



TALLINNA TEHNIKAÜLIKOOL

INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

TÖÖSTUSLIKU ENNUSTAVA HOOLDUSSÜSTEEMI UNIVERSAALSE TARKVARARAAMISTIKU ARENDAMINE

DEVELOPMENT OF A UNIVERSAL SOFTWARE FRAMEWORK FOR INDUSTRIAL
PREDICTIVE MAINTENANCE SYSTEM

BAKALAUREUSETÖÖ

Üliõpilane: Jaan Ojasild

Üliõpilaskood: 179176EAAB

Juhendaja: Tarmo Korõtko, teadur

Tallinn, 2020

(Tiitellehe pöördel)

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“20” mai 2020

Autor: Jaan Ojasild

/ allkiri /

Töö vastab bakalaureusetööle esitatud nõuetele

“.....” 201.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

“.....”201... .

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Jaan Ojasild

Lõputöö liik: Bakalaureusetöö

Töö pealkiri: Tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku arendamine

Kuupäev:

58 lk (lõputöö lehekülgede arv koos lisadega)

20.05.2020

Ülikool: Tallinna Tehnikaülikool

Teaduskond: Inseneriteaduskond

Instituut: Elektroenergeetika ja mehhatroonika instituut

Töö juhendaja(d): Tarmo Korõtko, PhD

Töö konsultant (konsultandid):

Sisu kirjeldus:

Käesoleva lõputöö eesmärgiks on edendada tööstuses kasutatavate ennustavate hooldussüsteemide arendust läbi universaalse tarkvararaamistiku, mis lihtsustab anduritelt näitude kogumist, mõõteandmete töötlemist, masinõppe algoritmide rakendamist ning andmeedastuse teostamist.

Lõputöö põhisisu koosneb levinumate ennustavate hooldussüsteemide lahenduste analüüsist, töö käigus arendatud universaalse tarkvararaamistiku kirjeldusest ning selle põhjal loodud ennustava hooldussüsteemi kirjeldusest.

Lõputöö koostamiseks kasutati peamiselt kirjanduse analüüsi, mõõteandmete analüüsi ning praktilisi mõõtmisi ja katsetusi.

Lõputöö põhiliseks tulemuseks on koostatud universaalne tarkvararaamistik, mis lihtsustab anduritelt näitude kogumist, mõõteandmete töötlemist, masinõppe algoritmide rakendamist ning andmeedastuse teostamist.

Märksõnad: hooldussüsteem, servaraalimine, masinõpe, andmetöötlus, tarkvararaamistik

ABSTRACT

Author: Jaan Ojasild

Type of the work: Bachelor Thesis

Title: Universal software framework development of industrial predictive maintenance system

Date: 20.05.2020

58 pages (the number of thesis pages including appendices)

University: Tallinn University of Technology

School: School of Engineering

Department: Department of Electrical Power Engineering and Mechatronics

Supervisor(s) of the thesis: Tarmo Korõtko, PhD

Consultant(s):

Abstract:

The aim of this bachelor thesis is to advance the development of the predictive maintenance systems currently used in industry, by creating a universal software framework that would simplify gathering data from sensors, processing it, relaying data to the user and applying machine learning algorithms for pattern recognition and component useful life estimation.

The main part of the thesis is made of an analysis of the most common predictive maintenance solutions, a description of the developed software framework and the predictive maintenance system created with it.

Main methods used in the making of this thesis were analysis of literature and measurement data alongside practical measurements and tests.

The main result of this thesis is a universal software framework for industrial predictive maintenance systems that simplifies gathering data from sensors, processing this data, storing processed data and results, sending data to the user and applying machine learning algorithms for pattern recognition and component useful life estimation.

Keywords: maintenance system, edge computing, machine learning, data science, framework

LÕPUTÖÖ ÜLESANNE

Lõputöö teema:	Tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku arendamine
Lõputöö teema inglise keeles:	Universal Software Framework Development of Industrial Predictive Maintenance System
Üliõpilane:	Jaan Ojasild, 179176EAAB
Eriala:	
Lõputöö liik:	Bakalaureusetöö
Lõputöö juhendaja:	Tarmo Korõtko
Lõputöö ülesande kehtivusaeg:	06.2020
Lõputöö esitamise tähtaeg:	20.5.20

Üliõpilane (allkiri)

Juhendaja (allkiri)

Õppekava juht (allkiri)

1. Teema põhjendus

Tänapäeval kasutatakse tööstuses kolme hooldussüsteemi: vajadusepõhist, ennetavat ja ennustavat. Erinevalt vajaduse põhisest ja ennetavast hooldussüsteemist, aitab ennustav hooldus maksimeerida süsteemi komponentide kasutusaega, ennetades samal ajal avariolukordade tekkimist. Ennustava hoolduse süsteemid võimaldavad ajatada seadmete hooldustööd, optimeerides sellega seadmete tööaega. Ennustava hoolduse süsteem aitab seadmetes vähendada nende hooldamiseks kuluvat aega ja raha, eemaldades enneaegsed või üleliigsed hooldustööd.

Efektivsete ennustavate hooldussüsteemide arendamine on eriti oluline komponentide jaoks, mille rikke maksumus on kõrge või mille puhul tekib otsene oht inimeste tervisele. Tänapäeval rakendatakse selliste komponentide puhul enamjaolt ennetavat hooldust, kuid see jätab märkimisväärse osa komponendi kasulikust elueast kasutamata, vähendades seeläbi seadme kulutõhusust.

Ennustava hooldussüsteemi loomisel kulub suur osa ressursidest tarkvara arendamisele. Käesolev töö keskendub universaalse tarkvararaamistiku loomisele, mis on kasutatav tööstuslike

ennustavate hooldussüsteemide arendamisel, vähendades seeläbi uute rakenduste valmistamiseks vajamineva töö mahtu.

2. Töö eesmärk

Bakalaureusetöö eesmärgiks on edendada tööstuses kasutatavate ennustavate hooldussüsteemide arendust läbi universaalse tarkvararaamistiku, mis lihtsustab anduritelt näitude kogumist, mõõteandmete töötlemist, masinõppe algoritmide rakendamist ning andmeedastuse teostamist.

3. Lahendamisele kuuluvate küsimuste loetelu:

- 1.** Millised on levinumad tööstuses kasutatavad ennustavad hooldussüsteemid ning millised on nende poolt tagatud funktsionaalsused?
- 2.** Kas tööstuses levinud ennustatavate hooldussüsteemide funktsionaalsust saab asetada universaalsesse tarkvararaamistikku?
- 3.** Kas arendatud universaalse tarkvararaamistiku baasil ehitatud tööstuslik ennustav hooldussüsteem on kasutatav süsteemi rikete ennetamisel?

4. Lähteandmed

Lähteandmetena kasutatakse:

- Avalikult kättesaadavad mõõteandmed tööstuslike seadmete tööst.
- Tehnoloogiliste seadmete mõõtmiste käigus kogutud mõõteandmed.

5. Uurimismeetodid

Uurimismeetoditena kasutatakse:

- Kirjanduse analüüs.
- Mõõteandmete analüüs (MATLAB, MS Excel).
- Praktilised mõõtmised ning katsetused.

6. Graafiline osa

Graafilise osana esitatakse:

- Ennustavate hooldussüsteemide analüüs.
- Hooldussüsteemi tarkvaraarhitektuur.
- Hooldussüsteemi tarkvararaamistiku struktuur.
- Hooldussüsteemi prototüübi elektriskeemid.
- Erinevad graafikud ja tabelid eksperimentide ja mõõtmiste tulemuste esitamiseks.

Graafiline osa asub enamjaolt töö põhiosas.

7. Töö struktuur

- 1.** Peatükk - Ennustavate hooldussüsteemide kirjeldus.
- 2.** Peatükk - Tööstusliku ennustava hooldussüsteemi tarkvararaamistiku loomine.
- 3.** Peatükk - Tööstusliku ennustava hooldussüsteemi tarkvararaamistiku arendamine.
- 4.** Peatükk - Tööstusliku ennustava hooldussüsteemi tarkvararaamistiku katsetamine prototüübil.
- 5.** Peatükk - Juhend tarkvararaamistiku rakendamiseks ning kokkuvõte.

8. Kasutatud kirjanduse allikad

Kirjanduse allikatena kasutatakse:

- Teaduspublikatsioone.
- Muud erialakirjandust.
- Tehnoloogiliste seadmete andmelehti.

9. Lõputöö konsultandid

Lõputöö juhendaja

10. Töö etapid ja ajakava

- 1.** Kirjanduse läbitöötamine (01.11.2019)
- 2.** Tööstuslikes ennustava hoolduse süsteemides kasutatava riist- ja tarkvara uurimine (01.11.2019)
- 3.** Tarkvararaamistiku loomine (01.12.2019)
- 4.** Tarkvararaamistiku arendamine (07.02.2020)
- 5.** Tarkvararaamistiku katsetamine prototüübil
- 6.** Lõputöö kirjaliku osa koostamine (07.03.2020)
- 7.** Järelduste kirjutamine (13.03.2020)
- 8.** Kokkuvõtte koostamine (27.03.2020)
- 9.** Töö esimene versioon valmis (31.03.2020)
- 10.** Juhendajale esimeseks läbilugemiseks saatmine (01.04.2020)
- 11.** Paranduste sisseviimine
- 12.** Juhendajale teiseks lugemiseks saatmine
- 13.** Töö lõplik versioon valmis.

SISUKORD

LÕPUTÖÖ LÜHIKOKKUVÕTE.....	3
ABSTRACT.....	4
LÕPUTÖÖ ÜLESANNE.....	5
EESSÕNA.....	11
SISSEJUHATUS.....	12
1.ENNUSTAVATE HOOLDUSSÜSTEEMIDE KIRJELDUS.....	13
1.1 Hooldussüsteemide tüübid.....	13
1.1.1 Vajaduspõhine hooldussüsteem.....	13
1.1.2 Ennetav hooldussüsteem.....	14
1.1.3 Ennustav hooldussüsteem.....	15
1.2 Hooldussüsteemi osad.....	17
1.2.1 Prognoosimine ja mustrite leidmine.....	19
1.3 Levinud hooldussüsteemid ja nende poolt pakutav funktsionaalsus.....	22
1.3.1 IBM.....	22
1.3.2 SIEMENS AG	27
1.3.3 SAP SE.....	30
1.4 Ennustavate hooldussüsteemide võrdlus.....	33
2.TÖÖSTUSLIKU ENNUSTAVA TARKVARARAAMISTIKU PROJEKTEERIMINE.....	35
2.1 Tarkvararaamistik.....	35
2.2 Tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku funktsionaalsused.....	36
2.3 Universaalse tarkvararaamistiku arhitektuur.....	37

3.TÖÖSTUSLIKU ENNUSTAVA TARKVARARAAMISTIKU ARENDAMINE.....	40
3.1 Osad.....	41
3.2 Kasutatud vahendid.....	42
4.TÖÖSTUSLIKU ENNUSTAVA HOOLDUSÜSTEEMI KATSETAMINE.....	43
4.1 Tarkvararaamistiku katsetamise tingimused.....	43
4.2 Prototüübi kirjeldus.....	43
4.2.1 Prototüübi üldine kirjeldus.....	43
4.2.2 Prototüübi komponendid.....	44
4.3 Hooldussüsteemi kirjeldus.....	49
4.3.1 Hooldussüsteemi rakenduse osad.....	50
4.4 Hinnang tarkvararaamistikule.....	53
KOKKUVÕTE.....	54
SUMMARY.....	55
KASUTATUD KIRJANDUS.....	56

EESSÕNA

Käesolev bakalaureuse lõputöö on tehtud teadur Tarmo Korõtko ettepanekul. Lõputöö eesmärgiks oli edendada tööstuses kasutatavate ennustavate hooldussüsteemide arendust läbi universaalse tarkvararaamistiku, mis lihtsustab anduritelt näitude kogumist, mõõteandmete töötlemist, masinõppe algoritmide rakendamist ning andmeedastuse teostamist. Lõputöö koosnes kolmest etapist: olemasolevate hooldussüsteemide lahendustega tutvumine, hooldussüsteemi tarkvararaamistiku arendamine ning loodud tarkvararaamistiku katsetamine, koostades, tarkvararaamistikku kasutades, ennustava hooldussüsteemi Tallinna Tehnikaülikooli poolt antud prototüübil. Töö praktiline osa toimus Tallinna Tehnikaülikooli laboris, koostöös Tarmo Korõtko ja Vahur Maaskiga.

Autor soovib tänada lõputöö juhendajat Tarmo Korõtkot asjatundliku nõu ja abi eest.

SISSEJUHATUS

Seadmete hooldus on ettevõtte tahk, mis ei pälvi suurt tähelepanu, kuid mis tagab seadmete korrektse töö ning nende poolt toodetavate toodete kvaliteedi. Ilma hoolduseta oleksid seadmete parandusest tulenevad kulutused suured ning seadme rikke poolt põhjustatud õnnetuse oht inimesele liiga suur. Just nimelt seadmete korrektse ja ohutu töö, toote kvaliteedi ning võimalikult väikeste paranduskulude tagamiseks on vaja seadmeid hooldada.

Nagu kõik teised tehnoloogiad, on ka hooldus läbi aegade arenenud. Viimaseim samm selles arengus on ennustav hooldus, mis võimaldab ette näha kui seade vajab hooldust või mõni komponent vajab välja vahetamist. Selleks kasutatakse erinevaid andureid, juhtseadmeid, andmebaase ning analüütilisi mudeleid. Nagu võib arvata, võtab selle kõige koostamine ja paigaldamine palju aega ja tööd. Universaalne tarkvararaamistik, aitaks vähendada vaja minevat töö hulka ning muuta süsteemi paigalduse hõlpsamaks ja odavamaks.

Käesoleva bakalaureusetöö eesmärgiks on edendada tööstuses kasutatavate ennustavate hooldussüsteemide arendust läbi universaalse tarkvararaamistiku, mis lihtsustab anduritelt näitude kogumist, mõõteandmete töötlemist, masinõppe algoritmide rakendamist ning andmeedastuse teostamist.

Töö on jaotatud kolmeks erinevaks osaks:

1. Esimeses peatükis on toodud lühike ülevaade hooldussüsteemide arnegust ning uuritud kolme enim levinud tööstusliku ennustava hooldussüsteemi lahendust ning millist funktsionaalsust nad pakuvad.
2. Teises ja kolmandas peatükis on kirjeldatud tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku loomine ja arendamine. Lõpptulemuseks on universaalne tarkvararaamistik, mis võimaldab esimeses peatükis vaadeldud lahenduste poolt pakututest sobilikud funktsionaalsused.
3. Neljandas peatükis on kirjeldatud ennustavat hooldussüsteemi, mis on koostatud kasutades antud töö käigus loodud tarkvararaamistikku.

1. ENNUSTAVATE HOOLDUSSÜSTEEMIDE KIRJELDUS

Tööstusettevõtetes kasutatavate seadmete plaanilise töö tagamisel on oluline roll nende hooldusel ja parandusel. Tootmise mahtude ja keerukuse kasv ning tehnoloogia areng on edendanud ka tööstusseadmete hooldamise ning remondi vahendeid ja meetodikaid. Tänapäevased ettevõtted kasutavad organiseeritud hooldus- ja parandustööde läbiviimisel hooldussüsteemide abi. Hooldussüsteem on seadmete ja meetodite kogum, mis kirjeldab ettevõtte üldist lähenemist seadmete hooldamisele ja parandamisele.

1.1 Hooldussüsteemide tüübid.

Kuna ettevõtte vajadused, võimalused ning hooldust vajavad seadmed on erinevad, on kasutusel ka eri tüüpi hooldussüsteeme. Eristatakse kolme tüüpi süsteeme: vajaduspõhine, ennetav ja ennustav hooldussüsteem. [1]

1.1.1 Vajaduspõhine hooldussüsteem

Seadmeid hooldatakse ja remonditakse kui tekib selle järgi vajadus. Lihtsaim ning algeline hooldussüsteemi tüüp. Hoolduse või remondi vajadus tuvastatakse seadme või selle komponentide anomaalsel käitumisel. Vajaduspõhise hoolduse rakendamiseks piisab, kui oodata mil seade lakkab töötamast, tuvastada töö seisaku põhjus ning seejärel see eemaldada. Vajaduspõhise hooldussüsteemi eelisteks on madalad rakendamiskulud ning võimalus maksimeerida seadme komponentide kasutusiga. Antud süsteemi puhul on oluliseks puuduseks seadme seisukorrast ülevaate puudumine, mis takistab tööseisakute planeerimist ja prognooside koostamist rikete kestuse, hoolduse või remondi maksumuse ja muu taolise kohta.

Vajaduspõhine hooldussüsteem sobib kasutamiseks lihtsamate seadmete ja süsteemide puhul, kus ühe komponendi rike teisi seadme detaile oluliselt ei kahjusta. Antud hooldussüsteemi kasutamine mitmetest komponentidest koosnevate komplekssete seadmete puhul on kulukas ning otstarbetu, kuna ühe komponendi purunemisega võib kaasneda märkimisväärne kahju tervele seadmele. Samuti ei ole vajadusepõhist hooldust otstarbekas kasutada ettevõtetes, kus tähtsustatakse tootmise planeerimist ning soovitakse vältida töö ettearvamatuid seisakuid.

1.1.2 Ennetav hooldussüsteem

Ennetava hooldussüsteemi eesmärgiks on ennetada igasuguseid rikkeid seadmes, teostades hooldustöid kindla ajavahemiku või tööhulga järel. Hooldustööde alla kuulub ka suure tähtsusega komponentide, mille rike võib põhjustada ulatuslikku kahju seadmele või seisakut seadme töös, välja vahetamist. Komponentid vahetatakse välja, vaatamata nende seisukorrale. Kuigi ennetav hooldussüsteem aitab vältida komponentide rikkeid, tõstab ta oluliselt hooldustööde maksumust. Enam ei ole võimalik kasutada iga komponendi kasutusiga maksimaalselt ära ning seetõttu suurenevadki hooldustega kaasnevad kulutused. Ennetavat hooldussüsteemi kasutuselevõtt vajab suuremat investeeringut kui vajadusepõhise hooldussüsteemi kasutuselevõtt, kuna on vaja perioodiliselt hankida uusi osasid vanade välja vahetamiseks. Ennetavat hooldussüsteemis on võimalik kasutada ka tarkvara kuna tekib vajadus hooldustööde planeerimiseks. Taolise hooldussüsteemi eelisteks ongi just võimalus planeerida, hoolduse tõttu tekkivaid, seadmete tööseisakuid. Näiteks on võimalik hooldustöid teostada aegadel millal seadmeid ei kasutata ning nende seisak on kavatsatud. Samuti on ennustava hooldussüsteemi eeliseks eespool mainitud võimalus vältida erakorralisi, komponentide kulumise poolt põhjustatud, seadmete rikkeid, mis võivad tekitada ulatuslikke purustusi seadme teistes komponentidest ning seeläbi tuua endaga kaasa pikki tööseisakuid ja kõrge maksuvusega parandustöid, mis võivad olla kallimad kui uue seadme hankimine. Ennetava hooldussüsteemi puuduseks on aga suurenenud hoolduskulud, sest komponente ei ole enam võimalik maksimaalselt ära kasutada ning seetõttu tekib olukord kus mitmeid välja vahetatud komponente saaks veel kasutada. Just nimelt selle puuduse eemaldamiseks ongi hakatud tänapäeval kasutama ennustavaid hooldussüsteeme.

1.1.3 Ennustav hooldussüsteem

Ennustava hooldussüsteemi eesmärgiks on kindlaks määrata millal mingi komponent rikneb. Selle eesmärgi teostamiseks kasutatakse erinevaid andureid, mis mõõdavad seadme komponentide parameetreid ning juhtseadmeid, kus püütakse mõõdetud andmete põhjal komponendi olekut kindlaks määrata ning selle muutusi tulevikus ette ennustada. Ennustav hooldussüsteem on ainuke kõigist kolmest tänapäeval kasutatavast hooldussüsteemist, mis kasutab lisaseadmeid mingi masina hooldamiseks. Ta ühendab endas vajaduspõhise ja ennetava hooldussüsteemide eelised ilma nendega kaasas käivate puudusteta. Ennustavat hooldussüsteemi kasutades on võimalik maksimaalselt kasutada igat komponenti, vältides samal ajal komponendi rikkest tingitud parandustöid ja seadme töö seisakuid ning ühtlasi omades võimalust planeerida hooldustöid ette ja sooritada neid aegadel, mis võimaldavad seadme tööaja maksimaalset kasutust. Ennustava hooldussüsteemi puuduseks on tema paigalduse kõrge maksumus, hooldussüsteemi enese vajadus hoolduse järele ning süsteemi üldine keerukus. Tema kõrge maksumus on tingitud erinevate andurite, ühendusseadmete, juhtseadmete, tarkvara ning inim-masina liideste järele. Samuti kulub ennustava hooldussüsteemi paigaldamiseks palju rohkem aega kuna selleks, et ennustav hooldussüsteem annaks võimalikult täpseid tulemusi, vajab ta suurel hulgal enne analüüsitud andmeid jälgitavate komponentide kohta. Just suurem hulk üles seadmiseks kuluvat tööd ning vajalikud lisaseadmed on need, mis teevad ennustavast hooldussüsteemist kõigist kolmest olemasolevast hooldussüsteemist kõige kallima. Ühtlasi kaasneb lisaseadmetega, nagu näiteks erinevad andurid, ka vajadus hooldada hooldussüsteemi ennast. Seda kulutust on võimalik minimaliseerida valides keskkonnale ning riskidele vastavalt kõige sobilikumad seadmed. Lisaks peab ennustav hooldussüsteem analüüsima suurel hulgal andmeid, hindamaks mingi komponendi seisukorda ning selle muutusi täpselt. seetõttu on vaja hooldussüsteemis kasutada juhtseadmeid, mis suudavad anduritelt saadud andmeid lugeda ning neid piisava kiirusega analüüsida. Vastavalt andurite tüübile võib tekkida ka vajadus lisa juhtseadmete jaoks, mis teostavad andmetega eeltöötlust enne kui neid saab komponendi seisukorra hindamiseks kasutada.

Erinevate hooldussüsteemi tüüpide omadused on toodud tabelis Tabel 1.1

Tabel 1.1 Erinevad hooldussüsteemi tüübid [2]

Nr.	Omadused	Vajaduspõhine	Ennetav	Ennustav
1	Põhimõte	Töö toimub rikkeni	Perioodiline hooldus	Süsteem jälgib komponendi seisukorda ja hooldab vastavalt
2	Eesmärgid	Kasutada komponendi kasutusiga maksimaalselt.	Planeerida hooldustegevusi kindla intervalliga	Planeerida hooldustööd vastavalt komponendi seisukorrale
		Parandada või vahetada välja komponendid, kui neis ilmneb rike	Parandada või välja vahetada komponendid enne kui ilmneb rike	Parandada või vahetada välja komponendid enne kui ilmneb rike
3	Sobivus	Väikesemõõdulised komponendid või seadmed	Kuluvad seadmed	Seadmed, mille rikke tekkimise protsess on juhuslik või teadmata
		Vähese tähtsusega seadmed	Tarbijakõlblikud seadmed	Kriitilise tähtsusega seadmed
		Komponendid, millel rikke esinemine on väikese tõenäosusega	Seadmed, millel rikke tekkimise protsess on teada	Mittekuluvad seadmed
		Ülearused süsteemid	Tootja poolt soovitatud	Süsteemid, milles võib vigane ennetav hooldus põhjustada rikke.
4	Eelised	Vähene maksuvus	Tasuv paljudes materjalimahukates protsessides	Suurem komponendi tööiga
		Nõuab vähe tööjõudu	Paindlikkus lubab hooldustöid perioodiliselt kohandada	Tööseisakutele kuluva aja langus
			Komponendi elutsükli pikenemine	Varuosadele ning hooldustöödele kuluva kapitali vähenemine
			Energiasäästlik	Suurem tootekvaliteet
			Väiksem rikete esinemise arv seadmetes	Töö- ja keskkonnoahutuse suurenemine
			Hinnanguliselt 12-18% kuluefektiivsem kui vajaduspõhine hooldussüsteem	Energiasäästlik
				Töötajate moraali suurenemine
5	Puudused	Suured tootmiskulud tänu ootamatutele seadmete riketele	Olemas suuri kahjusid tekitavate rikete võimalus	Suured paigalduskulud
		Suured tööjõukulud. eriti kui tekib vajadus ületundidele	Töömahukas	Suured kulutused töötajate koolitamiseks
		Ootamatud paranduskulud või vajadus seadmed välja vahetada	Sisaldab ootamatute hooldustööde sooritamist	Võimalus säästa ei ole selgelt nähtav
		Võimalikud, rikke poolt põhjustatud, kahjud seadmetele	Võimalus tekitada kahjusid komponenditele, sooritades ebavajalikke hooldustöid	
6	Levik	55% tööstusest	31% tööstusest	12% tööstusest
7	Maksuvuse võrrand	Rikke maksuvus =hooldustöö + tööseisak	Ennetava hoolduse maksuvus = hooldustööd + ennetava hooldussüsteemi poolt planeeritud tööseisak	Ennustava hoolduse maksuvus = hooldustööd + ennustava hooldussüsteemi poolt planeeritud tööseisak
8	Näide	Mootorite õlitamine kui suureneb müra või esinevad vibratsioonid	Pumpade õlitus iga 2000 töötunni järel	Pumpade ülevaatuse teostamine ning vajalike hooldustööde tuvastamine

Tabelis 1.1 on toodud kõigi kolme hooldussüsteemi tüübi omadused. Neid vaadeldes on märgata, et ennustav hooldussüsteem ühendab endas vajaduspõhise ja ennetava hooldussüsteemi eelised, kuid ilma nendepoolsete puudusteta. Ennustav hooldussüsteem võimaldab kasutada komponendi kasutusiga maksimaalselt, pakkudes samal ajal võimalust hooldustööde tegemise aega planeerida ning vähendada rikke tekkimise tõenäosust. Peamiseks puuduseks on süsteemi keerukus võrreldes teiste hooldussüsteemidega ning ka paigalduse keerukus ning suurem maksumus, mistõttu ei ole ennustav hooldus väga levinud. Tänapäeval kasutab seda kõigest 12% tööstusest. [2]

Ennustava hooldussüsteemi suur paigalduse kulu tuleneb osadest millest süsteem koosneb. Erinevalt vajaduspõhisest ja ennetavast hooldussüsteemist, kasutab ennustav hooldussüsteem erinevaid seadmeid komponentide seisukorra jälgimiseks ning nendes toimivate muutuste ennustamiseks. Järgnevalt on vaadeldud ennustava hooldussüsteemi osasid ning ehitust

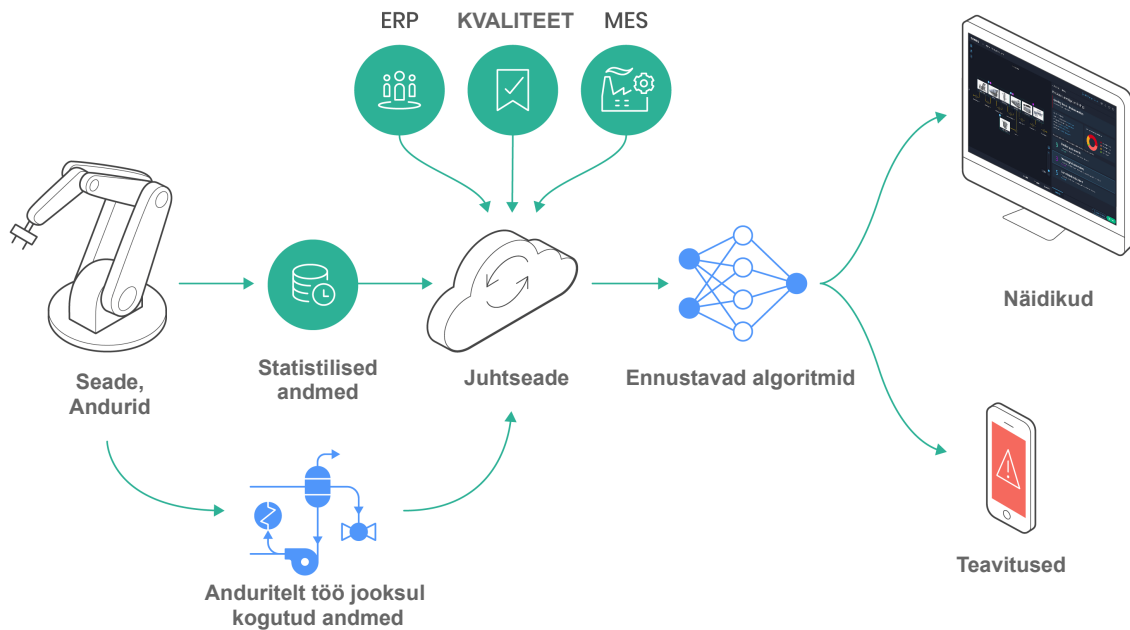
1.2 Hooldussüsteemi osad

Ennustava hooldussüsteemi rakendamiseks leitakse kõigepealt vaadeldava komponendi olulised parameetrid, mida hakatakse jälgima. Levinumateks parameetriteks on temperatuur, rõhk, vibratsioon jne. Vastavalt parameetritele valitakse sobilikud andurid. Anduritelt saadud andmed kogutakse juhtseadmesse kokku. Servaraalimise puhul teostatakse ka andmete analüüs samas juhtseadmes. Teistel juhtudel võidakse kohapeal teha andmete eeltöötlus või saata kogutud andmed serverisse kus toimub nende analüüs ja töötlus.

Ennustava hooldussüsteemi koostamiseks läheb vaja mitmeid erinevaid seadmeid ja tarkvara, millest peamised on järgmised:[3]

1. Andurid, mis mõõdavad erinevaid komponendi,seisukorda iseloomustavaid, parameetreid
2. Andmeside meetodeid jälgitava komponendi ja andmeid koguvate seadmete vahel.
3. Andmeid koguv juhtseade, mis töötleb, salvestab ja analüüsib andmeid.
4. Ennustavad algoritmid, mida kasutatakse komponendil asumatelt anduritelt kogutud andmetes mustrite leidmiseks ja hoiatuste loomiseks kui komponent hakkab liginema oma eluea lõpule.

Ennustava hooldussüsteemi arhitektuur on toodud joonisel Joonis 1.1.



Joonis 1.1 Ennustava hooldusüsteemi arhitektuur. [3]

Andmete kogumiseks komponendilt kasutatakse mitmeid erinevaid andurite tüüpe. Kõige levinumad on temperatuuri, rõhu, vibratsiooni ja ultraheli andurid. Andurite ühendamine andmeid koguva seadmega, sõltub andmevahetuse viisist, milleks iga andur suuteline on.

Anduritelt kogutud andmete säilitamiseks kasutatakse andmebaasi. Andmebaaside struktuur ja arv sõltub hooldussüsteemist endast. Anduritelt kogutud andmeid kasutatakse komponendi seisukorra analüüsimiseks ning selle põhjal seisukorra muutuste prognoosimiseks.

1.2.1 Prognoosimine ja mustrite leidmine

Prognoosimiseks ja mustrite leidmiseks kogutud andmetes kasutatakse mitmeid erinevaid meetodeid. Mõned levinumad meetodid on järgmised: [4]

Regressiooni mudelite abil komponendi kasuliku järelejäänud eluea prognoosimine - Komponenti kasuliku järelejäänud eluea ehk RUL-i (*Remaining Useful Life* ing. kl) arvutamiseks kasutatakse erinevaid regressiooni mudeleid. Mudel jälgib mõõdetavate suuruste ja komponendi seisukorra muutumise omavahelist suhet. Peamised kriteeriumid regressioonmudelite kasutamiseks on: [4]

- a) Komponenti käitumisest on olemas põhjalik andmekogum, mille abil on võimalik jälgida komponendi seisukorra muutust.
- b) Komponenti seisukorra halvenemise protsess on sujuv.
- c) Jälgitakse vaid ühte võimalikku rikke stsenaariumit. Mitme rikke tüübi puhul, milleni jõudmisel käitub komponent erinevalt, on vaja iga tüübi jaoks oma mudel.

Klassifikatsioonmudelite abil komponendi rikke prognoosimine - Teine meetod komponendi rikke ennustamiseks on sarnane regressiooni mudelite meetodiga. Peamiseks erinevuseks on, et klassifikatsioonmudelite korral ei proovita täpselt määrata millal komponendi rike toimub, vaid püütakse teada saada, kas komponendi rike toimub mingis ajavahemikus või mitte. Klassifikatsioonmudeli rakendamise kriteeriumid on sarnased regressioonmudelile. Peamisteks erinevusteks on: [4]

- a) Komponenti seisukorra halvenemise protsess ei pea olema eriti sujuv.
- b) Võimalik jälgida ühe mudeli abil mitut erinevat tüüpi rikke esinemist. Selleks liigitatakse erinevad rikked erinevatesse klassidesse
- c) Mudeli treenimiseks ja testimiseks on olemas piisavas koguses märgistatud andmeid igat tüüpi rikke jaoks.

Komponendi ebatavalise käitumise registreerimine - Antud lähenemine komponendi rikke tuvastamiseks nõuab vähem eelnevalt kogutud ja märgistatud andmete olemasolu, kui regressioon- ja klassifikatsioonmudelid mistõttu on ta levinud lahendustes kus puuduvad taolised andmekogud. Meetod jälgib andurite poolt vaadeldavaid suuruseid ning registreerib mõne parameetri väljumise määratud piiridest ehk komponendi ebanormaalse käitumise. Taoline universaalsus on antud meetodi suurim eelis, kuid ka peamine puudus kuna ebanormaalsele käitumisele ei pruugi alati järgneda komponendi rike. Samuti ei anna meetod teada, millal rike võib toimuda. Peamised kriteeriumid antud meetodi kasutamiseks on: [4]

- a) Puuduvad eelnevad statistilised andmed komponendi ning tema rikete kohta või taolisi andmeid on liiga vähe.
- b) Rikke võimalikke erinevaid tüüpe on väga palju.
- c) Võimalik määrata komponendi parameetrite normaalsed piirkonnad, millest väljumine eelneb rikkele.

Ellujäämismudelid - Ellujäämismudelite eesmärgiks on määrata rikke esinemise tõenäosus mingite komponendi parameetrite hetkeseisu juures. Teisisõnu sisaldab ellujäämismudel endas andmekogu sama tüüpi komponentide käitluste kohta ning kas ja millal erinevad rikked esinesid. Neid andmeid võrreldakse vaadeldava komponendi parameetritega ning leitakse rikke tõenäosus või teekond rikkeni, teades mis juhtus samasid parameetreid omavate komponentidega. Ellujäämismudeleid on võimalik rakendada ka komponendi gruppidele. Peamised kriteeriumid ellujäämismudeli kasutamiseks on:

- a) Andmed samat tüüpi komponentide käitluse kohta ning kas ja millal toimus nendega rike.

Erinevate prognoosimismeetodite võrdlus on toodud tabelis Tabel 1.2.

Tabel 1.2 Ennustavas hooldusüsteemi kasutatavad prognoosimismeetodid

Omadus	Regressiooni mudelid	Klassifikatsiooni mudelid	Ebatavalise käitumise tuvastus	Ellujäämismudelid
Põhimõte	Komponendi järeljäänud kasuliku eluea hindamine	Komponendi seisukorra klassifitseerimine mingis ajavahemikus.	Mõõdetavate suuruste mingis piirkonnas püsimise jälgimine	Komponendi seisukorra muutust võrreldakse teiste sarnaste komponentide seisukorra muutustega
Eesmärk	Rikkeni jäänud aja või tsüklite arvu määramine	Rikke toimumise määramine kindlas ajavahemikus	Komponendi mingi parameetri normaaltalitluse piiridest väljumise tuvastus	Määrata rikke esinemise tõenäosus mingi komponendi parameetrite juures
Kriteeriumid	Komponendi käitumisest põhjalik andmekogum, kus erinevad sündmused on märgitud	Komponendi seisukorra halvenemine ei pea olema sujuv	Puudub vajadus eelnevalt töödeldud andmete järgi komponendi seisukorra muutuste kohta	Võimalikult suur kogus eeltöödeldud andmeid samat tüüpi komponentide käitluse kohta, mis sisaldab andmeid rikete esinemise kohta
	Komponendi seisukorra muutus on sujuv	Võimalik jälgida mitut erinevat tüüpi rikke esinemist ühe mudeliga	Mitmeid erinevaid rikke tüüpe võimalik tuvastada ühe mudeli abil.	
	Üks mudel suudab jälgida ühte muutuse stsenaariumit.	Mudeli treenimiseks vaja eelnevalt töödeldud andmeid iga rikke tüübi kohta	Komponendi seisukord määratud parameetrite normaalpiirkondadena	

Tabelis 1.2 on näha, et peamised kasutatavad prognoosimise meetodid kasutavad, rikke esinemise protsessi ära tundmiseks, eelnevalt kogutud ja ette valmistatud statistilisi andmeid. Iga lähenemine komponendi seisukorra muutuste prognoosimiseks, sobib kasutamiseks erinevate lähteandmete või komponentide puhul ehk kasutatava prognoosimise meetodi valikul võetakse arvesse komponendi rikke esinemise protsesside iseärasusi ning olemasolevaid lähteandmeid.

1.3 Levinud hooldussüsteemid ja nende poolt pakutav funktsionaalsus

Ennustava hooldussüsteemiga tegelev turg on tänaseks päevaks suureks paisunud. Turu suurus 2019 aastal oli hinnanguliselt 3,0 miljardit USD ning kasvab 2024 aastaks oodatavalt 10,7 miljardi USD suuruseks [5]. Turul osalevad enam kui 180 ettevõtet, kes pakuvad erinevaid lahendusi ennustava hooldussüsteemi rakendamiseks. Pakutavateks lahendusteks võivad olla lihtsalt sobilik riistavara, andmeside, -salvestus ja -töötlus meetodid või ennustava hooldussüsteemi jaoks spetsialiseeritud platvormid ning andmete analüüsimiseks kasutatav tarkvara.

Kuna käesoleva uurimistöö eesmärgiks on arendada tarkvararaamistik, mida on võimalik kasutada tööstusliku ennustava hooldussüsteemi loomiseks, on järgnevalt uuritud kolme ettevõtte, IBM-i, Siemens-i ja SAP-i poolt pakutavaid ennustava hooldussüsteemi lahendusi ning on uuritud milliseid funktsionaalsusi nad on pakuvad.

1.3.1 IBM

IBM (*International Business Machines* ing. kl) on Ameerika Ühendriikidest pärit tehnoloogia ja infotehnoloogia valdkonnas tegelev ettevõtte, mille peamised tegevusalad on arvutite riistvara ning tarkvara tootmine ja müük ning erinevad infrastruktuuri, võrgunduse ja konsulteerimisega seonduvate teenuste pakkumine mitmetes valdkondades nagu serverid ja nanotehnoloogia [6]. IBM pakub veel lisaks pilvepõhiseid teenuseid ning masinõppe alaseid teenuseid [6].

IBM loodi 1911. aastal nelja erineva ettevõtte ühinemisel New Yorgis Charles Ranlett Flint-i poolt. Eialgu oli ettevõtte nimeks Computing-Tabulating-Recording Company, kuid 1924. aastal nimetati ta ümber IBM-ks. Kuigi enamuse oma ajaloost on IBM olnud peamiselt arvutustehnika infrastruktuuri pakkuja, liigub ta tänapäeval üha enam pilve ja andmete põhiste teenuste pakkujaks.[7]

IBM on rahvusvaheline ettevõtte, mis tegutseb enam kui 170 riigis ning temas töötab umbes 300 000 töötaja, tehes IBM-st maailma ühe suurima tööandja. [7]

Pakutav tööstusliku ennustava hooldussüsteemi lahendus

IBM pakub omalt poolt ennustava hooldussüsteemi loomiseks platvormi nimega Predictive Maintenance and Quality ehk PMQ [8]. PMQ on lahendus, mis võimaldab ettevõttel ennustada parimat aega hooldustööde läbiviimiseks vaadeldaval seadmel, leida puudujääke protsessis, mis mõjutavad lõpptoote kvaliteeti ning tuvastada seadme rikete algpõhjuseid [8]. Pakutav lahendus kasutab erinevaid tarkvarasid ja tooteid, et koguda, salvestada anduritelt saadud andmeid ning luua mudeleid, mille põhjal kogutud andmeid analüüsitakse. Kogutud andmete abil optimaliseerib platvorm ka hooldustööde aegasid, tootmisprotsesse ning parandab toote kvaliteeti.

Lahenduse komponendid

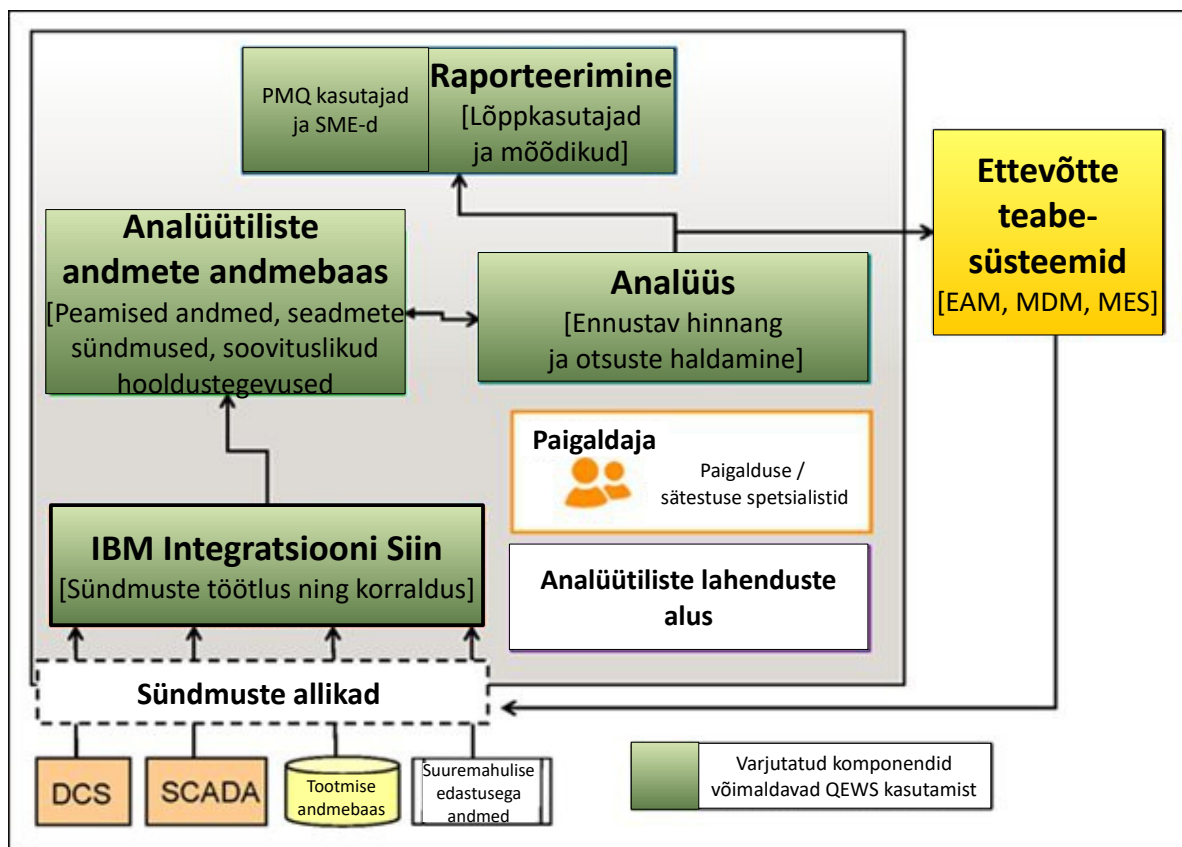
PMQ platvorm kasutab ära asjaolu, et paljud seadmed tänapäeva tootmises, koguvad juba andmeid oma erinevate parameetrite kohta nagu näiteks töötemperatuur, töö kestus, veakoodid jne. Neid andmeid kasutades saab PMQ ennustada, millal võib mingis seadmes tekkida rike. Lahendust on võimalik ka oma kohandada vastavalt oma vajadustele.

IBM-i poolt pakutav ennustava hooldussüsteemi lahendus sisaldab endas järgmisi, kõrge taseme komponente: [8]

- **Allikad** - Siia kuuluvad erinevad allikad, millest saadakse komponendi parameetreid mõõtvate andurite poolt väljastatud andmed. Selliste allikate hulka kuuluvad ka tehase SCADA või DCS süsteem. Allikate hulka kuuluvad ka erinevad andmebaasid, milles säilitatakse ajutisi andmeid ja välised süsteemid, mis saadavad pidevalt süsteemi töö kohta andmeid.
- **Integratsiooni siin** - Integratsiooni siin ühendab erinevad lahenduse sisesed ja välised komponendid omavahel. Siini peamine funktsioon on edastada komponendi parameetreid mõõtvatelt anduritelt saadud andmed analüüsivatesse komponentidesse, milleks on näiteks analüüsiv süsteem või hinnangut andev teenus. Andmete töötluste tulemused kantakse edasi vastavatesse komponentidesse.
- **Analüütiliste andmete andmebaas** - See on lahenduse andmebaas, kus säilitatakse kõik peamised andmed nagu sündmuste loetelud, soovitatud hooldustegevused jne.

- **Analüüsi kiht** - Lahenduse analüüsikihis toimub andmete peamine töötlus, kus püütakse leida erinevaid mustreid ning võimalikke probleeme, mis vajaksid hooldust. Peamisteks meetoditeks on hindamismudelid, mis kasutavad kogutud hooldusandmeid ja peamisi tegevuse indikaatoreid. Hindamismudelite tulemusteks on hinnangud, mille põhjal saab ennustada mingi seadme tulevast seisukorda ning teha soovitusi tema hoolduse kohta.
- **Ettevõtte teabesüsteem** - PMQ lahenduse väljundandmeid on võimalik edastada erinevatesse süsteemidesse nagu EAM, MDMD ja MSM süsteemid, milles oragniseeritakse ettevõtte hooldustöid, hoitakse põhilist teavet ettevõtte seadmete kohta ning mille kaudu juhitakse tehase üldist tööd.
- **Kasutajaliides** - Lahenduse äriteabe funktsionaalsus sisaldab endas mitmeid reaajas olukorrast andmeid jagavaid armatuure ning raporte, mille abil on võimalik vaadelda erinevaid sündmusi, hinnanguid ning soovitusi hoolduse kohta.
- **Kvaliteedi varajane hoiatusüsteem (QEWS)** - Antud komponent teostab tootmiskvaliteedi jägimist ehk tuvastab ebasobilikke muutusi toote kvaliteedis tootmisprotsessi ajal kogutud andmete põhjal ning teavitab nendest ning ühtlasi ka garantii jälgimist, mis jälgib komponentide käitlusindikaatoreid ning teostab garantii analüüsi.
- **Analüüsi lahenduse põhi** - Komponent pakub sobitamise võimalust hooldussüsteemi lahendusele, lisades võimekuse lahendust kohandada vastavalt ettevõtte vajadustele.
- **Paigaldus** - Paigalduse komponent, mis pakub automaatset paigalduse ning lahenduse tarkvara seadistamise võimalust.

Kõik eespool loetletud komponendid on kujutatud joonisel Joonis 1.2.



Joonis 1.2 Predictive Maintenance and Quality kõrgetasemelised komponendid [8]

Lahenduse arhitektuur

Predictive Maintenance and Quality ennustava hooldussüsteemi lahendus kogub mõõtmistulemusi ja teisi andmeid, mis iseloomustavad komponendi seisukorda, analüütiliste andmete andmebaasi. Andmeid kogutakse sinna mitmel erineval moel. Levinumaks meetodiks on .csv dokumentide kujul, mida võimaldab IBM Infosphere Master Data Management rakendus. Teiseks meetodiks on pidev andmete uuendus lahendusevälistelt süsteemidelt nagu näiteks IBM Maximo või mõni teine EAM süsteem. Andmebaasi salvestatakse ka põhilised käitlusindikaatorid ning parameetrite normaalkäitluse piirkonnad.[8]

Erinevad andmed seadmetelt edastatakse lahendusse mitmeid erinevaid protokolle kasutades. Näiteks võib tuua OPC, mida kasutatakse kas ILS DeviceWise või IBM Integrated Information Core adaptriga.[8]

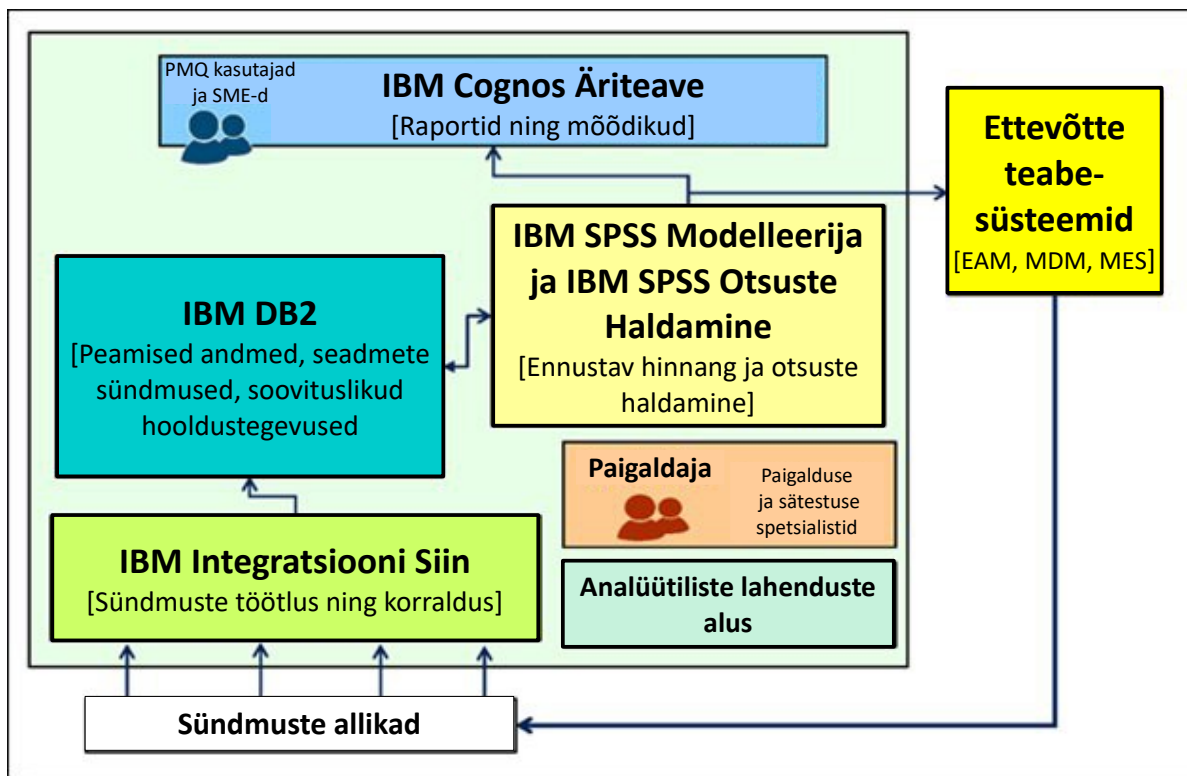
Andmebaas luuakse IBM DB2 tarkvara kasutades. Sinna salvestatud andmeid kasutatakse matemaatiliste mudelite sisendina. Mudelite loomiseks kasutatakse IBM SPSS MODELER tarkvara, mis hindab seadme või mõne selle komponendi rikke tõenäosust ning väljastab vastava hinnangu.

Mudeli väljund viiakse IBM SPSS Decision Management tarkvarasse, mis määrab etteantud kriteeriumite põhjal vajaliku tegevuse. Vajaduse korral väljastab süsteem kohese hooldusteatisse ettevõtte EAM süsteemi. [8]

Andmeid analüütilises andmebaasis on võimalik jälgida mitmel erineval moel, kasutades IBM Cognos Business Intelligence. Selle tarkvara abil saavad kasutajad luua mitmeid erinevaid näidikuid, graafikuid ja raporte, mis annavad ülevaate süsteemi seisukorrast. IBM Integration Bus tarkvara kasutatakse andmevahetuse organiseerimiseks. [8]

Paigalduskoode kasutatakse lahenduse paigaldamiseks tavapärasel topoloogias, kui on võimalik lahendust paigaldada vastavalt ettevõtte vajadustele. [8]

IBM Predictive Maintenance and Quality lahenduse arhitektuur on kujutatud joonisel Joonis 1.3.



Joonis 1.3 IBM-i tööstusliku ennustava hooldussüsteemi lahenduse arhitektuur [8]

1.3.2 SIEMENS AG

Siemens AG on Saksamaal asutatud ettevõtte, mille peamiseks tegevusalaks on elektriseadmete, tööstusautomaatika, elektroonkaseadmete ja meditsiiniliste elektriseadmete tootmine ja arendamine [9]. Siemens AG asutati 1847 aastal Werner von Siemensi ja mehaaniku Johann Georg Halske poolt. Ettevõtte algne nimi oli "Telegraphen-Bauanstalt von Siemens & Halske" [10]. Algusaastatel keskenduti telegraafseadmete tootmisele. Elektrimasinate arenguga hakkas Siemens tootma ka elektrimootoreid ja -generaatoreid. 1866. aastal avastas Werner von Siemens dünamoelektrilise printsiibi ning patenteeris oma leiutise 1967. aastal.[10]

Tänaseks päevaks on Siemensi poolt asutatud, telegraafe tootvast, ettevõttest saanud Euroopa juhtivaid tehnoloogiaettevõtteid, mis tegutseb enam kui 200 riigis üle maailma, andes tööd umbes 385 000 tuhandele inimesele [11]. Siemensi käive 2019. aastal oli umbes 86,8 miljardit eurot.

Pakutav tööstusliku ennustava hooldussüsteemi lahendus

Siemens pakub tööstusliku ennustava hooldussüsteemi loomiseks pilvepõhist IoT operatsioonsüsteemi, mille abil on võimalik erinevate komponentide ning süsteemide omavahelist ühendamist ning mis võimaldab seadmetelt kogutud andmete töötlust, kasutades analüütilist tarkvara. Samuti on Mindsphere avatud tarkvara, mis tähendab, et teda on võimalik vastavalt vajadustele kohandada, omapoolset koodi lisada ning tal on olemas kasutajate ja arendajate kogukond kus kogemusi ja teadmisi jagada. [12]

Mindsphere kogub reaajas andmeid erinevate parameetrite kohta nagu näiteks vibratsioon, temperatuur, tsüklite arv, koormus ja rõhk. Samuti jälgib ta antud parameetrid püsivalt normaalpiirkonnas ning normaalkäitluse korral esinevat parameetrite mustrit. Kasutades reaajas jooksvaid mudeleid, määrab lahendus, kas komponent töötab normaalselt ning kui tekib kõrvalekalle komponendi normaalseisukorrast, leiab lahendus mudelite abil anomaalia põhjuse. Vajaduse korral saadetakse kasutajale teavitust ja lahendus võib soovitada tegevusi anomaalia kõrvaldamiseks. Süsteem võimaldab ka mitme mudeli kasutamist ühe komponendi jälgimiseks [13].

Lahenduse komponendid

Mindsphere for Predictive Maintenance lahendus koosneb järgmistest komponentidest:[13]

- **Mindsphere Ennustav Õpe** - Komponent võimaldab ehitada ennustavaid mudeleid kasutades masinõppe meetodeid, võimaldades seeläbi optimaliseerida toote kvaliteeti. Mudelite ehitamist võimaldab Andmeteaduse Tööpink moodul, mis sisaldab Zeppelin keskkonda mudelite kirjutamiseks. Zeppelin pakub ka erinevaid näiteid. Masinõppe hõlpsamaks kasutamiseks, on Mindsphere-i lisatud ka Tensorflow, Keras, Theano ning MLib.

Mindsphere Ennustava Õppe OOTB (*Out Of The Box, ing. kl*) funktsioonid annavad kasutajale võimaluse suuri andmemahte kiiresti ning lihtsalt jälgida, analüüsida ning töödelda, koodi täiendamata. Taoline funktsionaalsus lihtsustab erinevate kasutajate vahelist koostööd.

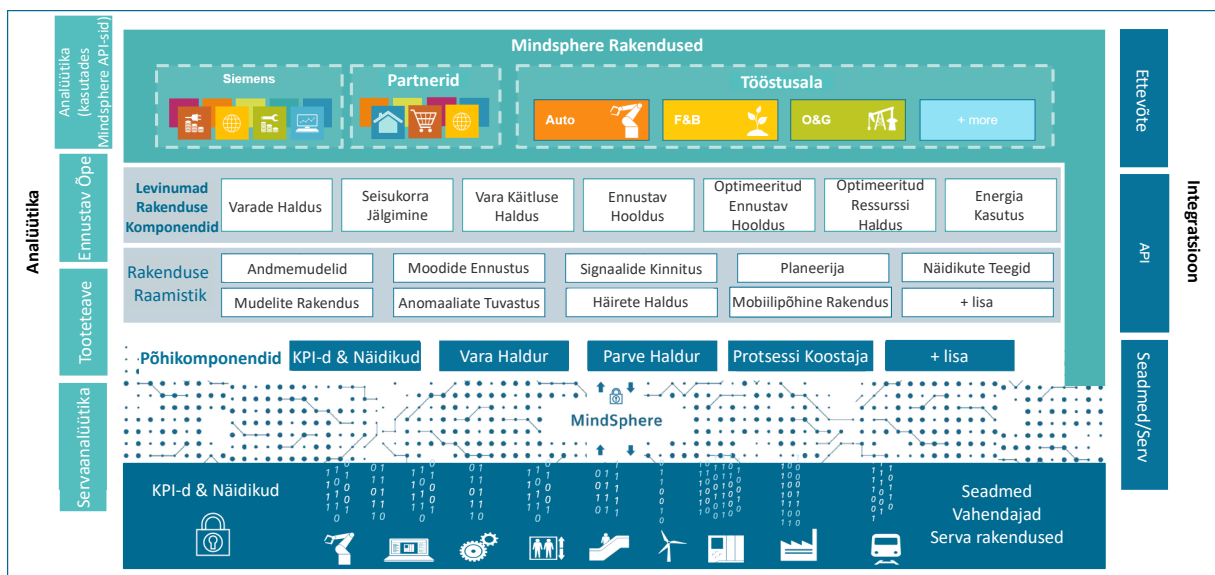
- **Mindsphere Vara Haldur** - Komponent annab võimaluse tööstuslikku protsessi modelleerida. Mudel luuakse erinevate erinevate seadmete digitaalseid teisendeid kasutades. Taoline mudel annab võimaluse hõlpsamalt tuvastada erinevaid, seadmete optimaalseks käitluseks, tähtsaid omadusi.
- **Mindsphere Parve Haldur** - Komponent on mõeldud seadmetest ülevaate andmiseks, Parve Halduris saab määrata iga seadme olulised omadused ning paika panna olukorrad, millest on vaja kasutajat teavitada. Samuti saab määrata andmete visualiseerimise meetodi.

Parve Halduri abil on võimalik luua tootmise 3D mudel ning siduda erinevaid hoiatusi sobilike komponentide parameetritega. Süsteem annab kasutajale koheselt teada kui mõni komponendi parameeter on ületanud lubatud piiri ning ruumiline mudel võimaldab jälgida tekkivaid hoiatusi terves tehases vastavalt nende komponentide asukohale.

- **Mindsphere Analüütilised teenused** - Komponent kujutab endast kogumikku erinevates API-dest, mis võimaldavad sooritada ülesandeid, erinevaid ajatempleid omavate tööstuslike andmetega. Mõned API-d on järgmised:
 - **KPI arvutus** - Võimaldab kasutada andurilt kogutud andmeid, komponendi põhiliste käitlusindikaatorite leidmiseks, mida saab kasutada komponendi seisukorra määramiseks.

- **Anomaaliat tuvastus** - Võimaldab kiirelt tuvastada komponendi ebanormaalset või ootamatut käitumist või mingit protsessi, mis võib viidata komponendi halvenevale seisukorrale.
- **Sündmuste analüüs** - Kasutab statistilist analüüsi, kuvamaks kümme põhilist komponendi riket, mida saab kasutada potentsiaalsete rikete tuvastamiseks. Saab kasutada ka vigade tuvastamiseks sündmuste logis.
- **Signaali arvutus** - Võimaldab leida puudulike andmeid, kasutades erinevaid meetodeid nagu näitkes interpolatsiooni. API koostab ülevaatliku kirjelduse mingi signaali väärtuste loetelust ning vajaduse korral tuletab, olemas olevate väärtuste põhjal, uued väärtused.
- **Signaali kinnitus** - Aitab tuvastada kõige levinumaid probleeme aegreastatud andmete juures. Mõned kasutatavad funktsionid on näiteks piirkonna kontroll, hüppe, sammu, müra ning kalde tuvastus. API-d saab kasutada ka andmete kvaliteedi optimaliseerimiseks.
- **Moodide tuvastus** - Võimaldab ennustada komponendi käitlust, kasutades lineaarset ja polünoomilist regressiooni.

Siemens Mindsphere-i arhitektuur on toodud joonisel Joonis 1.4



Joonis 1.4 Siemens Mindsphere-i arhitektuur [13]

1.3.3 SAP SE

SAP SE on Saksamaalt pärit ettevõtte, mis toodab rakendustarkvara ettevõtetele. Peamisteks tegevusaladeks on masinõpe, asjade internet (IoT) ja erinevad analüütika tarkvarad, mida kasutatakse mitmetel aladel nagu näiteks raamatupidamine, hooldus, ettevõtte administreerimine, planeerimine jne. [14]

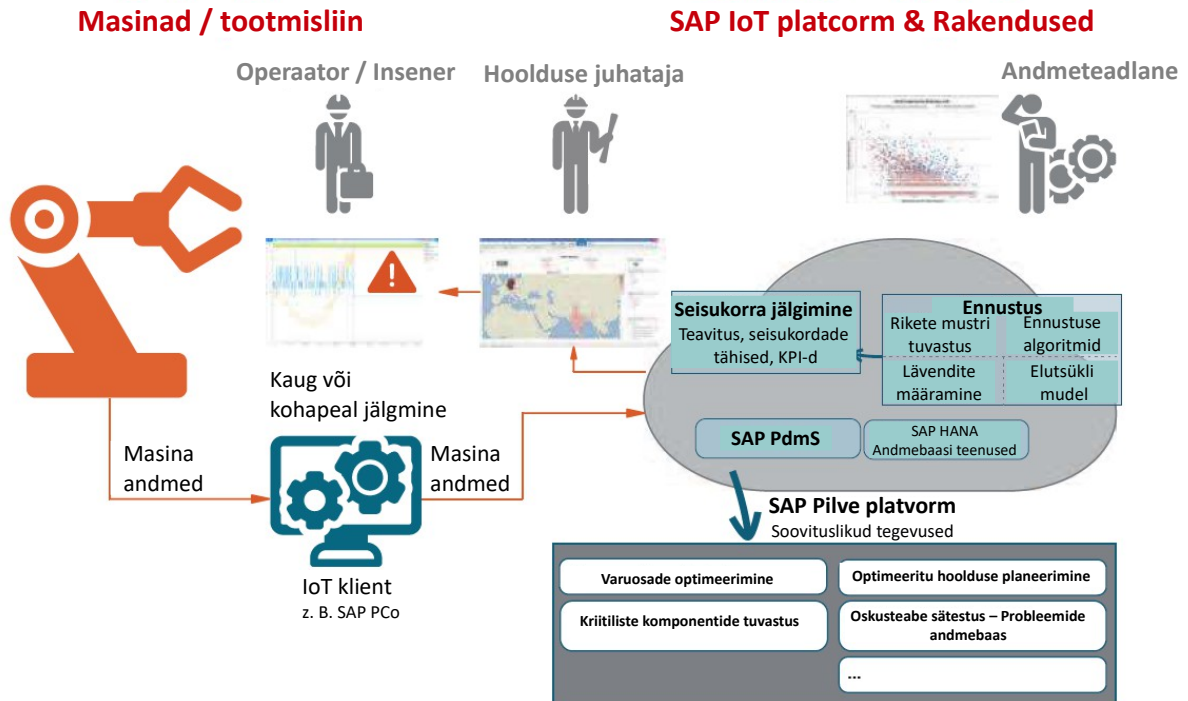
SAP SE asutati 1972. aastal Saksamaal. Asutajateks olid viis endist IBM-i töötajat: Dietmar Hopp, Hasso Plattner, Hans-Werner Hector, Klaus Tschira ja Claus Wellenreuther. Ettevõtte nimeks sai SAP (*SystemAnalyse Programmentwicklung*, saksa kl.). Ettevõtte eesmärgiks oli luua standard ettevõtete tarkvarade loomiseks ja arendamiseks, mis integreeriks erinevad äritegevusega seotud protsessid ning võimaldaks andmetöötlust reaajas.[15]

Tänaseks päevaks on SAP SE-st saanud rahvusvaheline ettevõtte, mis tegutseb enam kui 180, tehes koostööd enam kui 21 000 ettevõttega ning andes tööd umbes 101 150 inimesele. SAP SE käive 2018. aastal oli umbes 24,74 miljardit eurot. [14]

Pakutav tööstusliku ennustava hooldussüsteemi lahendus

SAP SE pakutav tööstusliku ennustava hooldussüsteemi lahenduseks on SAP SE Predictive Maintenance and Service tarkvaralahendus, mis pakub andmete seadmete seisukorra jälgimist, andmete salvestust, andmete analüüsi määramaks sobivaim aeg hoolduseks, varuosade tellimist ning optimaalset planeerimist ja hooldustööde graafikute loomist.[16]

Pakutava ennustava hooldussüsteemi lahenduse põhimõtteskeem on toodud joonisel Joonis 1.5



Joonis 1.5 SAP SE Predictive Maintenance and Service põhimõtteskeem [16]

Lahendus põhineb SAP toodetel, mis võimaldavad erinevate hooldussüsteemide rakendamist. Süsteemi on võimalik kasutada lihtsaks komponendi seisukorra jälgimiseks ning teostada selle põhjal hooldust või on võimalik kasutada lahenduse täielikku võimekust ja luua komponentide seisukordi ennustavaid stsenaariumeid, kasutades ennustamismudeleid ning põhjalikke analüüse.[16]

SAP SE Predictive Maintenance and Service pakub võimalust jälgida eemalt toodetavaid tooteid ning süsteeme, tuvastada seadmete rikkeid, parandada toote kvaliteeti ning seadmete käitlust läbi rikete varajase tuvastuse, vähendada mitteplaneeritud seisakutest tulenevaid kulutusi ning panna alus teenusele põhinevale ärimudelile.[16]

Lahenduse komponendid

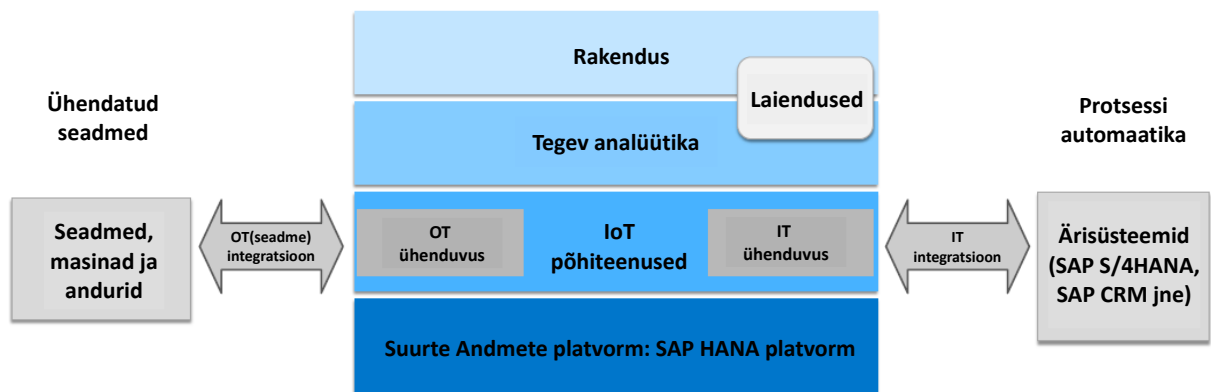
Lahendus koosneb mitmest erinevast SAP SE tarkvaralahendusest. Põhilise platvormina kasutatakse SAP pilve platvormi, mille abil kogutakse ja salvestatakse andurite mõõteandmeid. Kasutades Integrated Control Center-t, on platvorm võimeline seadmeid jälgima ning nende seisukorda analüüsima. [16]

Andmete kogumiseks pakub lahendus SAP Plant Connectivity tarkvara, mis võimaldab erinevaid andmeedastusprotokolle ja ühendamiste meetodeid. Standardprotokollina kasutatakse enamasti OPC UA-d. Andmed salvestatakse SAP HANA andmebaasi. [16]

Andmeid on võimalik analüüsida kahel erineval meetodil: kasutades servaraalimist või SAP pilve. SAP Predictive Maintenance and Service sisaldab endas erinevaid analüüsi meetodeid, mille abil on võimalik koostada ennustav mudel. Võimalik on kasutada tarkvara poolt automaatselt loodud mudeleid või enda poolt loodud süsteemi. Mudelite loomiseks pakub lahendus SAP HANA andmebaasiga kaasas käivat Predictive Analysis Library, mis sisaldab üle seitsmekümne erineva funktsiooni koos mälusisese tehnoloogiaga, võimaldades kiiret rakenduskiirust. Võimalik on kasutada ka SAP Cloud Platform poolt, pilvepõhiseid masinõppe teenuseid nagu näiteks mudeli korraldus, treenimine, hindamine ning anomaaliate tuvastus. [16]

Lisaks andmete kogumisele, salvestamisele ning analüüsile, pakub SAP SE Predictive Maintenance and Service ka võimalust integreerida ennustav hooldussüsteem erinevate SAP moodulitega nagu näiteks ERP ja Plant Maintenance (PM). Läbi taoliste ühenduste on võimalik tuvastada peatne seadme rike ning anda välja hoolduskäsk. Lahendust on võimalik integreerida ka ettevõtte vara haldamise süsteemiga, milleks SAP pakub oma, Asset Intelligence Network (AIN), lahendust ning automatiseerida ka varuosade hankimise toiming. [16]

SAP SE Predictive Maintenance and Service lahenduse arhitektuur on kujutatud joonisel Joonis 1.6



Joonis 1.6 SAP SE Predictive Maintenance and Service tööstusliku ennustava hooldussüsteemi lahenduse arhitektuur [17]

1.4 Ennustavate hooldussüsteemide võrdlus

Käesolevas töös on vaadeldud kolme enim levinumat tööstusliku ennustava hooldussüsteemi lahendust ning ettevõtteid, kes neid pakuvad. Lahenduste vaatlemisel võib märgata, et kõik kolm lahendust pakuvad ühiseid funktsionaalsusi ennustava hooldussüsteemi jaoks. Kõigi kolme lahenduse poolt pakutavad funktsionaalsused on järgmised:

1. Andmete kogumine anduritelt ja seadmetelt.
2. Kogutud andmete salvestamine süsteemis.
3. Kogutud andmete eeltöötlemine või formaadi muutmine ehk analüüsiks andmete sobilikuks muutmine.
4. Erinevate analüüsimudelite olemasolu hooldussüsteemis kasutamiseks või kasutajale oma mudeli loomise võimaluse pakkumine.
5. Analüüsi tulemuste ning kogutud andmete põhjal teatiste, raportite koostamine ning andmete graafiline kujutamine kasutajale.

Eespool kirjeldatud lahenduste pakutavad funktsionaalsused on toodud tabelis Tabel 1.3

Tabel 1.3 Kolme enim levinuma ennustava hooldussüsteemi lahenduste poolt pakutavad funktsionaalsused.

Pakutavad funktsionaalsused	IBM Predictive Maintenance and Quality	SIEMENS Mindsphere	SAP SE Predictive Maintenance and Services
Andmete kogumine anduritelt	Integratsiooni siin aitab erinevaid allikaid süsteemiga ühendada	Vajab lisaseadmeid andurite ühendamiseks	SAP Plant Connectivity tarkvara. Võimaldab erinevaid andmete kogumise meetodeid ja andmeedastuse protokolle
Andmete eeltöötlus	Teostatakse koos andmete analüüsiga	Erinevad rakendused ja API-d nagu näiteks KPI arvutus ning signaali kinnitus ja arvutus.	Teostatakse koos andmete analüüsiga
Andmete salvestus	Analüütiliste andmete andmebaas	Pilvepõhine andmesalvestus	SAP HANA andmebaas
Andmete analüüs	Analüüsi kiht, kus toimub mustrite ja probleemide leidmine	Mindpshere Analüütiliste Teenuste ja Ennustava Õppe poolt pakutavad funktsioonid ja tarkvara	Võimalik kasutada valmislahendust või luua oma mudel
Süsteemi ühildamine	Võimalik andmeid saata ettevõtte süsteemidesse nagu näiteks EAM, MDMD ja MSM süsteemid	Võimalik ühildada teiste süsteemidega	Võimalik ühendada teiste SAP moodulitega
Kasutajaliides	Võimalik koostada armatuure ja raporte, vaadelda sündmusi, hinnanguid ja soovitusi	Võimalik vaadata salvestatud andmeid ja koostada näidikuid	Võimaldab kaug- või kohapeal jälgimist
Süsteemi paigaldus	Eraldi paigalduse komponent	Paigaldustarkvara Mindpshere-i paigalduseks.	Kõik osad vaja eraldi paigaldada
Lisad	Kvaliteedi varajane hoiatussüsteem jälgib toote kvaliteeti ning teavitab ebasobilikest muutustest.	Erinevad rakendused tootmise modelleerimiseks ja seadmete haldamiseks	Võimalik integreerida ettevõtte vara haldamise süsteemiga

Tabelis 1.3 on toodud kolme vaadeldud ennustava hooldussüsteemi lahenduste võrldus. Võrreldud on nende meetodite pakutavaid funktsionaalsusi, milleks on andmete kogumine, eeltöötlus, salvestus ja analüüs, süsteemi ühildamine, kasutajaliides ning süsteemi paigaldamine lahendamiseks. Juurde on lisatud mõned näited erinevatest lisavõimalustest, mida iga lahendus pakub.

2. TÖÖSTUSLIKU ENNUSTAVA TARKVARARAAMISTIKU PROJEKTEERIMINE

2.1 Tarkvararaamistik

Tarkvararaamistik on platvorm, mida kasutatakse rakenduste kirjutamiseks. Raamistik moodustab aluse, millele arendajad saavad mingit konkreetset tüüpi rakendust luua. Tarkvararaamistik võib endas sisaldada eelmääratud klasse, funktsioone, teeke, API-d, kompilaatorit või teisi programme. Eesmärgiks on muuta konkreetse tarkvara arendus lihtsamaks, pakkudes arendajale võimalust vältida üldiste osade kirjutamist, mis aitab eemaldada ka palju koodi kirjutamisel tekkivaid vigasid ning vähendada tarkvara arendamiseks ja vigade eemaldamiseks kuluvat tööaega.[18]

Tarkvararaamistikud on oma olemuselt sarnased teekidele. Peamine erinevus nende kahe vahel on see, kuidas nad kasutaja poolt loodud programmiga suhtlevad ehk kes kelle välja kutsub. Teegid sisaldavad endas, sarnaselt tarkvararaamistikule, erinevaid funktsioone, API-sid ning koodijuppe, mida kasutaja saab kasutada oma rakenduse loomiseks, kuid teegi kutsub välja kasutaja programm. Tarkvararaamistik aga sisaldab endas, lisaks kõigele muule vajalikule, avatud või kasutamata objekte ja funktsioone, mida kasutaja saab oma programmi kirjutamiseks kasutada. Tarkvararaamistiku puhul ei kutsu kasutaja poolt koostatud kood teda välja, vaid tarkvararaamistik kutsub välja kasutaja poolt kirjutatud koodi.[19]

Kokkuvõtlikult võib öelda, et tarkvararaamistik on kogum rakendustest, teekidest ja abstraktsest koodist, mida saab kasutada mingi konkreetse rakenduse loomiseks. Ta on üldistus tarkvarast, millest kasutaja saab koostada endale sobiliku tarkvara.

2.2 Tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku funktsionaalsused

Käesoleva töö esimeses peatükis sai vaadeldud kolme, enim levinud tööstusliku ennustava hooldussüsteemi lahendust ning nende poolt pakutud funktsionaalsusi, mis olid järgmised:

1. Andmete kogumine anduritelt ja seadmetelt.
2. Kogutud andmete salvestamine süsteemis.
3. Kogutud andmete eeltöötlemine või formaadi muutmine ehk analüüsiks andmete sobilikuks muutmine.
4. Erinevate analüüsimudelite olemasolu hooldussüsteemis kasutamsieks või kasutajale oma mudeli loomise võimaluse pakkumine.
5. Analüüsi tulemuste ning kogutud andmete põhjal teatiste, raportite koostamine ning andmete graafiline esitamine kasutajale.

Tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku loomiseks on vaja tagada eespool loetletud funktsionaalsused mingis ulatuses. Kuna loodav lahendus peab olema universaalne, siis on kõige optimaalsem lähenemine luua tarkvararaamistik, mis võimaldab kasutajal endal koostada anduritelt kogumise, töötlemise, analüüsimise ja esitlemise tarkvara, kuna need ennustava hooldussüsteemi osad on spetsiifilised konkreetsele lahendusele. Tarkvararaamistik sisaldab endas osasid, mis on alati ühesugused nagu näiteks andmete salvestus, edastus erinevate hooldussüsteemi osade vahel ning hooldussüsteemi üldine struktuur. Seega on loodava tarkvararaamistiku üheks kriteeriumiks võimekus pakkuda kasutajale võimalust luua sobilikud hooldussüsteemi osad ise. Sellest lähtudes oleks tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku funktsionaalsused järgmised:

1. Võimaldab anduritelt loetud andmed sisestada hooldussüsteemi.
2. Võimaldab hooldussüsteemi andmeid salvestada.
3. Võimaldab sisendandmeid töödelda enne analüüsi.
4. Võimaldab lisada hooldussüsteemi kasutaja enda poolt kirjutatud analüüsikoodi
5. Võimaldab kasutajal väljastada hooldussüsteemi tulemusi ja andmeid kasutajale või teistesse süsteemidesse.

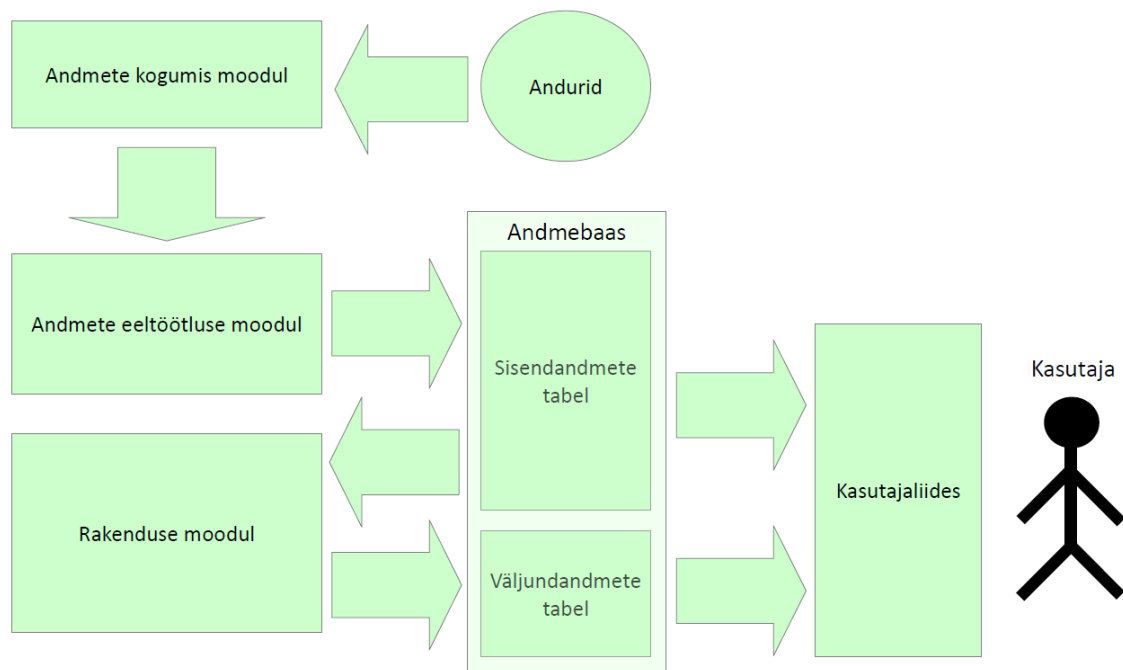
2.3 Universaalse tarkvararaamistiku arhitektuur.

Lähtudes eespool toodud vajalikest funktsionaalsustest on loodav tarkvararaamistik jaotatud järgmisteks mooduliteks:

1. Andmete kogumise moodul - Moodul loeb andurilt andmed, kasutades kasutaja poolt kirjutatud koodi. Andmed salvestatakse moduli siseselt ning edastatakse edasisesse tarkvararaamistiku osadesse.
2. Andmete eelöötluse moodul - Andmete kogumise moodulist loetakse andmed ning teostatakse kasutaja poolt kirjutatud eeltöötlus. Töödeldud andmed salvestatakse.
3. Andmebaasi moodul - Sisaldab endas andmebaasi andmete salvestamiseks hooldussüsteemis ning koodijuppe andmete salvestamiseks ja lugemiseks, mida kasutavad, kasutaja poolt kirjutatud koodid ning teised tarkvararaamistiku moodulid.
4. Rakenduse moodul - Teostab töödeldud andmete põhjal vaadeldava seadme või komponendi seisukorra ennustamist, kasutades selleks kasutaja poolt kirjutatud koodi. Tulemused salvestatakse andmebaasi.
5. Kasutajaliidese moodul - Väljastab kasutaja poolt soovitud andmed andmebaasist vastavasse süsteemi või seadmesse, kasutades selleks kasutaja poolt määratud koodi.

Andmed loetakse andmete kogumise mooduli poolt hooldussüsteemi ning edastatakse eeltöötlesse, kus neid vajadusel töödeldakse. Töödeldud andmed salvestatakse andmebaasis sisendandmete tabelisse. Rakenduse moodul loeb sisendandmete tabelist andmed ning teostab analüüsi, mille tulemus salvestatakse andmebaasi väljundandmete tabelisse. Hooldussüsteemist saab kasutaja andmeid küsida kasutajaliidese mooduli abil, mis loeb andmebaasist andmed ning edastab need soovitud kujul kasutajale.

Andmete liikumine, käesoleva töö käigus loodud tööstusliku ennustava hooldussüsteemi universaalset tarkvararaamistikku kasutavas hooldussüsteemis, on kujutatud joonisel Joonis 2.1.



Joonis 2.1 Andmevoog loodud tarkvararaamistikku kasutavas hooldussüsteemis.

Tarkvararaamistik sisaldab erinevaid osasid, mis jäävad igas paigaldatavas ennustavas hooldussüsteemis alati samaks. Nendeks osadeks on andmete edastus moodulite vahel, andmete formaat ning salvestus süsteemis ja süsteemi üldine struktuur. Hooldussüsteemi osad, mis võivad erinevate hooldussüsteemide vahel erineda nagu näiteks andmete kogumine anduritelt, andmete kuvamine kasutajale ning kasutatav analüütiline mudel, on jäetud iga kasutaja enda poolt lisatavateks osadeks. Kõik kasutaja poolt lisatavad osad tarkvararaamistikule on järgmised:

1. Anduritelt andmete kogumise kood - Annab võimaluse kasutajal kasutada tema poolt soovitud allikaid.
2. Andurite mõõteandmete eeltöötluste kood - Sõltub sellest missuguseid allikaid kasutatakse ning mida soovitakse nendega teha.
3. Andmeid analüüsiv mudel - Annab kasutajale võimaluse kasutada süsteemi vastavalt enese soovidele.
4. Sisend- ja väljundandmete tabelite vormistus andmebaasis - Annab kasutajale võimaluse salvestada andmed sobival kujul.
5. Andmete edastus kasutajale - Annab kasutajale võimaluse sobitada rakenduse kasutajaliides vastavalt soovidele.

Loodud tarkvararaamistik võimaldab hooldussüsteemi arenduseks kuluvat aega vähendada, kuna on paika pandud süsteemi üldine struktuur ning etapid, mida andmed läbivad. Kasutajal jääb üle vaid lisada need osad, mis on spetsiifilised koostatavale süsteemile nagu näiteks andmete kogumise tarkvara, mis sõltub sellest, milliseid andureid kasutatakse ning kuidas nendelt mõõteandmeid kogutakse.

3. TÖÖSTUSLIKU ENNUSTAVA TARKVARARAAMISTIKU

ARENDAAMINE

Arendatava tööstusliku ennustava hooldussüsteemi universaalne tarkvararaamistik on jaotatud mitmeks erinevaks programmiks, mis teostavad tarkvararaamistiku moodulite ülesandeid. Moodulite ülesanded võivad olla jagatud ka mitme erineva programmi vahel. Tarkvararaamistik sisaldab veel lisaks programmidele, ka andmete salvestamiseks kasutatavat andmebaasi ning täiendavat teeki, mida on võimalik hooldussüsteemi erinevates osades kasutada.

Antud tarkvararaamistiku põhjal koostatud hooldussüsteem käivitatakse ühest programmist, mis omakorda käivitab vastavas järjekorras kõik teised, hooldussüsteemi kuuluvad, programmid. Iga hooldussüsteemi moodulite ülesandeid täitev programm, töötab teistest sõltumatult. Programmide vahel saadetavad andmed salvestatakse andmebaasi vastavates tabelites.

Lisaks andmebaasile ning käivitamiseks ja tööks vajalikele programmidele sisaldab tarkvararaamistik veel täiendavat teeki, mida kõik hooldussüsteemi programmid saavad kasutada ning süsteemi seadistamisel kasutatavat programmi, mida saab kasutada süsteemi seadistamiseks vajamineva aja vähendamiseks.

3.1 Osad

Käesoleva töö käigus loodud tööstusliku ennustava hooldussüsteemi universaalne tarkvararaamistik koosneb järgmisest osadest:

1. **Eelandmed** - Kood, mis sooritab andmete kogumise ja eeltöötluse moodulite ülesandeid. Andmete kogumiseks lisab kasutaja vajalikud funktsioonid. Kood teostab andmete kogumist kasutaja poolt määratud perioodi järel ning muudab kogutud andmed analüüsiks sobilikumaks. Eeltöötluse meetodi lisab kasutaja. Töödeldud andmed salvestatakse andmebaasi.
2. **Andmebaas** - Teostab osaliselt andmebaasi mooduli ülesandeid. Kasutatakse süsteemi sisend- ja väljundandmete salvestamiseks. Andmebaasi kasutavad eelandmete programm, rakendusmudel ning kasutajaliides. Andmebaas luuakse eraldi seadistuse programmiga.
3. **Rakendusmudel** - Teostab rakenduse mooduli ülesandeid. Süsteemi peamine osa, mis analüüsib salvestatud sisendandmeid ning püüab ennustada tulevasi muutusi komponendi seisukorras. Mudeli väljundtulemus salvestatakse andmebaasis asuvasse väljundandmete tabelisse.
4. **Kasutajaliides** - Teostab kasutajaliidese mooduli ülesandeid. Edastab kasutajale soovitud andmed andmebaasi sisend- ja väljundandmete tabelitest.
5. **Pea** - Hooldussüsteemi osa, mida kasutatakse teiste osade käivitamiseks.
6. **Andmebaas moodul** - Teostab tarkvararaamistiku andmebaasi mooduli ülesandeid. Sisaldab funktsioone andmebaasi ja tabelite loomiseks ja vormistuseks ning andmete kirjutamiseks ja lugemiseks andmebaasist. andmebaasiga suhtlemise moodul on lisatud teistesse hooldussüsteemi osadesse.
7. **Seadistus** - Hooldussüsteemi osa, mida kasutatakse süsteemi paigaldusel. Loob andmebaasi ning vajalikud tabelid. Teostab osaliselt andmebaasi mooduli ülesandeid, kuid võib lisada ka teisi, süsteemi paigaldusel vajalikke, seadistusi.

3.2 Kasutatud vahendid

Tööstusliku ennustava hooldussüsteemi universaalse tarkvararaamistiku arendamise käigus on kasutatud kahte vahendit: sobiliku andmebaasi tarkvara ning programmeerimiskeelt, milles kõik tarkvararaamistiku osad on kirjutatud.

Andmebaasi tarkvaraks on valitud SQLite. SQLite on C baasil koostatud teek, mis võimaldab rakendada väikesemõõdulist, kiiret ja töökindlat SQL andmebaasi mootorit. SQLite-i on võimalik kasutada mitmesugustes erinevates masinates ning platvormidel. Käesolevas töös on SQLite-i kasutatud just nimelt tänu tema väikesemõõdulisusele, töökindlusele ja universaalsusele

Kõik tarkvararaamistiku koodid on kirjutatud, kasutades Python programmeerimiskeelt. Kasutatud on versiooni 3.7.3. Antud keel on valitud peamiselt kahel põhjusel. Esimeseks põhjuseks on asjaolu, et paljud seadmete tootjad pakuvad valmis kirjutatud mooduleid Pythonis. Teiseks põhjuseks on kasutaja poolt kirjutatud koodide lisamise lihtsus olemasolevatesse programmidesse. Nende omaduste tõttu, on Pythonis kirjutatud tarkvararaamistiku abil võimalik hõlpsalt koostada vajalik süsteem.

4. TÖÖSTUSLIKU ENNUSTAVA HOOLDUSSÜSTEEMI KATSETAMINE

4.1 Tarkvararaamistiku katsetamise tingimused

Käesoleva töö käigus loodud tööstusliku ennustava hooldussüsteemi universaalset tarkvararaamistikku on katsetatud koostades antud tarkvararaamistiku põhjal hooldussüsteemi ning hinnates kas seda kasutades on võimalik koostada ennustav hooldussüsteem. Hooldussüsteem on loodud komponendi temperatuuri, rõhku ja vibratsiooni jälgiva prototüübi jaoks, mille eesmärk on jälgida komponendi seisukorda ning, hooldussüsteemi tarkvara abil püüda ette ennustada, millal võib esineda rike.

4.2 Prototüübi kirjeldus

4.2.1 Prototüübi üldine kirjeldus

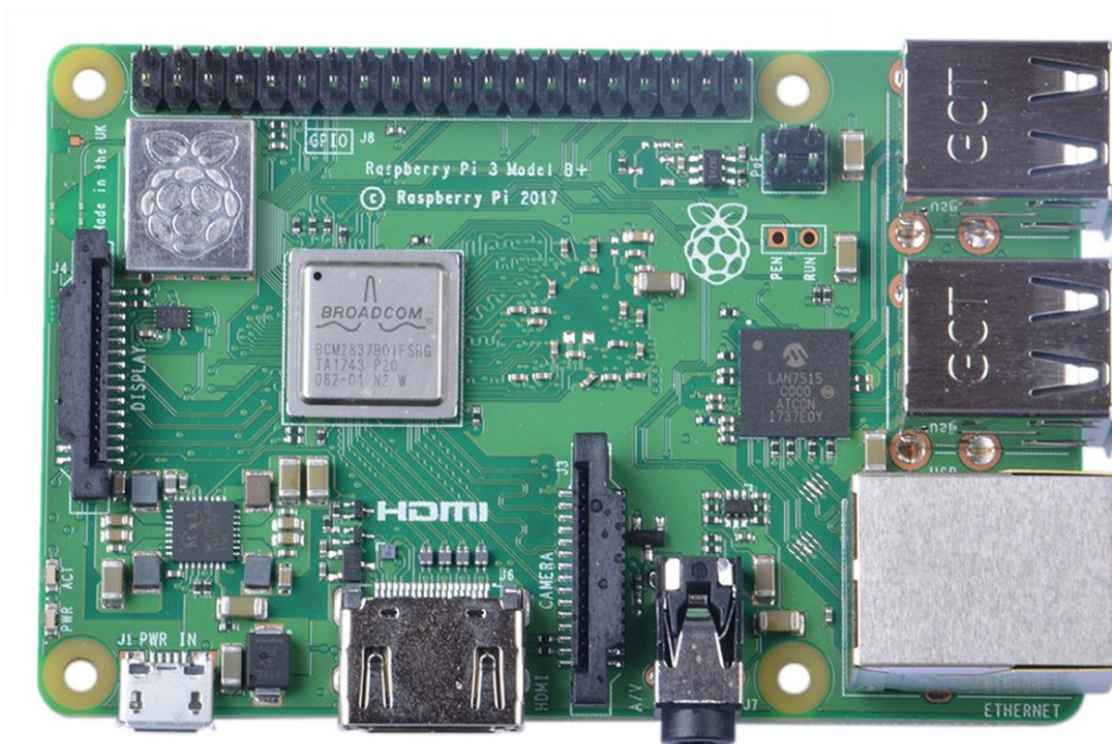
Prototüüp kasutab rõhu-, temperatuuri- ja vibratsioonandureid komponendi seisukorra määramiseks. Mõõteajaks on kümme sekundit, mille vältel loetakse ühe korra kõigi rõhu- ja temperatuurianturite väljundid. Vibratsioonianturite väljundeid loetakse mõõteperioodi kestel pidevalt, kiindla sagedusega, ning väärtused salvestatakse loggerites kust juhtseade nad alla laeb. Mõõdetud tulemused töödeldakse juhtseadmes ning hooldussüsteemi väljundid edastatakse kesksesse serverisse. Juhtseadmes asub ka hooldussüsteemi tarkvara.

Juhtseade kogub andmeid anduritelt. Andmeid kogutakse andurite lisamoodulitelt nagu ADC muundurid, takistuse mõõteplaadid ning loggerid, kasutades igat tüüpi anduri jaoks erinevaid andmesiine. Rõhuandurite mõõtetulemuste edastamiseks juhtseadmesse kasutatakse I2C andmesiini ning temperatuurianturite näitude jaoks kasutatakse SPI andmesiini. Vibratsioonianturite väljundeid loevad andmeloggerid, mis on ühendatud juhtseadmega, prototüübi sisesesse LAN võrku, mille kaudu edastatakse salvestatud andmed juhtseadmesse. Prototüübi sisest LAN võrku on võimalik ühendada ka välise arvutivõrguga, andes süsteemile võimaluse edastada hooldussüsteemi tarkvara tulemused ning anduritelt kogutud andmed, välisesse serverisse.

4.2.2 Prototüübi komponendid

Prototüüp koosneb erinevatest anduritest, nendega seotud lisamoodulitest, juhtseadmest ning seadme sisese LAN loomiseks vajalikest võrguseadmetest. Komponentide loetelu koos kirjeldustega on järgmine:

Juhtseade - Prototüübi keskne seade, mis juhib süsteemi tööd. Juhtseade kogub kõikidelt andurite moodulitelt mõõtetulemusi. Juhtseadmest asub ka ennustava hooldussüsteemi tarkvara, mis kogutud andmeid töötleb. Juhtseadmeks kasutatakse plaatarvutit Raspberry Pi mudelit B+.



Joonis 4.1 Raspberry Pi 3 mudel B+ [20]

Vibratsiooniandur - Süsteem kasutab komponendi keres esinevate vibratsioonide mõõtmiseks kahte kolmeteljelist vibratsiooniandurit, mis kasutavad vibratsioonide mõõtmiseks kõigis kolmes teljes, sobilikke kiirendusmuundureid. Andurite väljundid salvestatakse andmeloggeritesse. Vibratsioonianduriteks kasutatakse A/130/V.500 andureid, mida toodab DJB Instruments.



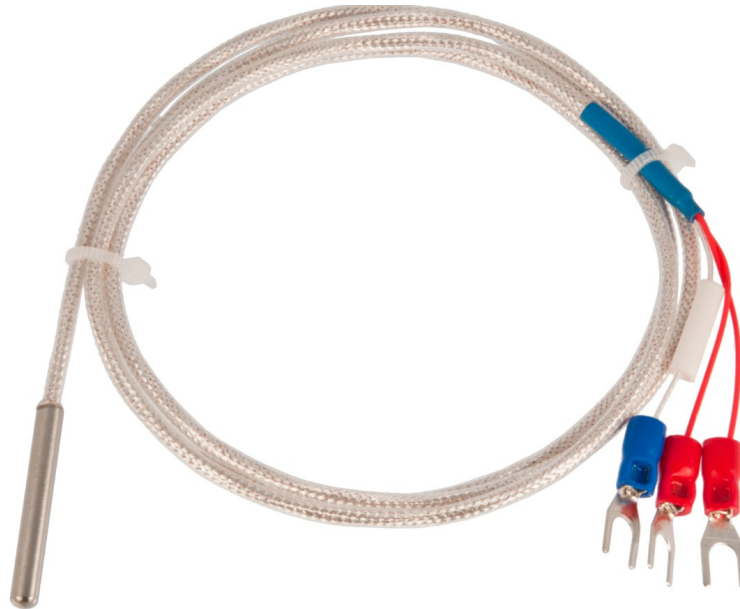
Joonis 4.2 A/130/V.500 kiirendusandur[21]

Andmelogger - Tänu vibratsiooni kui nähtuse omadustele ei ole võimalik saada teavet vibratsiooni kohta komponendis ainult teda mõõtvate andurite hetkenäitude pealt nagu on võimalik temperatuuri ja rõhuga. Ülevaatlike andmete saamiseks tuleb mõõta komponendis esinevaid vibratsioone mingi aja vältel võimalikult hea sagedusega ning seejärel, saadud andmeid töödeldes, leida vibratsiooni iseloomustavad suurused nagu näiteks suurim sagedus ning selle amplituud. Töötlemata andmete lugemiseks ja salvestamiseks kasutatakse prototüübis andmeloggereid, mis loevad andmeid ühe mõõteperioodi vältel ning salvestavad saadud andmed .wdd laiendiga binaarfaili, mis mõõteperioodi lõpus saadetakse juhtseadmesse. Andmeloggeriteks kasutatakse Raspberry Pi põhiseid WebDAQ 504 andmeloggereid.



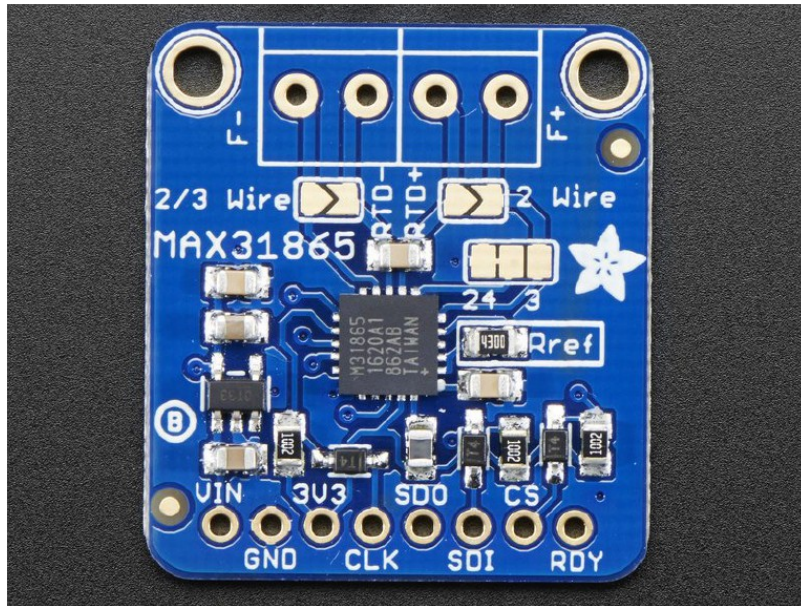
Joonis 4.3 WebDAQ 504 andmelogger [22]

Temperatuuriandur - Prototüüp kasutab komponendi temperatuuri mõõtmiseks kahteteist RTD andurit, mis paigutatakse sobilikesse kohtadesse komponendi kerel, andes seeläbi täpsema pildid komponendi kere erinevate kohtade koormuse kohta. Temperatuurianduriteks kasutatakse PT100 termotakisteid.



Joonis 4.4 RTD PT100 temperatuuriandur [23]

Temperatuurianduri muundur - RTD andurid ei oma konkreetset väljundit kuna nad ei koosne rohkemast kui takistusest, mille väärtuse sõltuvus temperatuurist on kindlalt teada ja sobilike omadustega. Anduri takistust on vaja lugeda ning seejärel, sellele vastav temperatuuri väärtus leida. Seda ülesannet sooritab temperatuurianduri muundur, mille sisendiks on, temperatuurianduriks olev takisti, ning mis edastab saadud temperatuuri väärtuse juhtseadmele. Andmete edastamiseks kasutab muundur SPI andmesiini. Temperatuurianduri muunduriteks kasutatakse prototüübis Adafruit MAX31865 muundurit.



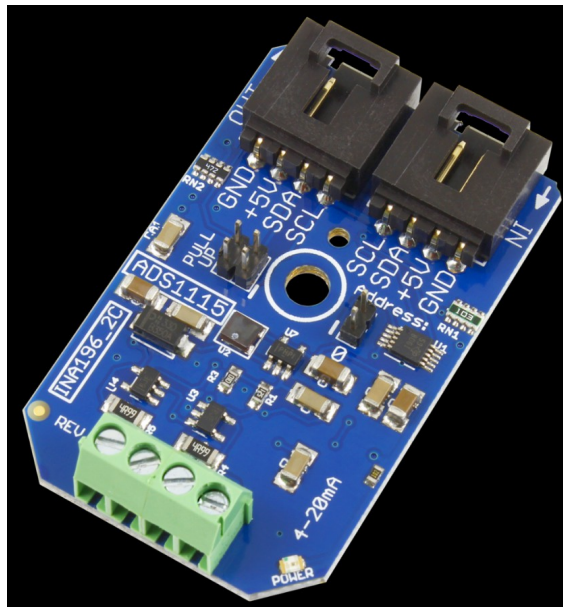
Joonis 4.5 Adafruit MAX31865 [24]

Rõhuandur - Prototüüp kasutab rõhu mõõtmiseks kahte rõhuandurit komponendi sisese rõhu mõõtmiseks. Andurite väljundsignaal on 4-20mA, mis tagab piisava töökindluse. Väljundsignaal muudetakse, enne juhtsignaali saatmist ADC muunduris. Rõhuanduriteks kasutatakse Gems sensors-i poolt toodetud 3500B02B5G05E000 rõhuandureid.



Joonis 4.6 Gems 3500B02B5G05E000 rõhuandur. [25]

Rõhuanduri ADC muundur - Enne hooldussüsteemis analüüsimist on vaja rõhuanduri väljundsignaal rõhu väärtuseks teisendada. Selleks on rõhuanduri väljund ühendatud ADC muunduriga, mis muudab anduri 4-20mA väljundi digitaalseks väärtuseks, mis saadetakse juhtseadmesse, kus see rõhu väärtuseks teisendatakse. Andmed edastatakse muundurist juhtseadmesse I2C andmesiini kasutades. Rõhuanduri väljundi ADC muunduriks kasutatakse Adafruit ADS1115 ADC muundurit.



Joonis 4.7 Adafruit ADS1115 ADC muundur [26]

Ruuter - Kuna vibratsiooniandurite andmed edastatakse juhtseadmesse Ethernet kaablit kasutades ning samuti peab olema võimalus prototüüpi ühendada arvutivõrku, on kõige optimaalsem lähenemine, luua prototüübis LAN võrk, mille kaudu saavad andmeloggerid saata andmeid juhtseadmesse, kuid mida saab ühendada ka välise seadmetega, andes süsteemile võimaluse saata andmeid teise süsteemi. Selle lahenduse jaoks on prototüüpi asetatud ruuter, mille abil on andmeloggerid ja juhtseade omavahel ühendatud ning mis pakub võimalust ühendada prototüüp mingi välise arvutivõrguga. Ruuteriks kasutatakse Teltonika RUT955 ruuterit.



Joonis 4.8 Teltonika RUT955 ruuter [27]

4.3 Hooldussüsteemi kirjeldus

Koostatud hooldussüsteemi eesmärk on koguda juhtseadmes anduritelt mõõteandmeid ning nende põhjal hinnata komponendi seisukorda, püüdes ette näha selle muutuseid. Hooldussüsteem on ehitatud, kasutades käesoleva töö käigus loodud universaalset tarkvararaamistikku, mistõttu on tal samasugune struktuur.

Hooldussüsteem kogub rõhu-, temperatuuri- ja vibratsioonanduritelt iga mõõteperioodi, milleks on määratud kümme sekundit, lõpus mõõteandmed. Vibratsioonandmete puhul on vajalik eeltöötlus. Töödeldud andmed salvestatakse andmebaasis sisendandmete tabelis kust hooldussüsteem loeb need rakendusmudelisse, mis on kasutaja lisatud. Mudeli tulemus salvestatakse väljundandmete tabelis, kust neid loeb kasutajaliides. Hooldussüsteemi koostamiseks on tarkvararaamistikule lisatud anduritelt andmete lugemise funktsioonid, kogutud andmete eeltötluse kood, rakendusmudel, kasutajaliides ning andmebaasis asuvate tabelite vormistus.

4.3.1 Hooldussüsteemi rakenduse osad

Anduritelt andmete kogumine

Igat tüüpi anduritelt kogutakse andmeid erinevate lisamoodulite ning meetodite abil, mistõttu on igale andurite tüübile koostatud eraldi moodul, mis sisaldab andmete lugemiseks koostatud funktsioone. Neid kasutatakse andmete lugemiseks tarkvararaamistiku andmete kogumise moodulis.

Rõhuanduritelt andmete kogumiseks kasutatakse tootja poolt pakutud lisamoodulit Adafruit_ADS1x15, mille abil on võimalik lugeda muunduri väljundid. Loetud väljundväärtused teisendatakse funktsioonis vastavaks rõhuväärtusteks, mis salvestatakse kahekohalises maatriksis. Kõik andmed kogutakse ühe funktsiooni sees, mis tagastab, andmeid sisaldava maatriksi, funktsiooni välja kutsunud programmi.

Temperatuuriandurite mõõteväärtuste lugemiseks kasutatakse mitut erinevat Adafruit-i poolt pakutavat lisamoodulit:

1. busio - Sisaldab erinevate andmesiinide kasutamiseks mõeldud klasse.
2. board - Sisaldab erinevate platvormide jalgade aadresse. Lihtsustab koodi kirjutamist kuna kaob vajadus uurida vaja minevate jalgade aadresse kasutataval platvormil.
3. digitalio - Võimaldab ligipääsu digitaalsetele sisenditele ja väljunditele.
4. adafruit_max31865 - sisaldab kasutatava muunduri kasutamiseks vajalikke funktsioone ja klasse.

Kuna temperatuuriandureid on kokku kaksteist, siis on nende käest andmete küsimine asetatud tsüklisse, mis salvestab muunduritest saadud temperatuuri väärtused kaheteistkohalisesse maatriksisse. Kõik andmed kogutakse ühe funktsiooni sees, mis tagastab välja kutsunud programmi andmeid sisaldava maatriksi.

Vibratsioonandurite andmed salvestatakse algul andmeloggeris asuvasse binaarfaili. Andmeid kogutakse terve mõõteperioodi kestel. Mõõteperioodi lõpus laeb hooldussüsteem, andmeid sisaldava dokumendi juhtseadmesse, kasutades selleks FTP ühenduse võimalust andmeloggeriga. FTP kasutamiseks on vastavaid funktsioone sisaldavasse koodi lisatud, Pythoni arendajate pool pakutav, lisamoodul ftplib. Dokumendi allalaadimise järel, loetakse sealt vajalikud andmed, mis salvestatakse kahemõõtmelisse maatriksisse, mille iga tulp vastab ühele mõõdetud teljele.

Andmete lugemiseks dokumendist, kasutatakse Pythoni arendajate poolt pakutavat lisamoodulit struct, mis võimaldab teisendada andmeid C keeles kirjutatud struktuuridest Pythonis asuvateks, baidi nime kandvateks väärtusteks. Mõlemast andurist loetakse andmed eraldi funktsiooniga, mis tagastavad välja kutsunud programmi, mõõteandmeid sisaldava kahemõõtmelise maatriksi.

Andmete eeltöötlus

Temperatuuri- ja rõhuanduritelt kogutud andmed ei vaja enne kasutamist töötlust mistõttu neid ka ei töödelda. Vibratsioonanduritelt kogutud andmed kujutavad endast aga suurt hulka kiirenduste hetkeväärtusi, mis ei oma mingit väärtust, mille põhjal saaks hinnata komponendi seisukorda. Selleks on vaja neid andmeid töödelda. Eesmärgiks on andmete taandamine paariks muutujaks, mis iseloomustavad kõige selgemalt komponendi vibratsioone. Laialt levinud meetodiks on kiire Fourier teisendus (FFT), mille abil viiakse kogutud andmed ajapõhisest ruumist sageduspõhisesse, kus on võimalik tuvastada suurima amplituudiga sagedus ning selle amplituudi suurus. Neid kahte väärtust kasutades on võimalik hinnata komponendi läbivaid vibratsioone, mistõttu on need eeltöötlemise tulemuseks.

Rakendusmudel

Käesoleva töö peatükis 1.2.1 loetletud erinevatest prognoosimismeetoditest sobib, koostatud hooldussüsteemi, komponendi ebatavalise käitumise registreerimine, kuna puudub varasemalt kogutud statistilised andmed, milles oleks komponendi seisukorra muutused jälgitavad ja rikked ära märgitud. Koostatud rakendusmudel jälgib kogutud andmeid ning annab teada kui need väljuvad normaaltalitluse piirkondadest, mis võib viidata komponendi seisukorra halvenemise ning peatse rikke tekkimiseni. Aja möödudes kui hooldussüsteem on kogunud piisava koguse andmeid jälgitava komponendi kohta, on võimalik kasutada täpsemaid mudeleid.

Rakendusmudeli väljundiks on hoiatusteaded, mis sisaldab normaaltalitluse piirkonna ületanud anduri nime. Normaalolukorras on rakendusmudeli väljundiks vastav teade "KOMP_OK".

Tabelite vormistus andmebaasis

Sisendandmete tabelisse sisetatavatele andmetele lisatakse juurde järjekorra number ja ajatempel, mis võimaldavad näha millal andmed on kogutud ning võimaldavad hõlpsalt tabelist sobilikke andmeid leida. Mõõteperioodi lõpus sisestatakse eraldi lahtritesse igalt rõhuandurilt saadud rõhu väärtus, temperatuurianduritelt saadud temperatuuri väärtus ning iga vibratsioonianduri telje mõõdetud suurima amplituudiga sageduse väärtus ning selle amplituut. Kokku sisaldab sisendandmete tabel kahtekümnet kaheksat tulpa ning iga tulba nime sisaldavat päist.

Sisendandmete tabeli vormistus on toodud joonisel Joonis 4.9.

Nr.	Aeg	Vibratsiooniandur_1						Vibratsiooniandur_2						Temperatuuriandurid												Rõhuandurid					
		X_f	X_a	Y_f	Y_a	Z_f	Z_a	X_f	X_a	Y_f	Y_a	Z_f	Z_a	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	P1	P2				
1																															
2																															
3																															
...																															
n																															

Joonis 4.9 Sisendandmete tabeli vormistus

Väljundandmete tabelisse salvestatakse ainult rakendusmodeli väljund. Sarnaselt sisendandmete tabelile lisatakse igale reale järjekorranumber ning aeg, millal andmed tabelisse salvestati. Kokku sisaldab väljundandmete tabel kolme tulpa ning iga tulba nime sisaldavat päist.

Väljundandmete tabeli vormistus on toodud joonisel 4.10

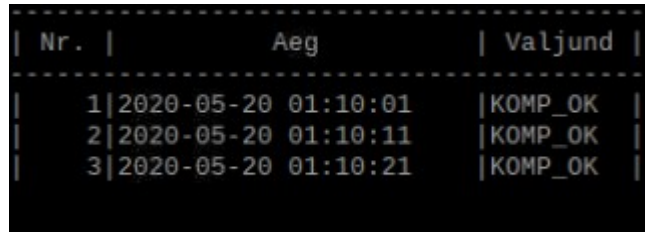
Nr.	Aeg	Rakendusmodeli väljund
1		
2		
3		
...		
n		

Joonis 4.10 Väljundandmete tabeli vormistus

Kasutajaliides

Koostatud hooldussüsteemi kasutajaliides kuvab juhtseadme ekraanile tabeli, milles on toodud väljundandmete tabelis olevad andmed, mis sisaldavad rea numbrit, rakendusmodeli väljundit ja ajahetke, millal mõõteperiood lõppes. Kasutajaliidese läbi on kasutajal võimalik jälgida komponendi seisukorda, andes ülevaate hooldussüsteemi tööst.

Kasutajaliidese väljund on toodud joonisel Joonis 4.11



Nr.	Aeg	Valjund
1	2020-05-20 01:10:01	KOMP_OK
2	2020-05-20 01:10:11	KOMP_OK
3	2020-05-20 01:10:21	KOMP_OK

Joonis 4.11 Kasutajaliidese väljund.

4.4 Hinnang tarkvararaamistikule

Käesoleva töö käigus loodud tarkvararaamistik täitis talle püstitatud eesmärgi lihtsustada anduritelt näitude kogumist, mõõteandmete töötlemist, masinõppe algoritmide rakendamist ning andmeedastuse teostamist. Tarkvararaamistiku kasutamine aitas eemaldada enamus süsteemi paigaldamiseks vajaminevast tööst, mis ei olnud spetsiifiline kasutatavatele seadmetele nagu näiteks andmete edastus erinevate hooldussüsteemi osade vahel, andmebaasi valik ning paigaldus. Universaalse tarkvararaamistiku abil loodud ennustav hooldussüsteem oli võimeline koguma andmeid, teostama eeltötlust, andmeid salvestama ning analüüsimeetodi abil hindama komponendi seisukorda. Samuti tuvastama selle muutusi. Koostatud hooldussüsteem oli ka võimeline edastama kasutajale soovitud andmed.

KOKKUVÕTE

Käesoleva bakalaureusetöö eesmärgiks oli edendada tööstuses kasutatavate ennustavate hooldussüsteemide arendust läbi universaalse tarkvararaamistiku, mis lihtsustab anduritelt näitude kogumist, mõõteandmete töötlemist, masinõppe algoritmide rakendamist ning andmeedastuse teostamist.

Töö käigus uuriti enam levinud tööstuslike ennustavate hooldussüsteemide lahendusi ning nende poolt pakutud funktsionaalsuseid. Nende põhjal koostati universaalse tarkvararaamistiku poolt pakutavad funktsionaalsused ning projekteeriti tööstusliku ennustava hooldussüsteemi tarkvararaamistik, mille toimimist katsetati, luues selle põhjal ennustav hooldussüsteem. Katse tulemusena leiti, et koostatud tarkvararaamistik täitis temale seatud eesmärgi.

Töö tulemuseks on tööstusliku ennustava hooldussüsteemi universaalne tarkvararaamistik, mis lihtsustab anduritelt loetud andmete sisestamist hooldussüsteemi, nende andmete salvestamist ja töötlemist enne analüüsi, võimaldab kasutada kasutaja enda poolt valitud andmete analüüsi meetodit ning väljastada hooldussüsteemi tulemusi ja andmeid teistesse süsteemidesse.

Töö käigus loodud tarkvararaamistik annab kasutajale põhilise ennustava hooldussüsteemi struktuuri ning vajaliku tarkvara, mida ta saab ise edasi arendada, lisades sellele enda poolt kirjutatud tarkvara.

Tulevikus annaks loodud tarkvararaamistikku muuta kasutajasõbralikumaks lisades sellele graafilise liidese, mille läbi ennustavat hooldussüsteemi koostada. Samuti võiks lisada olemasolevale veel, enam levinud andurite kasutamiseks mõeldud liideseid, standartseid analüüsimumudeleid, mida annab vastavalt vajadusele kohandada, tarkvara andmete graafiliseks esituseks, meetodi andmete graafiliseks kujutamiseks kasutajale jpm.

Autori enda hinnangul pakub koostatud tööstusliku ennustava hooldussüsteemi universaalne tarkvararaamistik võimalust paigaldada ennustav hooldussüsteem ilma põhjalike infotehnoloogiliste teadmisteta.

SUMMARY

The aim of this thesis was to advance the development of the predictive maintenance systems currently used in industry, by creating a universal software framework that would simplify gathering data from sensors, processing this data, storing processed data and results, sending data to the user and applying machine learning algorithms for pattern recognition and component useful life estimation.

In the course of this thesis, the most common industrial predictive maintenance system solutions and the functionalities they provide were analysed. Based on that analysis, the necessary functionalities were created and a universal software framework was developed that would provide these functionalities.

The developed software framework was tested by creating a predictive maintenance system with its use. As a result of the test, the development of a universal software framework for industrial predictive maintenance system was found to be successful.

As a result of this thesis a universal software framework for industrial predictive maintenance systems was created and developed that would simplify gathering data from sensors, processing, analysing and storing this data. Also relaying the stored data to the user or other systems.

The developed software framework presents the user with a structural design for the predictive maintenance system and the necessary software for its installation, that the user can modify and add their own software.

In the future the software framework can be made more user friendly, by adding to it a graphical interface for easier use, libraries for the most common sensors, software for analysing the data that the user can use or modify to make their own version, a method for displaying stored data in graphs etc.

In the opinion of the author of this thesis the developed universal software framework for industrial predictive maintenance systems provides an opportunity to install a predictive maintenance system without thorough knowledge of computer systems.

KASUTATUD KIRJANDUS

- [1] Wärja, Mathias. (2020). Maintenance Management of Complex Industrial Systems -a methodology for renewal strategies.
- [2] Mishra, Santosh & Mahapatra, Deepak & chandrasah,. (2015). Maintenance Strategy and Decision Making–AHP Method.
- [3] "INDUSTRY 4.0. PREDICTIVE MAINTENANCE. The Complete Guide" [Võrgumaterjal]
Address: <https://www.seebo.com/predictive-maintenance/> [Kasutatud 25.03. 2020]
- [4] "Machine learning for predictive maintenance: where to start?" [Võrgumaterjal]
Address: <https://medium.com/bigdatarepublic/machine-learning-for-predictive-maintenance-where-to-start-5f3b7586acfb> [Kasutatud 25.03.2020]
- [5] "Predictive Maintenance Market by Component (Solutions and Services), Deployment Mode, Organization Size, Vertical (Government and Defense, Manufacturing, Energy and Utilities, Transportation and Logistics), and Region - Global Forecast to 2024" [Võrgumaterjal]

Address: <https://www.marketsandmarkets.com/Market-Reports/operational-predictive-maintenance-market-8656856.html> [Kasutatud 30.03.2020]
- [6] "Company profile for IBM" [Võrgumaterjal]
Address: <https://www.theguardian.com/sustainable-business/profile-ibm>
[Kasutatud 30.03.2020]
- [7] "What is IBM?" [Võrgumaterjal]
Address: <https://www.cbronline.com/what-is/what-is-ibm-4950406/> [Kasutatud 30.03.2020]
- [8] IBM, "IBM Predictive Maintenance and Quality (Version 2.0)", 2014.
- [9] "Siemens - Statistics & Facts" [Võrgumaterjal]
Address: <https://www.statista.com/topics/1901/siemens/> [Kasutatud 01.04.2020]
- [10] "Company" [Võrgumaterjal]
Address: <https://new.siemens.com/global/en/company/about/history/company.html>
[Kasutatud 01.04.2020]
- [11] "About us" [Võrgumaterjal]
Address: <https://new.siemens.com/global/en/company/about.html> [Kasutatud 01.04.2020]

- [12] Siemens, "MindSphere: Enabling the world's industries to drive their digital transformations", 2018 [Valge paber]
- [13] Siemens, "MindSphere for Predictive Maintenance Solution brief", 2019.
- [14] "SAP Company Information" [Võrgumaterjal]
Address: <https://www.sap.com/corporate/en/company.html> [Kasutatud 02.04.2020]
- [15] "SAP: A 48-year history of success" [Võrgumaterjal]
Address: <https://www.sap.com/corporate/en/company/history.html> [Kasutatud 02.04.2020]
- [16] SAP, "Predictive maintenance with SAP", 2017
- [17] SAP, "Predictive Maintenance and Service", 2016
- [18] "Framework" [Võrgumaterjal]
Address: <https://techterms.com/definition/framework> [Kasutatud 06.04.2020]
- [19] "Software Framework vs Library" [Võrgumaterjal]
Address: <https://www.geeksforgeeks.org/software-framework-vs-library/>
[Kasutatud 06.04.2020]
- [20] "Raspberry Pi 3 Model B+ " [Võrgumaterjal]
Address: <https://www.seeedstudio.com/Raspberry-Pi-3-Model-B--p-3037.html>
[Kasutatud 16.05.2020]
- [21] DJB Instruments, "A/130/V Triaxial Piezo-Tronic IEPE accelerometer", 2020
- [22] "American NI MCC WebDAQ 504 vibration sound data recorder imported with high cost performance" [Võrgumaterjal]
Address: <https://www.ebuy7.com/item/us-ni-mcc-webdaq-504-vibration-sound-data-recorder-cost-effective-imported-583213650492> [Kasutatud 16.05.2020]
- [23] "Platinum RTD Sensor – PT100 – 3-wire 1m" [Võrgumaterjal]
Address: <https://www.electrokit.com/en/product/platinum-rtd-sensor-pt100-3-wire-1m-2/>
[Kasutatud 16.05.2020]
- [24] Adafruit Learning System, "Adafruit MAX31865 RTD PT100 or PT1000 Amplifier", 2020
- [25] "3500B02B5G05E000 - Pressure Sensor, 0...2.5 bar, 4...20 mA, Gems" [Võrgumaterjal]

Address: <https://www.elfadistrelec.ee/et/pressure-sensor-bar-20-ma-gems-3500b02b5g05e000/p/30024592> [Kasutatud 16.05.2020]

[26] "2-Channel 4-20 mA Current Loop Receiver 16-Bit ADS1115 I2C Mini Module" [Võrgumaterjal]

Address: <https://store.ncd.io/product/2-channel-4-20-ma-current-loop-receiver-16-bit-ads1115-i2c-mini-module/> [Kasutatud 16.05.2020]

[27] "RUT955" [Võrgumaterjal]

Address: <https://wiki.teltonika-networks.com/view/RUT955> [Kasutatud [16.05.2020]