



**LEGO-ROBOTIN LIIKKEENOHJAUksen TOTEUTTAMINEN JA
HAVAINNOLLISTAMINEN SÄÄTÖTEKNIIKAN OPETUKSESSA**

Lappeenrannan–Lahden teknillinen yliopisto LUT

Sähkötekniikan kandidaatintyö

2022

Miikka Rantanen

Tarkastaja: Apulaisprofessori Niko Nevaranta

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT
LUT Energiajärjestelmät
Sähkötekniikka

Miikka Rantanen

Lego-robotin liikkeenohjauksen toteuttaminen ja havainnollistaminen säätötekniikan opetuksessa

Sähkötekniikan kandidaatintyö

2022

31 s.

Tarkastaja: Apulaisprofessori Niko Nevaranta

Kandidaatintyön tavoitteena oli luoda havainnollistava ja samalla yksinkertainen esimerkki liikkeenohjauksen suunnitteluprosessista käyttämällä Lego EV3 Mindstorms-rakennussarjan avulla rakennettua lego-robotia. Tämän lisäksi työhön kuului kirjallisuuskatsaus, jonka tavoitteena oli kartoittaa opetus- ja tutkimustarkoitukseen rakennettuja lego-robotteja.

Liikkeenohjaukselle valittiin kaksi eri tavoitetta, jonka jälkeen legoista rakennettiin liikkeenohjausta havainnollistava robotti. Robotista luotiin simulaatiomalli ja sen siirtofunktio identifioitiin sovittamalla se Simulink-ohjelmistolla suoritettujen askelvastekokeiden pohjalta piirrettyyn askelvasteen kuvaajaan. Identifioidun 1. kertaluvun siirtofunktion perusteella järjestelmälle suunniteltiin P- ja PI-säätimet Ziegler-Nichols menetelmällä ja tarkasteltiin säätimien soveltuvuuksia molempiin liikkeenohjaustavoitteisiin. Säätöparametrien arvoja hienosäädettiin kokeellisesti ja järjestelmän stabiiliutta tarkasteltiin Bode-diagrammin avulla määrittämällä vaihe- ja vahvistusvarat. Säädetyin järjestelmän toimivuutta testattiin kokeellisesti lego-robotilla ja liikkeenohjaustavoitteet toteutuivat.

Työn tulokset osoittavat, että Lego EV3:lla toteutettua yksinkertaista liikkeenohjausesimerkkiä voitaisiin käyttää tukemaan säätötekniikan perusteiden opetusta.

ABSTRACT

Lappeenranta–Lahti University of Technology LUT
School of Energy Systems
Electrical Engineering

Miikka Rantanen

Implementation of motion control for a Lego robot and its usage as demonstration tool for control engineering education

Bachelor's thesis

2022

31 p.

Examiner: Assistant professor Niko Nevaranta

The objective of this bachelor's thesis was to demonstrate the design process of motion control for a simple Lego robot built by using Lego EV3 Mindstorms building-kit. In addition to this a literature review was made about Lego robots that have been built for both education and research purposes.

Two different targets were chosen for the motion control and then a Lego robot was built to demonstrate the implementation of those motion control targets. A simulation model for the robot was created and its transfer function was identified by fitting it to the system's step response. By using the identified first order transfer function a P and PI controllers were designed for the system. The controller parameters were chosen via Ziegler-Nichol's method and their suitability for the different motion control targets was analyzed. System's stability was examined by using Bode diagram and fine tuning to the control parameters was done. The functionality of the motion control was tested on a Lego robot and the motion targets were achieved.

The results of this thesis show that a simple motion control example can be implemented by using Lego EV3 and this example could be used to support the teaching of basic control engineering.

KÄYTETYT MERKINNÄT

φ	Kiertymä
τ	Aikavakio
C	Säädin (Controller)
G	Systeemi (Plant)
D	Derivaatta (Derivative)
I	Integraali (Integral)
K	Vahvistus
K_i	Integroivan osan vahvistus
K_p	Suhteellinen vahvistus
K_d	Derivoivan osan vahvistus
L	Viive
n	Kierrostaajuus
P	Suhteellinen (Proportional)
s	Laplace-muuttuja
t	Aika
T_d	Derivointiaika
T_i	Integrointiaika
T_s	Näytteenottoaika

LYHENTEET

IMC	Internal Model Control
PWM	Pulse-Width Modulation

SISÄLLYSLUETTELO

Tiivistelmä

Abstract

Käytetyt merkinnät ja lyhenteet

1. Johdanto	6
2. Katsaus lego-robotteihin opetusvälineenä.....	8
2.1. Lego EV3 Mindstorms	8
3. Liikkeenohjauksen toteutus Lego EV3-robotille	14
3.1. Pyörimisnopeuden säätö P- ja PI-säätimillä	15
3.2. Kulmansäätö	26
4. Yhteenveto ja johtopäätökset	30
Lähteet	31

1. Johdanto

Säätötekniikan opiskelun tukemiseksi voidaan sen opetuksessa käyttää erilaisia opetustarkoitukseen suunniteltuja laitteita ja robotteja. Matemaattisen ja teoriapainoiteisen säätötekniikan opettelu voi olla haastavaa opiskelijoille erityisesti, jos teoriaa ei havainnollisteta käytännön esimerkkien avulla. Tämän työn tavoite on edesauttaa säätötekniikan opetusta ja oppimista antamalla esimerkki yksinkertaisesta säätöteknisestä suunnitteluprosessista, jota havainnollistetaan Legon EV3 Mindstorms -robotilla. Lego EV3 Mindstorms-rakennussarjan (Kuva 1) avulla rakennetulle robotille määritetään tässä työssä kaksi erilaista liikkeenohjaustavoitetta ja näiden toteuttamiseen käytetään Simulink ja Matlab-ohjelmistoja. Liikkeenohjauksen tavoitteen saavuttamiseksi järjestelmään on suunniteltava sopiva säädinrakenne, jonka suunnitteluprosessiin otetaan kantaa tässä työssä. Työn pyrkimyksenä on havainnollistaa digitaalisen ja analogisen maailman välisiä yhteyksiä säätösuunnittelussa.



Kuva 1. Lego EV3 Mindstorms-rakennussarja.

Tutkielma alkaa kirjallisuuskatsauksella, jossa käydään läpi erilaisia jo olemassa olevia lego-robotteja ja kuinka robotit ovat toteutettu. Katsauksen tarkoituksena on antaa tiivistetty käsitys lukijalle lego-robottien käytöstä opetus- ja tutkimustarkoituksessa. Kirjallisuuskatsauksen jälkeen paneudutaan tutkielmaa varten rakennettuun lego-robottiin ja esitellään sen liikkeenohjauksen tavoitteet. Ohjainyksikön ohjelmointia ja käyttöönottoa havainnollistetaan kuvien avulla. Lego-robottia ohjelmoidaan Simulink- ja Matlab-ympäristössä, jotka pohjautuvat LabView-ohjelmointikieleen.

2. Katsaus lego-robotteihin opetusvälineenä

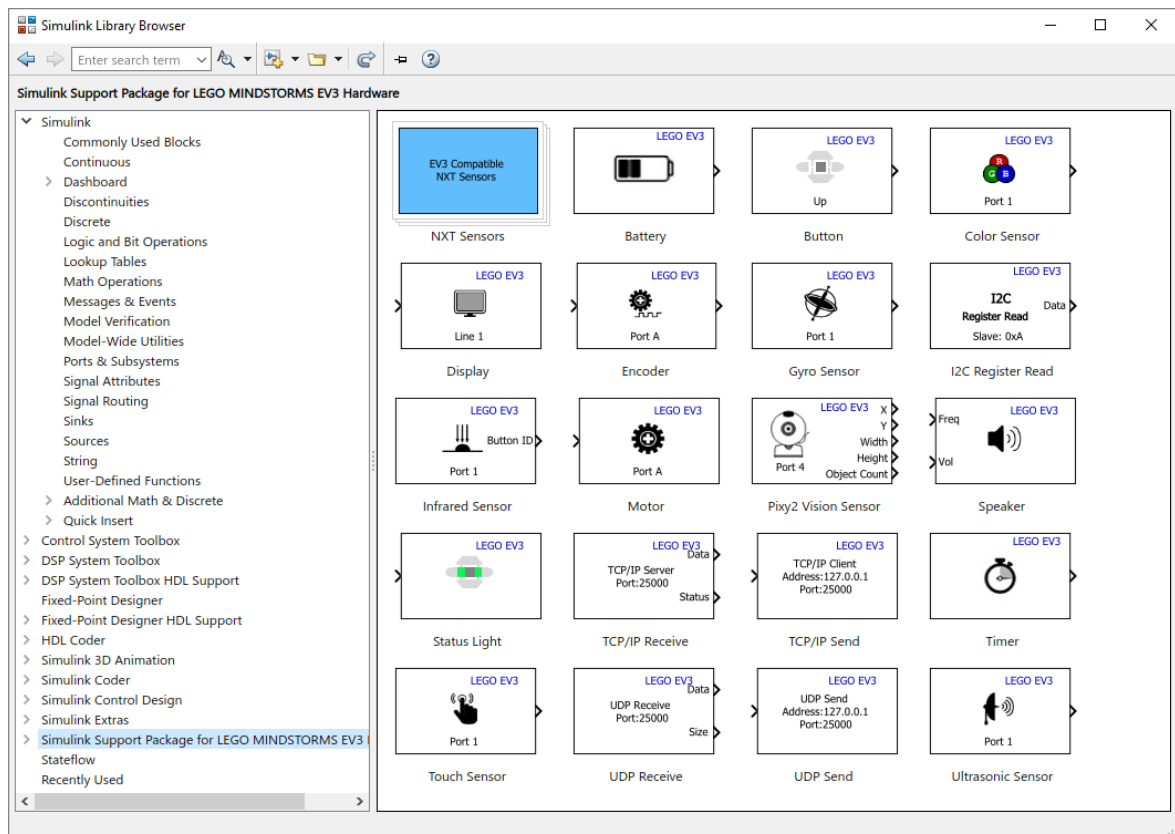
Ohjelmoitavia lego-robotteja käytetään opetusvälineenä niiden alhaisten kustannusten, yksinkertaisuuden ja joustavuuden takia [1]. Niiden avulla sekä peruskoululaiset että korkeakouluopiskelijatkin saavat käytännön kokemusta ohjelmoinnista ja robotiikasta. Säätötekniikan ja robotiikan opettelu voi osoittautua oppilaalle haastavaksi niiden opetuksessa käytettävien matemaattisten mallien ja teorioiden vuoksi. Säätötekniikan matemaattisesta luonteesta johtuen on tärkeää, että oppilas saa käsityksen millä tavalla teoreettiset mallit ovat yhteydessä reaali maailman fyysisiin järjestelmiin ja niiden säätöön. Kiinnostusta teknillisiä aineita kohtaan on pyritty herättämään kouluikäisissä, jotta useampi heistä päätyisi opiskelemaan aloja korkeakouluihin. Koululaisten kiinnostusta ohjelmointia kohtaan on pyritty edistämään myös Suomessa. Esimerkkinä tästä on vuonna 2016 käyttöön otettu valtakunnallinen opetussuunnitelma, joka määrää peruskoulun vuosiluokilla 3-6 opetettavaksi ohjelmointia graafisessa ohjelmointiympäristössä [2]. Ohjelmointia käytetään eri tekniikan aloilla ja sen kannalta on tärkeää, että oppilaan kyky ratkaista ongelmia myös kehittyy. Ongelmanratkaisussa käytetään opittujen mekaanisten taitojen lisäksi usein luovaa ajattelua. Tutkijat ovatkin suositelleet lego-robotteja käytettäväksi opetuksessa oppilaiden luovuuden kehittämiseksi [3].

2.1. Lego EV3 Mindstorms

Lego EV3 on modulaarinen alusta, jonka rakennussarja mahdollistaa sen käytön sekä tutkimus- että opetustarkoituksessa. Rakennussarja mahdollistaa suuren määrän erilaisia konfiguraatioita. Erilaisia projekteja on helposti löydettävissä lukuisia Internetin eri avointen yhteisöjen sivuilla. Lisäksi akateemisia julkaisuja erilaisista lego-roboteista ja niiden mallinnuksesta ja säätösunnittelusta löytyy myös paljon. Alustaa ja sen komponentteja ei voida kuitenkaan käyttää teollisissa ratkaisussa, koska sitä ei ole rakenteellisesti eikä toiminnallisesti optimoitu siihen tarkoitukseen [4].

Mindstorms on Legon oma graafinen ohjelmointikieli, joka toimii saman nimisessä ohjelmointiympäristössä. Tämän ympäristön ja Lego EV3-rakennussarjan avulla on mahdollista opettaa ohjelmointia ja robotiikkaa minkä ikäisille tahansa. Mindstorms-ohjelmointikieli ei kuitenkaan sovellu säätötekniikan opetukseen erityisen hyvin, sillä sen

sisältämien valmiiden rakennelohkojen ja säädinten parametreja ei ole mahdollista muokata [5]. Lego tarjoaa valmiita rakennelohkoja myös MATLAB-pohjaiseen Simulink-ohjelmointiympäristöön, joka on esitetty kuvassa 2. Ohjelman avulla voidaan mallintaa, simuloida ja analysoida erilaisia dynaamisia järjestelmiä. Legon oma ilmainen Simulink-tukipaketti sisältää rakennelohkojen lisäksi mahdollisuuden säätää parametrien arvoja ja mahdollistaa myös sensoreille tulevien signaalien monitoroinnin. EV3-keskukseksi voidaan ohjelmoida Mindstormsin lisäksi muissakin eri ohjelmointiympäristöissä kuten: LabView, IeJos ja ROBOTC. Erilaisten ympäristöjen lukumäärän ansiosta käyttäjillä on mahdollisuus valita sellainen ympäristö, joka pohjautuu käyttäjälle tuttuun ohjelmointikieleen. Tässä tutkielmassa käytetään LabView-pohjaisia MATLAB- ja Simulink-ohjelmistoja.



Kuva 2. Lego tukipaketin rakennelohkot Simulinkissä.

2.2. Erilaisia lego-robotteja ja niiden toimintaperiaatteet

Tähän kappaleeseen on valittu erilaisia lego-robotteja, joita on suunniteltu opetus- tai tutkimustarkoitukseen. Kaikki valitut esimerkit pohjautuvat Legon EV3- tai NXT-

rakennussarjoihin. Käydään seuraavaksi läpi lyhyesti muutamia kirjallisuudessa esitetyjä robotteja alkaen yksinkertaisimmasta esimerkistä.

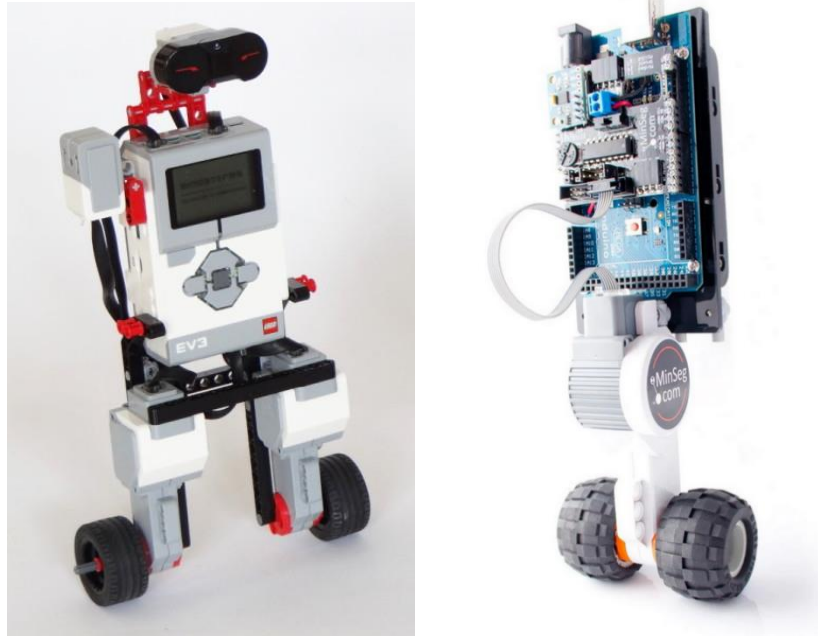
Opinnäytetyössä [6] on esitetty säätötekniikan opetuskäyttöön suunniteltu lego-robotti. Kyseisessä esimerkissä Legon EV3-ohjainyksikköön on yhdistetty ainoastaan servomoottori, joka pyörittää kahta rengasta kuten kuvassa 3 on esitetty. Vaikka järjestelmä vaikuttaakin yksinkertaiselta, on robotista tehty opinnäytetyö hyvä lähde havainnollistamaan PI-säätimen suunnittelua. Opinnäytetyössä on identifioitu lego-robotin servomoottorin dynamiikkaa kuvaava siirtofunktio käyttämällä tangenttimenetelmää, jonka jälkeen järjestelmälle on viritetty PI-säädin käyttäen Ziegler-Nichols, Chien-Hrones-Reswick ja Lambda-menetelmiä.



Kuva 3. Opetustarkoitukseen suunniteltu lego-robotti.

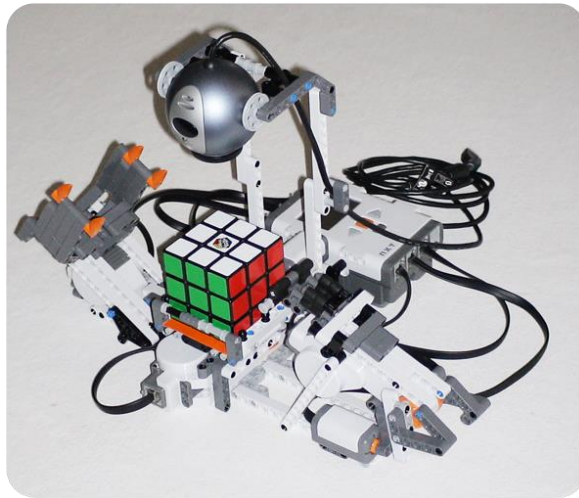
Astetta monimutkaisempi robotti on kahdella renkaalla itsensä tasapainottava ”Lego BALANC3R”, joka perustuu käänteiseen heiluriin. Tällaista systeemiä on havainnollistettu kuvassa 4. Käänteinen heiluri on monen mekaanisen järjestelmän perusosa [7] ja sen takia hyvin tyypillinen esimerkki säätötekniikan opetuksessa. Järjestelmän toimintaperiaate perustuu robotin rungon kallistuskulman mittaamiseen ja sen vakauttamiseen ohjaamalla servomootoreihin liitetyjä renkaita. Kallistuskulman arvoa mitataan robotin sivussa olevalla gyrosensorilla, joka lähettää dataa EV3-ohjainyksikölle. Kulmatiedon perusteella ohjainyksikkö pyörittää servomootoreihin yhdistettyjä renkaita, jotta kallistuskulma saadaan takaisin tasapainotilan arvoon. Säätötekniikan ongelman ratkaisun kannalta mielenkiintoista järjestelmässä on, että se on epästabiili ilman vakauttavaa säädintä. Samankaltaisia itsetasapainottavia robotteja on tutkittu myös opinnäytetöissä [8] ja [9].

Jälkimmäisessä työssä käytetyn MinSeg-robotin toimintaperiaate on pohjimmiltaan samanlainen, vaikkakin astetta kehittyneempi.



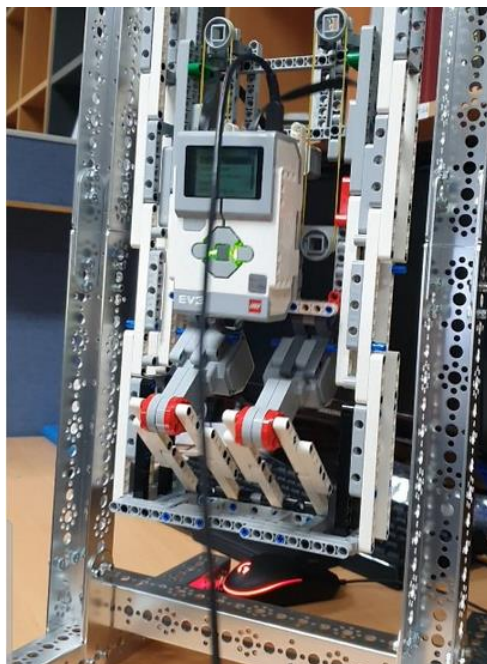
Kuva 4. Lego Balanc3r [10] ja MinSeg-robotti Arduinolla [11].

Kolmas esimerkki on lego-robotti, joka osaa ratkaista rubikin kuution [12]. Sen toimintaperiaate perustuu kuution jokaisen sivun tilan lukemiseen, jonka jälkeen robotti lähtee pyörittämään kuution sivuja NXT-ohjainyksikköön ohjelmoidun algoritmin mukaan. Ratkaiseminen voi kestää useita minuutteja riippuen kuution lähtökohdan tilasta. Robotti lukee kuution sivujen tilat kameran avulla ja hyödyntää kuution ratkaisemisessa visuaalista takaisinkytkentää kameran avulla, joka näkyy kuvassa 5. Kameran lisäksi ratkaisussa on käytetty kolmea servomootoria, jotka liikuttavat kuutiota ja sen sivuja. Kosketusanturit toimivat rajakytkiminä eli ne ohjaavat liikkuvien osien käynnistymistä ja sammuttamista. Valoanturi lukee kuution paikkatietoa, jolla varmistetaan että kuutio on oikealla paikalla. Tämänkaltaisen robottiesimerkki on herättänyt kiinnostusta akateemisissa ja harrastelijapiireissä. Esimerkiksi siitä on myöhemmin rakennettu kehittyneempi versio nimeltä ”CubeStormer II” [12].



Kuva 5. Rubikin kuution ratkaiseva Lego-robotti [12].

Vieläkin kehittyneempiä lego-robotteja on suunniteltu ja rakennettu. Tästä esimerkkinä Yonsei yliopiston koneoppimiseen erikoistuneen osaston suunnittelema lego-robotti, joka opettaa itsensä keinumaan tangossa [13]. Robotti asetetaan roikkumaan kuvan 6 mukaisesti tankoon, jonka jälkeen se aloittaa heilumisliikkeen. Koneoppimista ja tekoälyä hyödyntävä robotti lukee dataa sen heilurimaisesta liikkeestä ja ohjelmoi itsensä tietojen pohjalta alati kiihtyvään keinumisliikkeeseen. Lego-robotin EV3-ohjainyksikössä suoritettava algoritmi perustuu vahvistusoppimiseen. Tätä koneoppimisen alalajia on mahdollista soveltaa esimerkiksi itse-ajaviin autoihin [14].

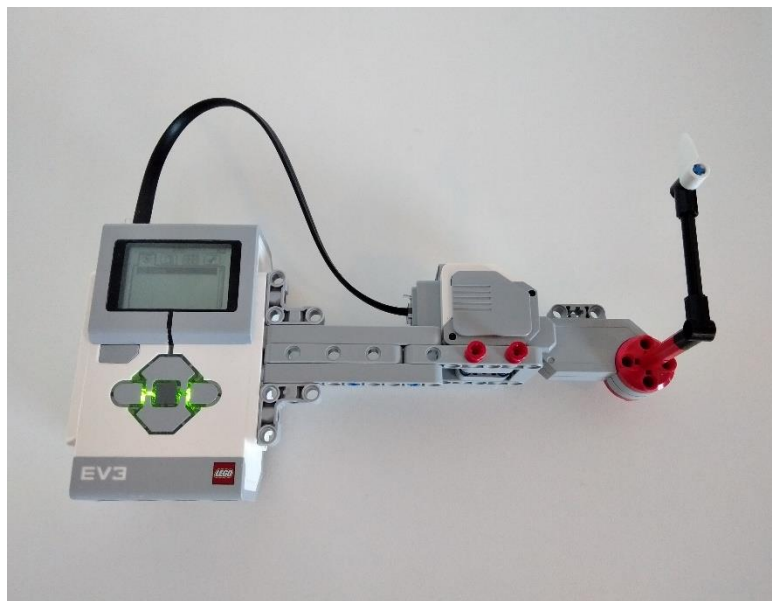


Kuva 6. Tangossa heiluva Lego-robotti [13].

Näiden esimerkkien pohjalta voidaan todeta, että Lego EV3 mahdollistaa ympäristön helppojen sekä monimutkaisten järjestelmien rakentamisen ja ohjaamisen. Se kykenee tukemaan perinteistä säätötekniikan opetusta, mutta se soveltuu myös ympäristöksi kehittyneiden algoritmien ja esimerkiksi tekoälyn soveltamiseen. Sovelluskohteita ja -mahdollisuuksia on monia, mutta robotin kehittyneisyyden sen ohjauksen näkökulmasta määrittää sen suunnittelijan tietotaito.

3. Liikkeenohjauksen toteutus Lego EV3-robotille

Tässä osiossa esitetään havainnollistava esimerkki liikkeenohjauksen toteuttamisesta Lego EV3 -robotilla. Havainnollistavuuden edesauttamiseksi esimerkki pidetään pelkistettynä ja yksinkertaisena. Liikkeenohjauksen tavoitteiksi on valittu ensiksi servomoottorin pyörimisnopeuden säätö halutulla kierrostaajuudella. Seuraava liikkeenohjauksen tavoite on servomoottorin kulman säätäminen halutulla kulmalla, joka syötetään järjestelmään asteina. Tarkoituksena on demonstroida nopeus- ja kulmasäädön toimintaa sekä eri säätörakenteiden vaikutuksia. Molempia tapauksia demonstroidaan kuvan 7 mukaisella lego-robottikonfiguraatiolla, joka on rakennettu Lego EV3 Mindstorms-rakennussarjaa hyödyntäen. Lego-robotin EV3-keskussyksikön ulostulo-porttiin on yhdistetty rakennussarjan suurempi servomoottori RJ12-tietoliikennekaapelilla. Servomoottoriin on myös kiinnitetty legoista rakennettu kulma-osoitin ja servomoottori on kiinnitetty lego-palikoilla EV3-yksikköön vakauden parantamiseksi.



Kuva 7. Lego-robotti, jolla liikkeenohjausta tehdään.

Lego-robotin käyttöä sekä ohjelmointia varten ohjausyksikön ja tietokoneen välille muodostettiin yhteys. Yhteyden voi luoda USB-kaapelin, Bluetoothin tai langattoman verkon (WiFi) avulla. Langattomien yhteyksien muodostaminen vaatii, että ohjainyksikön USB-porttiin liitetään rakennussarjan mukana tuleva vastaanotin Netgear dongle. Kun ohjainyksikkö on yhdistetty langattomaan verkkoon, täytyy siihen vielä ottaa yhteys Simulink-ohjelmiston kautta ja tämä vaatii EV3-tukipaketin asentamisen Simulink-

ohjelmistoon. Tukipaketin avulla voidaan lähettää ja vastaanottaa dataa EV3-keskusyksiköltä ja käyttää Simulink-ympäristössä Legon omia valmiita rakennelohkoja. Tässä työssä käytetään EV3-ohjainyksikön laiteohjelmiston (Firmware) versiota 1.09D. Jotkin saatavilla olevat laiteohjelmiston versiot eivät tue Simulink-ohjelmiston ja EV3-yksikön välistä yhteyden muodostamista.

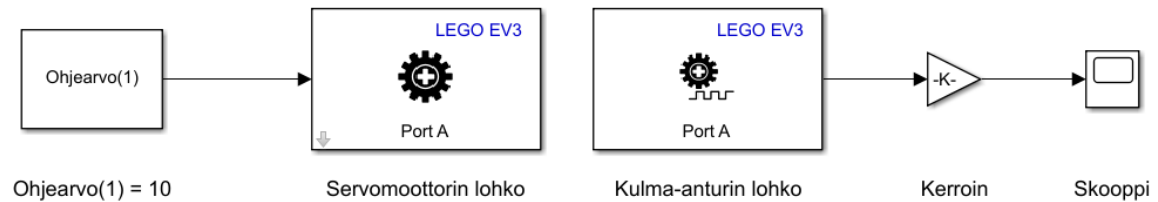
3.1. Pyörimisnopeuden säätö P- ja PI-säätimillä

Ennen kuin säätösuunnittelu aloitetaan, tutustutaan ensin lego-robotin servomoottoriin. Servomoottorin toiminnallisuutta tutkitaan aluksi kokeellisesti, jotta saadaan käsitys sen dynamiikista. Tämä tehdään rakentamalla yksinkertainen simulaatiomalli, joka käännetään koodiksi EV3:lle jonka avulla järjestelmälle voidaan syöttää erilaisia askelohjesignaaleja. Simulink-ympäristössä rakennettu simulaatiomalli kääntyy automaattisesti C-kieliseksi kun se lähetetään langattoman verkon kautta EV3-ohjainyksikköön, joka tämän jälkeen aloittaa ohjelman suorittamisen. Simulink-ympäristössä rakennettu malli hyödyntää Lego-tukipaketin valmiita rakennelohkoja. Ilman suljetun piirin säätörakennetta servomoottorin nopeutta saadaan säädettyä PWM-ohjauksen avulla antamalla ohjearvo, jonka minimiarvo on -100 ja maksimiarvo 100. Ohjearvon etumerkki määrittää kumpaan suuntaan moottori pyörii, positiivisilla arvoilla myötäpäivään, negatiivisilla vastapäivään ja nollassa moottori pysähtyy. Syötetään moottorille ohjearvoja väliltä 0 ja 100, jonka aikana tarkastellaan servomoottorin kulma-anturin antamia digitaalisia arvoja. Servomoottorin kulma-anturi antaa kulman arvoksi kiertymän φ asteina. Simulaatiomallin kulma-anturin lohkoa voidaan valita asetus, joka nolaa kiertymän arvon jokaisen näytteenottoajan T_s välein. Opinnäytetyössä [6] huomattiin, että liian nopea näytteistysaika aiheutti datan siirron kanssa ongelmia Simulinkin ja EV3-ohjainyksikön välillä. Tämän takia tässä työssä valitaan näytteenottoajaksi $T_s = 0.03$ s, joka todettiin hyväksi lähteessä [6].

Ennen säätimen suunnittelua päätetään missä yksikössä robotin pyörimisnopeutta halutaan ohjata. Nyt pyörimisnopeuden arvon halutaan olevan yksikössä kierroksia sekunnissa, se tarkoittaa, että kulma-anturista luettava kiertymän arvo täytyy skaalata sopivalla kertoimella. Tämän kertoimen arvo saadaan selvitettyä matemaattisesti kierrostaajuuden ja kiertymän välisestä yhteydestä:

$$n = \frac{\varphi}{T_s} * \frac{1}{360^\circ} \quad (1)$$

Lisäämällä (1) kuvan 8 simulaatiomallin kerroin-lohkon kohdalle, voidaan sen jälkeen Simulink'in monitorilohkon (Scope) avulla lukea servomotoorin pyörimisnopeus yksikössä kierroksia sekunnissa.



Kuva 8. Avoin järjestelmä lähtösignaalin skaalauksella.

Kun servomotoorin pyörimisnopeutta ohjataan kierrostaajuuden mukaan PWM-ohjauksen $[-100\dots100]$ sijasta, on järjestelmään lisättävä toinen skaalaus-kerroin. Kertoimen arvo lasketaan kuvan 8 avoimen järjestelmän lähtö- ja tulosignaalien suhteesta. Kerroin sijoitetaan järjestelmään heti ohjaussignaalin jälkeen kuvan 9 mukaisesti. Tämän jälkeen aloitetaan sopivan säätörakenteen suunnittelu järjestelmälle.

Säädön suunnittelu aloitetaan usein asettamalla järjestelmän dynaamisille ominaisuuksille tiettyjä kriteerejä, joiden halutaan täyttyvän. Näitä kriteerejä ovat esimerkiksi: nousuaika, asettumisaika, huippuaika ja huippuarvo. Säädinrakenteiden ja säätimen parametrien arvojen vaikutusta edellä mainittuihin kriteereihin tarkastellaan työn edetessä myöhemmissä kappaleissa. Jätetään kriteerien määrittäminen myöhemmäksi ja keskitytään säädön suunnittelun lähtökohtaan muodostamalla ensin siirtofunktio, joka kuvaa järjestelmän dynamiikkaa. Siirtofunktio identifioidaan sovittamalla käyrä avoimen järjestelmän askelvasteen kuvaajaan. Tällä menetelmällä muodostettu siirtofunktio ei kyseisen applikaation tapauksessa näytteenottotaajuuden rajoitusten vuoksi kuvaa järjestelmän dynamiikkaa täydellisesti, mutta menetelmä soveltuu käytettäväksi yksinkertaisten siirtofunktioiden identifioimiseen. Kuvan 7 systeemin yksinkertaisuudesta johtuen

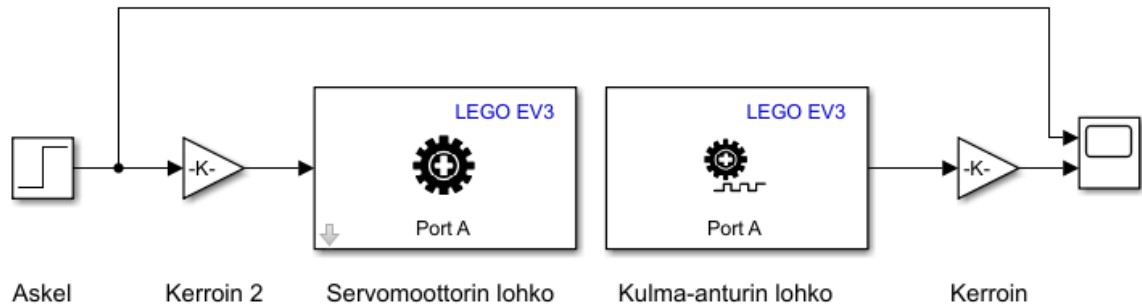
oletetaan, että järjestelmän dynamiikkaa voidaan mallintaa 1. kertaluvun viiveellisenä systeeminä seuraavan siirtofunktion mukaisesti:

$$G(s) = \frac{Ke^{-sL}}{\tau s + 1} \quad , \quad (2)$$

jossa K on järjestelmän vahvistus, τ on aikavakio ja L on viive.

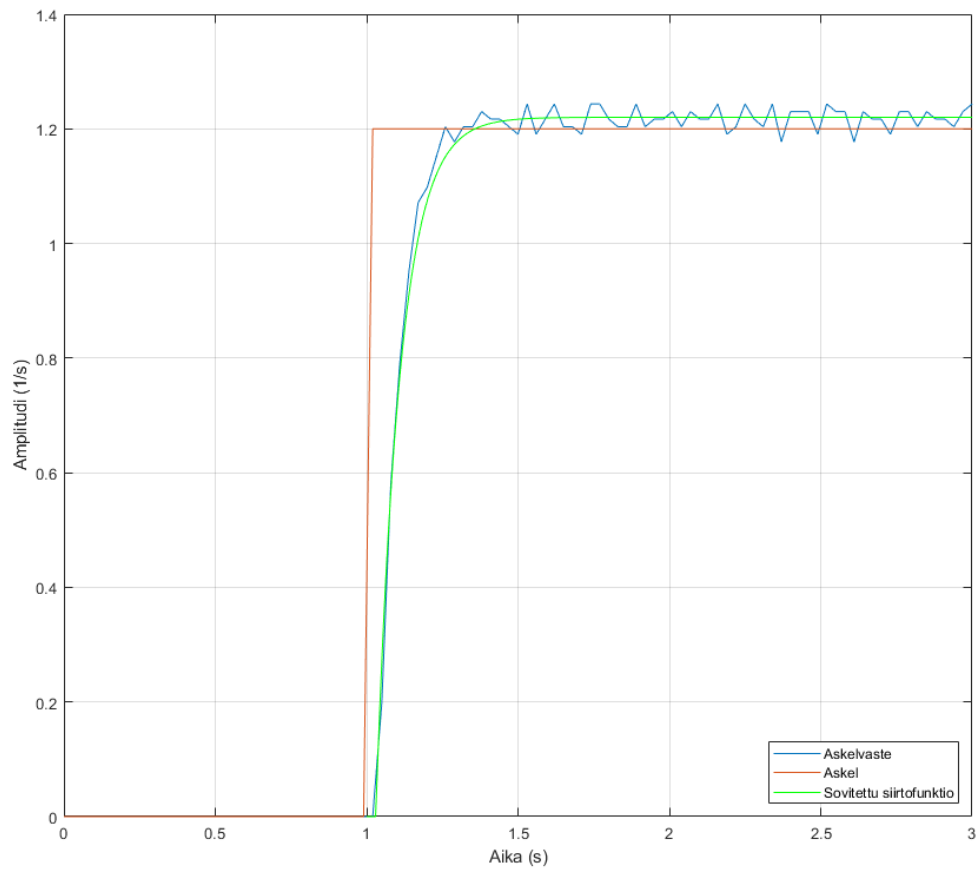
Askelvaste saadaan syöttämällä kuvan 9 mukaiseen avoimeen järjestelmään ohjaussignaalin askelfunktio ajan hetkellä 1 s, jonka jälkeen mittausdata eli ohjaus- ja tulossignaalien arvot tallennetaan Simulinkistä Matlabiin. Testin jälkeen Matlab'in avulla mittausdatasta piirretään askelvasteen kuvaaja. Askelvastekoe suoritetaan usealla eri amplitudin arvolla, jonka avulla mittausdatasta voidaan laskea suhteellisen tarkasti siirtofunktion vahvistus K . Tässä työssä vahvistuksen arvo lasketaan tulo- ja lähtösignaalien

suhteen seitsemän eri testin keskiarvon perusteella, jotka ajettiin eri kierrostaajuuden n arvoilla.



Kuva 9. Avoin järjestelmä, jolle suoritettiin askelvastekokeita.

Aikavakion τ sovittaminen suoritetaan silmämääräisesti kasvattamalla tai pienentämällä soviteen mallin aikavakion arvoa kunnes siirtofunktion askelvasteen kuvaaja vastaa järjestelmän askelvastetta mahdollisimman tarkasti. Viive L määritetään samalla periaatteella. Vaihtoehtoisesti aikavakion τ arvo on määritettävissä askelvasteen kuvaajasta siitä ajanhetkestä, jolloin askelvasteen arvo on 63% sen jatkuvuustilan arvosta. Mittausdatan ja kuvaajan epätarkkuuden vuoksi tätä menetelmää ei käytetä. Epätarkkuus mittausdatassa johtuu mahdollisesti servomootorin sisäisen takometrin ja näytteistysajan $T_s = 0.03s$ tarkkuuden rajallisuudesta. Kuvassa 10 on esitetty järjestelmään syötetyn seitsemän eri suuruisen askeleen keskiarvo ja näiden pohjalta laskettu askelvasteiden keskiarvo sekä sitä vasten määritetty sovite eli malli, jonka arvoiksi laskettiin $\tau = 0.08$ ja $L = 0.03$.



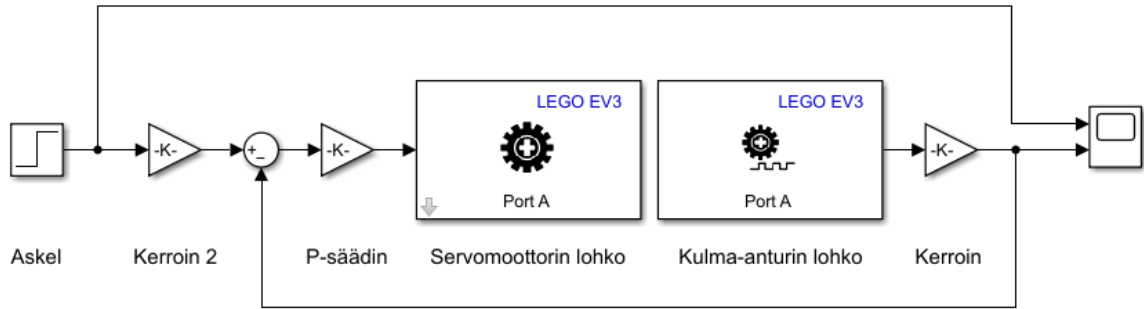
Kuva 10. Askelvasteiden keskiarvon kuvaajaan sovitettu siirtokuntio.

Säätimen parametrien virittäminen suoritetaan käyttäen Ziegler-Nicholsin askelvaste-menettelmää, jolla järjestelmälle saadaan suhteellisen optimoitu vasteaika [15]. Parametrien arvoja joudutaan usein vielä hienosäätämään käsin, mutta menetelmällä saadaan suuntaantavat arvot parametreille. Menetelmä soveltuu käytettäväksi silloin kun järjestelmän dynamiikkaa ei tunneta täysin. Ziegler-Nicholsin säännöt antavat P-, PI, ja PID-säätimille eri suuruiset kertoimet Taulukon 1 mukaisesti. Taulukon viritysäättö ei sisällä systeemin vahvistuksen kompensointia.

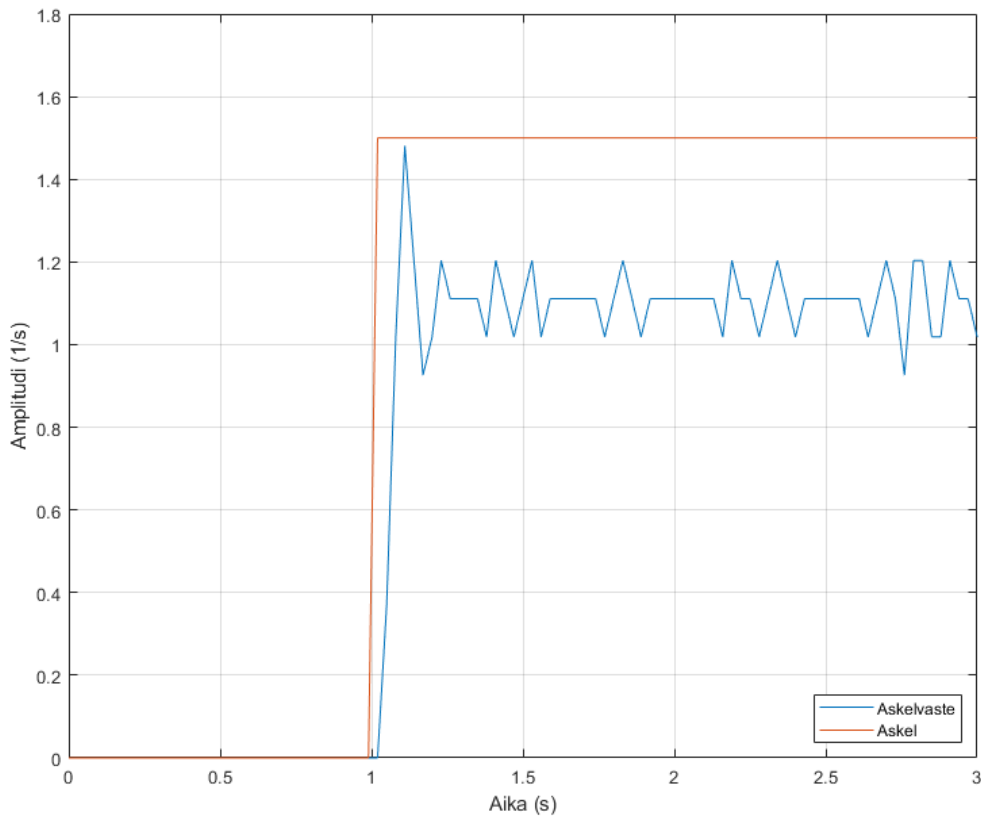
Taulukko 1: Ziegler-Nichols menetelmän kertoimet avoimelle järjestelmälle.

Säätimen tyyppi	K_p	$T_i = \frac{K_p}{K_i}$	$T_d = \frac{K_d}{K_p}$
P	$\frac{\tau}{L}$	∞	-
PI	$0.9 \frac{\tau}{L}$	$\frac{L}{0.3}$	-
PID	$1.2 \frac{\tau}{L}$	$2L$	$0.5L$

P-säädössä ohjaussignaalin arvo riippuu suoraan erosuureen arvosta [16]. Ohjaussignaali säätimelle muodostettiin sisään ja ulos tulevien signaalien erotuksesta, eli erosuureesta. Askelfunktio-lohko toimii järjestelmän tulosignaalina ja sen amplitudin suuruutta voidaan muuttaa. Järjestelmään täytyy lisätä negatiivinen takaisinkytkentä, jonka avulla voidaan laskea tulo- ja lähtösignaalien erotus. Erosuure syötetään P-säätimelle, jonka vahvistuksen K_p arvoksi laskettiin Taulukon 1 mukaisesti $K_p \approx 2.67$. Pelkällä P-säätimellä varustettu järjestelmä jättää lähtösignaaliin jatkuvuustilan virhettä eli käytännössä lähtösignaali ei saavuta haluttua arvoa, joka nähdään kuvasta 12. Lisäksi askelvasteesta nähdään että P-säätimen vahvistus on melko aggressiivinen, sillä se aiheuttaa vasteeseen värähtelyä, joka puolestaan tekee järjestelmästä huonosti vaimennetun. Tämä johtuu Ziegler-Nichols menetelmän tuottamasta parametriarvosta.



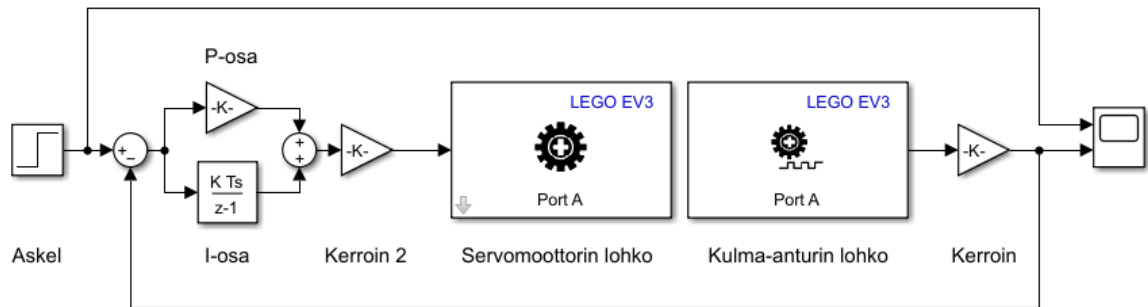
Kuva 11. Takaisinkytketty järjestelmä P-säätimellä.



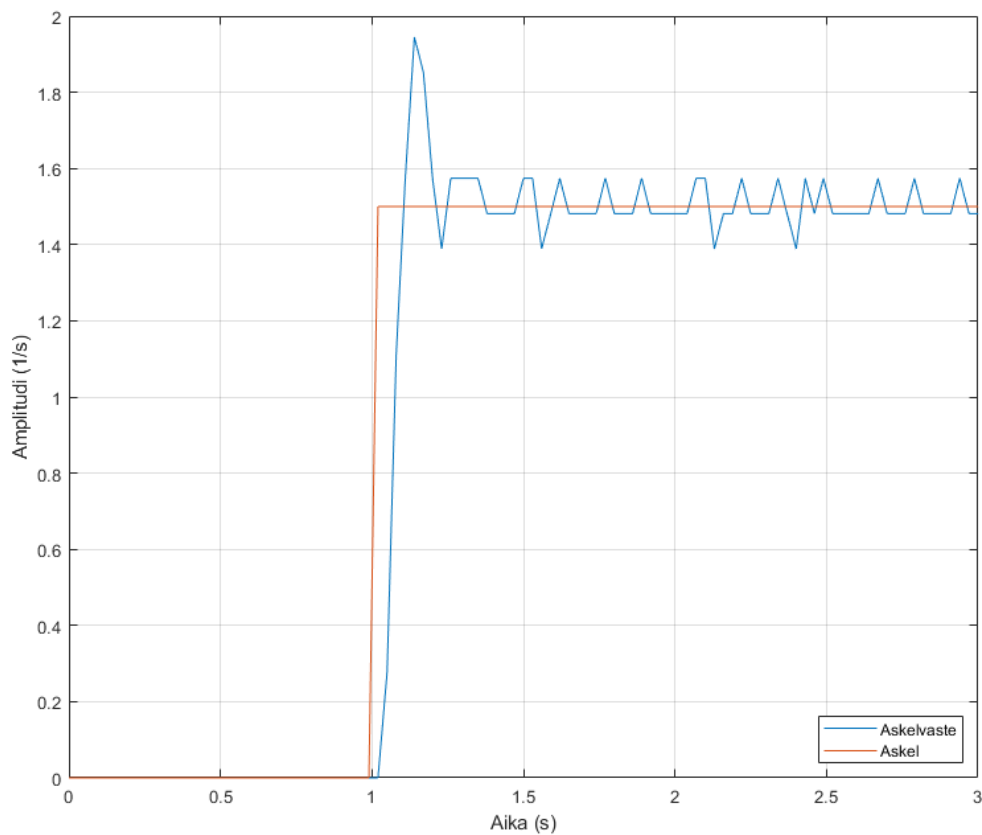
Kuva 12. Järjestelmän askelvaste P-säätimen kanssa.

Jatkuvuustilan virhe saadaan eliminoitua lisäämällä säädinrakenteeseen I-osa eli käyttämällä PI-säätöä. I-osan lisäämisellä on sen lisäksi järjestelmän stabiiliutta heikentävä vaikutus. Käytännössä se näkyy järjestelmän dynamiikassa asettumisajan ja ylityksen huippuarvon suurentumisella, sekä vaimennuksen heikentymisellä. PI-säätimen parametrien arvoiksi laskettiin Taulukon 1 mukaisesti: $K_p = 2.4$ ja $K_i = 24$. Kun PI-säädettyyn kuvan 13 järjestelmään syötetään askel, jonka amplitudin arvo on 1.5 ajanhetkellä 1 s, saadaan kuvan 14 kaltainen askelvasteen kuvaaja. Kuvan 14 kuvaajasta nähdään, että P-osan vahvistus on edelleen liian suuri ja se tekee järjestelmästä heikosti vaimennetun. Ziegler-Nicholsin

menetelmälle on ominaista, että se tuottaa aggressiivisen vahvistuksen, jonka takia parametrin K_p arvoa joudutaan hienosäätämään vielä kokeellisesti. Käytännössä P-osan vahvistuksen arvon pienentäminen kasvattaa nousuaikaa ja pienentää ylitystä. I-osan K_i termin pienentäminen aiheuttaa sen, että I-osa laskee hitaammin erosuureen integraalia, joka johtaa pidempään vasteaikaan. Suurilla K_i arvoilla vaste on nopeampi, mutta se puolestaan kasvattaa järjestelmän värähtelyä.

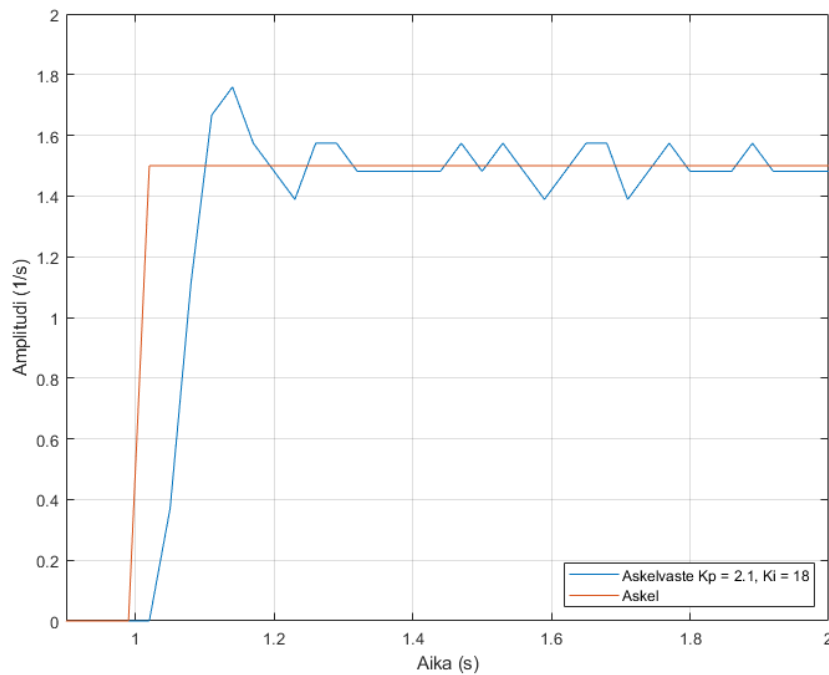
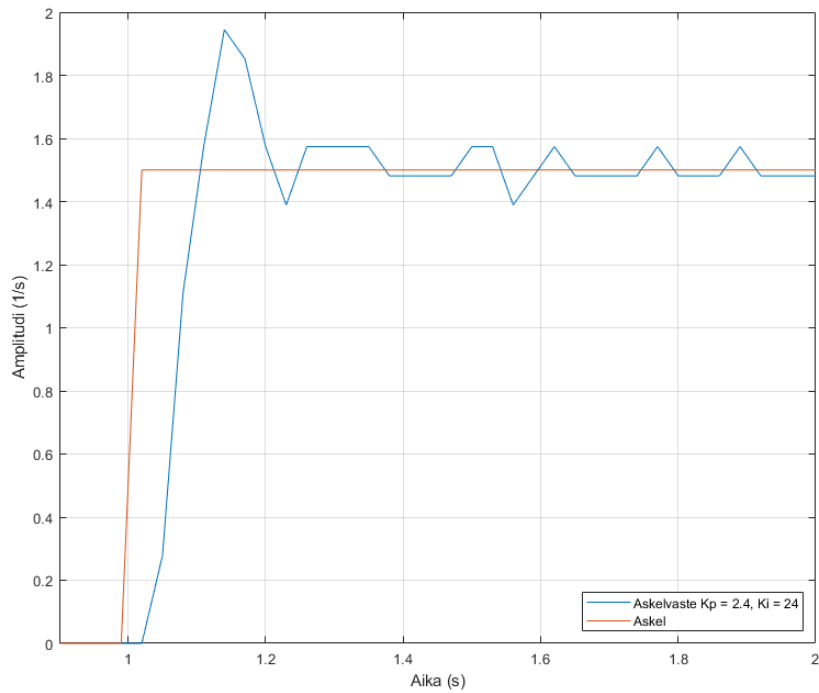


Kuva 13. PI-säädetty järjestelmä.



Kuva 14. Takaisinkytketyn järjestelmän askelvaste, jossa PI-säädin.

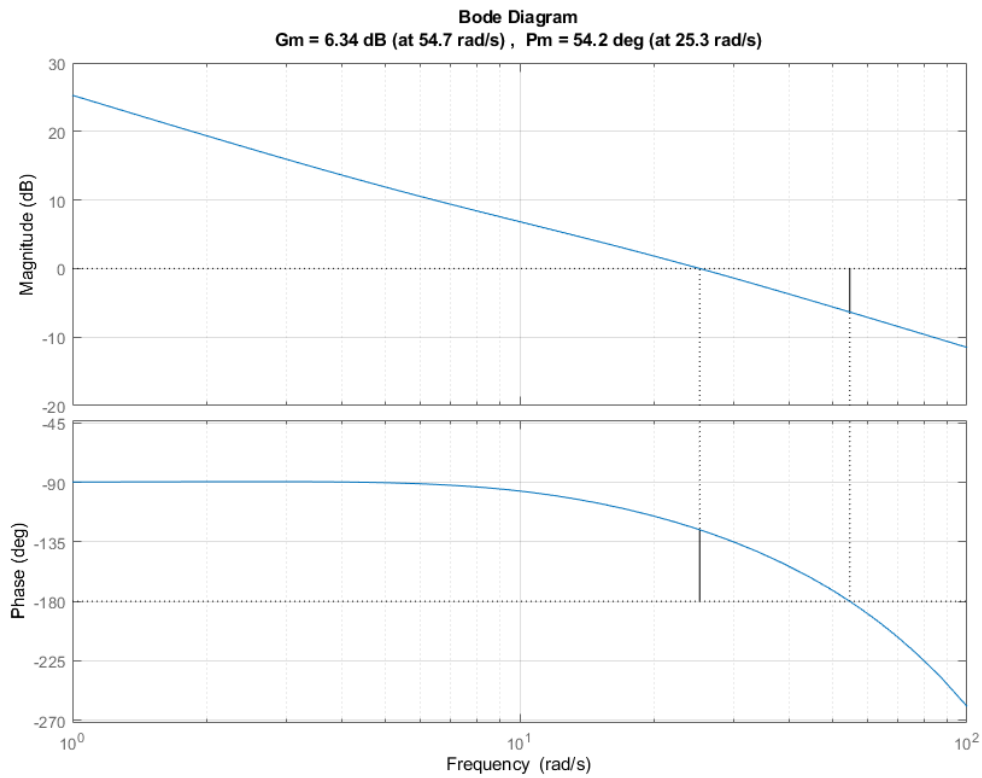
Seuraavaksi pyritään pienentämään kuvassa 14 havaittua ylitystä säätämällä kokeellisesti PI-säätimen P-osan vahvistustermin K_p arvoa. Järjestelmään syötetään edelleen tulosignaalina askel, jonka amplitudin arvo on 1.5 ajanhetkellä 1 s. PI-säätimen K_p termin arvoa pienennetään kokeellisesti 0.1 yksikön verran ja jokaisella kerralla suoritetaan askelvastekoe. Kokeiden perusteella havaittiin, että P-osan vahvistuksen K_p arvoa pienentämällä ylitystä saatiin vähennettyä. Tämän jälkeen säädettiin PI-säätimen K_i termin arvoa pienentämällä sitä kokeellisesti 1 yksikön verran ja jokaisella kerralla suoritettiin uusi askelvastekoe. K_p ja K_i arvojen kokeellisella pienentämisellä saatiin vähennettyä järjestelmän vasteessa havaittavaa ylitystä, joka näkyy kuvassa 15.



Kuva 15. PI-säätetyn systeemin askelvasteet Ziegler-Nichols menetelmän antamilla parametrien arvoilla ylhäällä sekä kokeellisesti käsin säädettyjen parametrien arvoilla alhaalla.

Tutkitaan seuraavaksi hienosäätetyn järjestelmän dynamiikkaa. Analysoidaan PI-säätetyn järjestelmän stabiiliutta määrittämällä vaihe- ja vahvistusvarat Bode-diagrammin avulla. Diagrammin piirtämiseen käytetään Matlabin margin-funktiota, joka piirtää järjestelmän

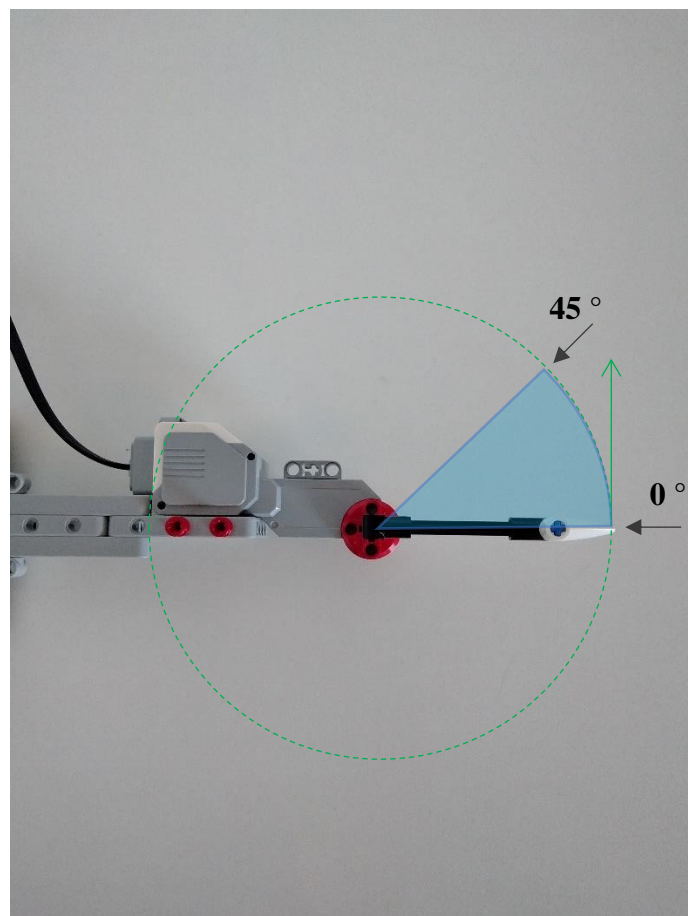
silmukkasiirtofunktion taajuusvasteen logaritmiselle desibeli-asteikolle. Silmukkasiirtofunktio on $L(s) = C(s)G(s)$ ja se koostuu säätimestä $C(s) = K_p + K_i \frac{1}{s}$ sekä siirtofunktiosta $G(s) = \frac{1.0168e^{-0.03s}}{0.08s+1}$. Bode-diagrammissa vahvistuksen (Magnitude) arvon ollessa 0 dB, täytyy vaihevaran (Phase margin) arvon olla positiivinen stabiilissa järjestelmässä. Vahvistusvaran (Gain margin) arvon täytyy puolestaan olla negatiivinen, kun vaiheen (Phase) arvo on -180 astetta. Kuvan 16 Bode-diagrammista nähdään, että aikajatkuvana tarkasteltuna järjestelmä on stabiili. Nämä rajat muuttuisivat tarkasteltaessa diskreettiä toteutusta.



Kuva 16. Avoimen PI-säädetyin järjestelmän Bode-diagrammi.

3.2. Kulmansäätö

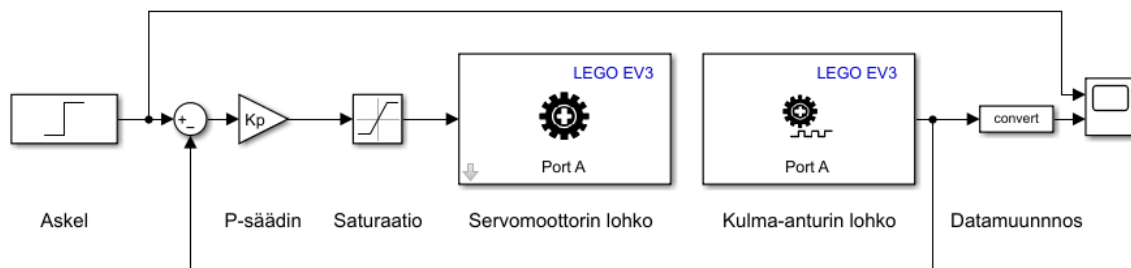
Tarkastellaan seuraavaksi kulmansäätöä käyttämällä kuvassa 17 näkyvää testilaitteistoa, jossa liikkeenohjauksen tavoitteena on siirtää osoitinta alkuasennosta 45 astetta. Kulmansäädön toteuttamista varten kuvan 18 simulaatiomallin kulma-anturin lohkoa valitaan asetus, joka muuttaa anturin antaman kiertymän arvon kumulatiiviseksi kiertymän arvoksi. Tämä tarkoittaa, että kulma-anturi lisää yhteen jokaisen kulmanmuutoksen näytteistysajan $T_s = 0.03$ s välein. Anturista luettava kulman arvo annetaan asteina ja koska järjestelmän tulosihtaalini halutaan myös olevan asteina, se tarkoittaa ettei pyörimisnopeuden säädössä käytettyä skaalaus-kerrointa (Kerroin 1) tarvita.



Kuva 17. Kulmansäätö tavoite.

Järjestelmän siirtofunktiona käytetään kappaleessa 3.1 identifioitua 1. kertaluvun viivästettyä siirtofunktiota $G(s)$. Kulma-anturin antaman kumulatiivisen kiertymän vuoksi järjestelmälle ei voida määrittää P- ja PI-säätimien parametrien arvoja perinteisen Ziegler-Nichols virityssäännön mukaisesti. P-säätimen suunnittelu aloitetaan rakentamalla kuvan 18 kaltainen simulaatiomalli. Kappaleessa 3.1 käytetty toinen skaalaus-kerroin (Kerroin 2)

korvataan Simulink:n saturaatio-lohkokalla, joka rajoittaa tulosignaalin arvon valitulle välille. Servomootoria ohjataan PWM-ohjauksella, joka toimii syöttämällä moottorille kokonaisluku väliltä $[-100, 100]$. Tästä johtuen saturaatio-lohkon minimi- ja maksimiarvoiksi valitaan -100 ja 100 . Mittausdatan tallentamista varten lähtösignaaliin lisättiin datamuunnos-lohko, jonka avulla Simulink määrittää automaattisesti sopivimman datatyypin ulostulevalle signaalille.

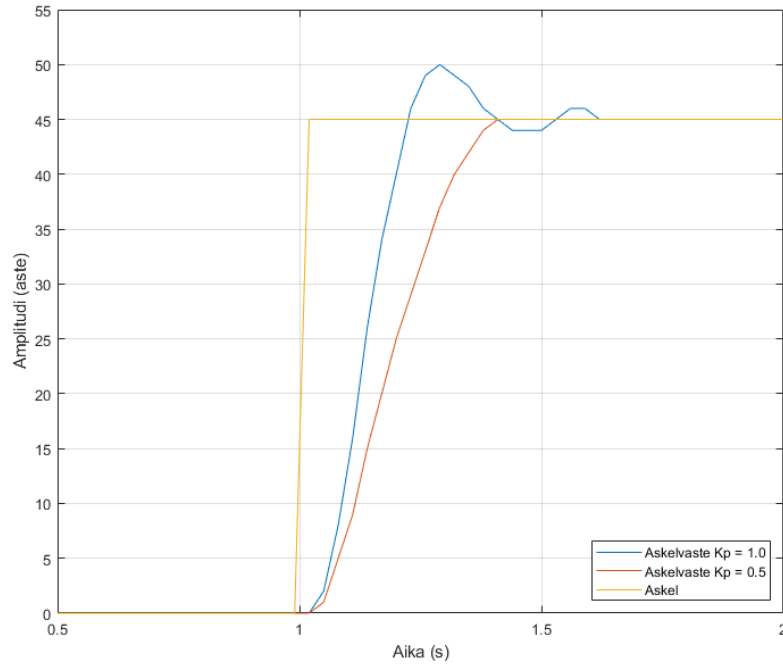


Kuva 18. Kulmansäätöä varten toteutettu simulaatiomalli P-säädöllä.

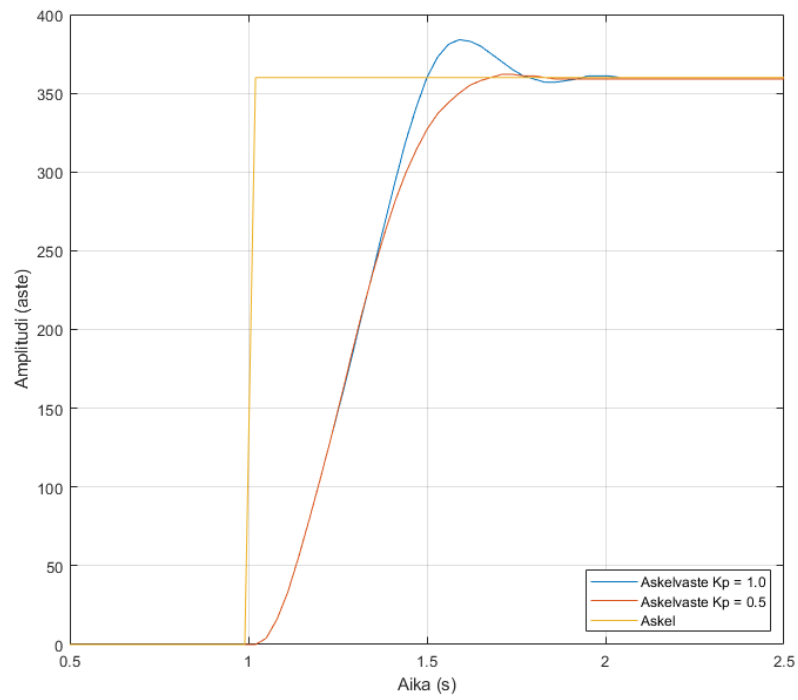
P-säätimen vahvistuksen arvoksi säädetään aluksi $K_p = 1$, jonka jälkeen järjestelmään syötetään askel, jonka amplitudi on 45. Kuvan 19 askeltestin tuottamasta askelvasteesta K_p :n arvolla 1 nähdään, että järjestelmä ajaa kulma-osoittimen haluttuun kulmaan nopeasti, mutta samalla vasteessa tapahtuu ylitystä. Pelkällä P-säätimellä toteutettu järjestelmä ei jätä nyt lähtösignaaliin jatkuvuustilan virhettä, koska kulma-anturin lohko laskee kumulatiivisen kiertymän arvoa eli järjestelmä on käytännössä integroiva prosessi. K_p :n vahvistuksen arvoa pienentämällä, voidaan järjestelmän askelvasteessa havaittua ylitystä vähentää. Suoritetaan toinen askeltesti vahvistuksen arvolla $K_p = 0.5$, jossa tarkoituksena on määrittää kokeellisesti P-säätimelle sellainen K_p :n vahvistuksen arvo, joka eliminoi ylityksen ja tuottaa suhteellisen nopean vasteen. Askelvastetestin tulosten perusteella havaitaan kuvassa 19, että ylitys eliminoituu ja vaste pysyy hyvänä P-säätimen vahvistuksen K_p pienetessä arvoon 0.5.

Tämän jälkeen testataan säätimen toimivuutta askelvastekokeella, jossa amplitudin arvo kasvatetaan 360 asteeseen eli lego-robotin osoittimella tehdään täysi kierros. Testin tulokset molemmilla P-säätimen arvoilla näkyvät kuvassa 20, josta havaitaan, että järjestelmän vaste käyttäytyy samalla tavalla kuin pienemmällä kulmaohjeella. Tutkitaan vielä säädetyn järjestelmän dynamiikkaa, jotta varmistutaan sen stabiiliudesta.

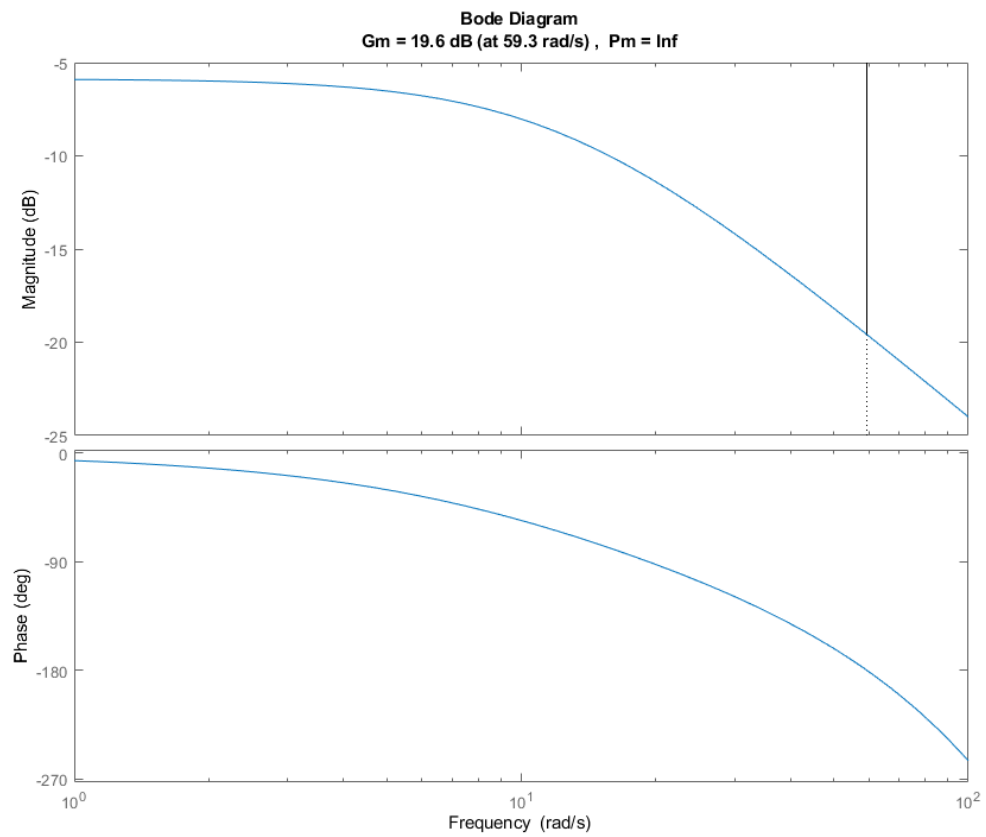
Silmukkasiirtofunktio $L(s) = C(s)G(s)$, jossa säädin on nyt $C(s) = K_p$ ja systeemi $G(s) = \frac{1.0168e^{-0.03s}}{0.08s+1}$. Kuvan 21 Bode-diagrammista nähdään, että järjestelmä on aikajatkuvana tarkasteltuna stabiili.



Kuva 19. Järjestelmän askelvaste P-säädön vahvistuksen eri arvoilla, kun amplitudin arvo on 45.



Kuva 20. Järjestelmän askelvaste P-säädön vahvistuksen eri arvoilla, kun amplitudin arvo on 360.



Kuva 21. Avoimen P-säädetyin järjestelmän Bode-diagrammi.

4. Yhteenveto ja johtopäätökset

Työssä tehtiin tutkimusta Legon EV3-robottien käyttämisestä opetustarkoitukseen kirjallisuuskatsauksen avulla, jossa käytiin läpi erilaisten lego-robottien toimintaperiaatteita ja säätötavoitteita. Esimerkit lego-roboteista valittiin katsaukseen niin, että ne havainnollistavat lego-robottien käyttömahdollisuuksia sekä tukevat tutkielman tavoitetta tehdä yksinkertainen lego-robottikonfiguraatio liikkeenohjauksen havainnollistamiseen.

Liikkeenohjauksen toteuttaminen aloitettiin rakentamalla yksinkertainen lego-robottikonfiguraatio, jolle luotiin simulaatiomalli Simulink:ssä ohjelmaksi käännettäväksi. Robotille valittiin kaksi liikkeenohjaustavoitetta, jotka olivat servomoottorin pyörimisnopeuden- ja kulmansäätö. Nämä tavoitteet soveltuvat liikkeenohjauksen havainnollistamiseen hyvin. Testijärjestelmän avulla suoritettiin askelvastekokeita, joiden pohjalta sen dynamiikalle identifioitiin 1. kertaluvun viivästetty siirtofunktio. Siirtofunktio malli sovitettiin käsin hyödyntämällä usean askelvastekokeen tuottamaa keskiarvoastetta. Mallin tarkkuutta olisi mahdollista parantaa käyttämällä tarkempia järjestelmän identifiointimenetelmiä (kuten pienimmän neliösumman sovite askelvasteeseen), mutta se ei ollut tämän tutkielman painopiste. Järjestelmille suunniteltiin P- ja PI-säätimet, joiden toimintaa tarkasteltiin kokeellisesti Lego-ympäristön avulla. Säädinten parametrit valittiin ensin Ziegler-Nichols viritysääntöjen perusteella, jonka jälkeen niiden arvoja muutettiin kokeellisten testien perusteella. Järjestelmien stabiiliutta analysoitiin määrittämällä niiden vaihe- ja vahvistusvarat Bode-diagrammin avulla, jonka perusteella todettiin säätimien olevan stabiileja. Säädön viritystavaksi olisi tässä työssä voitu ottaa jokin muu tapa, kuten IMC-perusteinen, jolloin suunnittelun olisi saanut kaistanleveysperusteiseksi. Toisaalta käytetyllä säätimellä saatiin työn päätavoite toteutettua eli demonstraatioympäristön toteutus yksinkertaisella säätörakenteella.

Lähteet

- [1] C. de Vries, Y. Wu, M. Ford ja O. Onipede Jr, ”122nd ASEE Annual Conference & Exposition” [Online]. Saatavissa: <https://peer.asee.org/using-lego-mindstorms-in-a-control-systems-lab-to-impact-next-generation-engineers-work-in-progress> [Viitattu 10 Maaliskuu 2022]
- [2] Opetushallitus, ”Perusopetuksen opetussuunnitelman perusteet 2014” [Online]. Saatavissa: https://www.oph.fi/sites/default/files/documents/perusopetuksen_opetussuunnitelman_perusteet_2014.pdf [Viitattu 28 Maaliskuu 2022]
- [3] M. Masril, B. Hendrik, H. Fikri, Al. Hazidar, B. Priambodo, E. Naf'an, I. Handriani, Z. Putra ja A. Nseaf, ”*Journal of Physics: Conference Series 1339 012082*” [Online]. Saatavissa: <https://iopscience.iop.org/article/10.1088/1742-6596/1339/1/012082> [Viitattu 10 Maaliskuu 2022]
- [4] F. Bevrnja, M. Bevrnja ja M. Petrovic, ”*Periodicals of Engineering and Natural Sciences*” [Online]. Saatavissa: <http://pen.ius.edu.ba/index.php/pen/article/view/1084> [Viitattu 10 Maaliskuu 2022]
- [5] A. Mikael ja L. Miao, ”*ACM Digital Library*” [Online]. Saatavissa: <https://dl.acm.org/doi/10.1145/3239576.3239624> [Viitattu 10 Maaliskuu 2022]
- [6] A. Ruusila, ”*Lego-servokäytön kokeellinen mallinnus ja säätö*” [Online]. Saatavissa: <https://lutpub.lut.fi/handle/10024/161658> [Viitattu 6 Toukokuu 2022].
- [7] Z. Shi, ”*Identification and Control of a LEGO EV3 Self-balancing Robot: An educational Tool in automatic control classes*” [Online]. Saatavissa: https://www.politesi.polimi.it/bitstream/10589/165586/1/shi_917971_thesis.pdf [Viitattu 10 Maaliskuu 2022].
- [8] P. Vanhanen, ”*Lego - robotti säätötekniikan opetuksen demonstraatiövälineenä*” [Online]. Saatavissa: <https://lutpub.lut.fi/handle/10024/159492> [Viitattu 1 Huhtikuu 2022].
- [9] T. Silver, ”*Control Synthesis for Balancing Robots*” [Online]. Saatavissa: <https://aaltodoc.aalto.fi/handle/123456789/32398> [Viitattu 1 Huhtikuu 2022].
- [10] Robotsquare, ”*Tutorial Building BALANC3R*” [Online]. Saatavissa: <http://robotsquare.com/2014/06/23/tutorial-building-balanc3r/> [Viitattu 6 Toukokuu 2022].
- [11] MinSeg, ”*MinSegShield M1V4 Single Axis*” [Online]. Saatavissa: <https://minseg.com/collections/minseg-kits/products/minsegshield-m1v3-2-single-axis-minseg-uno-mega-due-compatible> [Viitattu 6 Toukokuu 2022].
- [12] D. Benedetteli, ”*Lego Rubik Cube Solver*” [Online]. Saatavissa: <https://robotics.benedetteli.com/lego-rubik-cube-solver/> [Viitattu 6 Toukokuu 2022].
- [13] S. Lee, ”*Knowledge-Based Design Laboratory*” [Online]. Saatavissa: <https://kbd.yonsei.ac.kr/researches/lego> [Viitattu 6 Toukokuu 2022].

- [14] Mathworks, "Reinforcement Learning with MATLAB and Simulink" [Online]. Saatavissa: <https://se.mathworks.com/content/dam/mathworks/ebook/gated/reinforcement-learning-ebook-all-chapters.pdf> [Viitattu 28 Maaliskuu 2022].
- [15] J. a. N. N. Ziegler, *Optimum Settings for Automatic Controllers. Transactions of the ASME, vol. 64, 759-768.*, American Society of Mechanical Engineers, 1942.
- [16] T. Harju ja A. Marttinen, "Säätötekniikan koulutusmateriaali" [Online]. Saatavissa: https://www.automaatioseura.fi/site/assets/files/1426/pid_kirja_1-1.pdf [Viitattu 13 Toukokuu 2022]