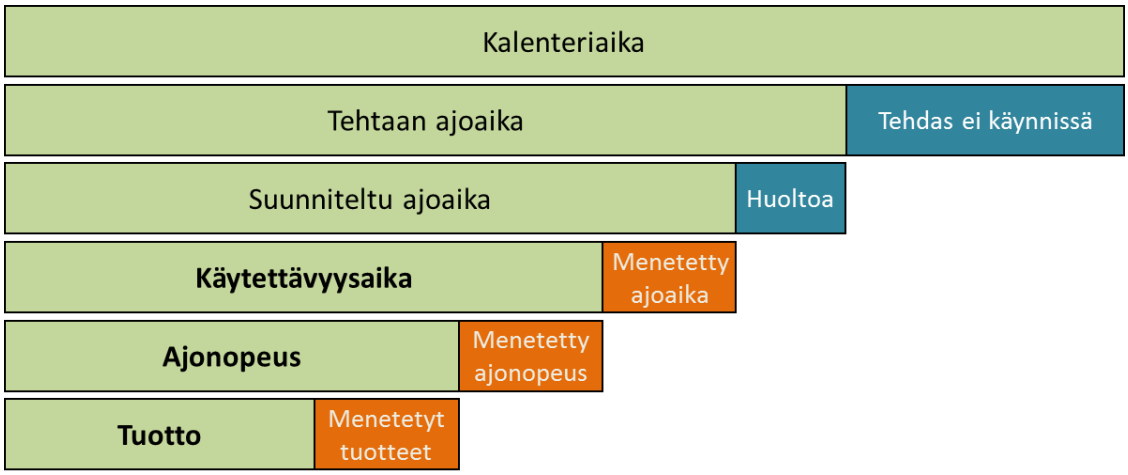
Työn teettäjä Orfer Oy on perheyritys, joka suunnittelee ja valmistaa Kawasakin ja Toshiba Machinen robotteja sisältäviä kappaleenkäsittelyjärjestelmiä. Merkittävä osa järjestelmistä on elintarviketeollisuuden lavaajasoluja, joissa käytetään melko kevyen kuorman nivelrobotteja ja noukintarobotteja. Esimerkkinä kuvassa (KUVA 1.) Kawasakin 130 kg nostokuorman nivelrobotti ja 2 kg noukintarobotti. (Orfer Oy. 2015.)

KUVA 1. Kawasaki ZX130L nivelrobotti ja YS002N noukintarobotti (Kawasaki. 2015.)

Orfer Oy:n liikevaihto on 10–20 miljoonan euron luokassa. Yhtiö työllistää noin 80 henkilöä päätehtaalla Orimattilassa ja lisäksi sivutoimipisteillä Keuruulla ja Pietarissa. Yhtiön toimitusjohtajana toimii Paul Stucki.

2.2 Järjestelmän kokonaistehokkuus

Overall Equipment Effectiveness (OEE tai suomeksi KNL) eli järjestelmän kokonaistehokkuus on ideaalitasolla järjestelmän tuotantoaika suhteessa kokonaisajoaikaan. Käytännössä kuitenkin menetykset tuotantonopeudessa ja tuotteen laadussa vaikuttavat laitteesta saatavaan hyötyyn (KUVIO 1.). Yleisesti järjestelmän kokonaistehokkuuden prosenttiluku lasketaan kertomalla käytettävyys-, suoritus- ja laatukertoimet keskenään (KAAVA 1.). (Vorne Industries. 2012.)



KUVIO 1. Järjestelmän tehokkuuden laskennan jaottelu

$$OEE = Käytettävyys \cdot Nopeus \cdot Laatu$$
Esimerkiksi: $OEE = 89{,}0\ \% \cdot 96{,}0\ \% \cdot 99{,}8\ \% \approx 85{,}3\ \%$ (1)

Käytettävyys tai saatavuus (Availability) on järjestelmän todellinen ajoaika suhteessa suunniteltuun ajoaikaan eli kuinka suuren osan suunnitellusta ajosta se kykenee suorittamaan ongelmitta (KAAVA 2.) (Vorne Industries. 2012.). Nopeus (Performance) ottaa huomioon järjestelmän todellisen suorituskyvyn suhteessa siihen mihin järjestelmän on luvattu kykenevän (KAAVA 3.) (Vorne Industries. 2012.). Toisaalta automaatiopäällikkö Luhdin mukaan (2015) nopeutta voi usein säätää myös käyttäjä, joka saattaa laskea prosessointinopeutta, jos esimerkiksi tarve ei ole sillä hetkellä niin suuri, että tarvitsisi ajaa täysillä. Nopeuden mittaaminen on kokonaistehokkuuslaskennan kannalta ehkä vaikein määrittää. Tuotteen laatu (Quality) kuvaa laatuehtojen täyttävien tuotteiden määrää suhteessa tuotettujen kappaleiden määrään (KAAVA 4.) (Vorne Industries. 2012.). Esimerkiksi paketointijärjestelmässä tuotteita saatetaan hylätä epämuodostuneen tai revenneen paketin perusteella.

$$Käytettävyys\ (\%) = Ajoaika\ /Suunniteltu\ ajoaika \quad (2)$$

$$\begin{aligned} &Nopeus\ (\%) \\ &= Ideaalinen\ syklinopeus/(Ajoaika/Tuotemäärä)\ tai \\ &= (Tuotemäärä/Ajoaika)/Ideaalinopeus \end{aligned} \quad (3)$$

$$Laatu\ (\%) = Hyväksytyt\ tuotteet/Tuotemäärä \quad (4)$$

Esimerkiksi ajoaikaa usein mitataan tunteina tai, jatkuvasti ajettavalla järjestelmällä pitkällä aikavälillä, vuorokausina. Tuotemäärää tarkastellaan tuotteesta riippuen kappalemääränä, massana tai tilavuutena. Syklinopeudella tarkoitetaan, kauanko määrätyn tuotemäärän valmistuksessa kestää. Vastaavasti nopeudella tarkoitetaan, kuinka paljon tuotetta pystytään valmistamaan määrätyssä ajassa.

Tässä työssä keskityttiin ensisijaisesti tiedon keräämiseksi saatavuuslaskentaa varten, sillä se on yleisesti määritettävissä kaikille järjestelmille. Laskentaan vaaditaan tieto järjestelmän tuotantoajosta eli automaattiajotilasta, häiriötilasta ja tuotannon ulkopuolisesta käsiajosta, jota voi olla esimerkiksi huolto tai päivittäinen pesu. Lisäksi vaaditaan tieto ajasta, jolloin järjestelmää tahdotaan ajaa. Tämä tieto voidaan saada esimerkiksi laitteesta, jolle tehdään erillinen tuotannon aloitus ja lopetus, tai tehtaan vuorolistasta.

Nopeus- ja laatulaskennan tiedot ovat hyvin prosessikohtaisia ja tulee määrittää erikseen jokaiselle järjestelmälle. Työssä luotuun logiikan tietorakenteeseen luotiin kuitenkin paikka nopeusarvon tallentamiseen ja eteenpäin siirtämiseen logiikalta laskennan palveluntarjoajalle. Kyseiseen paikkaan järjestelmän tunteva ohjelmoija kykenee lisäämään määrittämänsä koneen nopeusarvon.

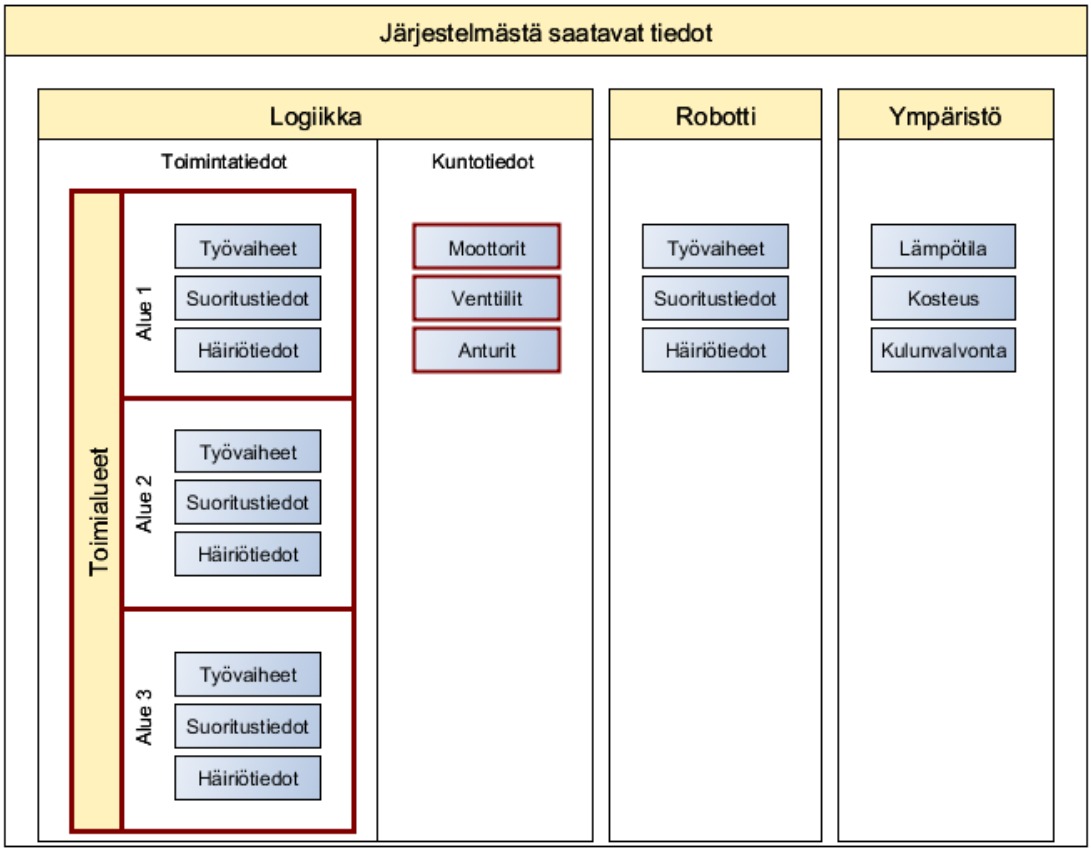
3 TIETORAKENNE

Tiedon sijainti, tyyppi ja sen merkitys tulee olla tiedossa, jotta sitä voidaan hyödyntää. Työn alussa määritettiin tiedonsiirron säännöt, kerättävä tieto sekä miten tieto jaotellaan ja sijoitetaan luettavaksi.

3.1 Jaottelun määrittely

Automaatiojärjestelmän voi jakaa pienempiin osiin monilla tavoilla. Lisäksi järjestelmän toimintaan liittyy myös muita asioita, joita voidaan seurata, esimerkiksi ympäristön lämpötila tai kosteus tai ihmisten sijainti järjestelmän valvomossa tai lähistöllä. Työn tiedonkeruun suunnittelu kohdistui ohjelmoitaviin logiikoihin ja niistä saatavilla olevaan tietoon. Logiikoita voi järjestelmässä olla useita ja yksi logiikka voi ohjata useita erilaisia toimintakokonaisuuksia. Näitä toimintakokonaisuuksia nimitetään tässä työssä koneiksi, ja niitä ovat esimerkiksi hitsaussolu tai lavaaja.

Koneessa on erilaisia osia, jotka toteuttavat yhtä tehtävää. Esimerkiksi lavaajassa voi olla tuotteiden syöttö eli kuljetin, joka saattaa kääntää tuotteet oikeaan asentoon helpompaa sijoittelua varten, lavaajarobotti, joka nostaa ja asettaa tuotteet lavalle, asettaa uuden lavan ja vaaditut välipahvit, sekä poisto, jolla valmiit lavat kuljetetaan eteenpäin. Näitä koneen osia nimitetään tässä työssä toimialueiksi ja niiden yksittäisten toimintojen suorittamisesta tahdottiin kerätä tilatietoja (KUVIO 2.). Koneiden ja niiden alueiden toimintatietojen lisäksi kerätään tietoa yksittäisten toimilaitteiden ja komponenttien toiminnasta. Nämä yksittäiset laitteet päätettiin kerätä erillisenä listauksena helpompaa erillistä kuntoanalyysiä varten.



KUVIO 2. Datarakenteen jaottelu

Tiedonkäsittelyn palveluntarjoajan datakeräin lukee tietoa suoraan Siemensin logiikan tietolohkoista (Data Block). Useita DB:itä voidaan siis käyttää erottamaan laajoja määriä tietoa ja siten selkeyttämään tietorakennetta. Kuvion (KUVIO 2.) mukaisesti päätettiin luoda DB yleiseen logiikan toimintaan ja tiedonkeräimen yhteyteen liittyville asioille, yksi DB jokaiselle koneelle ja niiden alueiden toiminnalle, sekä yksi DB yksittäisten toimilaitteiden ja komponenttien arvoille. Aina rakenteeltaan samanlaisiksi ja helposti monistettaviksi ajatellut osat on rajattu kuvassa punaisella.

3.2 Kerättävän tiedon määrittely

Ensisijaisena tavoitteena työssä oli kerätä tietoa järjestelmien käytöstä kokonaistehokkuuden laskemiseksi (luku 2.2). Tulevaisuudessa oli myös suunnitelmana tarkkailla osien kuntoa osana ennakoivaa huoltoa. Aluejaon määritelmän myötä käytönseuranta jakautui pienempiin osiin. Sen sijaan komponenttien kuntoa seurataan yksilötasolla.

Käytöstä tahdotaan tarkkailla, missä tilassa järjestelmä on: tekeekö se työtä, odottaako se jotain, ohjataanko sitä käsin tai onko jotain ongelmia. Erilaisista tiloista kerrotaan tarkemmin myöhemmässä luvussa (luku 3.3). Tilan lisäksi haluttiin tietää, mitä työvaihetta alue suorittaa ja kuinka kauan yhden vaiheen suoritus kestää. Työvaiheen seurannalla voidaan päätellä muun muassa missä työvaiheissa ilmenee eniten ongelmia tai miten esimerkiksi sylinterin hidastuminen vaikuttaa alueen toimintaan. Lisäksi päätettiin seurata alueen toimintanopeutta, jotta voitaisiin tulevaisuudessa pyrkiä parantamaan järjestelmän kapasiteettia parantamalla pullonkaula-alueita.

Jos järjestelmässä ilmenee käytön estävä ongelma, täytyy ongelma myös tunnistaa. Robotin häiriö päädyttiin ilmaisemaan lukuarvona, jonka mahdolliset arvot annetaan listana tiedonkäsittelyn palveluntarjoajalle analyysiä varten. Tämä on mahdollista, koska Kawasakin ja Toshiban roboteissa voi olla vain yksi häiriö kerrallaan. Logiikassa sen sijaan usein on useita hälytyksiä, joten niiden ilmaiseminen yhdellä lukuarvolla ei ole järkevää. Logiikkaohjelmissa hälytykset kerätään yleensä omaan DB:hen, joten niiden lukeminen sieltä on helppoa. Logiikan automaattisen toiminnan estävästä virheestä ilmaistaan antamalla häiriökoodina luku, joka ohjeistaa katsomaan hälytys-DB:stä.

Koneista kerätään prosessoitujen tuotteiden määrä ja tarkkaillaan, mitä tuotetta käsitellään. Esimerkiksi elintarviketeollisuudessa ajetaan erilaisia tuotantoeriä tarpeen mukaan, sillä samalla lavauslinjalla voidaan asetella useita tuotteita pienillä asetusten vaihdoilla. Myöhemmin saatetaan lisätä tiedonkeräykseen myös tuotteiden laadunvalvonnan seurantaa: esimerkiksi väärän paketin muodon perusteella hylätyt kappaleet. Laadunvalvonta on kuitenkin niin laaja ja prosessikohtainen aihe, että sitä ei tässä työssä määritetty perusrakenteeseen.

Komponenteista ja toimilaitteista kerätään niiden toimintaan ja kuntoon liittyviä tietoja eli käytännössä kaikki mitattavat suureet, mitä niistä saadaan irti. Esimerkiksi sylintereistä mitataan männän kulkuaikaa ja siten nopeutta, jonka hidastumisesta voidaan päätellä sylinterin olevan vaihdon tarpeessa, tai moottoria ohjaavista taajuusmuuttajista voidaan saada käynnistyskertojen tueksi moottorin nopeus, lämpötila ja momentti. Tulevaisuudessa näiden tietojen historian perusteella voidaan ennakoida osan rikkoutumista sen toiminnan heikentymisellä tai yhdistää jonkin arvon muuttuminen tiettyyn ongelmaan.

Työssä luodun tiedonkeräysmallin tuli olla helposti toistettavissa sekä pieniin, että suurempiin järjestelmiin. Laajennettavuuden ja uudelleen rakentamisen mahdollistamiseksi keräyksen täytyi koostua pienistä muuttumattomista osista. Muuttumattomaksi ajatellut osat on kuviossa (KUVIO 2.) rajattu punaisella. Tällaisiksi muuttumattomiksi osiksi voitiin määrittää alueet, joista koneet koostuvat. Koneita voi logiikan hallittavissa taas olla useita. Koneista kerättävän tiedon määrää ja rakennetta ei lyöty lukkoon mahdollisen laadunvalvonnan tai muun tulevaisuuden lisäyksen johdosta. Vastaavasti voidaan olettaa jokaisen perussylinteristä tai suoraohjattavasta moottorista saatavien tietojen olevan aina samanlaiset. Siksi jokaiselle laitetyypille voidaan määrittää perusrakenne, joista komponenttitietolista voidaan tulevaisuudessa rakentaa. Uudenlaiseen laitteen tullessa käyttöön sille luodaan oma tiedonkeräysrakenne tulevaisuuden käyttöä varten.

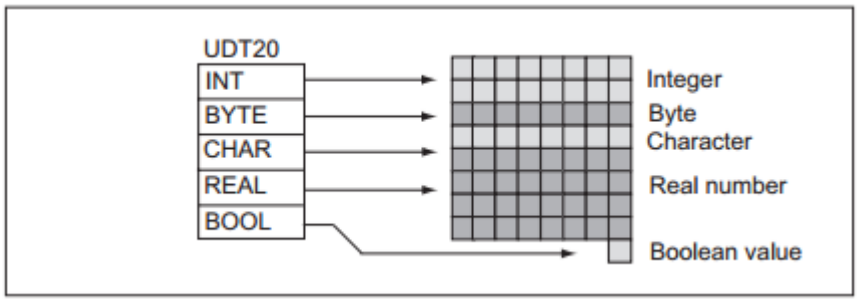
Tietojen tyyppien määrittely

Kerättävän tiedon tyypit valittiin siten, että tiedon tarkkuus- ja monipuolisuusehdot täyttyivät. Kuitenkin varsinkin monistettavien osuuksien koko pyrittiin pitämään mahdollisimman pienenä tilan säästämiseksi. Esimerkiksi alueen erilaisia tiloja ajateltiin olevan vajaa kymmenen, joten niin vähän erilaisen tilan ilmaisemiseen riittäisi kolme bittiä. Tilan ilmaisemiseen valittiin siten pienin riittävä Siemensin logiikan perustyyppi BYTE (TAULUKKO 1.). Työvaiheiden kestoajat taas tahdottiin mitata millisekunnin tarkkuudella, joten kun datatyypiksi valittiin 32-bittinen kaksoissana DWORD, voidaan tallentaa 2^{32} - 1 ms eli 4 294 967 295 ms pitkä ajanjakso, joka on 49 vrk 17 t 2 min 42,295 s. (Siemens. 2006a.)

TAULUKKO 1. Siemens STEP7 perustietotyypit (Siemens. 2006a.)

Type and Description	Size in Bits	Format Options	Range and Number Notation (lowest to highest value)_	Example
BOOL(Bit)	1	Boolean text	TRUE/FALSE	TRUE
BYTE (Byte)	8	Hexadecimal number	B#16#0 to B#16#FF	L B#16#10 L byte#16#10
WORD (Word)	16	Binary number Hexadecimal number BCD Decimal number unsigned	2#0 to 2#1111_1111_1111_1111 W#16#0 to W#16#FFFF C#0 to C#999 B#(0.0) to B#(255.255)	L 2#0001_0000_0000_0000 L W#16#1000 L word#16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (Double word)	32	Binary number Hexadecimal number Decimal number unsigned	2#0 to 2#1111_1111_1111_1111 1111_1111_1111_1111 DW#16#0000_0000 to DW#16#FFFF_FFFF B#(0,0,0,0) to B#(255,255,255,255)	2#1000_0001_0001_1000_ 1011_1011_0111_1111 L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (Integer)	16	Decimal number signed	-32768 to 32767	L 1
DINT (Integer, 32 bits)	32	Decimal number signed	L#-2147483648 to L#2147483647	L L#1
REAL (Floating-point number)	32	IEEE Floating-point number	Upper limit: ±3.402823e+38 Lower limit: ±1.175 495e-38	L 1.234567e+13
S5TIME (SIMATIC time)	16	S7 time in steps of 10 ms (default)	S5T#0H_0M_0S_10MS to S5T#2H_46M_30S_0MS and S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (IEC time)	32	IEC time in steps of 1 ms, integer signed	-T#24D_20H_31M_23S_648MS to T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (IEC date)	16	IEC date in steps of 1 day	D#1990-1-1 to D#2168-12-31	L D#1996-3-15 L DATE#1996-3-15
TIME_OF_DAY (Time)	32	Time in steps of 1 ms	TOD#0:0:0.0 to TOD#23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
CHAR (Character)	8	ASCII characters	'A','B' etc.	L 'E'

Siemensin logiikoissa yleisin perustietotyyppi on 16-bittinen sana eli WORD. Vaikka vähemmän tilaa vaativiakin tyyppejä voi käyttää, varaa logiikka tiedolle kahden tavun mittaisen alueen muistista, jos seuraava tieto vaatii sanan verran tai enemmän tilaa. Tämän takia 16-bittiä pienemmät tiedot kannattaa ryhmittää yhteen, jotta tilaa ei jää käyttämättä. Kuviossa (KUVIO 3.) on sijoitettu kaksi 8-bittistä muuttujaa, BYTE ja CHAR, peräkkäin. Jos kuvan rakenteen BOOL-muuttujan perässä oleva muuttuja on 16-bittinen tai suurempi, alkaisi muuttuja vasta 15-bittiä myöhemmin. (Siemens. 2006a.)

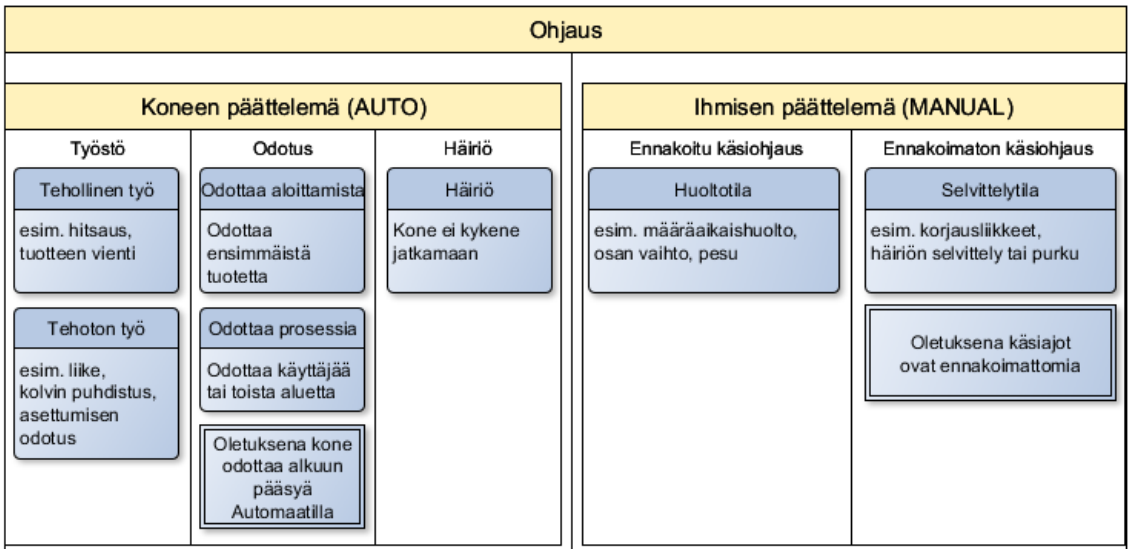


KUVIO 3. Muistin käyttö datarakenteessa (Siemens. 2006a.)

3.3 Tilahierarkian määrittely

KNL-laskennan kannalta olennaisinta on tietää, kuinka paljon konetta käytetään ja kuinka paljon se olisi käytettävissä. Lisäksi tulevaisuuden kehittämistä varten tahdottiin tarkkailla, kuinka suuri osa käytöstä on ns. rahallista arvoa tuottavaa eli tehollista työtä. Odotus on toimettomuutta, kun järjestelmä on käytössä. Myös odotus jaettiin kahteen osaan: odotetaan alkuun pääsyä, eli ensimmäistä tuotetta, tai odotetaan toista tuotantoon liittyvää prosessia, joko automaattista konetta tai käyttäjän manuaalityövaihetta. Esimerkkinä manuaalityövaiheesta on esimerkiksi täyden lavan vienti pois trukilla. Myös käsiajo jaettiin kahteen osaan: ennakoituun käsiajoon eli huoltoajoon ja ennakoimattomaan käsiajoon, jolla yleisimmin puretaan häiriön aiheuttamaa tilannetta. Esimerkiksi tukkeutunutta kuljetinruuvia peruutetaan taaksepäin tukoksen poistamisen mahdollistamiseksi.

Työssä oli tavoitteena luoda ohjelmoijalle selkeä määritystapa merkitä, milloin kone on missäkin tilassa. Mahdolliset tilat näkyvät kuviossa (KUVIO 4.), mutta työn tuloksena syntynyttä tilahierarkian päättelyketjun vuokaaviota ei teettäjän pyynnöstä ole sisällytetty tähän raporttiin.



KUVIO 4. Järjestelmän tilat

Konetta voidaan ohjata kahdella tavalla: automaattisesti, eli koneen päättelemää ohjausta, ja käsiajolla, eli ihmisen päättelemää ohjausta. Automaattisen järjestelmän tavoitteena on toimia itsenäisesti ilman ihmisen apua. Tästä voimme päätellä, että kone on käytettävissä, kun se kykenee automaattituotantoon. Kuitenkin tehtaita on erilaisia: joissain tehtaissa ajetaan kolmessa vuorossa vuorokaudessa eli kellon ympäri, kun taas toisissa ajetaan vain yksi vuoro päivässä eli kolmasosa vähemmän. Tämä käyttämättömyys vääristää saatavuuslaskentaa ja tulee leikata pois, sillä tuolloin koneen omistaja ei edes tahdo ajaa konettaan (KUVIO 1.). KNL-laskennassa käytettävyyden laskennan vertailukohteena on suunniteltu ajoaika. Suunniteltuun ajoaikaan ei myöskään sisälly järjestelmän suunniteltu huolto, kuten määräaikainen osan vaihto tai elintarvikejärjestelmän pesu. Jos järjestelmään on suunniteltu automaattitoimintoja huollon helpottamiseksi, tulee ne erottaa tuotannon automaattiajosta laskennassa.

Tilanseuranta päätettiin tehdä erikseen jokaiselle alueelle, jolloin pystytään seuraamaan syy-seuraussuhteita alueiden välillä ja etsimään prosessista mahdollisia pullonkauloja. Kuitenkin Orfer Oy:n valmistamissa koneissa robotti on pääasiallinen arvoa tuottava osa, joten usein vain yhden alueen suoriutuminen on olennaista varsinaisen KNL-laskennan kannalta. Nämä ovat prosessikohtaisia asioita, joihin ei syvemmin perehdytä tässä työssä.

3.4 Tiedonsiirron yhteyden määrittely

Tietoa kerättiin logiikasta sisäverkkoon kytketyllä erillisellä datakeräimellä, jonka toimitti tiedonkäsittelyn palveluntarjoaja. Tämä datakeräin kykenee lukemaan suoraan kokonaisia DB:itä, joiden tiedot se tallentaa puskuriin ja lähettää ne eteenpäin tietokantapalvelimelle. Datakeräin lukee suoraan logiikasta yhden tai useamman DB:n yhdellä lukukerralla. Lukutapahtuma voi tapahtua missä tahansa vaiheessa logiikan ohjelmakierron syklissä (Luhti. 2015). Työn alussa määritettiin logiikan ja datakeräimen kommunikoinnin ehdot, kuten tiedonsiirron tiheys ja varmistukset. Määrityksen valinnat vaikuttivat merkittävästi logiikan tietojen tallennuksen rakenteeseen.

Toteutetun tiedonkeräyksen tarkoituksena oli tuottaa materiaalia tuotteiden kokonaistehokkuuslaskentaan ja ennakoivaan huoltoon. Tiedon ei tarvitse olla reaaliaikaista, mutta tietovirran tulee olla aukoton, sillä tiedon menetys tekee analyysistä helposti käyttökelvottoman. Koska tiedon ei tarvitse olla reaaliaikaista, määritettiin tiedonsiirron kyselytiheys pitäen mielessä sen aiheuttama verkon ja laitteiden kuormitus ja logiikan puskurin vaatima muisti. Päätettiin, että datakeräin lukee logiikan tiedot 5 sekunnin välein.

Sovitun viiden sekunnin tarkkuuden todettiin olevan riittävä tilan tallennuksen tarkasteluväli tehokkuuslaskentaa varten. Koska tuotteiden työvaiheiden pituuksien arvioitiin olevan yleensä sekunnista minuutteihin, sovittiin logiikan tallentavan neljä edellistä työvaihetta muistiin. Koska työvaiheet ovat kestoltaan erilaisia, tulee merkitä, montako työvaihetta on kertynyt puskuriin viime lukukerrasta. Tämän toteuttamiseksi luotiin juokseva työvaiheiden laskuri, jonka arvosta lukuhetkellä voidaan päätellä uudet työvaiheet.

Datakeräimen ja logiikan välille luotiin yhteydenvarmistus siltä varalta, jos asiakas vaatii tuotannon pysäyttämistä, jos tiedonkeräys ei toimi. Esimerkiksi lääketeollisuudessa valmistuksen valvonta on erittäin tiukkaa ja tiedon menetys saattaisi tulla hyvinkin kalliiksi (Fimea 6/2011). Yhteyden tarkkailuun logiikkaan määritettiin yksi bitti, jota datakeräin kirjoittaa onnistuneen lukukerran jälkeen. Keräimen kirjoitustapahtuma on erillinen lukutapahtumasta (Luhti. 2015.). Logiikka nollaa kirjoitetun bitin ennen odotettavaa seuraavaa lukukertaa. Bitin avulla oli mahdollista luoda hälytys jos datakeräin ei ole käynyt lukemassa logiikan tietoja.

4 TALLENNUSKIRJASTO

Työn tavoitteena oli luoda helposti uudelleentoistettavissa oleva tiedonkeräysmalli. Myös tiedon tallennuksen luominen mahdollisimman helpoksi sisältyy tähän tavoitteeseen. Koska Orfer Oy käyttää tuotteissaan pääasiassa Siemensin 300/400-sarjan logiikoita, luotiin yhtiön käyttöön kopioitavissa oleva kirjasto tallennuksen toteuttavia funktioita. Lisäksi kirjoitettiin käyttöohje helpottamaan kirjaston käyttöä ja ilmaisemaan kirjaston aiheuttamat rajoitukset. Ohjelmointikirjasto on joukko valmiita ohjelman osia, funktioita tai rakenteita, joita voi siirtää ja käyttää useassa ohjelmassa. Ideaalinen kirjasto on täysin riippumaton tehtävän ohjelman toiminnasta ja erilaisten laitteiden ominaisuuksista. Tässä luvussa pohditaan hyvän ohjelmointikirjaston ominaisuuksia ja ongelmia, joita suunnittelussa tuli vastaan.

4.1 Funktioiden ja DB:iden numerointi

Siemensin Step 7 -ohjelmistossa jokaisella rakenteella ja funktiolla tulee olla identifioiva ja tyypissään yksilöllinen numero ohjelmassa. Esimerkiksi Simatic 300-sarjassa funktioiden mahdollinen numeroalue on 0–7999 ja DB:iden 0–16000 (Siemens. 2010). Myös kirjaston funktioille tulee antaa numerot. Numeroiden tulee olla mahdollisimman epätodennäköisesti käytössä yhdessäkään ohjelmassa, johon kirjastoa halutaan käyttää. Siemensin omat kirjastot usein käyttävät numeroita 0–200. Pyöreitä lukuja on myös hyvä välttää, sillä ne usein tulevat ihmisellä ensimmäisenä mieleen. Myös täysin loppupää on erittäin todennäköisesti valittu käyttöön jossain vaiheessa.

Koska tavoitteena oli käyttää kirjastoa kaikissa tulevissa tuotteissa, merkittiin valitut numerovaraukset yhtiön ohjelmointistandardiin, jotta päällekkäisyyksiä ei syntyisi. Edellä mainitut numeron valinnan kriteereitä pohdittiin kuitenkin, jotta vältyttäisiin päällekkäisyyksiltä aiemmin luotujen kirjastojen kanssa, jotka mahdollisesti jäivät huomioimatta. Kirjasto myös luotiin pyrkien välttämään riippuvaisuutta numeroinnista, joten myöhemmin niitä voidaan muuttaa tarpeen vaatiessa. Ainoa kirjaston osa, joka jäi riippuvaiseksi numerosta, oli muuttumattomaksi sovittu aluerakenteen UDT (User Defined Type). Kaikki rakennetta käsittelevät funktiot vaativat tarkan tiedon UDT rakenteesta ja sen kiertäminen funktion sisällä olisi ollut turhan työlästä.

4.2 Muuttujien sijainti

Siemens 300/400-sarjan logiikoissa kaikki pysyvä tieto tallennetaan DB:ihin. Kun tietoa käytetään, sitä osoitetaan antamalla sen sijainti kyseisessä DB:ssä. Kirjasto pyrittiin tekemään riippumattomaksi DB:iden numeroinnista ja esimerkiksi alueiden määrä ja siten myös tiedon määrä DB:ssä vaihtelee. Näistä syistä kirjaston funktioissa ei voinut suoraan kutsua muuttujia. Sen sijaan kirjaston käyttäjä, ohjelmoija, antaa funktioille niiden tarvitsemat tiedot. Aluetta käsitteleville funktioille syötetään alueen UDT-rakenne yksittäisten alueiden tietojen sijaan käytön yksinkertaistamiseksi.

4.3 Aikaan liittyvät toiminnot

Tiedonkeräyksessä mitataan aikaa eri tarkoituksiin, esimerkiksi yhteyden tarkistukseen ja työvaiheiden keston määrittämiseen. Logiikka sisältää valmista ajastinmuistia, mutta se on usein hyvin rajallista ja sitä tarvitaan myös prosessin hallintaan. Siemensin 300/400-sarjan logiikan Program Cycle Organization Block (OB1), josta kaikki ohjelmakierrot alkavat, sisältää INT muuttujan OB1_PREV_CYCLE, joka on tieto edellisen ohjelmakierron suoritusajasta millisekunnin tarkkuudella. Muuttuja sijaitsee OB1:n väliaikaismuistissa, joten sitä ei helposti voida lukea muissa funktioissa, sillä tarkan sijainnin määrittäminen Local-muistialueella on hankalaa ohjelman syvemmillä tasoilla. Ongelma ratkaistiin tekemällä merkintä yhtiön ohjelmointistandardiin, että OB1_PREV_CYCLE tulee siirtää Memory-muistialueelle OB1-lohkon suorituksen alussa. (Siemens. 2006b.)

Edellisen ohjelmakierron suoritusajan avulla voitiin luoda työvaiheille ja vastaaville ajastuksille laskenta lisäämällä aina edellinen suoritusaika muistipaikkaan. Testijärjestelmissä ohjelman suoritusaika oli 1–3 ms. Hyvin lyhyestä suoritusajasta johtuen millisekunnin tarkkuus aiheuttaa pientä epätarkkuutta ajastuksessa. Yhteyden varmistuksessa päädyttiin käyttämään logiikan ajastinta helpomman toteutuksen saavuttamiseksi vetohidastuksella. Kuitenkaan logiikasta ei varattu aina vakiota ajastinnumeroa, vaan kirjaston käyttäjä syöttää funktiolle yhden vapaan ajastimen.

5 POHDINTA

Työn tuloksena syntyi pohja Orfer Oy:n tulevalle tiedonkeruumallille. Työn aikana luodulla keräysjärjestelmällä saadaan logiikan tiedot koneen käytöstä, sen tiloista ja häiriöistä, alueiden työvaiheista ja niiden kestoista. Tietorakenteeseen pystyy myös järjestelmän tunteva henkilö määrittämään KNL-laskentaan vaaditun nopeusarvon. Näillä työkaluilla pystytään kattamaan jo kaksi kolmasosaa kokonaistehokkuuden laskennassa. Työssä luodun tilan päättelykaavion ja tallennuskirjaston avulla konetta suunnitteleva henkilö tulisi kyetä lisäämään työn tuloksena syntyneet seurantaominaisuudet parissa tunnissa ja vanhoihin järjestelmiin vain hieman pidemmässä ajassa. Työn aikana tilatiedon ja työvaihetiedon keräykset toteutettiin kahteen hitsaussoluun. Lisäksi eräs Orfer Oy:n logiikkaohjelmoija toteutti keräyksen yhteen asiakkaan lavaajaan ja loi samalla esiversion komponenttien kunnonvalvonnasta.

Kokonaistehokkuuden laskennan täydentämiseksi seuraava askel on luoda yhtenäisempi määritystapa nopeuden todellisen arvon ja vertailuarvon asettamiseksi. Lisäksi laadunvalvonnan toteuttamista tulisi suunnitella. Eräs yksinkertainen tapa olisi merkitä muistiin hylättyjen tuotteiden määrä ja hylkäysperuste. Kokonaistehokkuuden laskennan lisäksi koneiden komponenttien kunnonvalvonnan tallennusmoduuleja tulisi luoda sitä mukaa, kun tulee vastaan uudenlaisia osia, joista voi kerätä erilaisia tietoja. Näitä tietoja voidaan tulevaisuudessa hyödyntää ennakoivassa huollossa. Tiedonkeräystä voidaan myös laajentaa keräämällä muista lähteistä, kuten robotista tai ympäristöstä esimerkiksi ilmaston tai kulunvalvonnalla. Ylläpitämällä työn ohessa kirjoitettua tallennuskirjaston käyttöohjetta ja muuta dokumentointia varmistetaan myös osaamisen ja siten käytön tehokkuuden pysymistä.

LÄHTEET

Fimea. Lääkealan turvallisuus- ja kehittämiskeskuksen määräys apteekkien lääkevalmistuksesta 6/2011.

Kawasaki. Products. Large Payloads Robots. ZX130L Robot. Luettu 2.11.2015. https://robotics.kawasaki.com/en/products/robots/large-payloads/ZX130L/

Kawasaki. Products. Pick & Place Robots. YS002N Robot. Luettu 28.10.2015. https://robotics.kawasaki.com/en/products/robots/pick-place/YS002N/

Luhti, J. Automaatiopäällikkö. 2015. Opinnäytetyöohjaus syksyllä 2015. Orfer Oy. Orimattila.

Orfer Oy. Orfer Oy esite. Luettu 28.10.2015. http://orfer.fi/Portals/114/esitteet/ORFER_A4-esite.pdf

Orfer Oy. Referenssit. Luettu 28.10.2015. http://orfer.fi/suomeksi/ORFER/Referenssit/tabid/5359/language/en-US/Default.aspx

Siemens. 2006a. SIMATIC. Programming with STEP 7 Manual.

Siemens. 2006b. SIMATIC. System Software for S7-300/400 System and Standard Functions Volume 1/2.

Siemens. 2010. SIMATIC. Products for Totally Integrated Automation and Micro Automation. Catalog News ST 70 N.

Vorne Industries. 2012. World Class OEE ja Calculating OEE. Luettu 19.1.2016. http://oee.com/