

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kris Ilmar Kristenprun 213466IACB

Simulink keskkonnas koodi genereerimine Arduino platvormile

Bakalaureusetöö

Juhendaja: Andres Rähni

Magistrikraad

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kris Ilmar Kristenprun

14.05.2024

Annotatsioon

Käesoleva töö eesmärgiks oli prototüüpida erinevaid lahendusi Arduino platvormi jaoks kasutades Simulink keskkonda, et modelleerida ja selle põhjal programmikoodi genereerida, kaardistades erinevaid probleeme, mis võivad selle kasutamisel tekkida ning anda ka aluse tuleviku projektide jaoks. Lahendusteks sai valitud alalisvoolumootori kasutamine pöörpendli tasakaalustamiseks ning ka servomootori juhtimine valgusallika järgi.

Töö lõpptulemuseks oli see, et alalisvoolumootori lahendus jäi tegemata ühe tundmatu probleemi tõttu. Selle käigus aga saadi põhjalikult suhelda MATLAB toega, kes vastasid kiirelt ning andsid ka võimaliku lahenduse. Servomootori lahendus sai valmis ilma programmikoodi kirjutamiseta ning andis ka vajaliku kogemuse Simulink keskkonnaga.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 19 leheküljel, 4 peatükki, 16 joonist, 1 tabel.

Abstract

Code Generation For Arduino Using Simulink

The goal of this thesis is to use the Simulink block diagram environment to design different models that are then generated into code for an Arduino microcontroller. All of them have to be designed with minimal handwritten code. During this process, any problems with using the environment are documented, laying the foundation for any future projects using Simulink.

The end result is that one of the model prototypes was not able to be completed. The model was supposed to use a DC motor, but due to an unknown error, the motor would not work. This resulted in communications with MATLAB support, who responded quickly and eventually gave a solution, which unfortunately did not work. The model using a servo motor, however, was successful, without writing any code, which also gave the necessary experience to use Simulink for future projects.

The thesis is in Estonian and contains 19 pages of text, 4 chapters, 16 figures, 1 table.

Lühendite ja mõistete sõnastik

IMU	<i>Inertial Measurement Unit</i> , inertsiaalandur
MATLAB	<i>Matrix Laboratory</i> , programmeerimiskeel ja tarkvarakeskkond
I2C	<i>Inter-Integrated Circuit</i> , kahejuhteline andmete edastamise protokoll
SPI	<i>Serial Peripheral Interface</i> , neljajuhtmelise siini standard
IO	<i>Input/Output</i> , sisend/väljund
RTC	<i>Real-time clock</i> , reaalaega mõõtev elektrooniline seade
PWM	<i>Pulse-width modulation</i> , väljundpinge reguleerimine impulsside laiuse järgi [1]
EOL	<i>End of Life</i> , toode, mille müümine ja uuendamine on lõpetatud

Sisukord

1 Sissejuhatus	9
2 Kasutatud vahendid	10
2.1 MATLAB ja Simulink.....	10
2.1.1 Arduino tugi.....	11
2.1.2 MATLAB ja Simulink versioon	14
2.2 Arduino Mega 2560.....	14
2.3 Adafruit Motor Shield	15
2.4 LEGO EV3 Mootor	15
2.5 Andurid.....	16
3 Simulinkiga prototüüpimine	18
3.1 Alalisvoolumootori lahendus.....	18
3.1.1 Alalisvoolumootori plokk.....	22
3.2 Servomootori lahendus	25
4 Kokkuvõte	27
Kasutatud kirjandus	28
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	31
Lisa 2 – Arduino kontrollrite võrdlus	32

Jooniste loetelu

Joonis 1. MATLAB objekti näide	11
Joonis 2. Näide Arduino plokist ja selle parameetritest	12
Joonis 3. Simulink <i>External mode</i> režiim [6]	13
Joonis 4. Simulink <i>Connected IO</i> režiim [8]	14
Joonis 5. Adafruit Motor Shield v2.3 [11].....	15
Joonis 6. LEGO EV3 alalisvoolumootor [14]	16
Joonis 7. Alalisvoolumootori lahenduse 3 andurit ühel mudelil	19
Joonis 8. ICM20948 kiirenduse salvestamine eri nurkade all	20
Joonis 9. LIS3DH kiirendus mõõtmised paigalseisus	21
Joonis 10. ICM20948 kiirendus mõõtmised paigalseisus	22
Joonis 11. <i>Sensor Fusion</i> ploki müra parameetrid	22
Joonis 12. LEGO EV3 mootor ühendatud Arduino mikrokontrolleriga	23
Joonis 13. Alalisvoolumootori plokk	24
Joonis 14. <i>Connected IO</i> režiimis alalisvoolumootori kasutamisel tekkinud veateade..	24
Joonis 15. Servomootori lahendus.....	26
Joonis 16. Servo ja valgusanduri alamsüsteem	26

Tabelite loetelu

Tabel 1. Simulinki poolt toetatud andurid [16]	17
--	----

1 Sissejuhatus

Tänapäeva seadmetes võib tihti leida väikseid programmeeritavaid mikrokontrollereid. Ühte sellist kontrolleri saab kasutada selleks, et süsteemi juhtida. Seadme arendaja saab terve süsteemi koodi ise valmis kirjutada ning selle mikrokontrolleri peale laadida, aga arvestades seda, kui keerukad tänapäeva süsteemid olla võivad ning kui mitu rida koodi nende programmeerimiseks vaja läheb, siis see ei ole efektiivselt lähenemis viis. Parem lahendus oleks süsteem visuaalselt mudelleerida kasutades selleks loodud tarkvara ning siis lasta sellel tarkvaral loodud mudeli järgi kood genereerida, mida saab seejärel mikrokontroller kasutada.

Üks selline tarkvara on Simulink, mis on osa MATLAB keskkonnast ning mis võimaldab visuaalselt modelleerida erinevaid süsteeme, kasutades nii sissehitatud plokkide, kui ka kasutaja enda poolt loodud plokkide. Lisaks pakub Simulink mitmeid allalaaditavaid lisasi, mis näiteks lasevad kindla riistvara jaoks mudeleid luua. Üks nendest lisadest on *Simulink Support Package for Arduino Hardware*, mis võimaldab kasutada mudelite loomisel Arduino riistvara jaoks mõeldud plokkide ning ka koodi genereerida ja seda Arduino mikrokontrolleri peale laadida.

MATLAB pakub Simulinki kasutamiseks mitmeid erinevaid juhendeid ja kursuseid, mis aitavad keskkonna kohta õppida. Seetõttu võib pidada Simulinki algajatele sobilikuks töövahendiks. Selle töö eesmärgiks on seda uurida, prototüüpides erinevaid lahendusi Arduino platormi jaoks, mis näitavad lihtsamaid automaatjuhtimise põhimõtteid. Kõik lahendused saab tehtud kasutades Simulinki ning võimalikult palju nendest peaksid saama lahendatud ilma käsitsi programmikoodi kirjutamiseta. Õppeprotsessi käigus saab kaardistatud erinevad probleemid, mis võivad tekkida ning saab loodud alus, mille põhjal võib tulevikus Simulinki projektideks kasutada.

2 Kasutatud vahendid

Peamisteks töövahenditeks oli MATLAB, selle alla käiv Simulink, Arduino mikrokontroller ning *Adafruit Motor Shield*. Valitud sai kaks näiteülesannet: alalisvoolu mootori ja servomootori juhtimine. Alalisvoolu mootor tasakaalustab pöördpendlit ning servomootor muudab enda asendit valgustugevuse järgi. Nende lahenduste realiseerimiseks sai valitud erinevad IMUd ja ka üks valgusandur.

2.1 MATLAB ja Simulink

MATLAB on MathWorks'i poolt loodud programmeerimiskeel ja tarkvarakeskkond, mis annab kasutajale graafilise liidese, mille abil saab sama nimelise keelega skripte kirjutada [2]. Selle peamine kasutusala on erinevate süsteemide disainimine, loomine ja analüüsimine. Seda tihti kasutatakse erinevate teadusalade ekspertide ja inseneride poolt, tänu mitmekesisele teekite kogumile, mis pakuvad funktsioone andmete analüüsiks, sardsüsteemide ja automaatjuhtimissüsteemide arenduseks, algoritmide loomiseks ja kasutamiseks jne [2].

MATLAB keskkonnaga saab ka kasutada lisa tarkvara nimega Simulink, mis on visuaalne modelleerimiskeskond, mida saab kasutada süsteemide disainimiseks ja simuleerimiseks [3]. Mudelite põhjal on ka võimalik programmikoodi genereerida C, C++ või mõnes muus programmeerimiskeeles, mille saab pärast genereerimist otse riistvara peale üleslaadida [3]. Tänu sellele, et Simulink on MATLAB keskkonna peale ehitatud, saab MATLAB keeles kirjutatud käsklusi ja skripte Simulink mudelite loomises kasutada. Peale kasutaja poolt kirjutatud koodi, on võimalik ka kasutada olemasolevaid nn „plokke“, mis on visuaalselt esitatud funktsioonid, millel on tihti üks või mitu sisendit ja väljundit. Ühe sellise ploki väljundi saab teise ploki sisendiks ühendada ning selliselt mitme plokiga tehes on võimalik mudeli luua. Lisaks on olemas ka erinevad allalaaditavad lisad, mis võimaldavad kasutada rohkem plokke. Näiteks on olemas Arduino mikrokontrollerite jaoks loodud lisa, tänu millele saab juhtida Arduino seadmeid kasutades Simulink plokke. Plokke saab ka kasutaja ise luua, kirjutades MATLAB programmeerimiskeele abil funktsioone. Selle töö üheks tingimuseks on aga Simulinki

kasutamine ilma programmikoodi kirjutamiseta, mistõttu on töö käigus kasutatud ainult Simulinkis olemasolevaid plokkke.

2.1.1 Arduino tugi

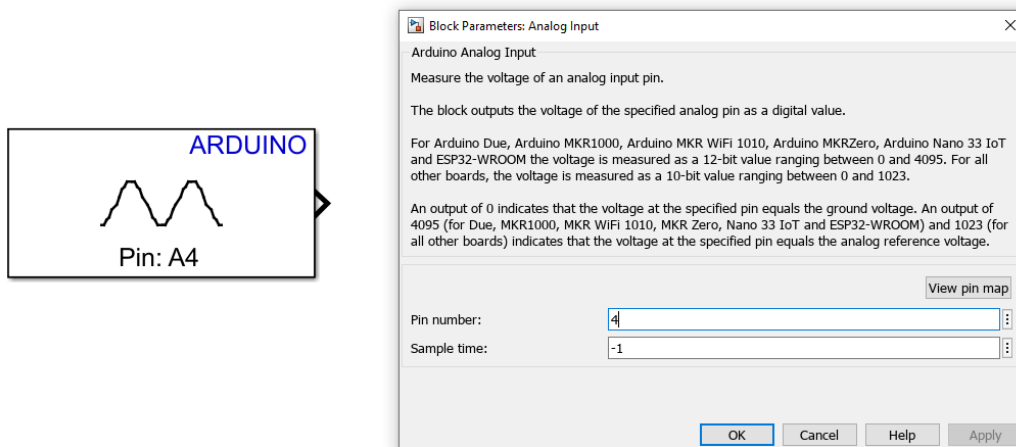
Simulinki ja MATLABi saab kasutada Arduino mikrokontrollerite juhtimiseks. Mõlemas keskkonnas on olemas selle jaoks loodud tarkvaralisa. MATLABis on selleks *MATLAB Support Package for Arduino Hardware* ning Simulinkis on selleks *Simulink Support Package for Arduino Hardware*. Nende kasutamiseks on vaja ühendada arvuti Arduino mikrokontrolleriga kas USB, Wi-Fi või Bluetoothiga [4]. Seejärel saab kasutada MATLABi või Simulinki, et suhelda erinevate seadmetega läbi näiteks I2C või SPI siini ning lugeda andmeid erinevatest anduritest nagu näiteks IMUdest [4]. Tarkvaralisa toetab ka *Adafruit Motor Shield* laiendusplaati [5].

MATLAB keskkonnas käib Arduino riistvaraga suhtlemine läbi käskude. Neid käske saab kirjutada ükshaaval MATLAB käsurea abil või saab mitu käsku korraga kirjutada kasutades skriptifaili. Mikrokontrolleri ja teiste seadmetega ühendamisel luuakse struktuurne muutuja, mis koosneb seadme erinevatest omadustest. Sellist muutujat kutsutakse MATLABis objektiks. Näide sellest on Joonis 1, kus on näha Arduino mikrokontrolleriga ühendamisel tekkinud objekti nimega „a“. Sellel objektil on erinevad atribuudid, mis vastavad arvutiga ühendatud Arduino Mega 2560 mikrokontrolleri omadustele. Objekti muutuja „a“ saab teistele käskudele argumendina juurde anda ehk objektis kirjeldatud informatsiooni saavad teised funktsioonid lugeda ja ka vajadusel üle kirjutada.

```
a =  
  
    arduino with properties:  
  
        Port: 'COM4'  
        Board: 'Mega2560'  
        AvailablePins: {'D2-D53', 'A0-A15'}  
        AvailableDigitalPins: {'D2-D53', 'A0-A15'}  
        AvailablePWMPins: {'D2-D13', 'D44-D46'}  
        AvailableAnalogPins: {'A0-A15'}  
        AvailableI2CBusIDs: [0]  
        AvailableSerialPortIDs: [1, 2, 3]  
        Libraries: {'I2C', 'SPI', 'Servo'}
```

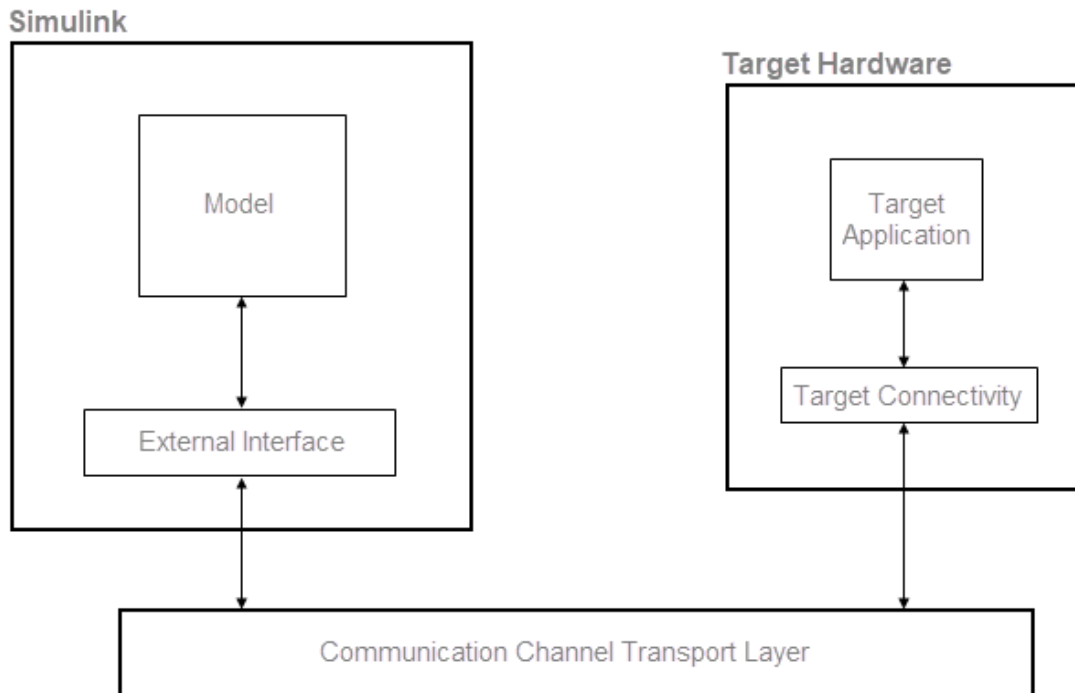
Joonis 1. MATLAB objekti näide

Simulink keskkonnas kasutatakse Arduino seadmete juhtimiseks plokkide. Plokkidel on olemas parameetrid, mis võimaldavad seadme omadusi muuta. Joonisel 2 on näidatud Arduino analoogsisendi plokk ja selle muudetavaid parameetreid. See plokk loeb andmeid parameetris etteantud analoogsisendist. Kui näiteks *Pin number* on 4, siis loetakse andmeid analoogsisendist A4. Parameetrite akna sees on ka plokk kirjeldus ja parameetrite seletused.



Joonis 2. Näide Arduino plokkist ja selle parameetritest

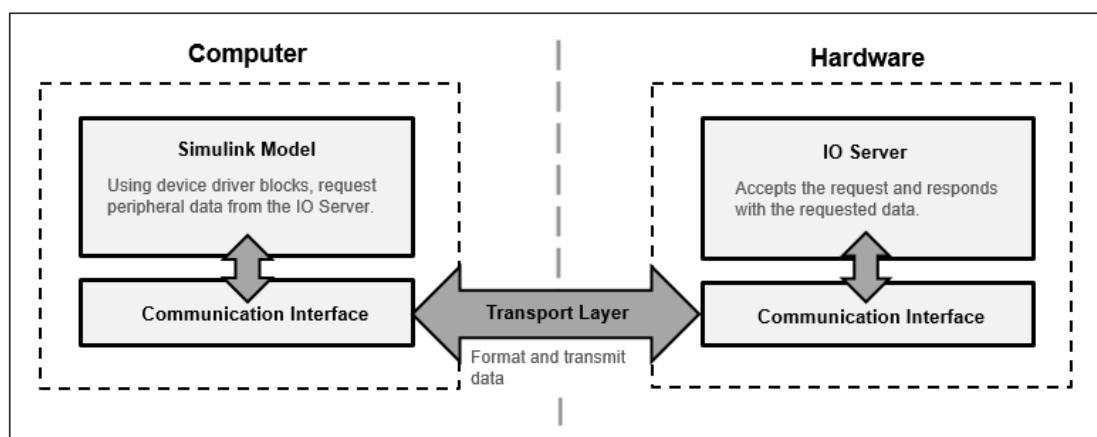
Pärast mudeli loomist saab selle põhjal programmikood genereerida ja see Arduino peale üleslaadida. Üleslaadimiseks on mitu võimalust. Esimene võimalus on kasutada välist režiimi ehk *external mode*. Selles režiimis luuakse andmevahetuskiht arvuti ja Arduino vahel, mis tegeleb üleslaetud programmiga (vt Joonis 3). Selle kihiga suheldes saab kasutaja muuta plokkide parameetreid ning jälgida ja salvestada signaale reaalsajas [6].



Joonis 3. Simulink *External mode* režiim [6]

Teine võimalus on luua eraldiseisev programm, mis laetakse Arduino mikrokontrolleri peale ning mis töötab isegi siis, kui kontroller ei ole arvutiga ühendatud. Programm töötab nii kaua, kuni Arduino on välja lülitatud või kuni kasutaja on ise programmi peatanud [7].

Olemas on ka režiim nimega *Connected IO (inputs/outputs mode)*. Režiim loob liidese, mis võimaldab Simulink mudelil ja riistvara peal oleva IO serveril üksteisega suhelda [8] (vt Joonis 4). See tähendab, et programmikoodi genereerimine ja üleslaadimine ei ole vajalik, sest liidese kaudu saab Simulink riistvara sisendite ja väljunditega otse suhelda, ilma, et programm töötaks. Kui plokk vajab mingisuguseid andmeid, siis neid saab tänu liidesele IO serverilt küsida. Andmeid saadetakse aga ainult siis, kui küsitakse ning simulatsioon töötab ainult kasutaja arvuti peal, mitte riistvara peal ehk kõiki simulatsiooni parameetreid ja andmeid ei saa täiuslikult reaajas jälgida, nagu on võimalik välises režiimis [8].



Joonis 4. Simulink *Connected IO* režiim [8]

2.1.2 MATLAB ja Simulink versioon

MATLABi ja Simulinki uuendatakse kaks korda aastas. Üks uuendus toimub märtsikuu vahemikus ning teine septembrikuu vahemikus. Erinevaid versioone nimetatakse selle aasta järgi, millal see versioon avaldati ning sellele on ka lisaks juurde pandud kas täht „a“, mis tähendab märtsikuu uuendust, või „b“, mis tähendab septembrikuu uuendust [9].

Selles töös sai kasutatud MATLAB ja Simulink versiooni 2023b ehk 2023. aasta septembris avaldatud versioon. Töö käigus avaldati ka uuem versioon, milleks on 2024a, aga kuna töö sai alustatud 2023b versiooniga, siis uuemat versiooni kasutusele ei võetud.

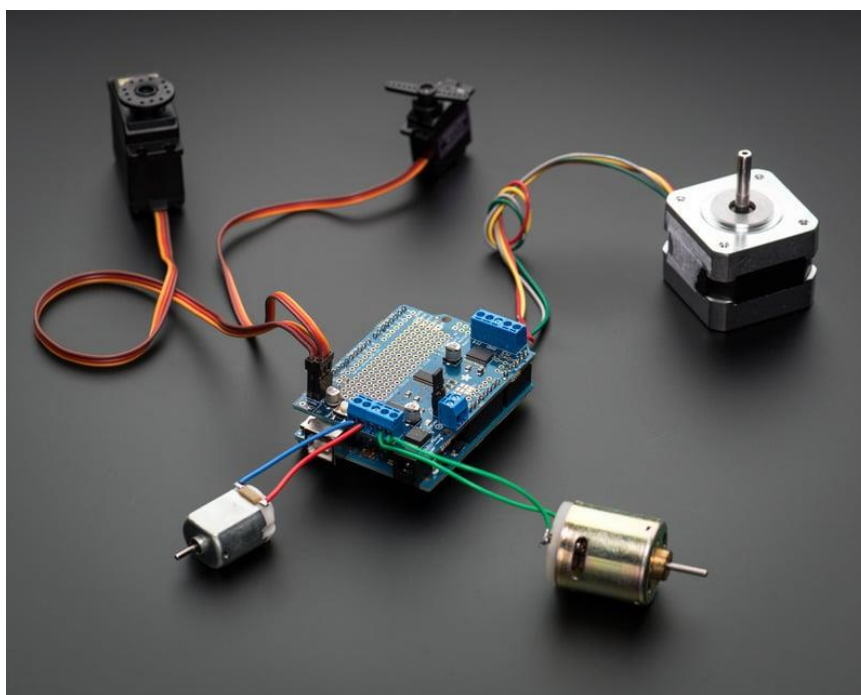
2.2 Arduino Mega 2560

Arduino mikrokontrollereid on mitmeid ning mõned on oma funktsionaalsuse suhtes lihtsamad ning mõned on keerulisemad. Selle töö jooksul aga ei olnud vajalik näiteks sissehitatud Bluetooth, RTC või Wi-Fi, mida lihtsamad kontrollid ei paku, aga mida edasijõudnutele mõeldud kontrollid pakuvad. Seetõttu jäi valikusse kontrollid nagu Uno, Nano ja Mega, millel on vähem funktsioone, aga on ka rohkem levinumad ja on sobilikumad sellisteks õppeesmärgiga projektideks [10]. Täpsemaid võrdlusi on näha Lisas 2. Nendest sai valitud Arduino Mega 2560, sest sellel on teistest suurem programmimälu ning sellel on ka kõige rohkem sisend-väljund viike. Ainsaks negatiivseks küljeks on selle suurus ja raskus, mis on rohkem, kui teistel kontrollritel.

2.3 Adafruit Motor Shield

Kasutusele sai ka võetud *Adafruit Motor Shield v2.3*, mis on Arduino jaoks mõeldud laiendusplaat (vt Joonis 5). Seda saab kasutada alalisvoolu- ja servomootorite juhtimiseks.

Mootoreid juhib sellel PWM kiip (PCA9685) läbi I2C siini [11]. Korraga võib olla ühendatud kuni 4 alalisvoolumootorit ning kuni 2 servomootorit, aga kui mitut laiendusplaati kokku panna, siis on võimalik rohkem mootoreid ühendada. Selle töö jaoks on aga vaja ainult ühte alalisvoolumootorit ja ühte servomootorit ühendada, mistõttu ei ole vajalik rohkem kui ühte *Motor Shieldi* kasutada.

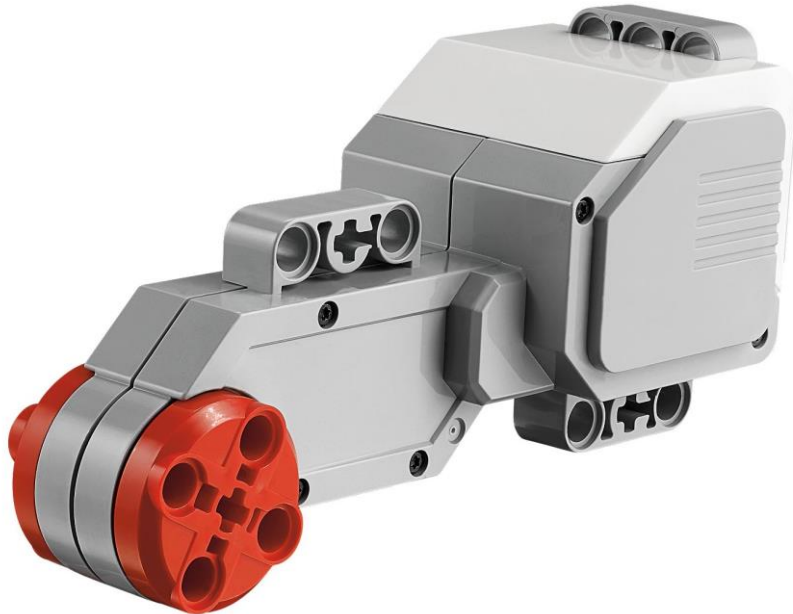


Joonis 5. Adafruit Motor Shield v2.3 [11]

2.4 LEGO EV3 Mootor

Alalisvoolumootoriks sai valitud LEGO EV3 suur mootor (vt Joonis 6). Mootori valiku juures sai peamiselt tähele pandud sellele, kas on olemas enkooder, mis võimaldab mootori hetke positsiooni tuvastada ja kasutajale väljastada. LEGO NXT mootorid olid selleks sobilikud, tänu sellele, et nendel on selline enkooder olemas [12]. EV3 mootor on

väga sarnane NXT mootorigega, aga *Motor Shieldi* kasutades on EV3 tunduvalt kiirem, kui NXT, mistõttu sai valitud EV3 [13].



Joonis 6. LEGO EV3 alalisvoolumootor [14]

2.5 Andurid

Andurid sai peamiselt valitud selle järgi, millised on Simulinki poolt toetatud (vt Tabel 1). Igal toetatud anduril on olemas plokk Simulinkis, mis võimaldab seda andurit mudelites ja programmides kasutada.

Kuna pöördpendli tasakaalustamisel on vaja positsiooni tuvastada, siis selleks sobib kõige paremini IMU ehk inertsiaalandur. Inertsiaalandur kombineerib kiirendusanduri, güroskoopanduri ning mõne IMU puhul ka magnetväljasensori [15]. Nende andurite andmete põhjal on võimalik määrata ära kui kiirelt mingi objekt liigub ning missuguses positsioonis see objekt on.

IMU võib olla kas 6-teljeline või 9-teljeline. 6-teljelise puhul on kolm omavahel ristuvat kiirendusandurit, mis arvutavad kulgliikumise kiirust ja suunast x-, y- ja z-telje sihis ning

samamoodi paigutatud kolm güroskoopandurit, mis mõõdavad nurkkiirendust [15]. 9-teljelise puhul on juures magnetväljasensor ehk kompass. Kuna kiirendusanduri on juba võimalik kulgliikumine kätte saada, siis saab ka kasutada 3-teljelist andurit IMUde asemel, kuigi sellise anduriga kaasnevad mürad võivad probleeme tekitada. Sellegipoolest sai valitud katsetuseks 3 andurit: 3-teljeline kiirendusandur LIS3DH, 6-teljeline IMU LSM6DSOX ning 9-teljeline IMU ICM20948.

Tabel 1. Simulinki poolt toetatud andurid [16]

MPU IMUd	LSM6DS IMUd	BNO055 IMU	APDS9960 valgusandur
LPS22HB rõhuandur	HTS221 niiskusandur	ADXL34x kiirendusandur	CCS811 õhukvaliteedi andur
ICM20948 IMU	LIS3DH kiirendusandur	BME68x gaasiandur	BMP280 rõhuandur
OV2640 kaamera	ADIS16505 IMU	VL53L0X <i>time-of-flight</i> andur	Pixy2 kaamera

ICM20948 sai valitud seetõttu, et see on asenduseks MPU9250 ja MPU6050 anduritele, mida võib tihti leida sarnastes pöördpendli tasakaalustamise lahendustest. Üks näide on MinSeg komplektid, mille seas näiteks *MinSegMega V4* [17] ja *MinSegShield M2V5 Dual Axis* [18] mõlemad kasutavad neid andureid. MPU9250 ja MPU6050 on aga EOL ning nende asenduseks on ICM20948 [19].

LIS3DH sai valitud selleks, et katsetada 3-teljelist kiirendusandurit selle ülesande puhul ning näha, kui palju efektiivsem on IMUde kasutamine. LSM6DSOX sai valitud seetõttu, et seda pakub Adafruit, kelle plaatide peal on pingeregulaatorid ja pingemuundurid, mis lubavad kasutada 5 V või 3 V toidet, kuigi anduri maksimum sisend-väljund toitepinge on väiksem [20]. Teised valitud andurid on samuti pakutud Adafruiti poolt, mistõttu on see lisa positiivne aspekt nende valimisel.

Peale IMUde, pidi valima ka servomootori lahenduse jaoks valgusanduri. Selleks oli APDS9960, mis on ainus valgusandur, mida toetab Simulink.

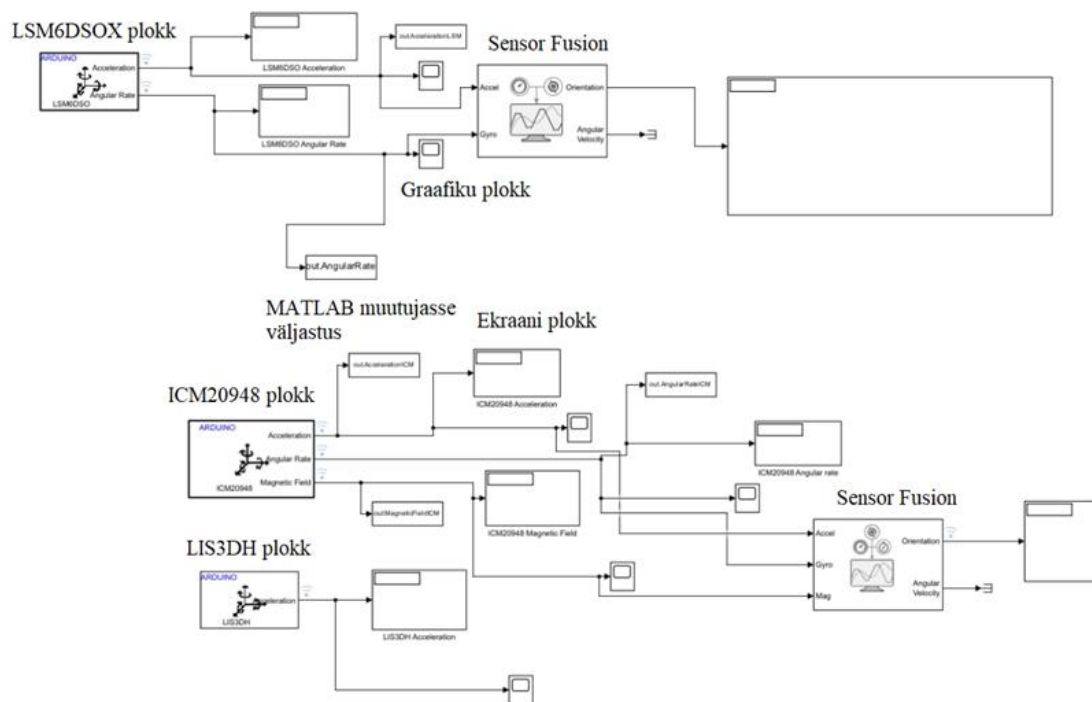
3 Simulinkiga prototüüpimine

Prototüüpimiseks sai valitud kaks lahendust. Üks on pöördpendli tasakaalustamine kasutades alalisvoolumootorit. Teine on servomootori juhtimine, suunates seda suurima valgusallika poole.

Enne prototüüpimisega alustamist sai läbi tehtud Simulinki *Onramp* kursuse, mis on algajatele loodud tarkvarasisene kursus. Selle kaudu oli võimalik õppida, kuidas plokkide mudelisse asetada, kuidas ühendusi erinevate plokkide vahel luua, millised plokkid olemas on ning andis ka näiteid sellest, kuidas neid plokkide saab kasutada erinevate algoritmide realiseerimiseks. Pakutud olid ka rohkem kursusi, mis käsitlesid teisi teemasid, nagu näiteks tagasisidestatud juhtimist, mis sai samuti tehtud alalisvoolumootori lahenduse ettevalmistuseks.

3.1 Alalisvoolumootori lahendus

Esialgset oli vajalik kõiki andureid katsetada, asetades nende plokkid Simulink mudelisse (vt Joonis 7). Iga anduri väljund on ühendatud ekraani plokkiga, mis näitab väljundi väärtust. Lisaks on ka ühendatud graafikute (*Scope*) plokkidega ning ka on väljastatud MATLAB muutujatena. Kasutusele on ka võetud Simulinki poolt pakutud *Data Inspector*, mis võimaldab signaali väärtusi salvestada graafikusse ning siis graafikut omakorda salvestada. *Data Inspector* on peamine töövahend, millega nende andurite tulemusi sai salvestatud.



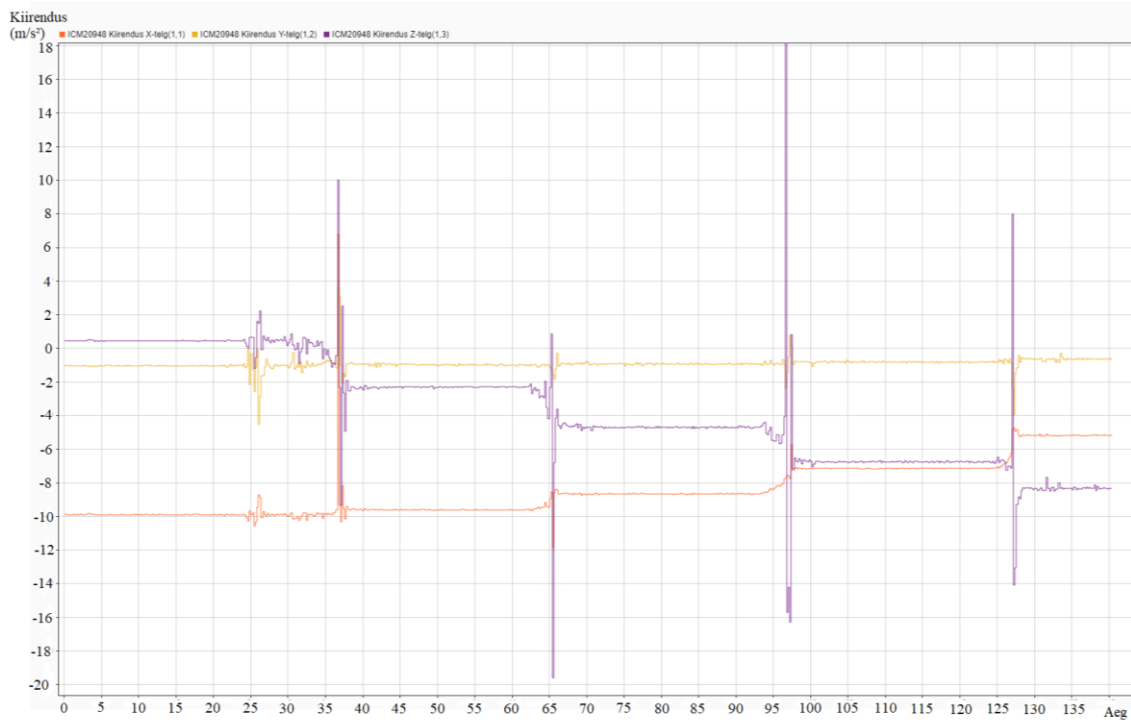
Joonis 7. Alalisvoolumootori lahenduse 3 andurit ühel mudelil

IMUde puhul on ka kasutusel *Sensor Fusion* plokid, mis kasutavad kiirendusanduri, güroskoopanduri ja kompassi andmeid, et väljastada orientatsiooni maatriksi. Selle maatriksi põhjal on võimalik määrata, mis asendis on hetkel mingi kolme dimensiooniline objekt.

Andurite töötamise mõistmise eesmärgil sai peamine tähelepanu peale pandud kiirendusele. Näiteks ICM20948 puhul sai mõõdetud kiirendus nelja eri nurga all, kui esialgne asend on vertikaalselt püsti seisev. Algne asend on 0 kraadi, teine on 20 kraadi, kolmas on 40 kraadi ning viimane on 60 kraadi. Anduri asend sai muudetud umbes iga 5 sekundi tagant. Tulemused said salvestatud kasutades *Data Inspectorit* (vt Joonis 8). X-telg on ajatelg, mis kasutab Simulinki enda ajaiühikuid. 5 sekundit vastab 30 ajaiühikule.

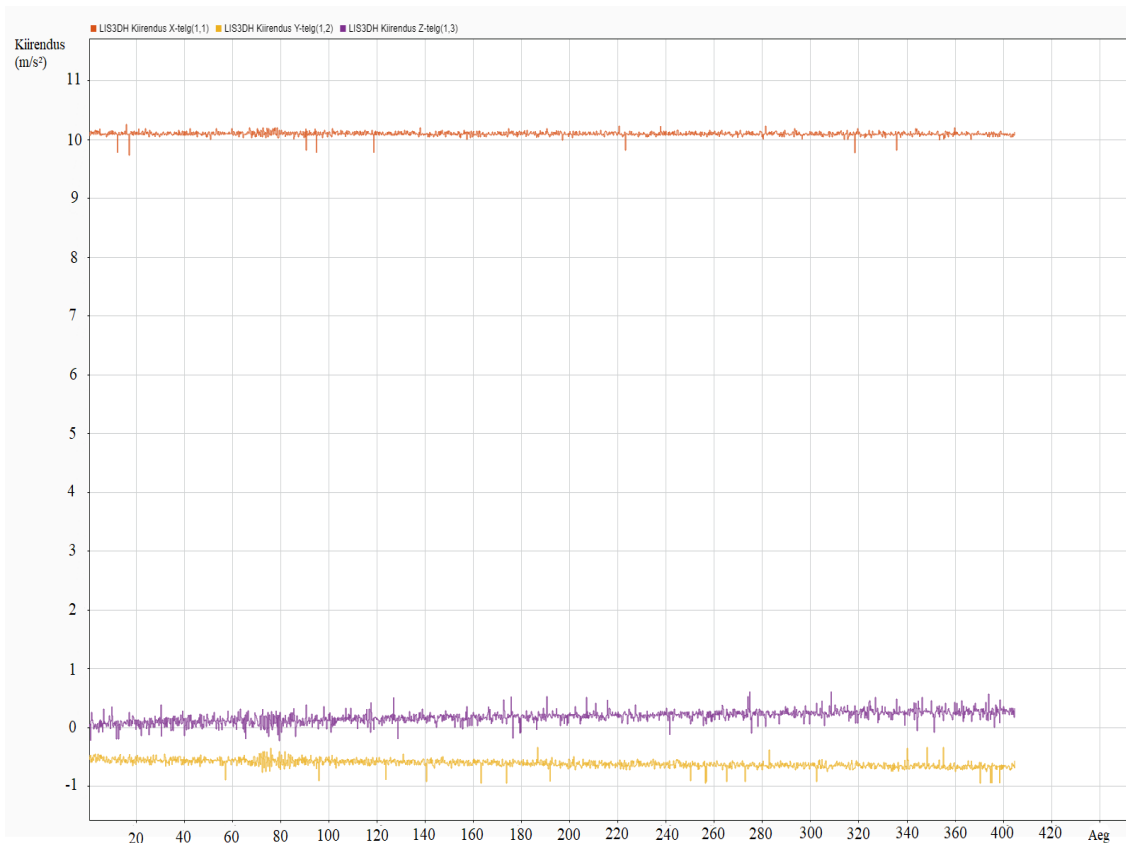
Jooniselt on näha, kuidas anduri asendit muutes toimub suur kiirenduse muutus. Näha on ka seda, et kiirendus ei ole igal teljel alati 0, kuigi andur on enne asendi muutmist liikumatu. Näiteks X-telje kiirendus on püsti seises -10 lähedal. Nurga alla pannes see kiirendus suureneb. Z-telje kiirendus puhul aga asendit muutes kiirendus väheneb. Näha on seda, et telgede kiirendused on nihutatud nulli suhtes ning see nihe on seotud anduri asendiga. Seda nähtust saab seletada gravitatsioonijõu kaudu. Anduri mõõtmisi mõjutab raskuskiirendus ($9,8 \text{ m/s}^2$) vertikaalil [21]. Mida rohkem telje kalle vertikaali suhtes

kaugeneb, seda vähem raskusjõud selle telje mõõtmistulemust mõjutab. Samas, kui kallutamise mõjul jõuab üks telgedest vertikaalile lähemale, siis on raskusjõu mõju selle telje puhul suurem.



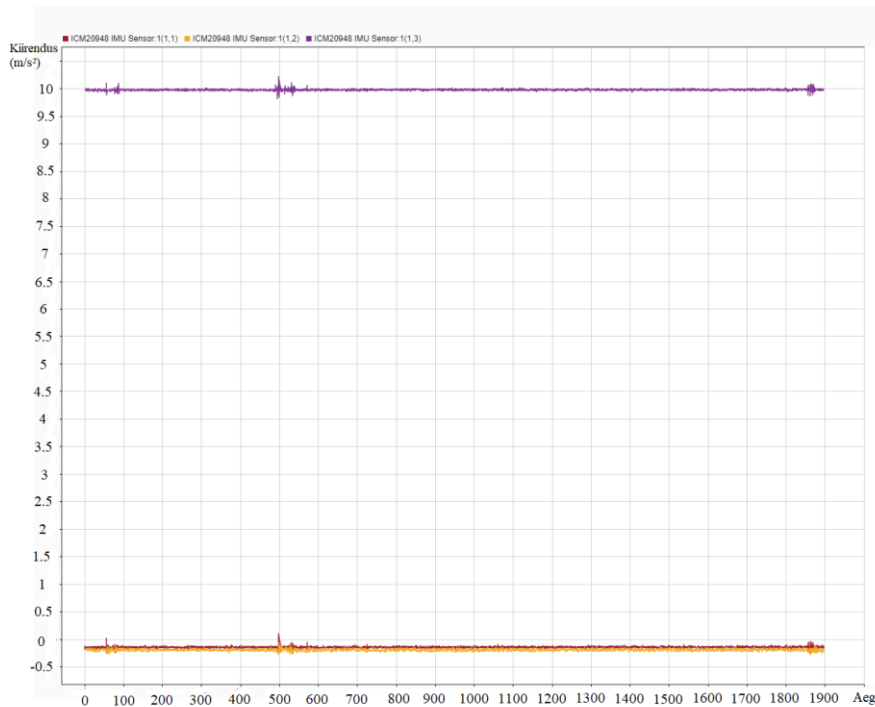
Joonis 8. ICM20948 kiirenduse salvestamine eri nurkade all

Veel toob näitena LIS3DH katsetuse, kus sai kiirenduanduri tulemus mõõdetud paigalseisus. Selle tulemusega on võimalik näha, kuidas anduri tulemused triivivad aja jooksul (vt Joonis 9). Sellel joonisel on samuti ka näha, kuidas sellel anduril esineb nulli suhtes nihe gravitatsioonijõu tõttu.



Joonis 9. LIS3DH kiirendus mõõtmised paigalseisus

Samasugune paigalseisus tehtud mõõtmine sai ka tehtud ICM20948 anduriga. Andur pandi lamama tasase pinna peale ning mõõtetulemused sai pandud graafikule (vt Joonis 10). Selle pealt on näha vähem triivimist, kui LIS3DH tulemustel. Kui aga oleks mingisugune nihe sees, siis oleks võimalik seda lahendada *Sensor Fusion* ploki parameetrite muutes. Nimelt, *Sensor Fusion* plokiga saab mõõtmis -ja keskkonnamüra muuta, et ploki arvutused oleksid võimalikult täpsed. Neid parameetreid on näha Joonisel 11. ICM20948 tulemused on piisavalt täpsed, et nende parameetrite muutmise vajalik ei ole. Lisaks, *Sensor Fusion* ploki ei saa kasutada anduriga LIS3DH, sest see ei ole IMU, mistõttu selle triivimist ei ole võimalik selliselt lahendada.



Joonis 10. ICM20948 kiirendus mõõtmised paigalseisus

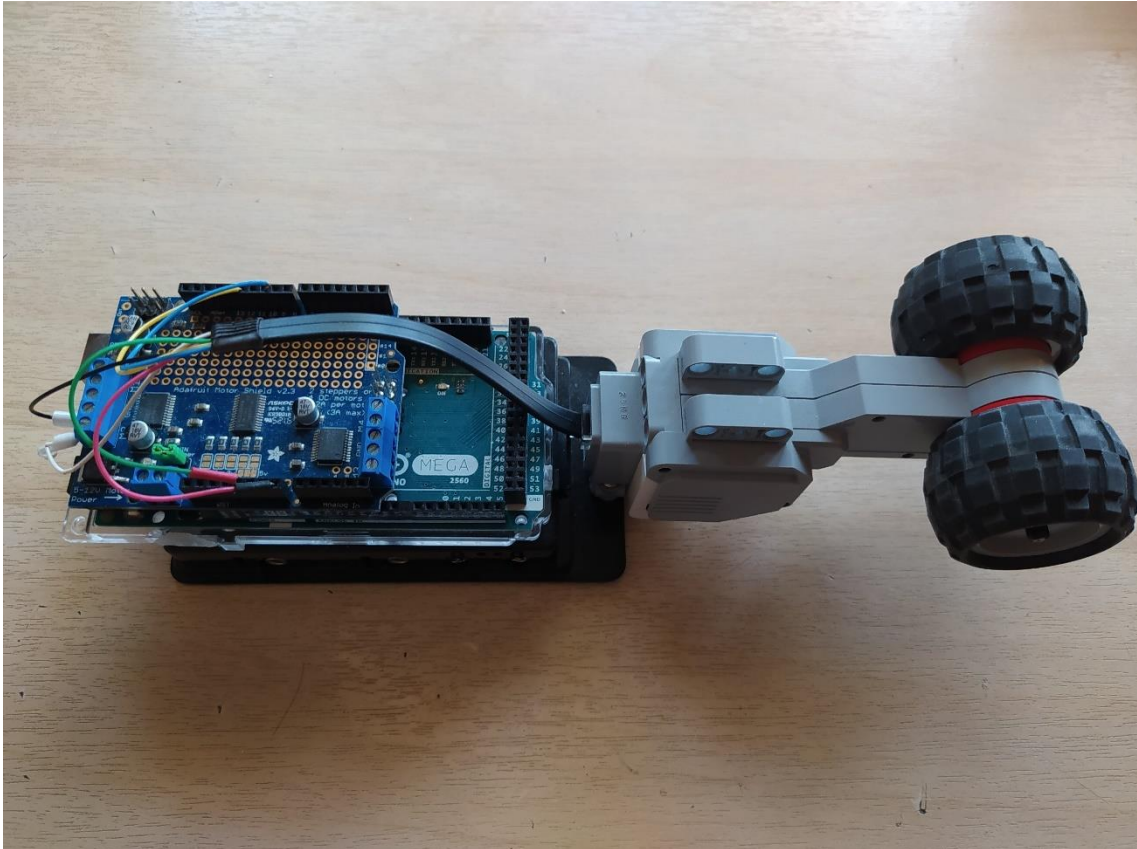
Main	Measurement Noise	Environmental Noise
Accelerometer noise ((m/s ²) ²):	0.0001924722	
Gyroscope noise ((rad/s) ²):	9.1385e-5	
Magnetometer noise ((μT) ²):	0.1	
Gyroscope drift noise ((rad/s) ²):	3.0462e-13	

Joonis 11. *Sensor Fusion* ploki müra parameetrid

*Data Inspector*it kasutades tekkis probleem graafikute esitamisega. Nimelt, ei olnud võimalik piisavalt graafikute visuaalset poolt muuta. Näiteks, ei ole võimalik X ja Y-telge ümber nimetada. Saadaval ei ole ka võimalust arvude suurust muuta, mistõttu graafikuid on pildi pealt raske lugeda. Mõlemad selle töö graafiku pildid said eraldi töödeldud, et arvud oleksid suuremad ja loetavamad ning et X ja Y-teljed oleksid nimedega.

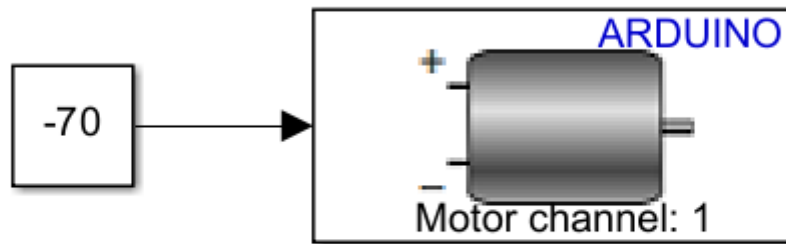
3.1.1 Alalisvoolumootori plokk

Enne ploki kasutamist sai ühendatud Arduino mootoriga. Selle tulemust on näha Joonisel 12.



Joonis 12. LEGO EV3 mootor ühendatud Arduino mikrokontrolleriga

Eraldi mudeli peal sai katsetatud alalisvoolumootori ploki. See plokk on nimetatud *Adafruit Motor Shieldi* PWM kiibi järgi, milleks on PCA9685. Plokk võtab sisendi -100 kuni 100-ni, mis määrab ära alalisvoolumootori kiiruse ja suuna. Positiivne arv on üks suund ning negatiivne on teine. Kasutusel on konstantse arvu plokk, et see arvuline sisend mootori ploki anda (vt Joonis 13).



Joonis 13. Alalisvoolumootori plokk

Kui selline mudel käivitada *Connected IO* režiimis, siis tekib veateade ning mootor ei käivitu (vt Joonis 14).

```
Caused by:
  • Current Data size is 72 bytes. Data size must be less than or equal to 58 bytes.
Component: Simulink | Category: Block error
```

Joonis 14. *Connected IO* režiimis alalisvoolumootori kasutamisel tekkinud veateade

Koodi genereerides ning välises režiimis kävitades ei lähe samuti mootor tööle, aga seekord ilma veateateta. Sama juhtub ka siis, kui luua eraldiseisev programm ja see Arduino peale laadida. Ainus töötav lahendus mootori käivitamiseks, oli kasutada MATLAB käsuri, mis kinnitab, et probleem on ainult Simulinki sisene.

Viga iseseisvalt uurida ei ole võimalik seetõttu, et ploki koodifaili sisu on kasutaja eest peidetud. Seda faili kutsutakse *P-code* failiks ning see on MATLABi loodud faili formaat [22]. Need failid on loodud selleks, et kasutajal ei oleks võimalik lähtekoodi lugeda.

Proovitud sai ka teist Arduino Mega ja *Adafruit Motor Shieldi*, aga sama problem toimus ka nendega.

Selle probleemi lahendamiseks oli sunnitud võtma kontakti MATLAB Support'iga, kellega algas mitu nädalat kestev vestlus. Selle käigus sai ka tehtud kõne MATLAB inseneriga, kes kogus selle käigus probleemi kohta rohkem infot.

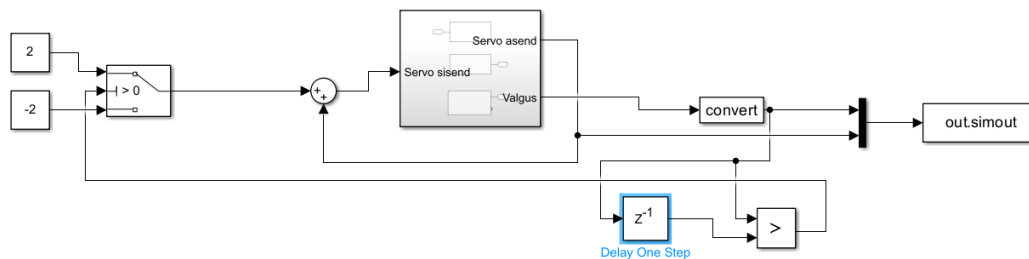
Pärast umbes kuu aega suhtlemist saatis MATLAB klienditugi probleemile lahenduse, milleks oli alalisvoolumootori plokki asendamine nende saadetud plokkiga. Kahjuks, isegi selle plokkiga ei hakanud mootor tööle. Kuna see vastus tuli töö lõpupoole, siis selle probleemi lahendamisega selle töö käigus ei jätkatud.

3.2 Servomootori lahendus

Eesmärgiks oli prototüüpida lahendus, tänu millele servomootor pööraks ennast alati suurima valgusallika poole. Kuna töö põhimõte on Simulinki katsetamine, siis selle lahenduse algoritm on lihtne.

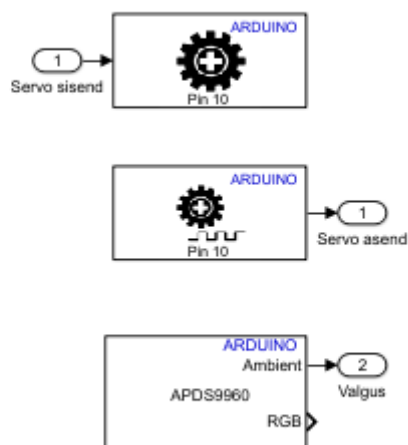
Servomootor liigub ühest asendist teise, muutes seda kahe võrra. Servomootor liigub 0 kuni 180 kraadini, nii et igal programmi sammul liigub kahe võrra edasi. Valgusandur jätab eelmise valgussuuruse näidu meelde ning võrdleb seda praeguse näiduga. Kui eelmine näit oli suurem, siis liigub kahe võrra tagasi ning võtab uue näidu. Kui uus näit on suurem, siis liigub kahe võrra edasi ning võtab uue näidu. Nii see jätkab lõpmatuseni.

Simulinkis sai see tehtud kasutades *Delay One Step* plokki, mis võimaldab viite tekitada mingi näidu jaoks. See on kokku pandud võrdlemis plokkiga ning ka tingimusliku plokkiga. Võrdlemis plokk võrdleb kahte näitu ning väljastab kas 0 või 1. Tingimuslik plokk otsustab selle 0 või 1 põhjal, kas liita või lahutada 2 kraadi servomootori asendist (vt Joonis 15).



Joonis 15. Servomootori lahendus

Servomootor ja valgusandur on eraldatud üheks alamsüsteemiks, mille üheks sisendiks on servomootori asendi määramine ning mille kaheks väljundiks on servo praegune asend ning valgusanduri mõõt. Alamsüsteemi sisemust on näha Joonisel 16.



Joonis 16. Servo ja valgusanduri alamsüsteem

4 Kokkuvõte

Töö käigus sai prototüübitud ainult üks lahendus kahest. Alalisvoolumootori probleem ei saadud lahendatud ning ei ole ka vastust, miks see probleem toimub. Teiste aspektidega, nagu plokkide kasutamine, koodi genereerimine ning andmete graafikutel kuvamisega probleeme ei tekkinud. Graafikute visuaalse poolega tekkis aga probleem sellega, et pildina graafiku väljastamisel ei olnud pilt loetav. Muidu oli õppeprotsess lihtne ning arusaam Simulinki kasutamisest on olemas. Tõestus sellest on servomootori lahendus, millega probleeme ei tekkinud.

Suurimaks komistuskiviks oli siiski alalisvoolumootor, mille probleemid andsid võimaluse rääkida MATLAB klienditoega ning näha, milline võib ühe tudengi kogemus nendega olla. Kogemus oli positiivne, vastused tulid vähemalt üks kord nädalas ning klienditoe suhtumine probleemile oli tudengile vastutulelik.

Töö eesmärki võib lugeda osaliselt saavutatuks. Üks prototüübitud lahendus sai valmis minimaalse koodi genereerimisega. Teine oli jäänud pooleli, aga andis võimaluse vähemalt ühe probleemi Simulinki kasutamisega kirja panna. Lisaks andis ühe lahenduse valmis saamine Simulinkiga rohkem kogemust, mis saab olla alus tuleviku projektide jaoks. Edasiarenduseks on veel võimalik alalisvoolumootori ploki probleemile lahenduse leida, jätkates ka klienditoega suhtlusi. Proovimata jäi ka Arduino Mega ja *Adafruit Motor Shieldi* asendamine. Näiteks Arduino Uno kasutamine Mega asemel või teise mootorit juhtiva laiendusplaadi kasutamine *Adafruit Motor Shieldi* asemel.

Kasutatud kirjandus

- [1] Vikipeedia, „Pulsilaiusmodulatsioon,“ 2024. [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Pulsilaiusmodulatsioon>. [Kasutatud 13 05 2024].
- [2] J. Schulze, „What Is MATLAB? Overview and FAQ,“ Coursera, [Võrgumaterjal]. Available: <https://www.coursera.org/articles/what-is-matlab>. [Kasutatud 15 03 2024].
- [3] MathWorks, „Simulink,“ [Võrgumaterjal]. Available: <https://se.mathworks.com/products/simulink.html>. [Kasutatud 15 03 2024].
- [4] MathWorks, „MATLAB Support Package for Arduino Hardware,“ [Võrgumaterjal]. Available: <https://se.mathworks.com/matlabcentral/fileexchange/47522-matlab-support-package-for-arduino-hardware>. [Kasutatud 15 03 2024].
- [5] MathWorks, „Control Motors Using Adafruit Motor Shield V2,“ 2024. [Võrgumaterjal]. Available: <https://se.mathworks.com/help/matlab/supportpkg/control-motors-using-adafruit-motor-shield-v2.html>. [Kasutatud 15 03 2024].
- [6] MathWorks, „Host-Target Communication with External Mode Simulation,“ 2024. [Võrgumaterjal]. Available: <https://se.mathworks.com/help/ecoder/armcortexa/ug/set-up-and-use-hosttarget-communication-channel.html>. [Kasutatud 10 04 2024].
- [7] MathWorks, „Standalone Target Computer Setup,“ 2024. [Võrgumaterjal]. Available: <https://se.mathworks.com/help/slrealtime/standalone-deployment-setup.html>. [Kasutatud 10 04 2024].

- [8] MathWorks, „Communicate with Hardware Using Connected IO,“ 2024. [Võrgumaterjal]. Available: <https://se.mathworks.com/help/rtw/nucleo/ug/connected-io.html>. [Kasutatud 10 04 2024].
- [9] MathWorks, „MathWorks Product Release Schedule,“ 2024. [Võrgumaterjal]. Available: https://se.mathworks.com/products/new_products/release_model.html. [Kasutatud 09 05 2024].
- [10] J. Joseph, „Types of Arduino Boards – Quick Comparison on Specification and Features,“ 19 03 2022. [Võrgumaterjal]. Available: <https://circuitdigest.com/article/different-types-of-arduino-boards>. [Kasutatud 16 04 2024].
- [11] l. ada, „Adafruit Motor Shield V2 Overview,“ Adafruit, 09 07 2024. [Võrgumaterjal]. Available: <https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/overview>. [Kasutatud 21 04 2024].
- [12] P. Hurbain, „LEGO 9V Technic Motors compared characteristics,“ [Võrgumaterjal]. Available: <https://www.philohome.com/motors/motorcomp.htm>. [Kasutatud 25 04 2024].
- [13] M. Giles, „Motors: EV3 vs NXT,“ [Võrgumaterjal]. Available: <http://www.mindsensors.com/blog/news/motors-ev3-vs-nxt>. [Kasutatud 26 04 2024].
- [14] LEGO, „EV3 Large Servo Motor,“ [Võrgumaterjal]. Available: <https://www.lego.com/en-us/product/ev3-large-servo-motor-45502>. [Kasutatud 26 04 2024].
- [15] Vikipeedia, „Inertsiaalandur,“ [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Inertsiaalandur>. [Kasutatud 28 04 2024].
- [16] MathWorks, „Arduino Sensor blocks,“ [Võrgumaterjal]. Available: <https://se.mathworks.com/help/simulink/sensors.html>. [Kasutatud 27 04 2024].

- [17] MinSeg, „MinSegMega V4,“ [Võrgumaterjal]. Available: <https://minseg.com/collections/minseg-kits/products/minsegmeg-v4>. [Kasutatud 30 04 2024].
- [18] MinSeg, „MinSegShield M2V5 Dual Axis,“ [Võrgumaterjal]. Available: <https://minseg.com/collections/minseg-kits/products/minsegshield-m2v5-dual-axis-balance-minseg-kit-new>. [Kasutatud 30 04 2024].
- [19] TDK-Invensense, „Migrating from MPU-9250 to ICM-20948,“ 09 2018. [Võrgumaterjal]. Available: https://invensense.tdk.com/wp-content/uploads/2018/10/AN-000146-v2.0-TDK_Migration_MPU_9250toICM-20948.pdf. [Kasutatud 30 04 2024].
- [20] Adafruit, „Adafruit LSM6DSOX + LIS3MDL - Precision 9 DoF IMU,“ [Võrgumaterjal]. Available: <https://www.adafruit.com/product/4517>. [Kasutatud 30 04 2024].
- [21] Science Buddies, „Accelerometer Technical Note,“ [Võrgumaterjal]. Available: <https://www.sciencebuddies.org/science-fair-projects/references/accelerometer>. [Kasutatud 20 05 2024].
- [22] MathWorks, „Create a Content-Obscured File with P-Code,“ 2024. [Võrgumaterjal]. Available: https://se.mathworks.com/help/matlab/matlab_prog/building-a-content-obscured-format-with-p-code.html. [Kasutatud 20 05 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Kris Ilmar Kristenprun

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Simulink keskkonnas koodi genereerimine Arduino platvormile“, mille juhendaja on Andres Rähni
 - 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.

14.05.2024

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Arduino kontrolleri vordlus

Algajatele moeldud Arduino kontrolleri vordlus [10]:

Board		Arduino Uno R3	Arduino Nano	Arduino Pro Mini	Leonardo	Micro	Nano Every	Mega2560 Rev3
Microcontroller		ATmega328p	ATmega328p	ATmega328p	ATmega32u4	ATmega32u4	ATmega4809	ATmega2560
FPGA		No	No	No	No	No	No	No
USB connector		USB-B	Mini-B USB	Mini-B USB	Micro USB (USB-B)	Micro USB	Micro USB	USB-B
I/O	Digital only I/O pin	14	14	14	20	20	14	54
	Analog input pins	6	8	8	12	12	8	16
	Analog output pins	0	0	0	0	0	0	0
	PWM Pins	6	6	6	7	7	5	15
Communication	UART	yes	yes	yes	yes	yes	yes	yes, 4
	I2c	yes	yes	yes	yes	yes	yes	yes
	SPI	yes	yes	yes	yes	yes	yes	yes
	CAN	No	No	No	No	No	No	No
	Bluetooth	No	No	No	No	No	No	No
WiFi	No	No	No	No	No	No	No	
Power	I/O voltage	5v	5v	5v	5v	5v	5v	5V
	Input nominal voltage	7-12V	7-12V	7-12V	7-12V	7-12V	7-21V	7-12V
	DC Current per I/O	20mA	20mA	20mA	10mA	10mA	20mA	20mA
	Power supply connection	Barrel Jack	GPIO header	GPIO header	Barrel Jack	GPIO header	GPIO header	Barrel Jack
	Battery Powered	No	No	No	No	No	No	No
Clock speed	Main processor	ATmega328P 16MHz	ATmega328P 16MHz	ATmega328P 16MHz	ATmega32u4 16MHz	ATmega32u4 16MHz	20MHz	16MHz
RTC		No	No	No	No	No	No	No
USB to Serial		ATmega16U2 16MHz	FT232RL	No onboard USB-TTL Converter	Native	Native	ATSAMD11D14A	ATmega16U2 16MHz
Memory	Flash	32KB	32KB	32KB	32KB	32KB	48KB	256KB
	SRAM	2KB	2KB	2KB	2.5KB	2.5KB	6KB	8KB
	EEPROM	1KB	1KB	1KB	1KB	1KB	256B	4KB
Dimensions	Weight	25 g	5 g	5 g	20 g	13 g	5 g	37 g
	Width	53.4 mm	18mm	18mm	53.3 mm	18 mm	18 mm	53.3 mm
	Length	68.6 mm	45mm	45mm	68.6 mm	48 mm	45 mm	101.5 mm

Edasijõudnutele mõeldud Arduino kontrolleri võrdlus [10]:

Board		Nano BLE	MKR Zero	UNO WIFI Rev2	Due	MKR Vidor 4000	Arduino Zero
Microcontroller		nRF52840	SAMD21 Cortex-M0+	ATmega4809	AT91SAM3X8E	SAMD21 Cortex-M0+	ATSAMD21G18
FPGA		No	No	No	No	Intel® Cyclone® 10CL016	No
USB connector		Micro USB	Micro USB	Micro USB	Micro USB	Micro USB	Micro USB
I/O	Digital only I/O pin	14	22	14	54	8	20
	Analog input pins	8	7	6	12	7	6
	Analog output pins	0	1	0	2	1	1
	PWM Pins	all digital pins	13	5	12	13	10
Communication	UART	yes	yes	yes	yes, 4	yes	yes, 2
	I2c	yes	yes	yes	yes	yes	yes
	SPI	yes	yes	yes	yes	yes	yes
	CAN	No	No	No	Yes(Ext Transceiver)	No	No
	Bluetooth	Yes	No	Yes	No	yes	No
	WIFI	No	No	Yes	No	yes	No
Power	I/O voltage	3.3V	3.3V	5V	3.3V	3.3V	3.3v
	Input nominal voltage	upto 21V	5V	7-12V	7-12V	5V	7-12V
	DC Current per I/O	15mA	7mA	20mA	9mA/3mA	7mA	7mA
	Power supply connection	GPIO header	GPIO header	Barrel Jack	Barrel Jack	GPIO header	Barrel Jack
	Battery Powered	No	Yes	No	No	Yes	No
Clock speed	Main processor	64MHz	48MHz	20MHz	84MHz	48MHz	48MHz
RTC		No	Yes	No	No	Yes	No
USB to Serial		Native	Native	ATMega32U4	ATmega16U2 16MHz	Native	Native
Memory	Flash	1MB	256KB+8KB	48KB	512KB	256KB	256KB
	SRAM	256KB	32KB	6KB	96KB	32KB	32KB
	EEPROM	none	None	256B	None	None	None
Dimensions	Weight	5 g	32 g	25 g	36 g	43.5 g	12 g
	Width	18 mm	25 mm	53.4 mm	53.3 mm	25 mm	53 mm
	Length	45 mm	61.5 mm	68.6 mm	101.5 mm	83 mm	68 mm