



TALLINNA TEHNIKAÜLIKOO

INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

TÖÖSTUSROBOTIL PÕHINEV ELEKTROMAGNETLÄHIVÄLJA SKANNER

ELECTROMAGNETIC NEAR-FILED SCANNER BASED ON INDUSTRIAL ROBOT

BAKALAUREUSETÖÖ

Üliõpilane: Vladimir Šulžik
/nimi/

Üliõpilaskood: 164109MAHB

Juhendaja: Madis Lehtla, dotsent
/nimi, amet/

Tallinn, 2020

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.

“.....” 202.....

Autor:

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

“.....” 202.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

“.....”202... .

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina Vladimir Šulžik (*autori nimi*) (sünnikuupäev: 12.10.1996)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Tööstusrobotil põhinev elektromagnetlähivälja skanner,

(*lõputöö pealkiri*)

mille juhendaja on Madis Lehtla,

(*juhendaja nimi*)

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

¹Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil.

_____ (*allkiri*)

_____ (*kuupäev*)

Elektroenergeetika ja mehhatroonika instituut

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Vladimir Šulžik 164109MAHB

Õppekava, peeriala: MAHB 02/13 Mehhatroonika

Juhendaja(d): dotsent, Madis Lehtla, tel. 620 3697

Konsultant: Heigo Mölder, insener (nimi, amet)

Tallinna Tehnikaülikool, 5264992, heigo.molder@taltech.ee (ettevõtte, telefon, e-post)

Lõputöö teema:

(eesti keeles) Tööstusrobotil põhinev elektromagnetlähivälja skanner

(inglise keeles) Electromagnetic Near-field Scanner based on Industrial Robot

Lõputöö põhieesmärgid:

1. Süsteemi (riistvara ja tarkvara) koostamine ja katsetamine elektromagnetväljade mõõtmiseks kasutades tööstusrobotit
2. Liikumistrajektoride planeerimine etteantud objektide (nt trükkplaadi või püsिमagnetiga masina) skaneerimiseks
3. Mõõteandmete salvestamise süsteemi kavandamine

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Lähteandmete kogumine	06.03.2020
2.	Roboti ja selle funktsioonide uurimine	09.03.2020 ja 20.03.2020 - 27.03.2020
3.	Info kogumine Internetist (seoses lähetusega teises projektis)	10 - 20.03.2020
4.	Elektromagnetvälja mõõtemetodite ja kasutatava anduri uurimine	20.03.2020 - 27.03.2020
5.	Töötava mudeli loomine ilma mõõteseadmeta	27.03.2020 - 15.04.2020
6.	Mõõteseadme valik	15.04.2020 - 18.04.2020
7.	Roboti ja mõõteseadmega ühendamine juhtseadmega (nt arvutiga) ja töötava mudeli loomine	24.04.2020
8.	Lõppkatsed	29.04.2020
9.	Tulemuste kirjeldamine lõputöös	29.04.2020 - 06.05.2020
10.	Töö esimene versioon valmis ja juhendajale läbilugemiseks saatmine	08.05.2020

11.	Paranduste sisseviimine ja juhendajale teiseks läbilugemiseks saatmine	13.05.2020
12.	Töö lõplik versioon valmis	18.05.2020

Töö keel: Eesti keel

Lõputöö esitamise tähtaeg: "22" mai 2020 .a

Üliõpilane: Vladimir Šulžik ".....".....202... .a
/allkiri/

Juhendaja: Madis Lehtla ".....".....202... .a
/allkiri/

Konsultant: Heigo Mölder ".....".....202... .a
/allkiri/

Programmijuht: ".....".....202... .a
/allkiri/

SISUKORD

EESSÖNA.....	9
LÜHENDITE JA TÄHISTE LOETELU.....	10
SISSEJUHATUS.....	11
1. ÜLEVAADE ELEKTROMAGNETLÄHIVÄLJA MÕÕTMISEST.....	12
2. KASUTATUD SEADMETE ÜLEVAADE.....	13
2.1 Anduri mikrokontrolleri plaat.....	13
2.2 Andurite ülevaade.....	14
2.2.1 Halli andur.....	14
2.2.2 Halli efektiga töötav andur.....	15
2.2.3 Halli anduri SS496A1 tööpõhimõte.....	16
2.3 Elektriskeem.....	17
3. TÖÖSTUSROBOTI ÜLEVAADE.....	18
3.1 Roboti seadistamine ja kinemaatikamudel.....	20
3.1.1 Seadistusskript MATLAB keskkonnas.....	20
3.1.2 Kinemaatikamudeli Denaviti ja Hartenbergi parameetreid.....	21
3.1.3 Roboti liigendite liikumisulatuste piiramine.....	23
3.1.4 Andmesideliides.....	24
3.2 Roboti juhtimine MATLAB keskkonnast.....	26
3.2.1 Olulised muutujad ja objektid.....	26
3.3 Tarkvara funktsioonide ülevaade.....	26
3.3.1 Kohandamisfunktsioonid.....	27
3.3.2 Juhtimisfunktsioonid.....	28
4. ELEKTROMAGNETLÄHIVÄLJA SKANNERI TÖÖPÕHIMÕTE.....	29
4.1 Roboti liikumine.....	29
4.2 Andmete lugemine.....	31

4.3 Värvitoonide valimine	32
5. ELEKTROMAGNETLÄHIVÄLJA SKANNERI PROGAMM.....	33
5.1 Põhifunktsioon.....	33
5.2 Funktsioon RobotHoming.....	35
5.3 Funktsioon New_Position.....	35
5.4 Funktsioon Read_From_Microcontroller	35
5.5 Funktsioon Value_to_RGB_color	36
5.6 Funktsioon RobotFinish	37
5.7 Mikrokontrolleri programm.....	37
6. SÜSTEEMI EDASIARENDUSVÕIMALUSED	38
6.1 Süsteemi eelised ja puudused	38
6.1.1 Puudused.....	38
6.1.2 Eelised	38
6.2 Andur.....	38
6.3 Kolmemõõtmeline mudel ja töö konveieril	40
6.4 Kinnitused	40
KOKKUVÕTE.....	41
SUMMARY	43
KASUTATUD KIRJANDUS.....	45
LISAD	48
Lisa 1 – Roboti juhtimisprogrammi põhifunktsioon.....	48
Lisa 2 – Funktsioon Value_to_RGB_color.....	51
Lisa 3 – Funktsioon Read_From_Microcontroller.....	52
Lisa 4 – Funktsioon New_Position	53
Lisa 5 – Funktsioon RobotFinish	54
Lisa 6 – Funktsioon RobotHoming	55

Lisa 7 – Seadistusklass MATLAB keskkonnas	56
Lisa 8 – Mikrokontrolleri programm	58

EESSÕNA

Idee magnetvälja mõõtmiseks tööstusrobotiga andis Tallinna Tehnikaülikooli elektroenergeetika ja mehhatroonika instituudi insener Heigo Mölder, kes oli varem kokku puutunud sarnaste elektromagnetvälja mõõtesüsteemidega. Varasemate lõputööde raames on katsetatud ka 3D-printeri kasutamist elektromagnetvälja skannerina. Instituudi elektromagnetilise ühilduvuse laboris on mitmesuguseid professionaalseid kõrgsageduselektromagnetvälja mõõteriistu, mille mõõteproovikuid saaks tulevikus robotsüsteemiga ühendada. Laboratoorsete kõrgsageduslike proovikute paigutamisel liigendkäega roboti tööriistaks tekib kõrgsageduskaablite roboti liigendkäest üle viimise probleem. Märksa lihtsam on roboti tööriistaks paigutada väiksemaid mõõteriistu tervikuna ja neid sidestada läbi standardse andmesidekaabli (USB või Ethernet). Pakutud lahenduses liidestati Halli-andur läbi mikrokontrolleri A/D-muunduri ja USB-kaabli arvutuspaketiga MATLAB.

Autor soovib tänada dotsenti Madis Lehtlat ja inseneri Heigo Mölderit juhendamise ja abi eest tööloomisel.

LÜHENDITE JA TÄHISTE LOETELU

USB	Universaalne järjestiksin (<i>Universal Serial Bus</i>)
GND	Maa või massiühendus (<i>Ground</i>)
RISC	Kärbitud käsustikuga arvuti (<i>Reduced Instruction Set Computer</i>)
RGB	RGB-mudel (<i>RGB color model</i>)
ADC	analoo-digitaalmuundur (<i>Analog-to-digital converter</i>)
CR	Rea algusesse tagasitulek ASCII-tekstil põhinevas protokollis (<i>Carriage Return</i>)
MM	disaini optimeerimine transistoride tiheduse suurendamisega (<i>More-Moore</i>)
MtM	multidistsiplinaarne disaini ja funktsioonide optimeerimine (<i>More-than-Moore</i>)
COM-port	sideliides operatsioonisüsteemis Windows arvuti välisseadmetega suhtluseks (<i>Communication port</i>). Tänapäeval toimub side füüsiliselt enamasti USB-liidese kaudu.
<i>B</i>	Magnetiline induktsioon
A1	Analoogsisend 1 st analoo-digitaalmuunduri kommutaatori sisend 1, mille kaudu mõõdetakse analoogsuuruste väärtusi
<i>U</i>	Elektriline pingesignaal
<i>bitt</i>	Digitaalse informatsiooni ühik (<i>bit</i>)
DH- parameetrid	manipulaatori kinemaatikamudeli kirjeldamiseks kasutatavad Denavit – Hartenbergi parameetrid
CSV	comma separated values

SISSEJUHATUS

Bakalaureusetöö eesmärgiks on süsteemi riistvara ja tarkvara koostamine ning katsetamine elektromagnetväljade mõõtmiseks kasutades tööstusrobotit. Selle eesmärgi saavutamiseks on otsustatud koostada ja programmeerida elektromagnetlähivälja skanner, mis põhineb Tallinna Tehnikaülikoolis asuval tööstusrobotil RV-1A[14].

Kaasaegses maailmas ilmub üha rohkem elektroonilisi seadmeid. Nende arvu suurenemisega kasvab nende kvaliteedi jälgimise vajadus, eriti elektromagnetilise ühilduvuse valdkonnas. Inimese tehtav monotoonne töö iga elektroonilise osa elektromagnetvälja mõõtmiseks võib olla väga kulukas ja ebatäpne. Seetõttu on vaja luua uusi seadmeid, mis võiks inimest asendada. Elektromagnetvälja skannerite tehnoloogia analooge on maailmas juba olemas. Kõigil nendega on võimalik teha ainult mõõtmisi. Tööstusrobotil põhineva skanneriga saab lisaks mõõtmisele teha ka muid toiminguid (töötada nagu tavaline robot). Muudeks toiminguteks saab kasutades roboti muid tööriistaid, näiteks haaratsit mõõteobjekti toomiseks või viimiseks. Käesolev töö aitab edasi liikuda selle tehnoloogia uurimisel ning automaatskaneerimiseks toimiva ja efektiivse mudeli loomisel.

Töö eesmärgi saavutamiseks seati järgmised ülesanded:

1. Liideste (jadaport) ja tarkvara (nt MATLAB) seadistamine tööstusroboti juhtimiseks välisest arvutist
2. Liikumistrajektooride planeerimine etteantud objektide (nt trükkplaadi või püsomagnetiga masina) skaneerimiseks
3. Kõigi protsesside haldamise programmi väljatöötamine ja kirjutamine
4. Skaneerimisprotsessi juhtiva arvuti (või kontrolleri) ühendamine elektromagnetvälja mõõteseadmega (A/D-muunduriga)
5. Mõõteandmete kuvamise süsteemi kavandamine (nt CSV vormingusse, MATLABi või EXCELisse)

Lõputöö alguses antakse lühiülevaade elektromagnetlähiväljade mõõtmisest. Teine osa räägib mõõtmiseks kasutatavatest elektroonikaseadmetest, eriti andurist ja mikrokontrollerist, ning nende tööpõhimõtetest. Kolmandas osas kirjeldatakse töös kasutatud tööstusrobotit, selle seadistusi, samuti tarkvara roboti juhtimiseks keskkonnas MATLAB. Neljas osa räägib roboti tööpõhimõttest, kuidas see liigub, kuidas analüüsitakse mõõteandmeid, kuidas on valitud värvitoonid magnetvälja tugevuse ekraanil kuvamiseks. Viies osa tutvustab programmi nii MATLAB keskkonnas kui ka mikrokontrolleril töötavaid funktsioone. Siin räägitakse nende töö loogikast ja sellest, mille jaoks on iga funktsiooni. Viimases osas on esitatud roboti edasised väljavaated, mis tulevikus võib aidata mõõterobotit edasi arendada.

1. ÜLEVADE ELEKTROMAGNETLÄHIVÄLJA MÕÕTMISEST

Elektroonikaseadmete üha suurem rakendamine inimtegevuse erinevates valdkondades seab uusi ülesandeid nende disaini ja kvaliteedi parandamiseks. Elektroonikaseadmed muutuvad kompaktsemaks ja nende töösagedused kõrgemaks. Moore seadusest tulenevalt mis prognoosis transistoride arvu kasvu mikrokiipidel on nüüd tulnud mikroelektroonikas ja integraallülitustes käibe uued suunad ja tehnoloogiad nii disaini mõõtmete, voolutarbe kui signaalide sageduslike omaduste optimeerimiseks nt More-Moore (MM) ja More-than-Moore (MtM). Signaali ja müra suhte vähenemine muudab uued elektroonikaseadmete põlvkonnad tundlikumaks elektromagnetilistele häiringutele. Elektroonikaseadmete arendajate jaoks põhjustab see üha suuremaid elektromagnetilise ühilduvuse probleeme mis vajavad lahendusi. [23]

Elektroonikaseadmete poolt emiteeritavad häiringud peavad vastama elektromagnetilise ühilduvuse normidele ja standarditele, selleks mõõdetakse elektroonikaseadmete elektromagnetilist kiirgust erinevates sageduspiirkondades. Samuti peavad seadmed taluma keskkonnas teiste seadmete poolt põhjustatavaid häiringuid vastavalt etteantud normidele. [22]

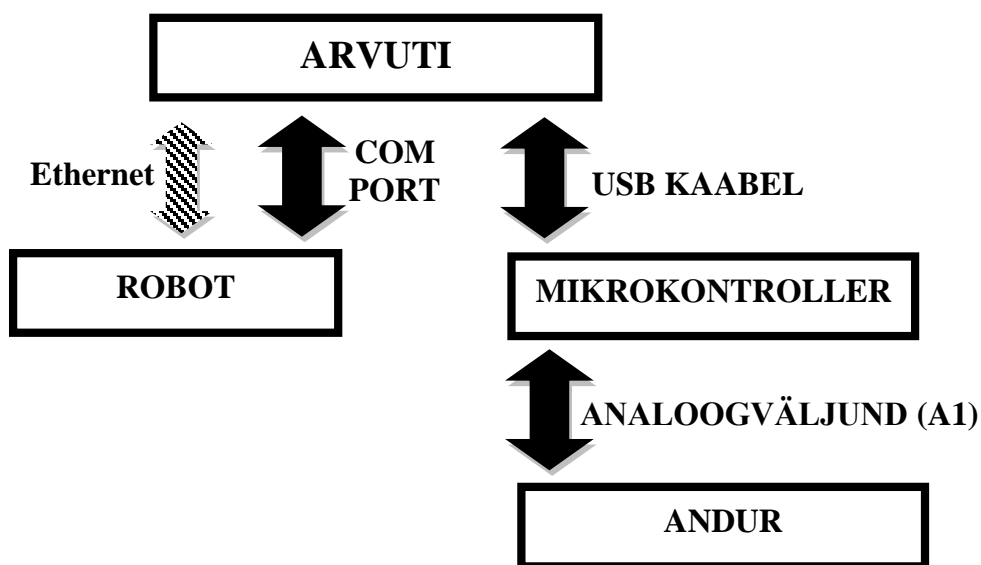
Testid viiakse sageli läbi kajavabades või pooleldi kajavabades kambrites, mis on rahaliselt kallis ülesanne. See omakorda tähendab valmistoodete maksumuse tõusu. Seega on üks täiendavaid võimalusi mõõtmiste automatiseerimiseks ning seeläbi inimese rutiinse töö vajaduse vähendamiseks tööstusrobotil põhinev elektromagnetlähivälja skanner. [22]

2. KASUTATUD SEADMETE ÜLEVAADE

2.1 Anduri mikrokontrolleri plaat

Anduri mikrokontrolleri plaat on vajalik analoogväljundiga elektromagnetvälja andurist saadava informatsiooni teisendamiseks digitaalkujule, töötlemiseks sh skaleerimiseks, filtreerimiseks ning kodeerimiseks edastamiseks sobivale kujule.

Anduri liidestamiseks valisin mikrokontrolleril ATmega32U4 põhineva mikrokontrolleri plaadi [8][9]. Mikrokontrolleri plaadil on 18 digitaalset sisend- ja väljundklemmi (neist 4 saab kasutada analoogsisendina), 16 MHz kvartskristallil põhinev ostsillaator ja mikro-USB ühenduspistik. See sisaldab kõike vajalikku, mida on vaja selles elektriskeemis. Analoogsisendid võimaldavad plaadi hõlpsalt analooganduriga ühendada. Samuti on anduri jaoks ideaalne 5-voldine tööpinge. Mikrokontrolleri plaat Arduino Pro Micro on mikro-USB-kaabli abil arvutiga ühendada väga lihtne. Seda tüüpi kaabel on väga levinud ning seda on lihtne leida ja asendada, samuti on see piisavalt paindub roboti manipulaatori külge kinnitamiseks. Samuti saab seda kaablit kasutada teistes süsteemides. Pealegi on see mikrokontrolleri plaat väga väike, kerge ning makettlaual (*breadboard*) hõlpsasti ühendatav anduri ja teiste seadmetega. Tänu nendele teguritele on plaati robotile väga lihtne kinnitada. [9]



Joonis 2.1 Süsteemi plokk skeem

Tabel 2.1 Anduri liidestamiseks kasutatud mikrokontrolleri plaadi omadused [10][11][12]

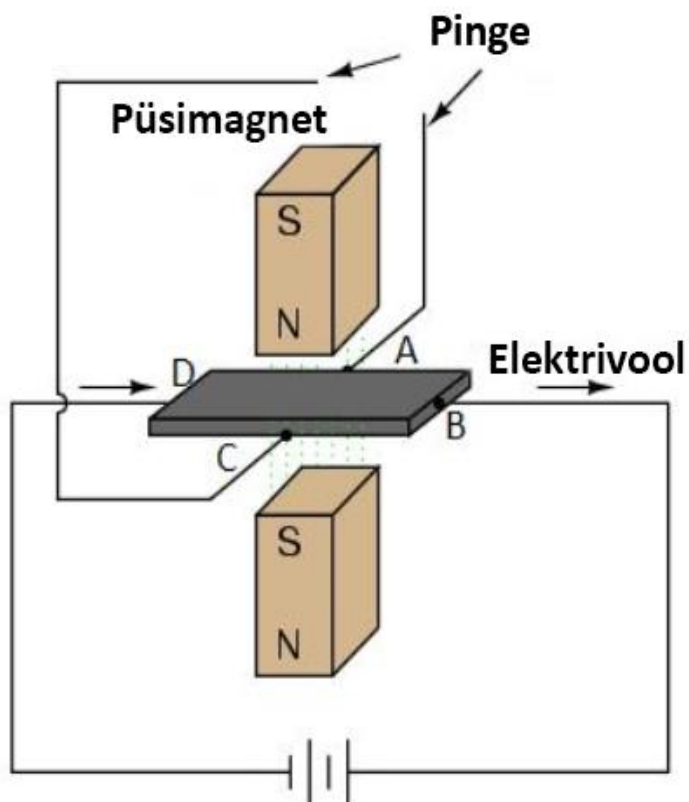
Omadus	Tehniliste näitajate väärtused	Ühik
Mikrokontrolleri kiip	ATmega32U4	
Toitepinge	5	V
Taktsagedus	16	MHz
Maksimaalne koormusvool klemmi kohta	40	mA
Soovitatav koormusvool klemmi kohta	20	mA
Mõõtmed	33x18	mm
Kaal	5	g

2.2 Andurite ülevaade

Anduri valimine on minu töö üks olulisemaid osi. Andur aitab määrata magnetvälja kindlates mõõtepunktides. Mõõteandmete kvaliteet sõltub anduri täpsusest. Andur peab olema ka väike, et seda saaks robotile paigaldada. Samuti peaks andur töötama madala pingega korral, nii et seda saaks ühendada 5V toitepingega töötava mikrokontrollerplaadiga. Mõlemad alltoodud andurid pakuvad neid omadusi.

2.2.1 Halli andur

Oma töös otsustasin kasutada Halli andurit. See on üks tõhusamaid meetodeid magnetväljade määramiseks. See võimaldab määrata ka magnetvälja tugevuse. Klassikaline Halli sensor on õhuke mitmekihiline (ABCD) pooljuht lehtmaterjal. Elektrivool juhitakse läbi plaadi. Plaadil olevad elektronid liiguvad küljelt B küljele D. Pärast seda kui viiakse püsomagnet plaadi suhtes risti tekib niinimetatud Lorentzi jõud, mis on suunatud mööda plaati lendavate elektronide küljele. Selle tõttu tekib plaadi pindade A ja C vahel potentsiaalne erinevus. See potentsiaalide erinevus, teisisõnu, pinge on otseselt võrdeline magnetilise induktsiooniga. [6][13]



Joonis 2.2 Halli tajuri tööpõhimõte

2.2.2 Halli efektiga töötav andur

Tööanduriks valiti andur SS496A1. Sellel anduril on analoogsensor, mis aitab kindlaks teha mitte ainult magnetvälja olemasolu, vaid ka selle suurust. Sellel anduril pole moodulit, kuid mul õnnestus see makettlaual (*breadboard*) mikrokontrolleriga abil ilma probleemideta ühendada.



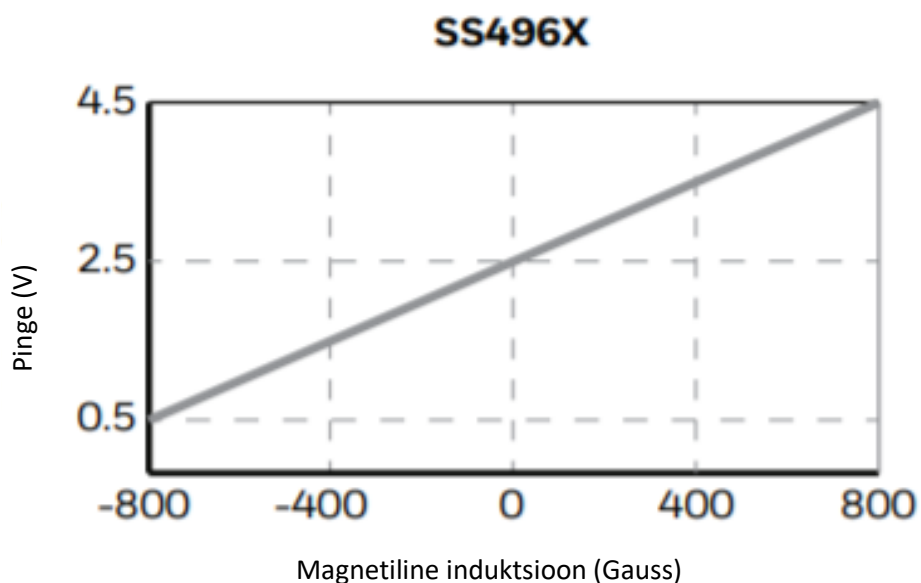
Joonis 2.3 Hall'i anduri reguleeritav moodul

Tabel 2.2 Halli anduri SS496A1 omadused [2]

Omadus	Tehniliste näitajate väärtuste vahemik	Ühik
Toitepinge	4,5 .. 10,5	V
Toitevool (temperatuuril T = 25°C)	tavaoludes 7,0 maksimaalne 8,7	mA
Töötemperatuur	-40 .. +150	°C
Möötepiirkond	tavaoludes ± 840	Gauss
Tundlikkus (temperatuuril T _A = 25°C)	tavaoludes 2500	mV/Gauss
Väljundi tüüp	sisendsuurusega proportsionaalne ehk ratiomeetriline (<i>ratio metric</i>)	

2.2.3 Halli anduri SS496A1 tööpõhimõte

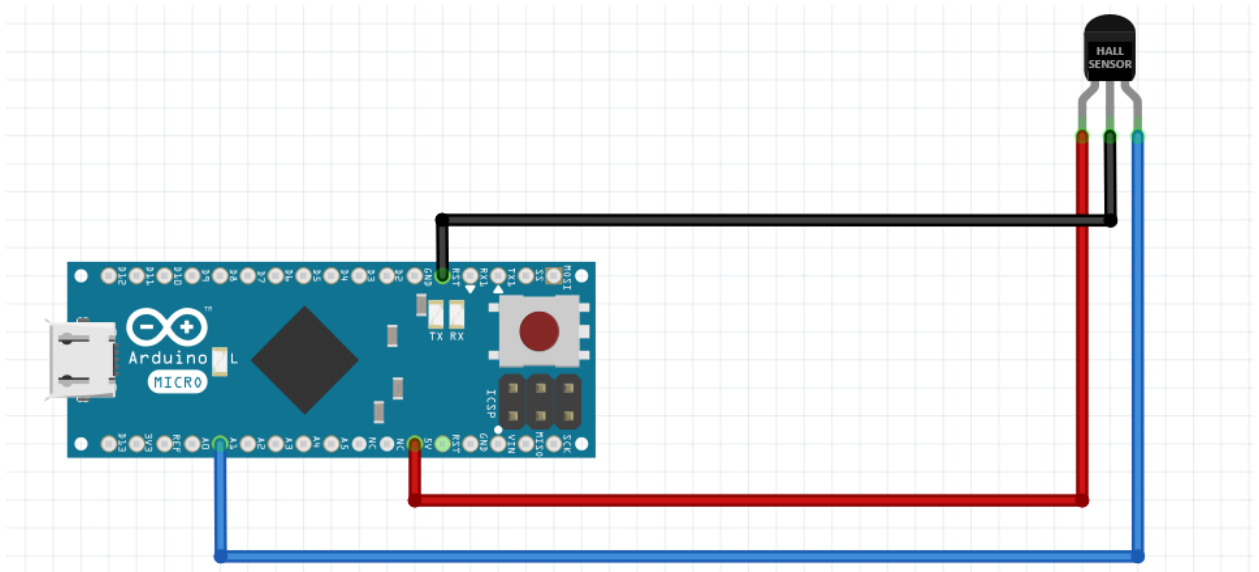
Selle anduri väljundsuurus sõltub magnetinduktsioonist. Rahulikus olekus ($B = 0$) tagastab andur poole temale edastatud pingest. Niipea kui andur siseneb magnetvoolu tsooni, hakkab see pinget vähendada või suurendama, sõltuvalt magnetinduktsioonist. Magnetinduktsiooni maksimaalse negatiivse väärtuse korral annab andur minimaalse pinget ja maksimaalse positiivse magnetinduktsiooni väärtuse korral annab andur maksimaalse pinget. [2]



Joonis 2.4 Anduri SS496X pinge sõltuvus magnetilisest induktsioonist

2.3 Elektriskeem

Andur ühendatakse mikrokontrolleri plaadiga 3 juhtmega ühenduse kaudu. Andurikiibi SS496A1 toitepinge ühendatakse vasaku klemmiga. Keskmine klemm ühendatakse kokku mikrokontrolleri A/D-muunduri massiühendusega GND. Kolmandaks juhtmeks on signaalijuhe, mis ühendab sensori parema klemmi mikrokontrolleri plaadi analoogsisendiga A1, mille kaudu andmeid sensorilt loetakse.



Joonis 2.5 Anduri ühenduskeem mikrokontrolleri plaadiga ilma vahevõimenditeta.

3. TÖÖSTUSROBOTI ÜLEVAADE

Tööstusrobotina kasutati firma Mitsubishi tööstusrobotit MELFA RV-1A, mis oli olemas ülikooli õppelaboris NRG-423. See on kuue pöördteljega liigendkäega robot. Seda robotit on mugav kasutada ning seda saab programmeerida kas välise puldi või arvuti abil. Antud robot oli püstitatud ülesande jaoks piisavalt kiire ja täpne (vt. tabeli 3.1). Mõõdetavate objektidena kasutati väiksemat mootorit ja püsimagnetit. Robot teeb suure hulga samme väikeses vahemaas, liikumise standardne samm antud projektis oli 1 mm. Sellega seoses vajab robot head täpsust. Positsioneerimise hajuvuse ulatus (ehk positsioonide korratavus) on selle robotil $\pm 0,02$ mm. Positsioneerimise korratavus tähendab roboti võimet naasta alati samale (programmeeritud) positsioonile samast suunast. Hea positsioneerimise korratavus on eriti oluline, kuna mõõtmiste ajal liigub robot pidevalt sirgjooneliselt ja läbib iga kord samu punkte. See aitab robotil jõuda täpselt soovitud koordinaatideni ilma kõrvalekalleteta.



Joonis 3.1 Tööstusroboti ülevaade

Tabel 3.1 Roboti RV-1A omadused [14]

Omadus	Tehniliste näitajate väärtused	Ühik
Struktuur	Vertikaalne, mitme liigendlüluga	
Ajamisüsteem	Vahelduvvoolu servoajam (liigenditel J1 .. J3 50W koos piduritega, liigenditel J4 ja J6 15W ilma piduriteta, liigendil J5 15W koos piduriga)	
Asendi määramise meetod	Absoluutasendi väljundiga impulss-pöördkoodrid (varutoide patareidelt)	
Maksimaalne kiirus	Ligikaudu 2200	mm/s
Tõstetav koormus	1	kg
Asendi korratavus	± 0.02	mm
Mass	Ligikaudu 19	kg
Liigendkäe haardeulatus (Arm reachable radius, front p-axis center point)	418	mm

Servomootorid on varustatud patareidel töötavate absoluutkoodritega, mis välistab vajaduse roboti sisselülitamisel seda igal korral lähtepunktis kalibreerida. Roboti juhtseadmes kasutatakse 64-bitist RISC-protssessorit, millel on paremad täiustatud funktsioonid, jõudlus ja laiendatavus. Roboti keeled MELFA-BASIC IV ja *MoveMaster Command* kuuluvad standardkonfiguratsiooni. Robotit on väga mugav juhtida selliste programmide abil nagu CIROS® Studio [7] või MATLAB. Selle töö jaoks valisin programmpaketi MATLAB. See võimaldab robotil andmete mõõtmise ajal hõlpsalt uusi koordinaate luua, mitte ainult eelnevalt salvestatud koordinaatidega töötada ja rakendada väljatöötatud lahendust tulevikus ka teistel robotitel. Kontrolleriga CR1-571 saab juhtida kõiki 6 telge, samuti meelde jätta kuni 88 programmi, 2500 positsiooni ja 5000 sammu. [14]



Joonis 3.2 Roboti juhtseadme juhtpaneel

3.1 Roboti seadistamine ja kinemaatikamudel

3.1.1 Seadistusskript MATLAB keskkonnas

Roboti juhtimiseks läbi MATLAB-keskkonna on võimalik kasutada lisapaketti *Mitsubishi Melfa Robot Control Toolbox* [17]. Selle tarkvara kasutamiseks loodi programm, mis võimaldab muuta kõik vajalikud parameetrid süsteemi korrektseks tööks selle robotiga. Koostatud programme saab muuta ja kohandada ka mõne teise selle sarja roboti jaoks. Esmalt on vaja seadistama põhisätteid, näiteks roboti nimi, selle seeria ja kirjeldus. Kõige tähtsam on anda failile ja funktsioonidele lihtne nimi, kuna see on ühendatud kogu tarkvaraga. Funktsiooni nimed koostatakse kahest osast st eesliitest mmRobot , mille järel on roboti nimi.

Koodinäide:

```
mmRobot_RV1A
```

Iga roboti jaoks on oluline määrata ka õige liigendite arv. Sellel robotil on 6 liigendit. [14]

Koodinäide:

```
robot.DOF = 6;
robot.joints = 'RRRRRR';
```

3.1.2 Kinemaatikamudeli Denaviti ja Hartenbergi parameetreid

Roboti juhtimiseks kasutatakse kinemaatikamudelis Denaviti ja Hartenbergi parameetreid. Denavit ja Hartenbergi maatriksmeetod on kinemaatilise ahela iga lüliga seotud koordinaatsüsteemide järjestikuse konstrueerimise meetod. Seda kasutatakse külgnevate ühenduslülide vaheliste pöörlevate ja kulgliikumisega (translatiivsete) ühenduste kirjeldamiseks. Denavit - Hartenbergi meetodi eesmärk on moodustada homogeenne teisendusmaatriks, millel on 4×4 mõõde ja mis kirjeldab iga lüli koordinaatsüsteemi asukohta eelmise lüli koordinaatsüsteemi suhtes. [15][16]

Iga koordinaatsüsteem moodustatakse järgmise kolme reegli alusel: [15][16]

- 1) z_{i-1} telg on suunatud piki i -nda liigendi telge;
- 2) x_i -telg on z_{i-1} -teljega risti ja suunatud sellest eemale;
- 3) y_i -telg täiendab x_i , z_i -telgi, et teha parempoolse Descartes'i koordinaatsüsteemi.

Tahkete ühenduslülide DH-meetod sõltub igale lülile vastavast neljast geomeetrisest parameetrist. Need neli parameetrit kirjeldavad täielikult pöörlevaid liikumisi või kulgliikumisi (translatsiooniliikumisi) ja määratakse järgmiselt: [15][16]

- α_i on nurknihe - nurk, mille abil z_{i-1} telge tuleb pöörata ümber x_i telje nii, et see muutuks z_i teljega ühesuunaliseks (märk määratakse vastavalt parema käe reeglile);
- a_i on lineaarne nihe - z_{i-1} telje ja x_i telje ristumiskoha ja i -nda koordinaatsüsteemi alguse vaheline kaugus, arvestatuna piki x_i -telge, see tähendab, et z_{i-1} ja z_i -telgedel vahel on lühim kaugus;
- θ_i on määratud nurk, mille peal tuleb x_{i-1} telge pöörata ümber z_{i-1} telje, nii et see joonduks x_i teljega (märk määratakse vastavalt parema käe reeglile);
- d_i on z_{i-1} -telje ja x_i -telje ristumiskoha ja $(i-1)$ koordinaatsüsteemi algpunkti vaheline kaugus, mõõdetuna piki z_{i-1} -telge.

Pöörlevate lülide puhul on parameetrite d_i , a_i ja α_i väärtused teatud tüüpi roboti korral konstantsed. Samal ajal on θ_i muutuja, mis muutub i -nda lüli liikumise (pöörlemise) ajal $(i-1)$ -ga võrreldes. [15][16]

DH-parameetrid salvestatakse objekti „robot“ muutujas robot.denavitHartenberg. Roboti RV-1A jaoks Denaviti ja Hartenbergi parameetrid on näitud Tabelis 1. Kõik andmed on võetud spetsifikatsioonide juhendist [14] (vt. joonis 1.9.) ja CIROS® Studio robotimudelist [7].

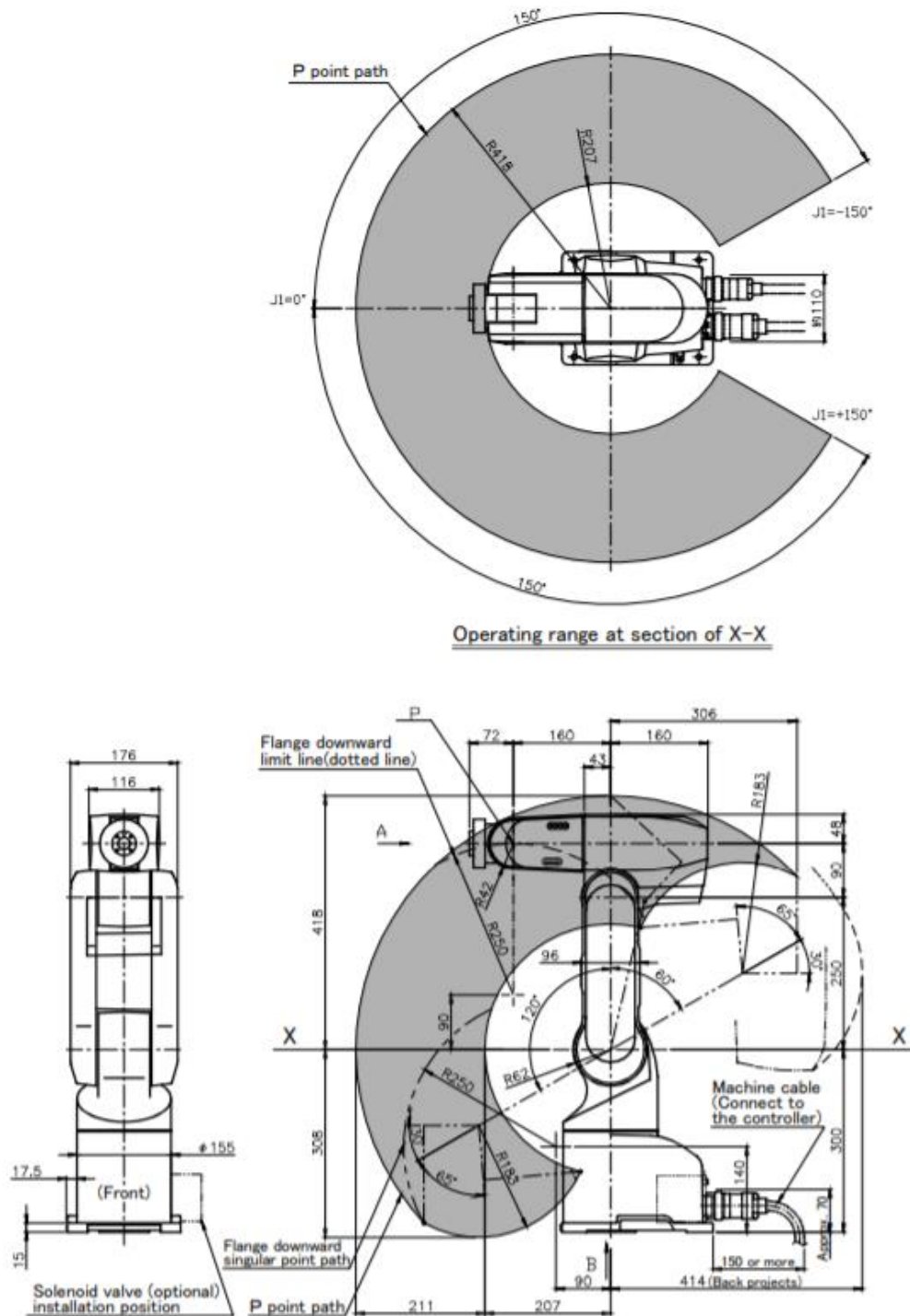
Tabel 3.2 Kinemaatikamudeli parameetrid

	liigend 1	liigend 2	liigend 3	liigend 4	liigend 5	liigend 6
α	$-\pi/2$	0	$\pi/2$	$-\pi/2$	$\pi/2$	0
a	0	250	-90	0	0	0
θ	0	$-\pi/2$	$\pi/2$	0	0	0
d	300	0	0	160	0	72

Koodinäide MATLAB-keskkonnas:

```
robot.denavitHartenberg =
```

```
[-pi/2,      0,      pi/2,  -pi/2,   pi/2,    0;  %alpha
      0,      250,    -90,     0,     0,     0;  %a
      0,    -pi/2,    pi/2,     0,     0,     0;  %theta
      300,     0,     0,    160,     0,    72]; %d
```



Joonis 3.3 Tööstusroboti Mitsubishi RV-1A töötsoon [14]

3.1.3 Roboti liigendite liikumisulatuste piiramine

Veel üks oluline parameeter on liigete töövahemik. Ta on vajalik roboti juhtimiseks, määrates iga liigendi pöördenurga. Samuti see parameeter on vaja piiride seadmiseks, milles robot saab liikuda, ja mitte seda murda. See parameeter salvestatakse objekti „robot“ muutujas robot.jointLimits.

Tabel 3.2 Roboti RV-1a töövahemik kraadides (degrees) [14]

Liigend	Liikumisulatus kraadides
J1	300 (-150 .. +150)
J2	180 (-60 .. +120)
J3	95 (+60 .. +155)
J4	320 (-160 .. +160)
J5	180 (-90 ..+90)
J6	400 (-200 .. +200)

Koodinäide:

```
robot.jointLimits = [-150, -60, 60, -160, -90, -200 %minimum
                    150, 120, 155, 160, 90, 200];%maximum
```

3.1.4 Andmesideliides

Viimased parameetrid, mida on vaja programmi sisestama, on kontrolleri ja pordi spetsifikatsioonid. Neid on vaja ühenduse loomiseks MATLAB programmi ja robotit juhtiva kontrolleri vahel. Kõik pordi sätted tuleb sisestada funktsiooni mmComprops_Serial parameetritena.

Tabel 3.2 Selle kontrolleri pordi vaikesätted [14]

Omadus	Tehnilise näitaja väärtus	Ühik
Andmeside kiirus	9600	Bit/s
Andmesõna pikkus	8	Bit
Paarsuskontroll	paaris (even)	
Stopp-bittide arv	2	
Lõppseadme valmisoleku kontroll (<i>Data Terminal Ready, DTR</i>)	aktiivne	
Saatmispäring (<i>Request to send, RTS</i>)	aktiivne	
Realõpukood	CR (<i>carridge return</i>)	

Koodinäide MATLABis:

```
robot.com = [];
robot.comtype = 'Serial';
robot.portname = 'COM1';
robot.controller = 'CR1-571';
robot.robotNo = 1;
robot.slotNo = 1;
robot.comprops= mmComprops_Serial(9600,8,'even',2,'on','on','CR');
robot.pincodes = 0;
```

3.2 Roboti juhtimine MATLAB keskkonnast

Roboti jaoks on Internetis kättesaadav MATLABi teek „Mitsubishi Melfa Robot Control Toolbox“, mis on Tšehhi tehnikaülikoolis Prahas välja töötatud tarkvara. Selle autor on Martin Meloun. Tarkvara võimaldab juhtida roboteid otse MATLAB keskkonnast jadapordi kaudu.. Roboti liikumist saab ette anda nii Descartes'i koordinaatides kui ka liigendi pöördenurga abil. [17] [4]

Selle tarkvara abil roboti juhtimiseks peab olema saadaval roboti määratlusfail. Teegi autori Martin Melouni poolt on määratlusfail tehtud robotitele Mitsubishi Melfa RV-6S ja Mitsubishi Melfa RV-6SDL. Sarnase faili valmistasin ka RV-1A roboti jaoks ning seda saab kohandada ja modifitseerida ka teiste robotite jaoks. [17]

Mitsubishi Melfa Robot Control Toolbox vajab paketti MATLAB 7.6 (R2008a) või uuemat versiooni koos jadapordiobjektide toega (x64 jaoks on vaja uuemat versiooni). Mitsubishi Melfa Robot Control Toolbox on avatud lähtekoodiga tarkvara. Seda saab kasutada vastavalt GNU Lesser General Public Licence tingimustele, mille on avaldanud Free Software Foundation, kas litsentsi versioonile 3 või vabal valikul ka selle hilisemad versioonid. [17]

3.2.1 Olulised muutujad ja objektid

Vastus 'r' - kogu tarkvara kasutab seda tagastamisväärtust teatisena, et side robotiga oli edukas. Sellega kontrollitakse ainult seda, kas kommunikatsioonis oli viga, nt robotile käsu minna väljaspool oma tööpiirkonda pole seetõttu kommunikatsiooni viga. Mõnel funktsioonil on 'turvaline' variant; see tähendab, et pärast käsku tehakse korrektne vigade kontrollimine (tagastab tavaliselt 'r' väärtuse asemel 'rs'. 'r' edu edukuse korral on 0, muu väärtus on viga). [17]

Objekti 'robot' kasutatakse roboti kõigi andmete salvestamiseks. See on vajalik peaaegu kõigis selle tarkvara funktsioonides. See luuakse funktsioonides mmOpen või mmOpen2, kui roboti nimi on täpsustatud, ja see keelatakse mmClose abil. [17]

3.3 Tarkvara funktsioonide ülevaade

Funktsioonid on omavahel tihedalt seotud. Üks funktsioon võib avada palju teisi, kus on salvestatud mitmesuguseid andmeid või tingimusi käsu täitmiseks. Sellega seoses on väga oluline salvestada funktsioone ühes kaustas koos põhikoodiga. Järgnevalt on toodud näited funktsioonidest, mida kasutasin oma koodis kõige rohkem või mis aitasid lõputöö lõpule viia. Teiste funktsioonide loetelu leiab lisast.

3.3.1 Kohandamisfunktsioonid

mmOpen(robotname, varargin)

Funktsioonid mmOpen avab side Melfa robotiga. Kõige olulisem argument on roboti nimi. See peab vastama faili nimele, kuhu selle seaded salvestatakse. Varargini sisend on paarid 'propertyName', 'propertyValue' ja need paarid on määratud robotile.

Koodinäide: `robot = mmOpen('RV1A');`

mmCntlOn(robot) ja mmSrvOn(robot)

Funktsioon mmCntlOn on vajalik roboti juhtimise võimaldamiseks. Selle funktsiooni kõne võimaldab käskude saatmist kontrolleri pordi kaudu.

Funktsioon mmSrvOn on vaja roboti servomootorite sisselülitamiseks. Samuti saab käsku 'mmWaitForServoOn' kasutada, et oodata, kuni servomootor sisse lülitatakse.

Koodinäided: `mmCntlOn(robot);`

`mmSrvOn(robot);`

mmSetBaseSpeed(robot, speed) ja mmSetJogSpeed(robot, speed)

Need kaks funktsiooni aitavad määrata vastavalt roboti baasi ja püsikiiruse. Nende argumentid on roboti seaded ja kiirus, millega robot peaks liikuma.

Koodinäided: `mmSetBaseSpeed(robot, 50);`

`mmSetJogSpeed(robot, 100);`

mmClose(robot)

See funktsioon lülitab mootorid ja roboti kontrolli välja, samuti katkestab side robotiga ja sulgeb pordi. See on vajalik iga koodi lõpus, vastasel juhul pole robotil võimalik uut programmi käivitada enne, kui MATLAB taaskäivitatakse.

Koodinäide: `mmClose(robot);`

3.3.2 Juhtimisfunktsioonid

mmGetPos(robot, type)

Selle funktsiooni abil saate teada roboti asukoha koordinaadid. Koordinaate võib olla kahte tüüpi - ristkoordinaadid ja liigese pöördekoordinaadid. Ristkoordinaadid on esitatud millimeetrites, liigese pöörde koordinaadid - kraadides. Argument 'type' määrab koordinaatide tüübi, milles on vaja andmeid saada (see võib olla "P" XYZ koordinaatide korral, "J" liigese pöördenurga jaoks ja "X" XYZ ja liigese pöörenurga koordinaatide jaoks).

Koodinäide: `mmGetPos (robot, 'X');`

mmMov(robot, type, pos)

See funktsioon liigub roboti uuele kohale kasutades liigese interpoleerimist. See on ristkoordinaadistikus roboti liigutamiseks vajalik põhikäsk sellest tarkvarast. Argument 'type' määrab koordinaatide tüübi, mille peal robot liigub (see võib olla "P" XYZ koordinaatide korral, "J" liigese pöördenurga jaoks ja "X" XYZ ja liigese pöörenurga koordinaatide jaoks). Argument 'pos' on koordinaattelje vektori veerg, mis näitab roboti liikumise asukohta. Liikumise rakendamine on asünkroonne, selle pärast on vaja kasutama funktsiooni 'mmWaitForStop', et oodata, kuni liikumine on lõppenud (või veateateni).

Koodinäide: `mmMov (robot, 'P', [160; 0; 568; 180; 0; 180]);`

mmExec(robot, tmplt, varargin)

See funktsioon saadab robotile Melfa BASICu käsu ja loeb selle vastuse. Seda funktsiooni minu programmis kasutati roboti lõpus klambrite avamiseks ja sulgemiseks. Funktsiooni argumendiks 'tmplt' on robotile saadetav käsk. See võib olla ka stringi kujul (koos vastavate muutuvate argumentidega). Varargini sisend on paarid 'propertyName', 'propertyValue' ja need paarid on määratud robotile.

Koodinäide: `mmExec (robot, 'HOPEN 1');`

4. ELEKTROMAGNETLÄHIVÄLJA SKANNERI TÖÖPÕHIMÕTE

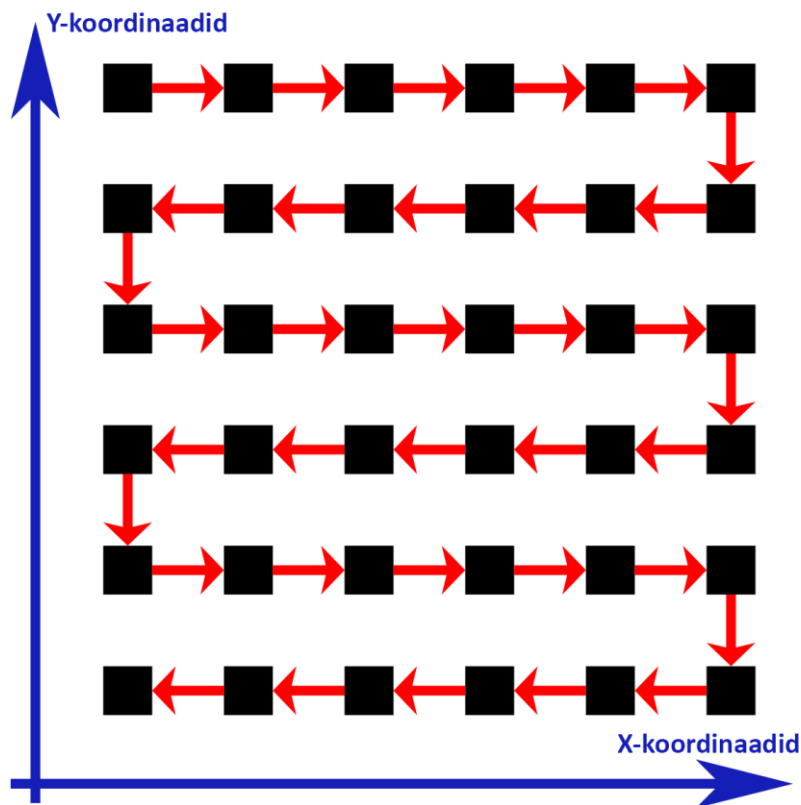
4.1 Roboti liikumine

Enne tööd oli üks peamisi küsimusi, kuidas robot peaks kõige paremini liikuma. Robot pidi läbima kogu mõõdetud objekti pindala ja mõõtma maksimaalset arvu punkte, et saada kõige täpsemaid andmeid. Nende eesmärkide täitmiseks oli kaks liikumise ideed - ümmarguselt või sirgjooneliselt.

Ümmarguse liikumise korral alustaks robot ringi keskpunktist ja liiguks pöörlevalt ümber objekti, suurendades pidevalt raadiust. Kuid selle meetodi puhul oleks uute koordinaatide arvutamise süsteem väga keeruline.

Sirgjoonel liikudes alustab robot mõõdetava objekti nurgal asuvast kindlast punktist ja liigub seejärel mööda X-telge, kusjuures iga uue tsükliga ta liigub lähtepunktist eemale Y-telge. See meetod on palju lihtsam ja kiirem, kuna see ei nõua uute koordinaatide määramiseks keerukaid funktsioone.

Algselt oli kavas, et iga uue tsükliga naaseb robot X-koordinaatide lähtepunkti. Kuid tööprotsessis otsustati, et iga uue tsükli korral alustab robot samast X-koordinaadist, millega ta lõppes, ja läheb vastupidises suunas (vt. joonis 4.1). See kiirendas roboti tööprotsessi tunduvalt, kuna vähendas läbimiseks vajalikku kaugust. Teisest küljest tegi see roboti programmeerimise keerukaks, sest nüüd tuli arvestada sellega, millisel viisil see peaks liikuma. Kui võrrelda protsessi täitmise kiiruse eelist ja mitte väga keerulist liikumissuuna hindamise funktsiooni, on parim variant see, milles iga tsükkel algab saamal koordinaadil X, milles eelmine tsükkel lõppes. Selles programmis üks samm on 1 mm, kuid sammu saab muuta.



Joonis 4.1 Roboti liikumise model

Teine oluline küsimus on, kuidas panna roboti liikuma. Iga koordinaadi fikseerimise ja meeldejätmise roboti jaoks protsess oleks väga pikk ja ebaefektiivne ning raskendas ka kiirete muutuste tegemist. Lisaks soovisin, et roboti liigutused saaksid muutuda sõltuvalt mõõdetava objekti suurusest. Selleks oli vaja kaaluda süsteemi, mis saab ise määrata roboti liikumise koordinaate. Pakett MATLAB oli nendeks eesmärkideks ideaalne variant, sest seal on tsüklike funktsioon. Tarkvara Mitsubishi Melfa Robot Control Toolbox võimaldab viia roboti kindlatele koordinaatidele. Nüüd oli ülesandeks luua tsükkel, et määrata igal sammul uued koordinaadid. Kuna tegemist on sirgjoonelise liikumisega, võtab programm algpunkti ja lisab iga sammuga eelmisele väärtusele ühe X-koordinaadi väärtuse. Niipea kui robot läheneb mõõtmispiirile, lisab programm Y-koordinaadile ühe väärtuse ja liikumissuuna muutmise funktsioon aktiveeritakse. Mõõtmispiirid kehtestatakse eelnevalt. Seda jätkatakse seni, kuni robot läbib kogu mõõtmiseks vajaliku ala.

4.2 Andmete lugemine

Enne iga uuele punktile üleminekut skaneerib programm anduri abil andmeid. Andmed tulevad kahendkoodi kujul. Koodi väärtus on otseselt võrdeline pingega, mille andur edastab analoogsisendisse A1. Koodi maksimaalne väärtus on 1023, see vastab pingele 5 V. Neid andmeid kasutades süsteem saab arvutada pinget, mida andur edastab (vt. valem (3.1)). Rahulikus olekus annab seade 512 bitti (± 10 bitti).

$$U = \text{bitt} * 5/1023, \quad (4.1)$$

kus U – pinge,

bitt - bittide väärtus.

Järgmisena on vaja arvutada magnetiline induksioon andurilt saadud andmete põhjal. Antud anduri pinge ja magnetilise induksiooni graafiku põhjal (vt. joonis 3.2) saab arvutada valemi pinget leidmiseks magnetilise induksiooni kaudu [2]. Teades, et kui magnetilist induksiooni suurendatakse 800 võrra, suureneb pinge 2 V ja et magnetilise induksiooni nullväärtuse korral on pinge 2,5 V, on võimalus luua järgmise funktsiooni (vt. valem (3.2)).

$$U = 0.0025 * B + 2.5, \quad (4.2)$$

kus U – pinge,

B – magnetiline induksioon.

Selle funktsiooni pöördfunktsioon (3.3):

$$B = U * 400 - 1000, \quad (4.3)$$

kus B – magnetiline induksioon,

U – pinge.

Valemite (3.1) ja (3.3) põhjal on võimalik koostada valem magnetilise induksiooni sõltuvuseks programmi sisseloetavast koodist (vt. valem (3.4)).

$$B = 2000 * \text{bitt}/1023 - 1000, \quad (4.4)$$

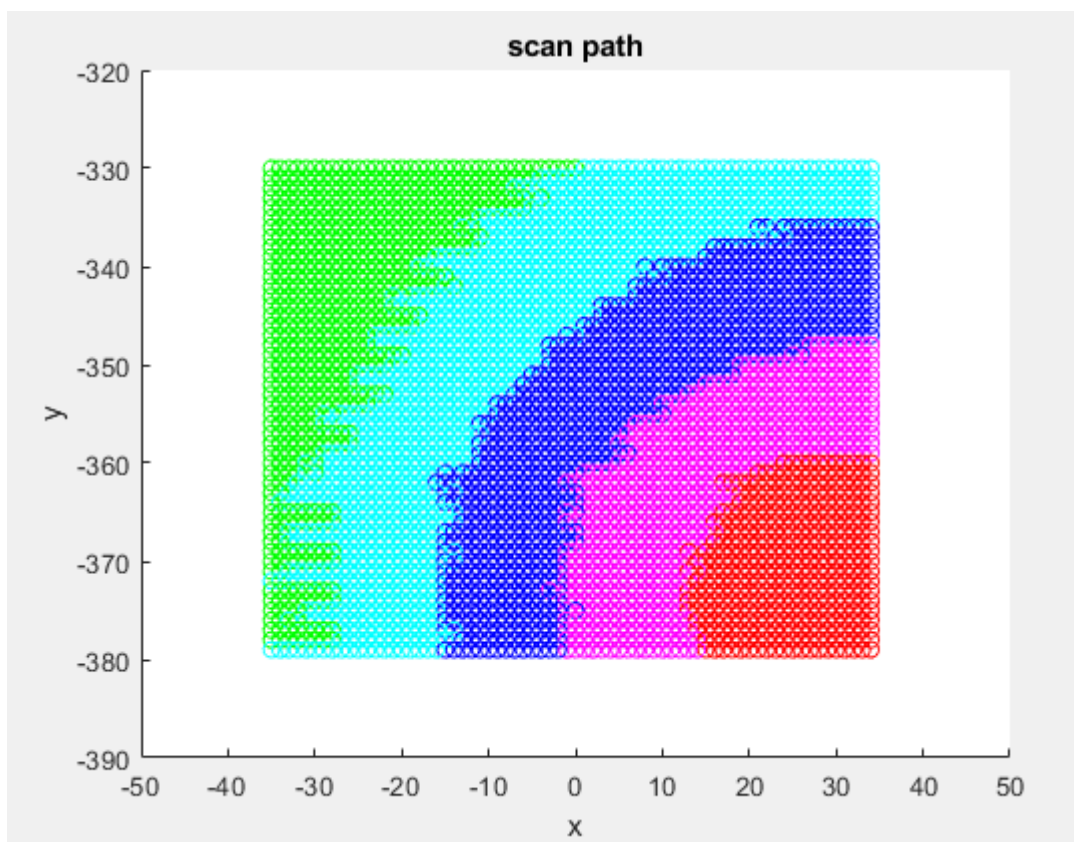
kus B – magnetiline induksioon,

bitt - bittide väärtus.

Magnetilise induksiooni andmed kirjutatakse kahemõõtmelise massiivi, mis pärast programmi lõppu kuvatakse tabeli kujul.

4.3 Värvitoonide valimine

Minu töö üheks oluliseks ülesandeks oli magnetvälja kuvamine ekraanil. Selleks märkisin graafikul erinevate värvidega punktid, mis vastasid koordinaatidele, kus mõõtmine tehti. Värvus sõltub magnetilise induktsiooni tugevusest, mida tumedam on värv, seda suurem on magnetiline induktsioon. Funktsioon, mis vastutab värvivaliku eest, korreleerib just saadud magnetilisi väärtusi antud värvidega. Iga värvi jaoks on määratletud tsoon ja kui magnetiline induktsioon langeb antud tühikusse, kuvatakse see värv ekraanil. Värvitsoon on esitatud protsendina maksimumväärtusest, mida andur suudab lugeda. Värvide väljund graafikul toimub reaajas. Maksimaalset väärtust, värvitsooni ja ka saadaolevaid värve saab muuta.



Joonis 4.2 Tulemuse näide

Pärast magnetilise induktsiooni andmete lugemist naaseb robot algasendisse.

5. ELEKTROMAGNETLÄHIVÄLJA SKANNERI PROGAMM

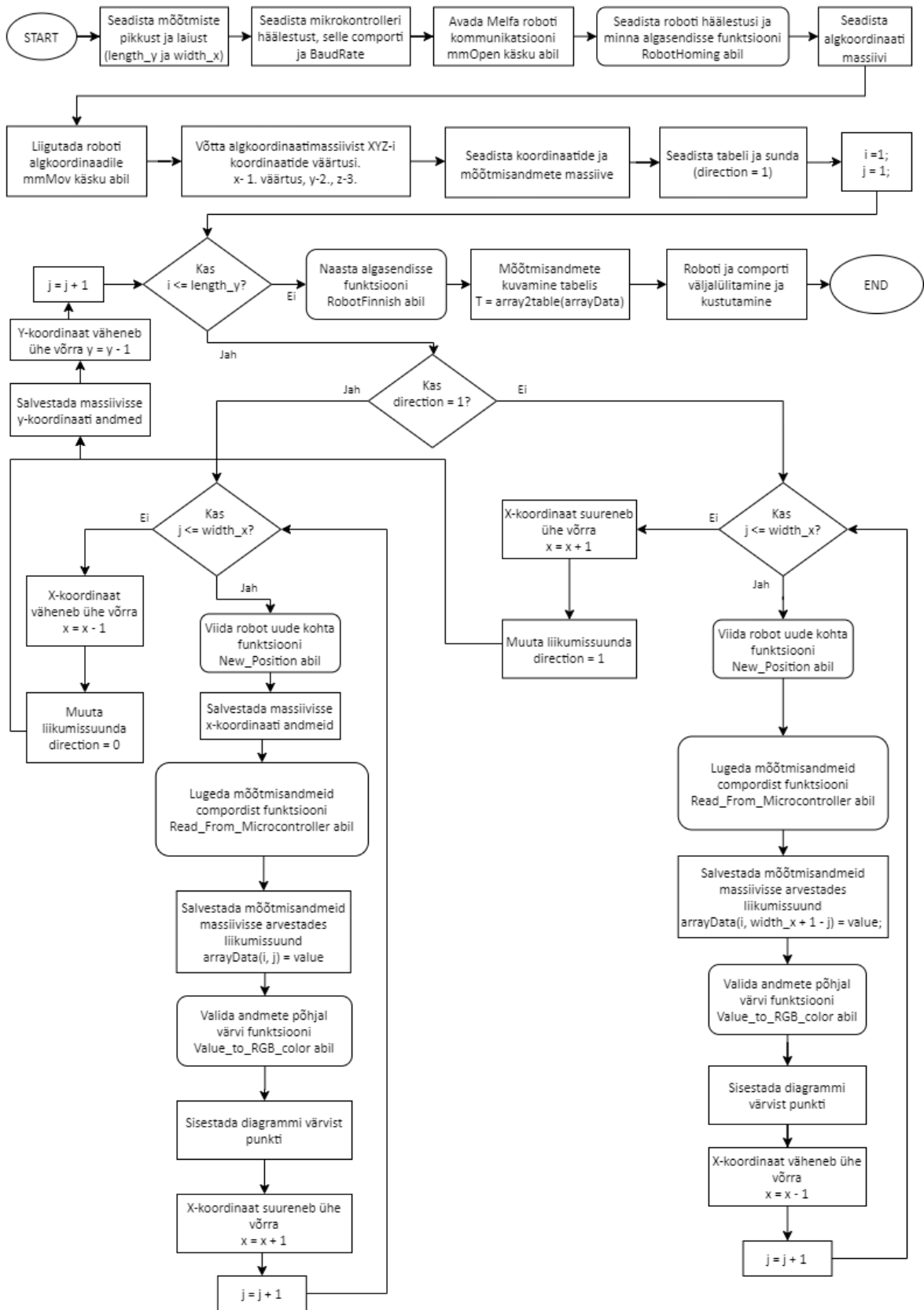
5.1 Põhifunktsioon

Põhifunktsiooniga on teostatud roboti juhtimine. Siin on kirjutatud kood nii liikumise koordinaatide määramiseks kui ka mõõtmisandmete graafikul kujutamiseks.

Funktsiooni alguses seadistakse põhisätted, näiteks mõõtmispiirkonna mõõtmed, mikrokontrolleri ja selle pordi häälestused, ning avatakse side robotiga. Funktsiooni „RobotHoming“ kasutades muudetakse roboti põhisätteid ja viiakse robot tagasi algassendisse. Pärast seda sisestatakse mõõtmiste algkoordinaat ja robot liigub sinna. Kõik koordinaadid sisestatakse kuuekohalisse massiivi, kus kolm esimest väärtust on XYZ-koordinaadid ja teised - tööriista suund. Mõõtmiste suuruste ja esialgse koordinaadi seadmisel tasub meeles pidada roboti töötsooni piire. Seejärel tõmbab funktsioon algkoordinaadist XYZ-koordinaadid ja salvestab need eraldi. Luuakse graafik ja määratakse algne liikumissuund - 1, kui on vaja liikuda piki X-telge, või 0, kui on vaja vastu X-telge.

Pärast seda algab tsükkel, mis teostab koordinaatide muutmise piki Y-telge. Selle sees toimuvad programmi põhitegevused. Iga ringi alguses määratakse roboti liikumise suund piki X-telge. Seejärel algab alamtsükkel, mis teostab koordinaatide muutmise piki X-telge. Selle alamtsükli alguses liigub robot vastavalt praegustele koordinaatidele uude asukohta, kasutades funktsiooni „New_Position“. Selle järel loetakse andmed funktsiooni „Read_From_Microcontroller“ kasutades mikrokontrollerilt ja salvestatakse andmemassiivi. Andmete põhjal valitakse sobiv värvitoon kuvatavale punktile funktsiooni abil. Iga mõõtepunkt kuvatakse diagrammil. Iga alamtsükli ringi lõpus suureneb või väheneb X-koordinaat ühe võrra, sõltuvalt liikumissuunast. Pärast alamtsükli lõppu muutub liikumissuund ja Y-koordinaadi väärtus vähendatakse ühe võrra ja salvestatakse massiivi.

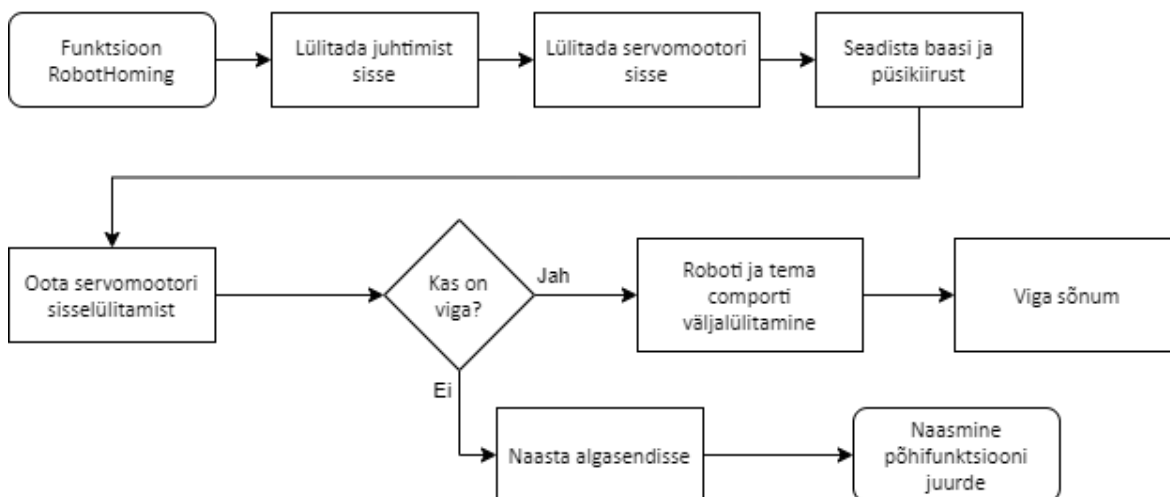
Pärast kõigi tsüklite lõppu naaseb robot funktsiooni „RobotFinish“ kasutades algasendisse. Mõõtmiseandmed kuvatakse ekraanil tabeli kujul. Programmi lõpus lülitatakse robot ja andmesidepordid välja.



Joonis 5.1 Põhifunktsiooni algoritmi plokkskeem.

5.2 Funktsioon RobotHoming

Funktsioon RobotHoming on vajalik roboti tööks ettevalmistamiseks. See lülitab roboti sisse ja viib manipulaatori algasendisse. Siin seatakse roboti põhiparameetrid. Funktsioon lülitab sisse roboti juhtimist ja servomootoreid, seab roboti baasi- ja piirkiiruse. Ta kontrollib, kas sisse lülitamise hetkel on robotil juhtseadmes viga aktiivne. Vea ilmnemisel kuvatakse ekraanil sellekohane teade, kui mitte, naaseb robot algasendisse ja programm naaseb põhifunktsiooni.



Joonis 5.2 Funktsiooni „RobotHoming“ algoritmi plokkskeem.

5.3 Funktsioon New_Position

Funktsioon viib roboti uude asukohta vastavalt antud koordinaatidele ja ootab siis, kuni robot lõpetab liikumise. Funktsioon sobib ainult Descartes'i koordinaatidega liikumiseks.

5.4 Funktsioon Read_From_Microcontroller

Funktsioon loeb USB-COM pordi kaudu mõõtmisandmeid mikrokontrollerilt, millega on ühendatud andur. Andmed arvutatakse ümber valemiga (4.4) ja naastakse põhifunktsiooni. Mõõtetulemus skaleeritakse (normeeritakse) vahemikku 0 kuni 1000 et see oleks suhtelistes ühikutes ja ei sõltuks A/D-muunduri kahendkoodi bittide arvust. Mõõtetulemus antakse seega protsendina maksimaalsest väärtusest. See võimaldab sama MATLAB programmi kasutada koos teiste mikrokontrollerite ja analoog-digitaalmuunduritega. Kuna andur mõõdab bipolaarset signaali, siis asutatakse selles funktsioonis valemit (5.1).

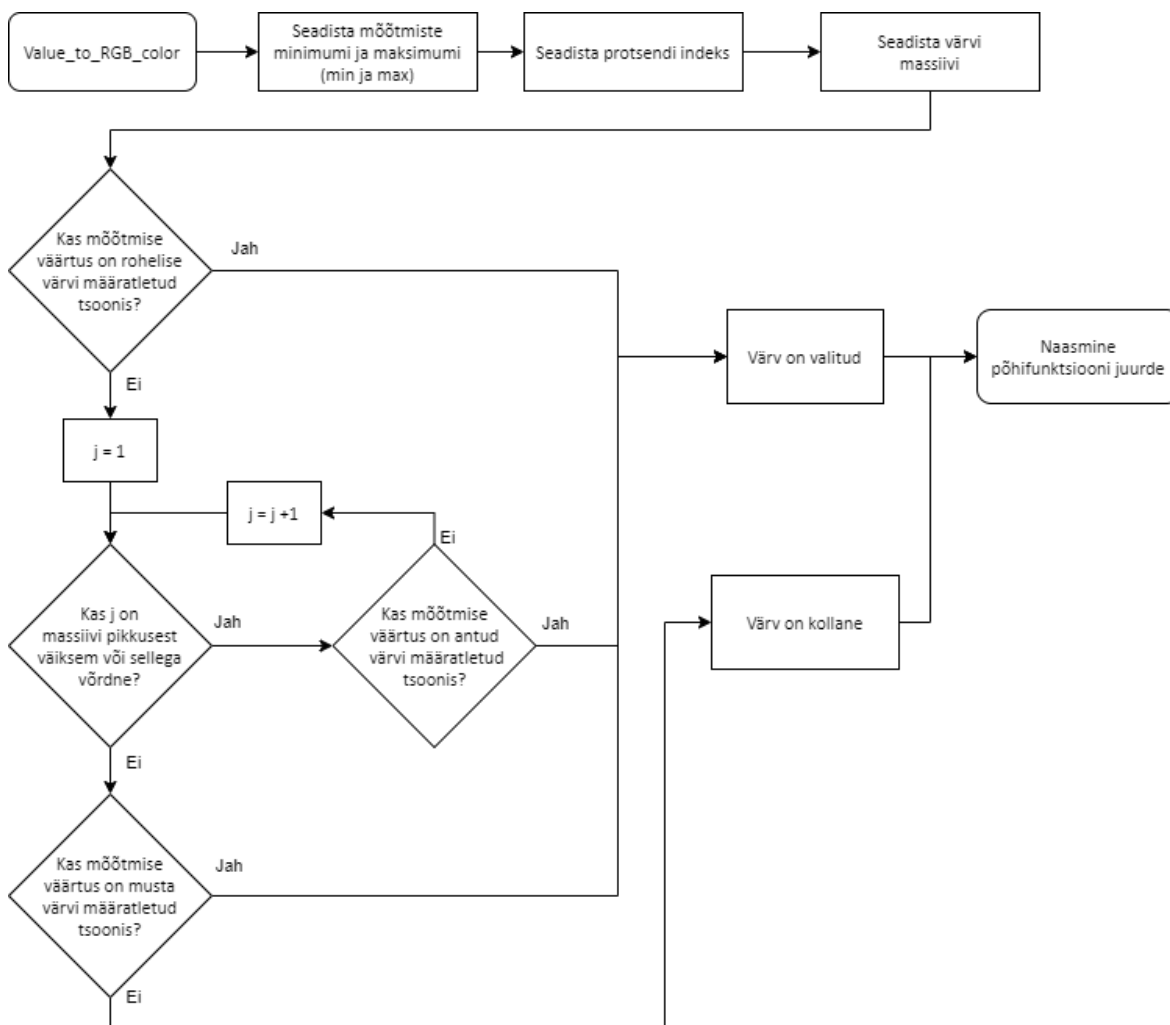
$$B = 2000 * x/1000 - 1000, \quad (5.1)$$

kus B – magnetiline induktsioon,

x – mikrokontrollerist saadud mõõtetulemus ühikuga 0,1 %.

5.5 Funktsioon Value_to_RGB_color

Funktsioon määrab diagrammi punkti värvitooni sõltuvalt mõõdetud andmete väärtusest. Funktsiooni alguses seadistakse indeks ja magnetilise induktsiooni maksimaalne ja minimaalne väärtus, mida andur saab mõõta. Indeks on protsent maksimaalsest väärtusest, mis näitab, kui palju magnetiline induktsioon peaksid muutuma, nii et punkt graafikul muudab värvi. Pärast seda seadistatakse kõigi kasutatud värvide massiiv (palett). Kõige heledam värv ja kõige tumedam pole massiivis. Pärast seda väljastab funktsioon jooksevat magnetilisele induktsiooni väärtusele vastava värvitooni koodi. Niipea, kui jooksevad mõõtmisandmed langevad ühte tsooni, registreeritakse selle punkti jaoks värviväärtus. Kui väärtus ei kuulu ühte tsooni, siis valitakse kollane, see näitab, et sellel punktil on ilmnenud viga. Värviväärtus tagastatakse põhifunktsiooni.



Joonis 5.3 Funktsiooni „Value_to_RGB_color“ algoritmi plokk skeem.

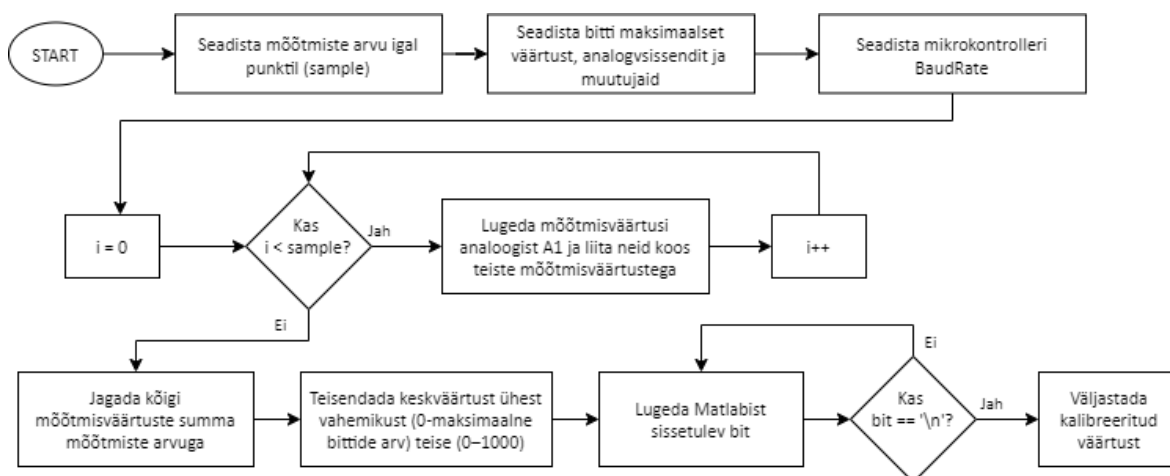
5.6 Funktsioon RobotFinish

See funktsioon paneb roboti võtma mõõteobjekti (magneti) ja naasma algasendisse. Selleks tõuseb ta kõigepealt üles ja läheb ettevalmistavale positsioonile. Kõik see tehakse selleks, et mitte kahjustada andurit, mis on paigaldatud otse roboti haaratsile. Pärast seda ta liigub magneti juurde ja võtab seda. Selles versioonis on robot programmeeritud vabastama mõõteobjekti (magnetit) paar sekundit pärast selle tõstmist, kuid vajadusel saab seda muuta nii, et robot eemaldab magneti konkreesse kohta. Pärast kõigi toimingute tegemist naaseb robot algasendisse.

5.7 Mikrokontrolleri programm

Mikrokontrolleril olev programm loeb andurilt andmeid analoogsisendi A1 kaudu, arvutab jooksva keskmise väärtuse (filtreerib) ja edastab seda väärtust keskkonnas MATLAB töötavale programmile.

Programmi alguses seadistakse põhihäälestused, näiteks mõõtmiste arv igal punktil, maksimaalne bittide arv, analoogsisend ja andmesideliidese seadistused. Igas mõõtmispunkti teeb andur antud arvu mõõtmisi ja arvutab keskmise väärtuse. Pärast seda teisendatakse andmed vahemikust 0 kuni maksimaalsele väärtusele vastav kood vahemikku 0 – 1000. Arvule 1000 vastab 0,1%. Niipea kui tuleb käsk keskkonnast MATLAB, edastatakse kalibreeritud andmed mikrokontrollerilt põhiprogrammi. [18]



Joonis 5.4 Mikrokontrolleri programmi algoritmi plokk skeem.

6. SÜSTEEMI EDASIARENDUSVÕIMALUSED

6.1 Süsteemi eelised ja puudused

6.1.1 Puudused

Sellel süsteemil on kaks peamist puudust: kiirus ja vajadus andurit kalibreerida pärast mitut mõõtmist (anduri magneetumise tõttu püsिमagneti väljas). Tänu andurile on võimalus küll häid pilte elektromagnetlähiväljast, kuid pärast mitut protsessi läheb andur segadusse ega näita täpseid andmeid. Selle probleemi lahendamiseks on vaja läbi viia täielik mõõtmistsükkel ilma magnetita (anduri demagneetimiseks), siis töötab andur jälle nagu peaks. Tulevikus on võimalus kasutada mõnda muud andurit, samuti valmistada kinnitusi.

Teine probleem on kiirus. Mõõtepunktide suure arvu tõttu on robot väga aeglane. Selle probleemi lahendamiseks on võimalus punktide vahelist kaugust suurendada. Punktide vaheline standardkaugus on 1 mm. Testid viidi läbi ka 2 mm kaugusel. Sel juhul töötas robot kaks korda kiiremini, kuid pilt muutus vähem täpseks, kuna mõõdetud punktide arv vähenes.

6.1.2 Eelised

Skannerit saab ühendada ka roboti muude seadmetega. Näiteks kontrolliti katsete ajal skanneri ja haaratsi ühist tööd. Täiusliku koostöö jaoks tasub teha kinnitus, mis aitab erinevatel seadmetel üksteist mitte segada. Tööstusrobotil põhinev skanner saab lisaks mõõtmisele töötada nagu tavaline robot ja teostada muid toiminguid, kasutades selleks oma muid tööriistasid, näiteks tuua ja viia mõõteobjekti.

Skanneri veel üks suur eelis on selle kohandamise võimalused. Mõõtmete suurusi, andmete ekraanil kuvamise parameetreid ja muid sätteid on väga lihtne muuta. Samuti tänu sellele, et mõõtmisandmeid loetakse mikrokontrollerist protsentidena, sai võimalikuks anduri asendamine mõne teisega, isegi nendega, mis on mõeldud teist tüüpi mõõtmiseks.

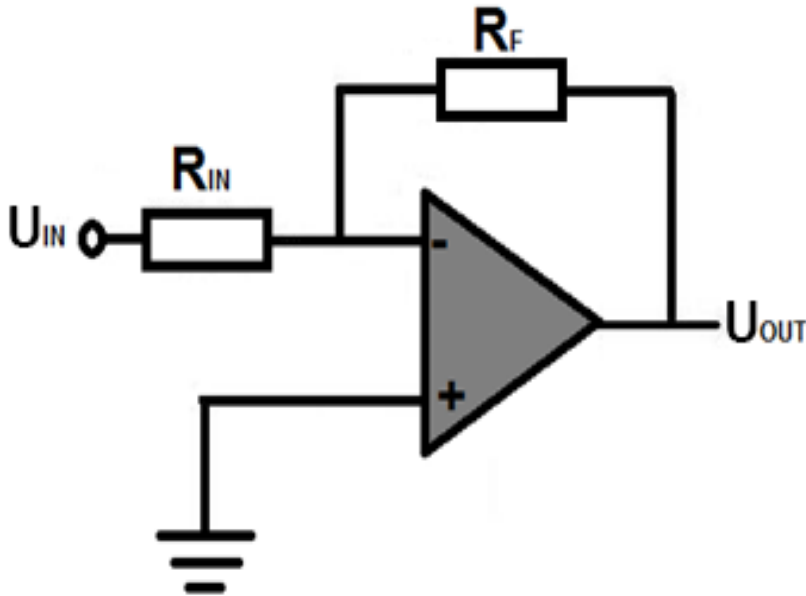
6.2 Andur

Skanneri edasisel arendamisel saab andmete mõõtmiseks kasutada tundlikumat andurit, näiteks SS494B. Selle tööulatus on väiksem, kuid rohkem tundlikkust. See võimaldab tal paremini reageerida muutustele elektromagnetväljas, mis sobib ideaalselt nõrga elektromagnetväljaga objektidega töötamiseks. [2]

Omadused: [2]

1. Mõõtepiirkond (*Magnetic Range, Gauss*): tavaoludes ± 420
2. Tundlikkus (*Sensitivity, mV/Gauss* temperatuuril $T_A = 25^\circ\text{C}$): tavaoludes 5.00

Andurist tuleva pinge võimendamiseks on võimalus kasutada operatsioonvõimendit. Operatsioonvõimendi saab konfigurereida mitteinverteeriva võimendina või inverteeriva võimendina. Inverteeriva operatsioonvõimendi konfiguratsiooni ahelad on näidatud joonisel (6.1).



Joonis 6.1 Inverteeriva operatsioonvõimendi skeem. [19]

Inverteeriva operatsioonvõimendil on sisendtakisti (R_{IN}) ja tagasisidetakisti (R_F). Võimendi suurendab sisendpinget vastavalt võimendustegurile (vt. valem (6.2)). Inverteeriva operatiivvõimendi võimendustegur on takisti R_F ja takisti R_{IN} suhe. Kui takistuse väärtus R_F on võrdne takistuse väärtusega R_{IN} , siis võimendustegur on 1. Kui suureneb takistus R_F , suureneb ka võimendustegur. Teguri saab leida valemi (6.2) abil. [19][20]

$$U_{OUT} = K * U_{IN}, \quad (6.1)$$

kus U_{OUT} – väljundpinge,

U_{IN} - sisendpinge,

K – võimendustegur.

$$K = R_F / R_{IN}, \quad (6.2)$$

kus K – võimendustegur,

R_{IN} – sisendtakisti,

R_F - tagasisidetakisti.

6.3 Kolmemõõtmeline mudel ja töö konveieril

Selles versioonis teeb robot mõõtmisi ainult ühel tasapinnal. Mõõtmispilt kuvatakse kahemõõtmelise graafikuna. Programmil on käsk lugeda Z-koordinaati alguspunktist ning samuti on võimalik robotit pöörata teise nurga alt. Kõik see võimaldab objekti mõõta erinevatest külgedest ja koostada mõõtmiste kolmemõõtmeline pilt. Sel juhul võite teha mitu pilti erinevates koordinaattasapindades.

Samuti on süsteem väga kohanemisvõimeline ja seda saab väga hõlpsalt muuta. On võimalus muuta andurit, mõõtmiste suuruseid, roboti saab asendada teise Mitsubishi Melfa seeria robotiga. Saab panna roboti võtma mõõteobjekte konkreetselt kohast ja pärast mõõtmist neid teise kohta tagasi panema. Kõik see võimaldab kasutada robotit konveieril.

6.4 Kinnitused

Üks olulisi edasiarendamise võimalusi on heade kinnituste loomine, et andur robotil paremini fikseerida. Need peavad olema tugevad ja usaldusväärsed ning robotile hästi kinnitatud, et mitte häirida mõõtmisi liikumise ajal. Need peaksid sobima ka roboti tööriista jaoks, et mitte segada nende kasutamist. Näiteks kui robotile on paigaldatud haaratsid, ei tohiks segada anduri kinnituste liikumist. Kinnitusmaterjalina ei tohiks kasutada metalle, mis võivad magnetväljade mõõtmisi mõjutada.

KOKKUVÕTE

Bakalaureusetöona sai valmis tööstusrobotil põhinev elektromagnetlähivälja mõõtmise skanner. Skanner loodi olemasolevate tööstusrobotite kasutamiseks õpetamise protsessis ning mõõtmisprotseduuride läbiviimiseks. Töö eesmärgini liikumise käigus tuli ühendada robot, mikrokontroller ja andur ühte süsteemi, ette valmistada tarkvara roboti juhtimiseks programmi MATLAB kaudu ja kirjutada programmikoodid roboti ja mikrokontrolleri jaoks.

Esimene osa räägib elektromagnetlähivälja mõõtmistest. See kirjeldab, miks need on kaasaegses maailmas väga vajalikud. Samuti selgitakse seda, millised raskused tekivad uute elektroonikaseadmete väljatöötamise ja elektromagnetlähivälja mõõtmisega.

Bakalaureusetöö teises osas on esitatud skanneri loomiseks kasutatud seadmed. Tutvustati mikrokontrollerit ja andurit ning kirjeldati nende omadustest. Teises osas on näidatud ka elektriskeem ja süsteemi plokk skeem. Samuti on vaadeldud Halli anduri ja eriti lõputöös kasutatud anduri tööpõhimõtet.

Kolmandas osas kirjeldatakse töös kasutatud tööstusrobotit, selle seadistusi, samuti tarkvara roboti juhtimiseks keskkonnas MATLAB. Selles osas kirjeldatakse selle roboti omadusi ja eeliseid, miks tasub seda lõputöökseks valida. Siin räägitakse sellest, kuidas robotit konfigureerida ja tööks ette valmistada, peamistest parameetritest, mida on vaja seadistada, nagu DH parameetrid, liigendite liikumisulatuste piirangust ja andmeside. Samuti kirjeldatakse roboti juhtimiseks vajalikku tarkvara ja selle kõige olulisemaid funktsioone.

Bakalaureusetöö neljas osa tutvustab roboti põhimõtet. See näitab roboti liikumise põhimõtet ja seda, kuidas ta uusi koordinaate valib. Samuti on arvatud valem magnetilise induksiooni ja pingearvutamiseks, kasutades mikrokontrolleri andmeid. Kuvatakse diagrammis kuvatava punkti värvide valimise loogika.

Viiendas osas kirjeldatakse programmi funktsioone, mis koostati nii keskkonnas MATLAB kui ka mikrokontrolleris keeles C. Siin räägitakse nende töö loogikast ja sellest, mille jaoks on iga funktsiooni. Selgitatakse funktsioonide vahelist suhet ja seda, kuidas keskkonnas MATLAB töötav programm ja mikrokontroller üksteisega suhtlevad. Samuti on kirjeldatud funktsioonide algoritmi plokk skeemid.

Viimases osas on esitatud roboti edasised väljavaated, mis tulevikus võib aidata robotit arendada. See kirjeldab anduri värskendamise kahte võimalust. Siin selgitatakse, kuidas teha kolmemõõtmelise mudelit ja mida tähele panna kinnitusdetailide valmistamisel. Samuti on selles osas kirjeldatud süsteemi eeliseid ja puudusi, näiteks kiirus ja vajadus kalibreerida. Roboti

liikumiskiiruse suurendamiseks saab tulevikus kasutada positsioonide etteandmiseks Etherneti võrku RS232 jadapordi asemel, sest sellel robotil on olemas 10Mbit/s kiirusega Etherneti kaart.

Lõputöö eesmärk on peamiselt saavutatud, sest roboti ja mikrokontrolleri tarkvara, töö algoritmid ja elektroonika on koostatud ja katsetatud. Kõik skanneri osad olid omavahel ühendatud ja testitud oli ka skanneri töö. Skanneri väljatöötamist on kavas jätkata, tulevikus on võimalik arendada kinnitusi, et vähendada anduri liikumist roboti töö ajal ja lisada ka funktsiooni, nii et skanner teeb kolmemõõtmelisi mõõtmisi. Seda skannerit saab tulevikus kasutada Tallinna Tehnikaülikoolis roboti juhtimise ja elektromagnetlähivälja mõõtmiste demonstreerimiseks õppelaboris.

SUMMARY

Throughout the course of thesis research, an electromagnetic near-field scanner based on industrial robot was designed and created. This scanner was made with the intention to be used on an existing industrial robot to perform electromagnetic field measurements. In order to achieve the set goal, the robot was combined with a microcontroller and a sensor in one system. Robot control software was written in MATLAB, as well as, other codes for the robot and microcontroller.

In the first part, one is presented with relevant theory behind the electromagnetic field measurements, explaining their significance in the modern world. As well as, the potential difficulties related to new electrical devices and their near-electromagnetic field measurements.

The second part of the research work presents equipment used in the scanner creation, description and characteristics of microcontroller and sensor, as well as, the flowchart and electrical circuit of the system. In this section, a summary of Hall sensor operation principles were also presented.

An overview of the industrial robot used in the research; settings and software allowing its control through MATLAB is given the third part. Here one could also read about the distinctive characteristics and advantages of such robot and why it was chosen for this research work. A concise description of robot settings that need to be installed for proper functioning. Such as DH parameters, joints movement range values limit and data transfer. As far as controlling the robot is concerned, a software that was used to manipulate it and its most important functions are also depicted in this chapter.

In part four, one is introduced with the principles behind rob functioning. How it chooses its path of movement and sets of new coordinates. A new formula was derived to calculate magnetic flux and voltage based on microcontroller data. The thinking behind the colour choice for each point shown in the graph is explained.

Part five describes the software functions both in MATLAB and in the microcontroller using C language, giving the background behind their working principles and their general purpose. Moreover, presenting their interaction with one another and function flowchart algorithm description.

The last part speculates about perspectives for future work that could improve further development of the system. Such as the improvement of two sensors and new 3D scans. It also talks about the advantages and disadvantages of the system such as speed and the need to calibrate. In order to increase the speed of the robot, it is possible to use an Ethernet network

instead of an RS232 serial port to feed positions in the future, because this robot has an Ethernet card with a speed of 10 Mbit / s.

Overall, the aims of the research project were achieved. The software was successfully implemented into the microcontroller, work algorithms developed and electronics tested. All parts of the scanner were interlinked and a full run-through was performed. It is thought, that potential further development of the scanner could be performed. This includes a new scanner attachment mounting, to reduce detector vibration during the scan, implementation of novel functional, allowing to perform measurements in 3D. This scanner could be used for electromagnetic field measurements and as an existing robot functional demonstration in university.

KASUTATUD KIRJANDUS

- [1] Solid State Hall Effect Sensors Datasheet: [võrgumaterjal] Available:
<http://www.farnell.com/datasheets/1676927.pdf>
[Kasutatud 01.05.2020].
- [2] Linear Hall-effect Sensor ICs Datasheet: [võrgumaterjal] Available:
<https://sensing.honeywell.com/honeywell-sensing-sensors-linear-hall-effect-ics-ss490-series-datasheet-005843-2-en.pdf> [Kasutatud 01.05.2020].
- [3] SS496A1 - Hall Effect Sensor, Linear, 1.5 mA, TO-92, 3 Pins, 4.5 V, 10.5 V: [võrgumaterjal]
Available: <https://ee.farnell.com/honeywell/ss496a1/magnetic-sensor-hall-effect-to/dp/2445921> [Kasutatud 01.05.2020].
- [4] Martin Meloun, Tomáš Pajdla, Inverse Kinematics for a General 6R Manipulator. Prague: Czech Technical University, 2013. [Võrgumaterjal] Available:
<ftp://cmp.felk.cvut.cz/pub/cmp/articles/meloun/Meloun-TR-2013-29.pdf> [Kasutatud 19.05.2020].
- [5] SENSITIVE HALL-EFFECT SWITCHES FOR HIGH-TEMPERATURE OPERATION Datasheet:
[võrgumaterjal] Available:
https://www.digchip.com/datasheets/download_datasheet.php?id=58427&part-number=3144
[Kasutatud 01.05.2020].
- [6] Датчики Холла. Виды и применения. Работа и подключения: [võrgumaterjal] Available:
<https://electrosam.ru/glavnaja/jelektrooborudovanie/ustrojstva/datchiki-kholla/>
[Kasutatud 01.05.2020].
- [7] CIROS® Studio – Creating virtual learning environments: [võrgumaterjal] Available:
<https://www.festo-didactic.com/int-en/learning-systems/software-e-learning/ciros/ciros-studio-creating-virtual-learning-environments.htm> [Kasutatud 01.05.2020].
- [8] ATmega16U4/ATmega32U4 Datasheet: [võrgumaterjal] Available:
http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf [Kasutatud 01.05.2020].
- [9] Pro Micro Datasheet: [võrgumaterjal] Available:
<https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/ProMicro16MHzv1.pdf>
[Kasutatud 01.05.2020].

- [10] Arduino Pro Micro: [võrgumaterjal] Available: <https://radioprogram.ru/shop/merch/6>
[Kasutatud 01.05.2020].
- [11] Pro Micro & Fio V3 Hookup Guide: [võrgumaterjal] Available:
<https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/hardware-overview-pro-micro> [Kasutatud 01.05.2020].
- [12] Keyes Pro Micro with Atmega32U4 Datasheet: [võrgumaterjal] Available:
<https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/hardware-overview-pro-micro> [Kasutatud 02.05.2020].
- [13] Павелъев В.Г., Оболенский С.В., Волкова Е.В., Савинов Д.А., Тарасова Е.А., Чурин А.Ю.,
ЭФФЕКТ ХОЛЛА: Практикум, Нижний Новгород: Нижегородский госуниверситет, 2015. –
14 с.: [võrgumaterjal] Available: http://www.lib.unn.ru/students/src/Halls_effect.pdf
- [14] Mitsubishi Industrial Robot RV-1A/RV-2AJ Series Standard Specifications Manual. Versioon
BFP-A8050-F.
- [15] Е.С.Шаньгин, УПРАВЛЕНИЕ РОБОТАМИ И РОБОТОТЕХНИЧЕСКИМИ СИСТЕМАМИ, Уфа,
2015.: [võrgumaterjal] Available: <https://studfile.net/preview/985240/page:9/>
[Kasutatud 03.05.2020].
- [16] Борисов О.И., Громов В.С., Пыркин А.А., Методы управления робототехническими
приложениями. Учебное пособие. СПб.: Университет ИТМО, 2016. — 108 с.:
[võrgumaterjal] Available: <https://books.ifmo.ru/file/pdf/2094.pdf>
- [17] Mitsubishi Melfa Robot Control Toolbox pro MATLAB - Návod použití: [võrgumaterjal]
Available: https://cw.fel.cvut.cz/wiki/help/common/robot_mitsubishimelfa_toolbox
[Kasutatud 04.05.2020].
- [18] EMF Detector: [võrgumaterjal] Available: <http://www.aaronlai.com/emf-detector>
[Kasutatud 10.05.2020].
- [19] Op Amp Gain Calculator: [võrgumaterjal] Available:
<http://www.learningaboutelectronics.com/Articles/Op-amp-gain-calculator.php#answer1>
[Kasutatud 11.05.2020].
- [20] Инвертирующий операционный усилитель: [võrgumaterjal] Available:
<http://kipiavp.ru/pribori/invertiruyushiy-usilitel.html>
[Kasutatud 11.05.2020].

- [21] Measurement of industrial robot pose repeatability: [võrgumaterjal] Available:
https://www.matec-conferences.org/articles/mateconf/pdf/2018/103/mateconf_itep2018_01015.pdf
[Kasutatud 11.05.2020].
- [22] Dr. M.A. Romashchenko, Ma. A.A. Pryakhin, Methodology for scanning data analysis of near electromagnetic field. Voronezh: Voronezh State Technical University, 2013.: [võrgumaterjal]
Available: <https://cyberleninka.ru/article/n/metodika-analiza-dannyh-skanirovaniya-blizhnego-elektromagnitnogo-polya/viewer> [Kasutatud 11.05.2020].
- [23] Dr. Christian Hedayat, “Electromagnetic Near-Field Measurement Techniques”. Fraunhofer Institute for Electronic Nano Systems ENAS: [Võrgumaterjal] Available:
<https://www.enas.fraunhofer.de/en/departments/ASE/forschungsschwerpunkte/electromagnetic-near-field-measurement-techniques.html> [Kasutatud 11.05.2020].

LISAD

Lisa 1 – Roboti juhtimisprogrammi põhifunktsioon

```
function []= Function()
% The main function for controlling the robot. A code written
% here is to determine the coordinates of the movement, as well as
% to illustrate the measurement data on the plot.

% Here are the measurement length and width of the robot along
% the X and Y coordinates.
% Borders can be changed depending on the measured object.
% XYZ coordinates are in millimetres, joint coordinates are in
% degrees.
% X coordinates must be in the range [-80;80] and Y in the range
% [-400;-330]
length_y = 50; %length of measurements
width_x = 80; %width of measurements

% Setting comport data for working with the Arduino board
fclose ('all');
comport = serial('COM4');
set(comport, 'BaudRate', 9600);
% Against error "A timeout occurred during the write operation"
InputBufferSize = 115200; % for max baudrate
Timeout = 5;
set(comport , 'InputBufferSize', InputBufferSize);
set(comport , 'Timeout', Timeout);
fopen(comport);

% Opening the robot communication with default properties
robot = mmOpen('RV1A'); % robot settings
RobotHoming(robot);

% Moving to the starting coordinate
% starting coordinate
startingcoordinate = [-40; -330; 150; 180.0000; 0; 180.0000];
mmMov(robot, 'P', startingcoordinate);
mmWaitForStop(robot);

x = startingcoordinate(1); %x coordinate
y = startingcoordinate(2); %y coordinate
z = startingcoordinate(3); %z coordinate
arrayX = []; % array for X coordinates
arrayY = []; % array for Y coordinates
arrayData = []; % array for measurement data

% Determine the size of the plot
figure
hold on
xlim ([-50 50]) % the scale of the plot is 1.3 m
ylim ([-390 -320]) % the scale of the plot is 1.3 m
xlabel('x'), ylabel('y'), title('scan path')
```



```

% Variable for determining the movement of the robot along
% the coordinate axis X or against
direction = 1;

% the cycle that sets the coordinates, moves the robot,
% reads the values and draws a plot
for i = 1:length_y

    if direction == 1
        for j = 1:width_x
            % Moves the robot to a new position
            New_Position(x,y,robot);
            arrayX(j)= x; % writes X coordinates to an array

            % Reads data, determines color depending on its value
            % and draws a point on the chart
            value = Read_From_Microcontroller(comport); % Reads data
            arrayData(i,j)=value; % writes measurement data to an array

            rgb_color = Value_to_RGB_color(value); % determines color
            h=plot(x,y,'o'); % draws a point in plot
            % Paints a point in the desired color
            set(h, {'color'}, num2cell(rgb_color, 2) );

            x = x + 1;
        end
        direction = 0; %change direction
        x = x - 1;

    else
        for j = 1:width_x
            % Moves the robot to a new position
            New_Position(x,y,robot);

            % Reads data, determines color depending on its value
            % and draws a point on the chart
            value = Read_From_Microcontroller(comport); % Reads
            data
            % writes measurement data to an array
            arrayData(i, width_x + 1 - j) = value;

            rgb_color = Value_to_RGB_color(value); % determines
            color
            h=plot(x,y,'o'); % draws a point in plot
            % Paints a point in the desired color
            set(h, {'color'}, num2cell(rgb_color, 2) );

            x = x - 1;
        end
        direction = 1;
        x = x + 1;

    end
    y = y - 1;
    arrayY(i) = y; % writes Y coordinates to an array
end

```

```
% The robot picks up the magnet and returns to its home position.
RobotFinish(x,y,robot);
% Measurement data is displayed in table form.
T = array2table(arrayData)

% Disabling the communication ports of robot and microcontroller
mmClose(robot);
fclose(comport)
% delete(comport)
clear comport
end
```

Lisa 2 – Funktionsion Value_to_RGB_color

```
function [rgb_color]=Value_to_RGB_color(color)
    % The function determines the color of a point on the plot
    % depending on the value of the measured data.
    %
    % Arguments:
    %   color: value of the magnetic field
    %
    % Returns color of a point

    % maximum value of the magnetic field
    max = 800;

    % minimum value of the magnetic field
    min = -800;

    % index showing the percentage of the maximum value
    % through which the color changes
    index = 0.04;

    % used colors
    colorName = 'y';
    arrayColor = ['c';'b';'m';'r'];

    %the loop determines the well color for a given value
    if (color >= index * min && color <= index * max)
        colorName='g';
    end;
    for j = 1:length(arrayColor)
        if (color >= (j+1) * index * min && color < j * index * min)
            colorName = arrayColor(j);
        end;
        if (color <= (j+1) * index * max && color > j * index * max)
            colorName = arrayColor(j);
        end;
    end;
    if (color >= min && color < (j+1) * index * min)
        colorName='k';
    end;
    if (color <= max && color > (j+1) * index * max)
        colorName='k';
    end;

    rgb_color = colorName;
    return
end
```

Lisa 3 – Funktionsion Read_From_Microcontroller

```
function [value]=Read_From_Microcontroller(comport)
% The function reads the measurement data from
% the microcontroller to which the sensor is connected.
%
% Arguments:
%   comport: Arduino comport data settings
%
% Returns the value of the magnetic field

% writeline(comport,'\n');    % for MATLAB 2019 and up only
fprintf(comport, '\n');

% valuestring = readline(comport); % for MATLAB 2019 and
% up only reads data from comport
valuestring = fscanf(comport);
% disp (valuestring);    %for debugging

value = sscanf(valuestring, '%f');
%counts magnetic field value using comport data
value = value * 5 * 400 / 1000 - 1000;
% disp (value);    %for debugging
end
```

Lisa 4 – Funktionsion New_Position

```
function []=New_Position(x,y,robot)
    % The function moves the robot to a new position.
    %
    % Arguments:
    %   x: x coordinate
    %   y: y coordinate
    %   robot:  robot settings

    mmMov(robot, 'P', [x; y; 150; 180; 0; 180]);
    %wait until finished
    mmWaitForStop(robot);
    % pause (0.001);
end
```

Lisa 5 – Funktions RobotFinish

```
function []=RobotFinish(x,y,robot)
    % The function makes the robot pick up the magnet and return
    % to its home position.
    %
    % Arguments:
    %   x: x coordinate
    %   y: y coordinate
    %   robot:  robot settings

    % Points for raising the magnet and returning home.
    % For better robot movements, joint coordinates are used.
    preparationPos = [-80; 80; 90; 0; -90; 45];
    homePosJoint  = [0; 0; 90; 0; 90; 0];
    capturePos    = [-80; 116; 65; 0; -90; 45];

    % The robot rises up from the last point
    mmMov(robot, 'P', [x; y; 300; 180; 0; 180]);
    % wait until finished
    mmWaitForStop(robot);

    % The robot moves to the preparation point.
    mmMovSafe(robot, 'J', preparationPos);
    mmWaitForStop(robot);

    % The robot moves to the capture point and picks up the object.
    mmMovSafe(robot, 'J', capturePos);
    mmWaitForStop(robot);
    mmExec(robot, 'HCLOSE 1'); %picks up the measured object.

    % Robot moves back to the preparation point and releases the
    % measured object
    mmMovSafe(robot, 'J', preparationPos);
    mmWaitForStop(robot);
    pause(2);
    mmExec(robot, 'HOPEN 1');%releases the measured object

    % The robot moves back to the home position.
    mmMovSafe(robot, 'J', homePosJoint);
    mmWaitForStop(robot);
end
```

Lisa 6 – Funktions RobotHoming

```
function []=RobotHoming(robot)
    % Function in order to prepare the robot for work.
    % Turns on the robot and returns it the home position.
    %
    % Arguments:
    %   robot:  robot settings

    % turn on control of the robot
    mmCntlOn(robot);

    % turn on servomotor of the robot
    mmSrvOn(robot);

    % set base speed coefficient
    mmSetBaseSpeed(robot, 100);

    % set jog speed coefficient
    mmSetJogSpeed(robot, 100);

    homeposition = [160.0000; 0; 568.0000; 180.0000; 0; 180.0000];

    % wait for the servomotor to turn on
    r = mmWaitForServoOn(robot);

    % check if there is an error when turning on the servomotors
    if (r)
        mmClose(robot);
        error('Failed to turn on servomotors.');
```

end

```
    % the robot moves back to the home position
    mmMov(robot, 'P', homeposition);
    mmWaitForStop(robot);
end
```

Lisa 7 – Seadistusklass MATLAB keskkonnas

```
classdef mmRobot_RV1A < mmRobotDef
% MMROBOT_RV1A Robot specification: Mitsubishi RV-1A.
% This configuration class is designed to be 1:1 compatible with
% Mitsubishi Melfa Robot Control Toolbox v1.4 (C)2011, Martin Meloun
% under MATLAB and PC, but is applicable with other software
% platforms, libraries or programming languages.
% Author:                Revision:        Date:
Reason:
% Vladimir Sulzik                1        29.4.2020 16:30
Initial revision

methods

function robot = mmRobot_RV1A()

% Robot name
% -----
robot.name = 'RV1A';
robot.description = 'Mitsubishi Melfa RV1A';
robot.series = 'Melfa IV';

% Parameters of the robot kinematic model
% -----
robot.DOF = 6;
robot.joints = 'RRRRRR'; % Types of manipulator joints
robot.denavitHartenberg = [-pi/2,0,pi/2,-pi/2,pi/2,0; %alpha
                           0, 250, -90, 0, 0, 0; %a
                           0,-pi/2, pi/2, 0, 0, 0; %theta
                           300, 0, 0, 160, 0, 72]; %d

robot.denavitHartenbergParameters = zeros(4,6,'double');

robot.base = eye(4); % no baseframe: 4x4 identity matrix
robot.tool = eye(4); % no tool def.: 4x4 identity matrix
robot.jointLimits = [-150, -60, 60, -160, -90, -200; %min
                    150, 120, 155, 160, 90, 200]; %max

% Using 4 parameters for 6 axis
robot.A76 = mtxTranslate([0; 0; robot.denavitHartenberg(4, 6)]);
```



```

% Parameters for the robot controller serial communication
% -----
robot.com = [];
robot.comtype = 'Serial';
robot.portname = 'COM1';
robot.controller = 'CR1-571';
robot.robotNo = 1;
robot.slotNo = 1;
robot.comprops=mmComprops_Serial(9600,8,'even',2,'on','on', 'CR');
robot.topVariableSign = '<'
robot.pincodes = 0;

% Inverse kinematic solver settings
robot.ikt = @mmIkt_RV6S_Series;
robot.testKinematics = @mmTestKinematics_RV6S_Series;
robot.getflags = @mmGetFlags_RV6S_Series;

% Debugging (debugging printout disabled)
robot.verbose = 0;
robot.thrError = 0;

% HardHome
% -----
robot.hhflag = 2;

% Userdata (none)
% -----
robot.userdata = [];

% Security (none)
% -----
robot.safetyData = [];
robot.secureMovement = 0;
robot.secureSpeed = 0;
robot.secureAccel = 0;

end

end

end

```

Lisa 8 – Mikrokontrolleri programm

```
#define sample 300          // how many samples the device takes per reading

int inPin = A1;           // analog input A1
int val = 0;              // value from analog-to-digital converter
int bits = 1023;
int array1[sample];      // creates an array with number of elements
                        // equal to "sample"

unsigned long averaging; // this variable is the sum of all array values

void setup() {

    Serial.begin(9600);

}

void loop() {

    //reads values fom sensor and fill each element in the array
    for(int i = 0; i < sample; i++)
    {
        array1[i] = analogRead(inPin);           // reads in the values from
                                                // analog input A1

        averaging += array1[i];
    }

    val = averaging / sample;                   //the program divides the sum of
                                                //all values by the number of
                                                //elements "sample"

    uint8_t inByte;
    val = map(val, 0, bits, 0, 1000);           // Re-maps a number from one
                                                // range (0-Maximum bit number)
                                                // to another (0-1000)

    // Asynchronous output values when needed (asked via terminal) - can be
    // tested via serial monitor
    if (Serial.available())
    {
        inByte = Serial.read();

        if (inByte == '\n')
        {
            Serial.println(val);               // outputs calibrated value
        }
    }
    averaging = 0;                             //resets the averaging
}
```