

Ossi Koski

TARKKUUDEN PARANTAMINEN ROBOTILLA TARJOILTAVIEN JUOMIEN KAATOMÄÄRISSÄ

Käytännön koe Drinkkirobotti-sovelluksella

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Lionel Hulttin
Huhtikuu 2021

TIIVISTELMÄ

Ossi Koski: Tarkkuuden parantaminen robotilla tarjoiltavien juomien kaatomääriissä
Kandidaatintyö
Tampereen yliopisto
Teknisten tieteiden kandidaatin tutkinto-ohjelma
Huhtikuu 2021

Avainsanat: robotiikka

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

ABSTRACT

Ossi Koski: Otsikko englanniksi
Bachelor of Science Thesis
Tampere University
Bachelor's Programme in Engineering Sciences
April 2021

Keywords: robotics

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tampereella, 13. huhtikuuta 2021

Ossi Koski

SISÄLLYSLUETTELO

1	Johdanto	1
2	Lähtökohdat	2
2.1	Drinkkirobotti 5.0	2
2.2	Vanha kaatoratkaisu	4
2.3	Vanhan kaatoratkaisun ongelmat	6
2.4	Mahdollisten ratkaisujen kartoitus	8
3	Ratkaisu	12
3.1	Uusi kaatoratkaisu	12
3.2	Uuden kaatoratkaisun analysointi	16
4	Tulokset ja vertailu	19
5	Yhteenvedo	21
	Lähteet	22

LYHENTEET JA MERKINNÄT

ry	rekisteröity yhdistys
TAU	Tampere University
TUNI	Tampere Universities
UDP	User Datagram Protocol
UML	Unified Modeling Language

1 JOHDANTO

Robotiikan käyttö maailmanlaajuisesti yleistyy koko ajan huomattavalla nopeudella. Roboteista tulee yhä älykkäämpiä ja taloudellisesti kannattavampia. Reilusti suurin osa roboteista on teollisuusrobotteja, jotka ovat kätketty tuotantolaitoksiin [1]. Palvelurobotiikka kuitenkin kasvattaa myös osuttaan, ja esimerkiksi robottiruohonleikkurit ja -imurit ovat ai-kaisempaan verrattuna suhteellisen tuttuja näkyjä kotitalouksissa. Myös esimerkiksi sote-aloilla palvelurobotteja käytetään jo melko laajalti [2]. Yhä innovatiivisempia tapoja käyttää robotteja palvelualalla tulee esiin. Yksiä näistä ovat ravintola-alan sovellukset.

Juomia tarjoilevat robotit ovat toistaiseksi vielä melko uniikkeja. Juomien tarjoilu robotilla saattaa joissain tilanteissa nopeuttaa tarjoilua, mutta suurin myyntivaltti siinä on sen näytävyys. Juomien automaatisessa tarjoilussa ongelmia aiheuttavat nesteen ominaisuudet. Usein robotit käyttävät hyväksi jonkinlaista juoma-automaattia [3] tai sitten pullot roikkuvat väärin päin esimerkiksi robotin päällä ja niissä on erityinen korkki, jota painamalla ja näin korkin venttiilin avaamalla robotti voi laskea pullossa juomaa [4]. Näissä ratkaisuissa hyvä puoli on se, että nesteen virtaus tarjoiluastiaan on lähellä vakiota ja täten kokonaisjuomamäärä on helposti säädetävissä. Tässä kandidaatintyössä käsitellään tapausta, jossa robotti kaataa juomaa pullossa mukiin. Tällöin juomankaatotehtävä voi antaa käytännössä mille tahansa robotille tarvitsematta juoma-automaatteja tai kokonaista robottisoluta, jossa pullot roikkuvat robotin päällä. Pullossa kaataessa kuitenkin nesteen virtauksen ja pullon kallistuksen mukana tulevat muuttujat tekevät juomamäärän säätelystä hankalampana.

Tässä kandidaatintyössä kerrotaan Pullonkaula ry:n Drinkkirobotti-sovelluksesta, ja siitä miten sillä on aikaisemmin toteutettu juomien kaato avonaisesta pullossa. Tämä tapa on sisältänyt ongelmia juuri kaatomäärän tarkkuuden suhteen, ja näitä ongelmia käsitellään alaluvussa 2.3. Sen jälkeen käydään läpi sitä, miten on pyritty kehittämään parempi tapa ohjata kaodon määrää sovelluksessa. Lopuksi vertaillaan kahta tapaa keskenään. Vertailu on pyritty tekemään sekä juuri tämän robottisolun ominaisuuksiin liittyen että myös yleisemmällä tasolla. Nämä tämän työn pohjalta on mahdollista soveltaa tuloksia muihin samankaltaisiin sovelluksiin.

2 LÄHTÖKOHDAT

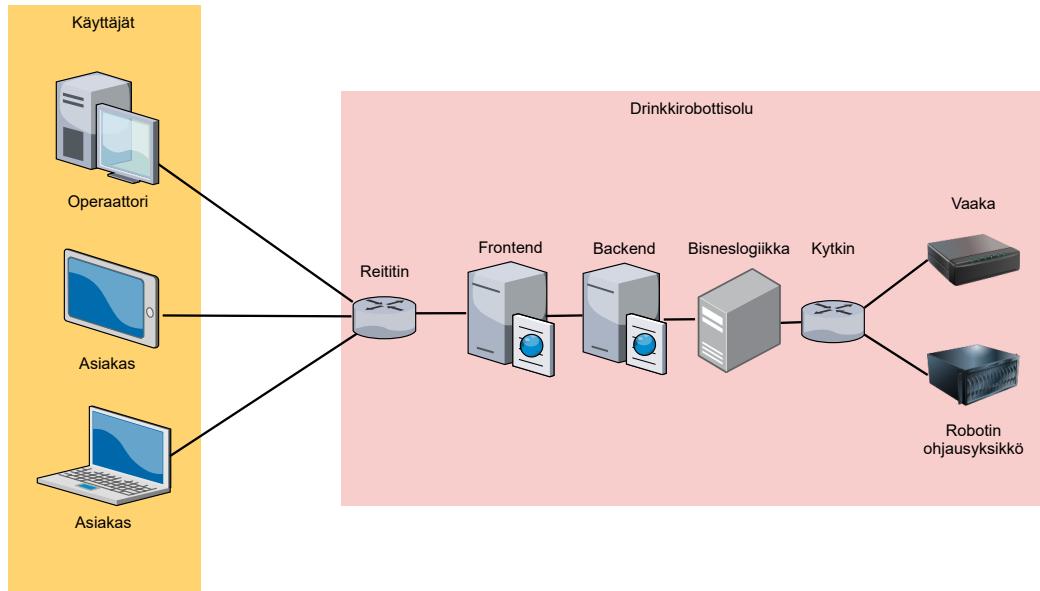
2.1 Drinkkirobotti 5.0

Drinkkirobotti 5.0 on Pullonkaula ry:n eli Tampereen yliopistossa toimivan tuotantoteknikan ammattiainekerhon opiskelijoiden vapaa-ajalla tekemä projekt. Robottisolu koostuu Yaskawan HC-10-monitoimirobotin You Teach Me -solusta, ja sen ympärille rakennetuista baaritiskeistä. Drinkkirobottisolu näkyy kuvassa 2.1. You Teach Me on Yaskawan valmiiksi valmisteltu opetuskäyttöön tarkoitettu robottisolukokonaisuus [5]. Robottisolun Baaritiskeissä on sisäpuolella pullohylyt, ja kun robotilta tilataan juomaa, se hakee oikeat pullot hyllystä ja valmistaa niistä monikomponenttijuomia kaatamalla juomaa lasiin. Baaritiskin päällä on neljä mukipaikkaa, joihin robotti voi kaataa juomia.



Kuva 2.1. Drinkkirobottisolu

Robotilta juomien tilaaminen tapahtuu tabletille tehdyn käyttöliittymän avulla. Yaskawan HC-10 on yhteistyörobotti, eli se sisältää voima-antureita ja näiden avulla ihmisen ja robotin välinen yhteistyö on helpottunut. [6]



Kuva 2.2. Ohjelmistorakenne [7, s. 11]

Drinkkirobotin ohjelmisto koostuu neljästä tasosta. Ohjelmistorakennetta on havainnollistettu kuvassa 2.2. Front end sisältää React.js:llä tehdyt käyttöliittymät operaattoria ja asiakasta varten. Operaattorinäkymästä voi esimerkiksi hallita robottisolun pullohyllyssä olevia pulloja lisäämällä tai poistamalla niitä ja valita robotin idle-tilassa tekemiä liikkeitä. Asiakasnäkymästä voi selata tarjolla olevia juomia ja niiden sisältämää komponentteja ja tilata juomia kerrallaan 1–4 kappaletta. Back end toimii Node.js:llä, ja se sisältää tiedon meneillään olevasta tapahtumasta, missä juomia tarjoillaan, ja siinä tapahtumassa tarjolla olevista juomista, sekä niiden komponenteista ja määristä.

Itse robottikoodi on suhteellisen yksinkertaista. Siinä käytetään Yaskawan Inform-ohjelmointikieltä, ja robottia ohjaan Yaskawan YRC-1000-ohjausyksikkö. Robottikoodi sisältää jojeja eli tehtäväohjelmia, jotka sisältävät robotin liikeratoja ja niiden parametrien määrittelyjä ja joissakin väleissä lyhyitä odotusaikoja. Robottisolun aivoina voidaan pitää sen bisneslogiikkatasoa, joka on kirjoitettu C#-kielellä. Logiikka ohjaan robottia kertomalla, mikä robotin tehtäväohjelma milloinkin suoritetaan. Se siis käsitteli front endiltä saadut tilaukset ja päätää esimerkiksi, mikä pullo haetaan ja mihin lasiin juomaa kaadetaan. Logiikassa on myös reaalialkainen tieto pullohylyn tilanteesta, eli mitä pulloja hyllyssä on ja kuinka paljon juomaa missäkin pullossa on jäljellä. Kun operaattori lisää pullon, hän arvioi pullossa olevan juoman määrän ja merkitsee sen käyttöliittymään, jolloin juoma-määrä välitetään logiikkaan. Kun pullossa kaadetaan juomaa, päivittyy uusi tieto pullossa olevasta juomasta. Jos pullossa on liian vähän juomaa jäljellä minkään drinkin tekoon, robotti poistaa pullon hyllystä automaattisesti. Näiden lisäksi robotin sisällä on Raspberry

Pi -tietokone, joka ohjaa robotin tarttujaa. Tarttujaa voidaan ohjata robottikoodista siihen tarkoitetuilla tehtäväohjelmilla, jolloin robottikoodi kutsuu Raspberry Pitää.

Uusimpana lisäyksenä robottisoluun on implementoitu vaaka, joka on niin sanotulla pullonvaihtopisteellä. Vaa'an toiminta on hyvä esimerkki robotti-bisneslogiikkarajapinnasta, sillä vaa'an voi kalibroida ja sen jokaisella hetkellä näyttämää arvoa voi kysyä logiikan avulla. Tämän vaa'an avulla robotti tunnistaa, onko pullonvaihtopisteellä pulloa. Tätä käytetään esimerkiksi uuden pullon lisäämisessä: Kun robotti hakee uutta pulloa hyllyyn, se pysähtyy ensin pullonvaihtopisteen eteen. Jos vaaka kertoo, että pullonvaihtopiste on tyhjä, logiikka ei käske robottia hakemaan olematonta pulloa, vaan käskee sen odottamaan tietyn ajan. Robotti hakee pullon vasta, kun se lisätään pullonvaihtopisteelle. Tämän toiminnallisuuden avulla robotti ei siis vie olematonta pulloa hyllyyn, jolloin back endissä olisi tieto pullossa tietyllä kohdalla pullohylyyä, vaikka pullo ei oikeasti ole. Samoin, jos robotti on poistamassa pulloa pullohylystä, ei se vie sitä pullonvaihtopisteelle, jos siellä on jo pullo. Näin robotti ei työnnä vanhaa pulloa vaihtopisteeltä alas, vaan odottaa, että se poistetaan sieltä. Valmisteilla on myös ominaisuus, jossa logiikka kertoo back endille, miten paljon juomaa lisättyssä pullossa on, ja operaattorin tehtäväksi jäädä vain tarkistaa lukema, eikä itse arvioda lisättävässä pullossa olevaa juoman määrää.

2.2 Vanha kaatoratkaisu

Tässä alaluvussa käsitellään useamman vuoden käytössä ollutta ratkaisua juomien kaatamiseen Drinkkirobotilla. Ratkaisussa juomien kaadolle robottikoodissa on yksittäinen tehtäväohjelma, joka on esitetty alla.

```

1 NOP
2   ' PROGRAM RUNNING
3 SET I000 1
4   ' CALCULATE POUR PARAMETRES
5 IFTHENEXP I003>3
6   SET I011 I003
7   MUL I011 60
8   SUB I011 171
9 ELSE
10  SET I011 1
11 ENDIF
12  CALL JOB:GOCENTRALPOS
13  SET I005 0
14  *LABEL10
15 IFTHENEXP I005=I004
16  JUMP *LABEL999
17 ENDIF
18  CALL JOB:CALCULATEPOUR
19  ' MOTION TO POURTARGET
20  MOVL P021 V=150.0 DEC=20

```

```

21 MOVL P023 V=50.0 DEC=20
  'POUR
23 IMOV P008 V=40.0 TF ACC=20 DEC=20
  TIMER T=I011
25 ' LEVEL BOTTLE AND MOVE AWAY
  MOVL P023 V=40.0 ACC=20 DEC=20
27 IMOV P009 V=40.0 TF ACC=20 DEC=20
  'ADD COMPLETED POUR TO COUNTER
29 ADD I005 1
  'JUMP BACK TO POUR NUMBER CHECK
31 JUMP *LABEL10
*LABEL999
33 'MOTION TO CENTRAL POSITION
  CALL JOB:GOCENTRALPOS
35 'PROGRAM COMPLETED
  SET I000 0
37 END

```

Ohjelma 2.1. POURDRINKS-tehtäväohjelma

Kaadon tarkkuutta tarkasteltaessa merkittävä osa tehtäväohjelmassa on ensimmäinen if-else-haara, joka alkaa ohjelman 2.1 riviltä 5. Se käsittelee kaadossa rivillä 24 käytetystä ajastimelle annetun arvon laskentaa. Haluttu kaodon määrä senttilitroina on asetettu muuttujaan I003. POURDRINKS-tehtäväohjelma laskee alussa ajan, jonka mukaan robotti kaataa juomaa. Kuten ohjelmasta 2.1 nähdään, kaodon aika sekunteina saadaan kaavalla

$$t = 60 \cdot V - 171, \quad (2.1)$$

jossa V on muuttuja I003 eli haluttu tilavuus senttilitroina. Kyseessä on lineaarinen yhtälö, jossa kulmakerroin 60 ja vakiotermi -171 on määritetty kokeellisesti kaatamalla eri määriä juomia. Ensimmäisen if-elsen tehtävä on myös tarkastaa, onko haluttu kaatomääri alle kolme senttilitraa. Tätä pienemmällä määrittä yllä kuvattu lineaarinen funktio antaisi negatiivisen ajan. Niinpä kaikilla määrittä, jotka ovat alle kolme senttilitraa, kaatoajaksi on määritelty yksi sekunti.

Tehtäväohjelma toimii siis pääpiirteittäin siten, että robotti liikkuu kohteena olevan mukin kohdalle ja kallistaa pulloa. Kun pullo on kallistunut kaatoasentoon, käynnistyy ajastin, jonka kesto on määritelty muuttujaan I011.

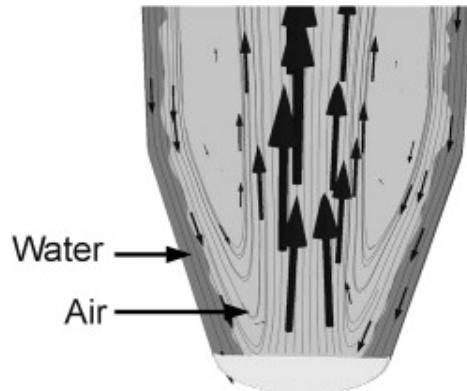
Tämä tehtäväohjelma hoitaa kaodon useammalle mukille samalla, sillä siinä käytetään silmukkaa LABEL10- ja LABEL999-lippujen sekä muuttujien I004 ja I005 avulla. Muuttuja I005 on lasin numero, johon robotti sen hetkisellä iteraatiolla kaataa. Muuttuja I004 taas on haluttujen lasillisten määrä. Tehtäväohjelma CALCULATEPOUR laskee koordinaatit, joissa kohdemuki on ja tallettaa sen P-alkuisiin paikkamuuttuihin. Tässä se käyttää apuna

muuttujaan l005. Mukipaikat ovat vaakasuorassa rivissä, joten siirryttääessä mukilta toiseen riittää vain lisätä robotin koordinaatiston x-koordinaattia mukien etäisyyden verran.

2.3 Vanhan kaatoratkaisun ongelmat

Edellisessä alaluvussa kuvattu ratkaisu on toiminut kohtalaisten hyvin useamman vuoden ottaen huomioon sen yksinkertaisuuden. Siinä on kuitenkin selkeitä heikkouksia. On selvää esimerkiksi, että alle kolmen senttilitran kaatoja tällä toteutuksella ei käytännössä voi tehdä. Tämä toteutus ei myöskään ota huomioon mitään muita muuttujia mitä juomien kaadossa voi olla halutun juomamäärän lisäksi. Tässä alaluvussa käydään läpi erilaisia ongelmia, joita vanhassa kaatoratkaisussa on ollut.

Eri muuttujien vaikutusta juoman kaatoon on yritetty vähentää käyttämällä aina samanlaisia pulloja, ja sen lisäksi pulloissa kaatonokkia. Tämä helpottaa myös kaatoliikkeen ohjelointia robotille, kun pullossa tuleva juoman virtaus osuu mukiin helpommin. Virtaus on myös tasaisempi kaatonokkaa käytettäessä. Kaatonokassa on pieni korvausilmaputki, joka tasaa painetta pullon sisä- ja ulkopuolella. Kaatoja pullossa ilman kaatonokkaa ovat simuloineet Geiger et al [8]. Simulaatiosta huomataan, miten normaalisti pullon suusta kaadettaessa ilman pitää virrata kaatoaukosta välillä sisäänpäin. Tämä aiheuttaa nesteen virtaukseen nykivää liikettä, jota ei kaatonokkaa käytettäessä tule korvausilmaputken ansiosta.



Kuva 2.3. Juoman virtaus [8]

Varsinkin kapeilla ulostuloputkilla ilmavirtaus juoman virtausta vastaan tekee juoman virtauksesta huomattavasti turbulentiseman, mikä vaikuttaa juoman virtausnopeuteen [8]. Kapea ulostuloputki on siltä kannalta parempi, että sellaista käytettäessä robotin on paljon helpompia osua lasiin juomaa kaataessa, kun virtaus on kooltaan kapeampi.

Juoman kaatamisessa robotti on kuitenkin omat ongelmansa, vaikka käytössä olisikin kaatonokka. Suurin ongelmista tässä sovelluskohteessa on se, että korvausilmaputkeen saattaa mennä pisara nestettiä. Näin pullon sisään ei tule korvausilmaa, ja kaatonokasta ei virtaa nestettä liian suuren paine-eron takia, vaikka pullo olisi kokonaan ylös salaisin.



Kuva 2.4. Kaatonokka, johon on merkitty ulostuloputki ja korvausilmaputki

Tällaista tilannetta ei robotti ole osannut havaita. Sen takia asiakkaan lasi on jäynyt käytännössä tyhjäksi ja juomatilaus on jouduttu tekemään uudelleen. Tavoite uudelle kaatoratkaisulle on tällaisen tilanteen tunnistaminen ja siitä palautuminen.

Juomien kaatoon muuttujia tuo huomattavasti lisää myös se, millaista juomaa kaadetaan. Veden, mehujen ja muiden kuohumattomien ja viskositeetiltään samankaltaisten juomien kaataminen onnistuu vielä melko hyvin lineaarisen funktion mukaan, mutta ongelmia alkaa tulla varsinkin kuohuvien juomien kanssa. Tällöin saman lineaarisen funktion käyttö ei enää toimi, sillä esimerkiksi kuplat aiheuttavat turbulenssia kaodon aikana. Vettä paljon suuremman viskositeetin omaavat nesteet ovat dynamiikaltaan erilaisia, sillä korkeampi viskositeetti pienentää virtausnopeutta [9]. Tämän takia niiden kaataminen ei myöskään onnistuisi vanhalla toteutuksella.

Pullossa jäljellä oleva juoman määrä vaikuttaa kaadossa siihen, miten paljon juomaa virtaa pullostaa. Kun pullo on täysin kaatoasennossa, juomaa virtaa vakiomäärä, mutta ero tulee kallistus- ja suoristusliikkeiden aikana. Kun pullo on täynnä, juomaa alkaa virtaamaan pulosta paljon pienemmällä kallistuskulmalla kuin siinä tapauksessa, jos pullossa on enää vähän juomaa jäljellä. Tämän vaikutusta kaatomäärään on pyritty pienentämään tekemällä kallistus- ja suoristusliikkeet nopeasti.

Juomien kaatomäärän tarkkuus on tärkeää monestakin syystä. Varsinkin jos tarjoiltava

juoma sisältää alkoholia, vaikuttaa asiaan Suomen lainsäädäntö ja alkoholilaki. Alkoholi-laissa on kirjattu alkoholijuomien perusannokset: "Väkevien alkoholijuomien perusannos on 4 senttilitraa, enemmän kuin 15 tilavuusprosenttia etyylialkoholia sisältävän miedon alkoholijuoman perusannos on 8 senttilitraa, enemmän kuin 8 mutta enintään 15 tilavuus-prosenttia sisältävän miedon alkoholijuoman perusannos on 12 senttilitraa ja muun mie-don alkoholijuoman perusannos on 33 senttilitraa." [10] Lakisääteisyyden lisäksi on monta komponenttia sisältävässä juomasekoituksissa maun kannalta tärkeää, että komponentte-ja juomasekoituksessa oikea määrä. Usein asiakkaat ovat myös kiinnostuneet siitä, onko robotti kaatanut tasaisesti ja näin he vertailevat juomalaseja keskenään.

Asiakkaan näkökulman lisäksi ongelmat kaatojen määrässä kertautuvat, kun robotin back end saa väärää tietoa pulloista kaadetuista määristä. Jos robottia kutsutaan kaatamaan esimerkiksi kymmenen senttilitraa juomaa, ja juomaa kaatuukin senttilitra liikaa, niin back end laskee pullossa lähteneen silti kymmenen senttilitraa. Kun tämä tapahtuu tarpeeksi monta kertaa, niin back endin tieto pulloissa jäljellä olevista juomista ei pidä enää paik-kaansa. Pahimmissa back endissä voi olla tieto, että pullossa riittäisi juomaa vielä yhteen lasilliseen, mutta todellisuudessa kyseinen pullo on lähes tyhjä. Tällöin jää operaattorin vastuulle huomata, että juoma pullossa ei riitä. Näin voidaan taas joutua tilanteeseen, jossa asiakas tilaa juomaa, mutta saakin vain tyhjän lasin.

Robotin logiikkassa on ominaisuus, joka tarkistaa, onko pullossa liian vähän juomaa jäljellä ja se kutsuu suoraan pullon poistoa solusta. Jotta ei päädyttäisi tilanteeseen, missä lä-hes tyhjästä pullossa yritetään tarjoilla juomaa, on jouduttu määrittelemään palautettavan pullon jäljellä olevaksi juomamäärän rajaksi melko korkea määrä. Tällöin robotti poistaa solusta pulloja, jossa juomaa olisi vielä tarpeeksi, koska ei haluta että joudutaan tilantee-seen, joka haittaa asiakasta. Uudella kaatoratkaisulla tavoitellaan tilannetta, jossa back endin tieto pulloissa jäljellä olevista juomamääristä on koko ajan melko tarkka. Tällöin robotti voisi myös toteuttaa kaadot siten, että se kaataa vanhan pullon tyhjäksi, hakee uuden pullon ja jatkaa kaatoa siitä.

2.4 Mahdollisten ratkaisujen kartioitus

Tässä alaluvussa kartoitetaan mahdollisia ratkaisuja alaluvussa 2.3 esitetylle ongelmille ja tehdään lyhyt kirjallisuuskatsaus aikaisempiin aiheeseen liittyviin tutkimuksiin, sekä esi-tellään ideoita ongelman ratkaisemiseksi. Tämän lisäksi valitaan työssä käytetty ratkaisu ja perustellaan sen valinta.

Kuten alaluvussa 2.3 todettiin, myös Suomen alkoholilaki määrittää alkoholin perusannok-set. Lakisääteisyyys alkoholin tarjoilussa on saanut aikaan erilaisia ratkaisuja mitata tarjoil-tavaa juomaa ihmistenkin tarjoilemana. Jopa englanninkielinen Wikipedia-artikkeli mää-rittelee alkoholijuomien mittamistavat omaan luokkiinsa [11]. Mitta-astioiden ja tavallis-ten, kuvassa 2.4 esitetyjen kaatonokkien lisäksi on olemassa juoman virtausta mittavia

kaatonokkia. Kuten johdantoluvussa todettiin, on korkkeja, joissa on pieni tila johon juomaan virtaa tietty määrä, ja sitten korkin venttiilin avaamalla tuon määrän voi laskea tilasta ulos. Tämän jälkeen tila täytyy uudelleen pullossa olevalla juomalla. Tämä ratkaisu vaatii kuitenkin pullon olevan ylösalaisten jonkinlaisessa telineessä tai esimerkiksi juuri katossa roikkumassa. Tämän lisäksi on olemassa kaatonokkia, jotka sulkeutuvat automaattisesti kun juomaa on virrannut tarpeeksi. Niissä voi esimerkiksi olla sisällä pieni metallinen kuula, joka nosteen ansiosta nousee tietyllä nopeudella ylöspäin ja lopulta tukki juoman virtauksen [12]. Tällaisia kaatonokkia on ollut testattavana Pullonaula ry:llä, mutta niitä ei ole koskaan saatu käytettyä luotettavasti, ja näin niiden käytöstä on luovuttu.

Nesteiden kaatamista astiasta toiseen on tutkittu paljon metallin valannan yhteydessä ja useamman tutkimuksen ovat tehneet Noda et al. [13][14][15]. Metallin valannassa tarvitssee kaataa nestemäisessä olomuodossa olevaa metallia sisältävästä astiasta materiaalia muottiin tarkasti astiaa kallistamalla. Kaadolta vaaditaan nopeutta ja tarkkuutta sekä virtaavan metallin määrän että muottiin osumisen suhteen. Robotin käyttö on tällaisessa tapauksessa hyödyllistä, koska sula metalli luo vaarallisen työmpäristön. Robotille voisi opettaa jokaisen kaatotapauksen erikseen, mutta se on aikaa vievää ja hankalaa, sekä altista muuttujille. Verrattuna juomien tarjoiluun, metalli saattaa esimerkiksi paakkuuntua kaatoastian reunaan, jolloin se vaikuttaa virtaukseen. Sen takia on kehitetty edistyneempiä tapoja hallita kaatoa robotilla. Esimerkiksi [13]:ssa Noda et al. kehittivät epälineaarisen mallin kaadolle ja käyttivät sen ohjaamiseen etukompensaatiota. Myöhemmin [15]:ssa he käyttivät laajennettua Kalman-suodatinta kaatoprosessin tilojen, eli nesteen korkeuden kaatoastian suulla ja nesteen massan kohdeastiassa, estimointiin. Kaatoastian suoristuksen aikana virtaavan nesteen määrää mallinnettiin ja siitä saatettiin laskettua tarkempi ajotus suoristuksen aloittamiselle. Menetelmää testattiin vedellä ja sillä päästiin keskimäärin 1,7 %:n tarkkuuteen, ja virhe oli kaikissa tapauksissa korkeintaan 3 %.

Yksi tapa saada juomien kaadosta tarkka on mitata kaadettua määrää jollain tavalla. Tätä määrän mittausta voidaan käyttää ikään kuin takaisinkytkenänä kaodon lopettamiselle tai säättämiselle. Kaadetun määrän mittaamiseen voidaan käyttää visuaalista dataa kameroiden tai 3D-skannerien avulla. Myös voima-antureilla joko robotin akseleilta tai kohdeastian alta mitattua dataa voidaan käyttää. Jopa audion [16], tai audio- ja voima-anturidataan yhdistelmää [17] kaadetun määrän arvioimiseen ovat tutkineet Liang et al.

Kameroilta saatavan datan käyttöä juomien kaadossa ovat tutkineet mm. Schenck ja Fox, jotka käyttivät konvoluutioneuroverkkoja arvioimaan nesteen määrää kohdeastiasta saadusta kuvamateriaalista. He käyttivät tästä mittausta yksinkertaisen PID-säätimen takaisinkytkenänä, ja säädin ohjasi robotin rannenivelen eli uloimman akselin kulmaa ja näin pystyi säätämään juoman kaatoa. Tulokset olivat keskimäärin hyviä, mutta ne vaihtelivat melko paljon, jopa sadalla millilitralla. [18]

Tässä työssä käytetty Yaskawan HC-10 on monitoimirobotti, mikä tarkoittaa sitä, että sen

akseleissa on voima-anturit. Näitä voima-antureita käytetään esimerkiksi törmäysten havaitsemiseksi, jolloin robotti voi törmättyään pysähtyä, tai siihen että operaattori voi käsin työntämällä ohjata robottia. Yaskawan HC-10:n käsiohjaimen käyttöliittymässä on sivu, jolta näkee kaikkiin akseleihin sekä robotin päähän kohdistuvan voiman eri koordinaatti-suunnissa reaalialkaisesti. Nämä voimat on laskettu kaikkien kuuden voima-anturin datan, sekä akselien asennon pohjalta. [19, s.10-11] Aluksi hypoteesina oli, että näiden voima-anturien dataa voisi käyttää robotin tarttujassa olevan massan mittaamiseen. Tämän tiedon massasta voisi integroida osaksi robotin tehtäväohjelmaa, jolloin robotti pysyisi itse saamaan tiedon siitä, paljonko juomaa tarttujassa olevasta pullossa on jo kaadettu. Tällöin esimerkiksi tilanteessa, jossa kaatonokka tukkeutuu, tilanne voitaisiin havaita ja silloin yrittää uutta kaatoa. Tämän ratkaisun esteeksi tuli itse voima-anturien tarkkuus HC-10:ssä. Esimerkiksi pystysuoraan koordinaattisuuntaan vaikuttava voima heittelee jatkuvasti usean Newtonin verran käyttöliittymästä nähtävän datan perusteella. Lisäksi epätarkkuutta lisää se, jos robottikäsi tai pullossa oleva juoma on vähänkään muuttuvassa liikkeessä johonkin suuntaan.

Robotin nivelen voima-anturia kaodon apuna ovat käytäneet esimerkiksi Rozo et al. Tässä tapauksessa ei kuitenkaan käytetty robotin akselien sisäisiä voima-antureita, vaan yksittäistä ulkoista voima-anturia asetettuna robotin ranneniveleen. Heidän lähestymistapansa hyödynsi robotin opettamista demonstraation avulla. Ensin ihmisen ohjaa haptisella etäohjaimella robottia kaatamaan juomia, ja näistä kaadoista kerätään dataa voima-anturilta. Tältä anturilta saadaan myös arvio tarttujassa olevan pullon sisältämästä juomamäärästä. Tätä dataa käytetään parametrisen Markovin piilomallin avulla luomaan yhteys voima-anturien lukemien ja kaadetun juomamäärän välillä. Sen jälkeen tämän avulla robotti jäljentää aikaisempia ihmisen opettamia kaatoja ja pyrkii kaatamaan tietyn määrän juomaa. Tapaa testatessa ei kuitenkaan kaadettu nesteitä, vaan pieniä metallikuulia, joilla simuloitiin nestettä. Tulosten esittelyssä keskityttiin enemmän siihen, että robotti saatiin seuraamaan ihmisen opettamia kaatoja, eikä tarkennettu, olivatko kaadot määritiltään tarkkoja vai eivät. [20]

Yksi tapa mitata kaadettua massaa on yksinkertaisesti asettaa jonkinlainen painoanturi tai vaaka kohteena olevan mukin alle ja käyttää sen antamaa lukemaa apuna. Tässä työssä on valittu tämä tapa ratkaisuki kaatomäärän tarkkuuden parantamiseen. Tämän ratkaisun etuna on sen yksinkertaisuus, koska se ei aseta suuria vaatimuksia käytettävälle robotille tai robottisolulle. Vaa'an käyttöön voisi yhdistää kaatokulman säädön, kuten Schenck ja Fox tekivät [18]. Samoin pullon suoristuksen aikana virtaavaa nestettä voisi eri tilanteissa mallintaa tarkemmin, kuten Noda et al. tekivät [14]. Tässä työssä on kuitenkin käytetty yksinkertaisempaa, tapahtumapohjaisuuteen perustuvaa ratkaisua ja suoristuksen aikana virtaavaa nestettä approksimoitiin vakiolla. Massan mittaamista painoanturilla helpottaa myös se, että Drinkkirobotin robottisolussa on jo käytössä vaaka pullonvaihtopisteellä, ja se kommunikoi robotin logiikan kanssa. Tätä vaakaa voidaan käyttää

testaamaan tavan toimivuutta ja sitten jälkeenpäin hankkia kaikkien kaatopisteiden alle erilliset vaa'at ja käyttää niitä samalla tekniikalla.

3 RATKAISU

3.1 Uusi kaatoratkaisu

Tässä alaluvussa käsitellään uutta kaatosysteemiä, jonka avulla pyritään ratkaisemaan alaluvussa 2.3 käsitellyjä ongelmia. Kaatosysteemin käsitteily sisältää sekä logiikan että robottikoodin ja siinä kuvataan myös näiden kahden kommunikaatiorajapintaa.

Uudessa kaatosysteemissä käytetään ikään kuin takaisinkytkentänä vaakaa, jolloin saadaan tietoon mukissa oleva massa kaodon aikana. Tässä vaakana on käytetty alaluvussa 2.1 esiteltyä pullonvaihtopisteellä käytettyä vaakaa. Vaaka koostuu HX711-mittakortista ja siihen kytketyistä 5 kg:n painoanturista, ja näitä ohjaaa arduino-kirjastoja apuna käyttäen ohjelmoitu NodeMCU V3. Kommunikaatio logiikkaan toimii wlan-yhteydellä lähetettävillä UDP-paketeilla. Vaakaa pystyy siis siirtämään langattomuuden ansiosta, ja niin se soveltuu hyvin myös juomamäärien mittaamiseen. Mikä tahansa samalla toimintaperiaatteella toimiva painoanturi kävisi tähän tarkoitukseen. Vaa'an NodeMCU:ltä voidaan pyytää painoanturin osoittamaa massaa tietyllä merkkijonolla. Vaaka voidaan myös taarata toisella siihen tarkoitettulla merkkijonolla. Robotin logiikkaan on tehty oma ScaleCommunication-taso, joka ottaa yhteyden vaakaan. Sen metodeilla getWeight() ja Tare() pystyy suoraan käyttämään vaakaa logiikasta.

Kun käytöliittymästä tilataan juomaa, logiikka käsittelee tilauksen, käskyttää robottia hakemaan oikean pullon ja kaatamaan oikean määärän juomia. Logiikkakerros kutsuu funktiota PourBottle, joka käsittelee kaatoa yksittäisestä pullossa. Nyt uutena logiikkaan on toteutettu funktio HandlePour, joka on kuvattu alla ohjelmassa 3.1. Tämä funktio pyytää vaa'alta massalukemia ja käskee robottia lopettamaan kaodon, kun lukema on tarpeeksi suuri.

```

1  private async Task HandlePour(int pourAmount)
2  {
3      const string funcName = nameof(HandlePour);
4      await _robot.StartPour(pourTimes, startingLocation);
5      _PourScale.Tare();
6      System.Threading.Thread.Sleep(5000);
7
8      int pourWeight = (pourAmount * 10) - 27;
9      int currentWeight = _PourScale.GetWeight();
10
11     double maxTime = 13;
12     DateTime currentTime = DateTime.Now;
13     double secondsPassed = 0;
14
15     while (currentWeight < pourWeight && secondsPassed <= maxTime)
16     {
17         System.Threading.Thread.Sleep(10);
18         if (currentWeight < 10 && secondsPassed > 2
19             && secondsPassed < 2.1)
20         {
21             Log.InfoEx(funcName, $"2 seconds passed with no weight:
22                 Scale reports {currentWeight}g.
23                 Attempting to pour again.");
24             await _robot.ShakeBottle();
25             maxTime += 6;
26             System.Threading.Thread.Sleep(3000);
27         }
28         currentWeight = _PourScale.GetWeight();
29         secondsPassed = DateTime.Now.Subtract(currentTime)
30             .TotalSeconds;
31     }
32     if (currentWeight >= pourWeight)
33     {
34         Log.InfoEx(funcName,
35             $"Pour amount reached with weight of {currentWeight}");
36     }
37     else
38     {
39         Log.InfoEx(funcName,
40             $"Pour amount reached by timeout of {maxTime} seconds");
41     }
42
43     await _robot.StopPour();
44 }
```

Ohjelma 3.1. HandlePour-funktio

HandlePour kutsuu ensin RobotCellLayerin funktiota StartPour. RobotCellLayer vastaa

robotin kanssa kommunikoinnista. StartPour kutsuu robotin tehtäväohjelmaa, joka on tällä hetkellä nimellä NEWPOUR.

```

    NOP
2   ' PROGRAM RUNNING
    SET I000 1
4   CALL JOB:GOCENTRALPOS
    *LABEL10
6   IFTHENEXP I005=I004
        JUMP *LABEL999
8   ENDIF
    CALL JOB:CALCULATEPOUR
10  ' MOTION TO POURTARGET
    MOVL P021 V=150.0 DEC=20
12  MOVL P023 V=50.0 DEC=20
    ' POUR
14  IMOV P008 V=40.0 TF ACC=20 DEC=20
    WHILEEXP I014<>1
16      TIMER T=0.01
        IFTHENEXP I014=2
            CALL JOB:SHAKEBOTTLE
        ENDIF
20  ENDWHILE
    ' LEVEL BOTTLE AND MOVE AWAY
22  MOVL P023 V=40.0 ACC=20 DEC=20
    IMOV P009 V=40.0 TF ACC=20 DEC=20
24  ' ADD COMPLETED POUR TO COUNTER
    ADD I005 1
26  ' JUMP BACK TO POUR NUMBER CHECK
    JUMP *LABEL10
28  *LABEL999
    ' MOTION TO CENTRAL POSITION
30  CALL JOB:GOCENTRALPOS
    ' PROGRAM COMPLETED
32  SET I000 0
END

```

Ohjelma 3.2. NEWPOUR-tehtäväohjelma

Ohjelman 3.2 NEWPOUR-tehtäväohjelma on robotin liikkeitä ohjaavilta komennoltaan hyvin samantapainen kuin alaluvussa 2.2 esitellyssä ohjelmassa 2.1 esitetty POURDRINKS. Aluksi se määrittää kaodon kohteena olevan mukin sijainnin CALCULATEPOUR-tehtäväohjelmalla. Sitten robottia käsketään siirtymään kaatopisteelle ja robotti kallistaa pulloa aloittaen kaodon. Uutta on kuitenkin se, että robotti jäää while-silmukan ansiosta odottamaan, että muuttuja I014 on 0. I014 on nimetty robotissa nimellä "pour ready signal".

HandlePour-funktio laskee halutun massan kertomalla halutun nesteen tilavuuden, joka on senttilitroina, kymmenellä, jolloin saadaan haluttu massa grammoina. Tästä vähenne-

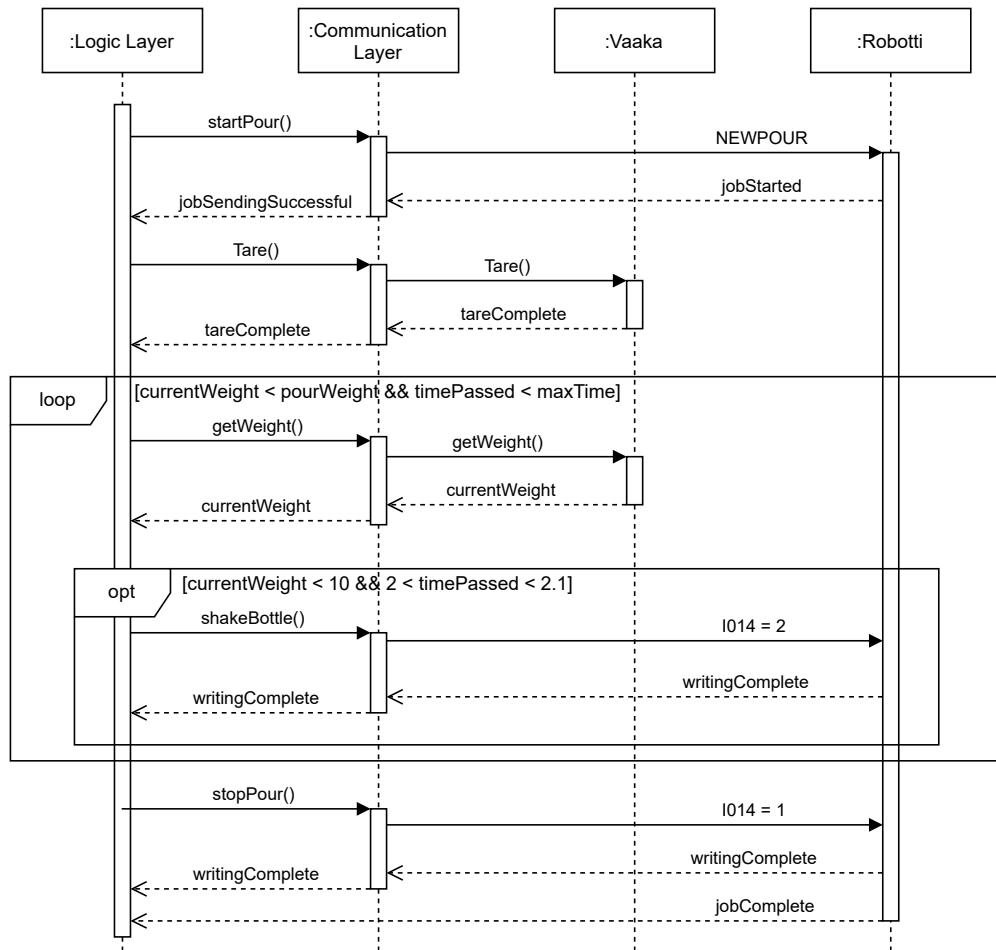
tään vielä vakio 27. Tämä vakio on määritetty testaamalla ja se kuvaa sitä viivettä, joka kestää siitä kun vaaka tunnistaa massan olevan sopiva, siihen kun nestettä ei enää tule ollenkaan mukiin.

Funktiossa on while-silmukka, joka odottaa joka iteraatiolla kymmenen millisekuntia, kysyy uutta vaa'an havaitsemaa massaa, ja sen jälkeen tarkistaa onko havaittu massa yhä pienempi kuin haluttu massa. Samalla silmukka tarkistaa, ettei kaato ole kestänyt yli maxTime-vakiossa määriteltyä aikaa. Kun while-silmukasta on tultu ulos, niin tulostetaan konsoliin tieto saavutetusta massasta tai mahdollisesti ajasta ja kutsutaan RobotCellLayerrin funktiota StopPour.

HandlePour-funktion while-silmukan sisällä oleva if-lause tarkastaa, onko kahden sekunnin kuluttua kaodon aloittamisesta mukiin tullut jo kymmenen grammaa juomaa. Jos näin ei ole, niin voidaan päättellä kaatonakan tukkeutuneen. Tällaisessa tilanteessa kaatoliike tehdään uudelleen. Kaatoliikkeen tekeminen uudelleen on toteutettu muuttamalla muuttuja I014 arvoon 2. Myös robotin NEWPOUR-tehtäväohjelmassa on if-lause, joka tunnistaa I014:n muuttumisen ja kutsuu tehtäväohjelmaa SHAKEBOTTLE. SHAKEBOTTLE sisältää robotin liikkeet pullon suoristamiseksi ja uuden kaodon aloittamiseksi.

Funktiossa on myös määritelty vakio maxTime, joka on 13 sekuntia. Tämä vakio on määritetty sen perusteella, että alaluvussa 2.2 esitellyn kaatoajan funktion mukaan saadaan 13 sekunnin kaadolla n. 25 senttilitraa juomaa. Drinkkirobotin tarjoilussa tavallisest käytämät lasit ovat hieman yli 25 senttilitraa tilavuudeltaan. Tätä maxTime-vakiota käytetään siihen, ettei mahdollisessa vaa'an virhetilanteessa juomaa kaadeta siten, että se läikkyisi lasista yli.

StopPour-funktion tehtävänä on ainoastaan muuttaa robotin muuttuja I014, eli "pour ready signal" arvoon yksi. Tällöin robotin tehtäväohjelma tulee omasta while-silmukastaan ulos, jolloin robotti suoristaa pullon ja näin lopettaa kaodon.



Kuva 3.1. Sekvenssikaavio uudesta kaatosysteemistä

Yllä olevassa sekvenssikaaviossa on havainnollistettu UML-kielessä uuden kaatosysteemin toimintaa.

3.2 Uuden kaatoratkaisun analysointi

Logiikka laskee halutun juoman massan kaavalla

$$\text{pourWeight} = \text{pourAmount} \cdot 10 - 27, \quad (3.1)$$

eli halutusta massasta vähennetään vaa'an hitautta kuvaava vakio 27. Hitaus johtuu muun muassa tietoliikenneyhteyden hitaudesta ja silmukoiden sisällä olevista odotusajoista. Esimerkiksi robotin tehtäväohjelmassa olevan timer-komennon tai logiikassa olevan sleep-komennon arvoa nostamalla tuo vakio 27 kasvaisi. Tämä hitaus tarkoittaa sitä, että pullon suoristusliikkeen aikana tapahtuvien muuttujien ei pystytä enää vaa'an avulla vaikuttamaan. Käytännössä siis 27 grammaa juomaa kaatuu vielä sen jälkeen, kun logiikka havaitsee juomaa olevan tarpeeksi. Tämä on kuitenkin vain pieni osa koko kaatoliikettä. Esimerkiksi kaatonokan vaihtaminen sellaiseen kaatonokkaan, jossa virtausnopeus on eri, aiheuttaa sen, että tuo vakio muuttuu hieman.

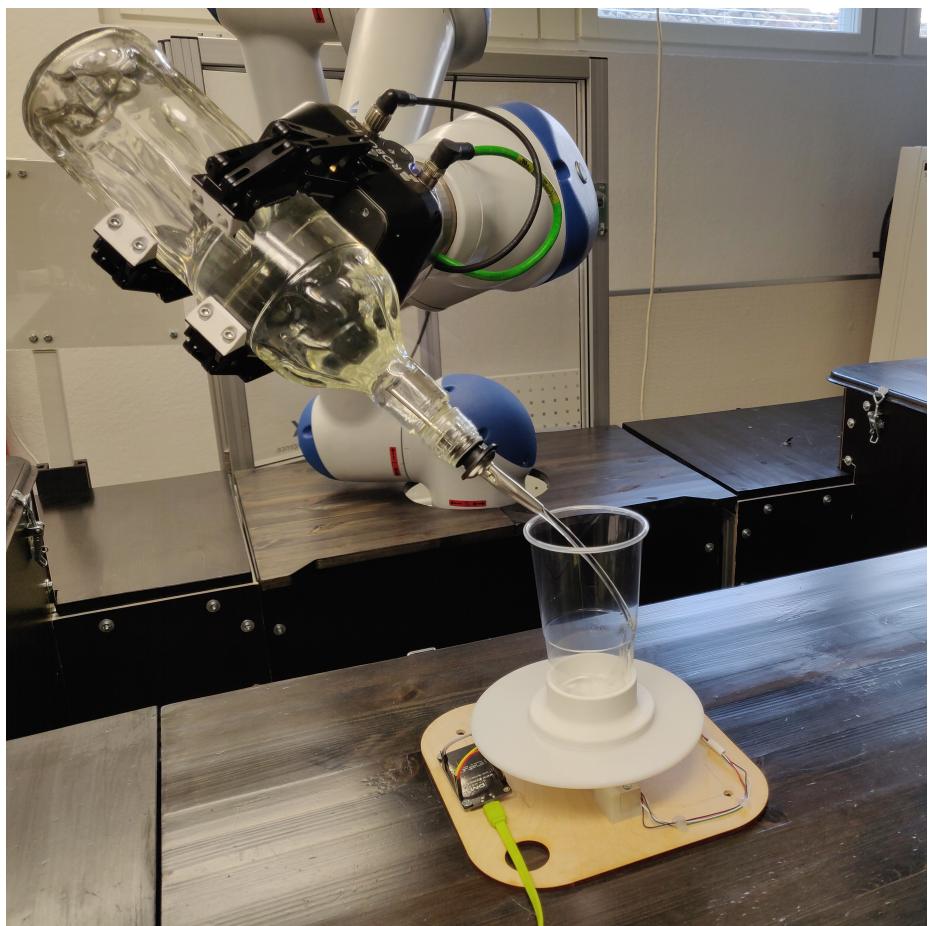
Painoanturin tarkkuutta testattiin vertailemalla vaa'an antamia tuloksia tavallisen keittiövaa'an näyttämän painon kanssa. Kaikissa tapauksissa painoanturin ja keittiövaa'an tulokset erosivat maksimissaan yhdellä grammalla toisistaan. Vaaka osoittaa hieman liian suuria lukemia johtuen siitä, että juomavirta osuu lasin pohjaan ja osumisen impulssi aiheuttaa voiman. Kuten Fox ja McDonald [21, s.197-198] johtivat, Newtonin II laista saadaan voima, joka aiheutuu nestevirtauksen osumisesta pintaan. Tämän voiman suuruus yhteen koordinaattisuuntaan on

$$F_z = \rho Q \delta v_z, \quad (3.2)$$

jossa ρ on nesteen tiheys, Q on virtausnopeus ja Δv_z on nesteen nopeuden muutos tuohon koordinaattisuuntaan. Koska juomaa kaadetaan niin matalalta, sen nopeus ei ehdi missään kohtaan olemaan kovin suuri. Lisäksi virtausnopeus on varsinkin kaatonokalla kaadettaessa melko pieni, luokkaa 20 millilitraa sekunnissa. Tämän takia nesteen virtauksesta vaakaan aiheutuva voima on hyvin pieni ja se vastaa vaa'an lukemassa grammojen epätarkkuutta. Tätä kuitenkin kompensoi yllä kuvattu vakio 27.

HandlePour-funktio taaraa vaa'an ennen jokaista kaatoa. Tämän ansiosta useamman komponentin juomasekoituksia on mahdollista kaataa tällä systeemillä. Funktiossa oleva currentWeight on siis mukiin tullut paino sen jälkeen, kun kyseisen komponentin kaato on aloitettu. Taaraus mahdollistaa myös mukitelineen pitämisen vaa'an päällä ilman etää sen massaa erikseen vähennetään saadusta painosta. Mukiteline auttaa siihen, ettei muki liikahda juomavirran vaikutuksesta. Samoin voitaisiin myös käyttää erilaisia, eri painoisia mukeja ilman että se vaikuttaa kaatoihin. Kuvassa 3.2 näkyy robotti kaatamassa vaa'an päällä olevaan mukiin, joka on mukitelineessä.

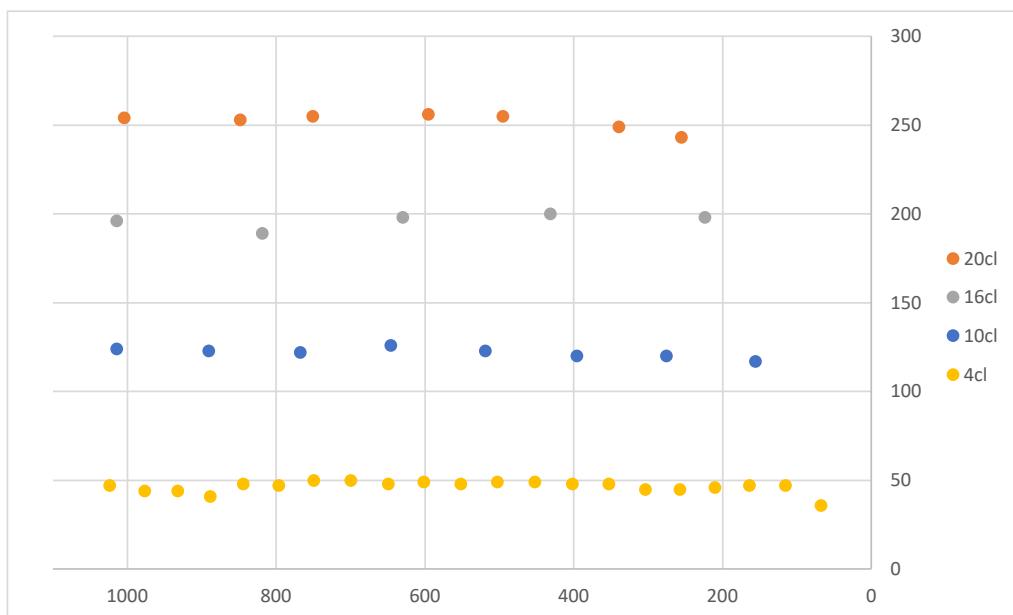
Uusi kaatosysteemi ei tällaisenaan mahdollista alle kolmen senttilitran kaatoja robotilla. Tämä johtuu siitä, että pelkkä kallistus- ja suoristusliikkeen aikana pullostava virtaava juoma on tilavuudeltaan noin kolme senttilitraa, vaikka robotti ei pysyisi kaatoasennossa paikallaan. Pienempiä kaatoja olisi luultavasti mahdollista saada muokkaamalla kaatoasennon kulmaa pienemmäksi. Alaluvussa 2.3 luetellut lakisääteiset alkoholijuomien perusannokset eivät kuitenkaan vaadi alle neljän senttilitran kaatoja. Tämän takia pienien kaatojen mahdolisuutta ei lopulta ole tässäkään työssä toteutettu.



Kuva 3.2. Robotti kaatoasennossa kaatamassa vaa'an pääällä olevaan mukiin

4 TULOKSET JA VERTAILU

Työssä tutkittiin sekä vanhan että uuden kaatoratkaisun pohjalta saatuja tuloksia kaatojen määrisä, jotta niitä voitaisiin vertailla. Kaatoja tehtiin 4 cl, 10 cl, 16 cl ja 20 cl kokoisilla tilauksilla. Samalla jokaisen kaodon välissä punnittiin pullossa jäljellä oleva neste, jotta voidaan huomioida sen vaikutus kaodon määrään. Ideaalisesti pullossa olevan nesteen määrä ei vaikuttaisi kaodon määrään.

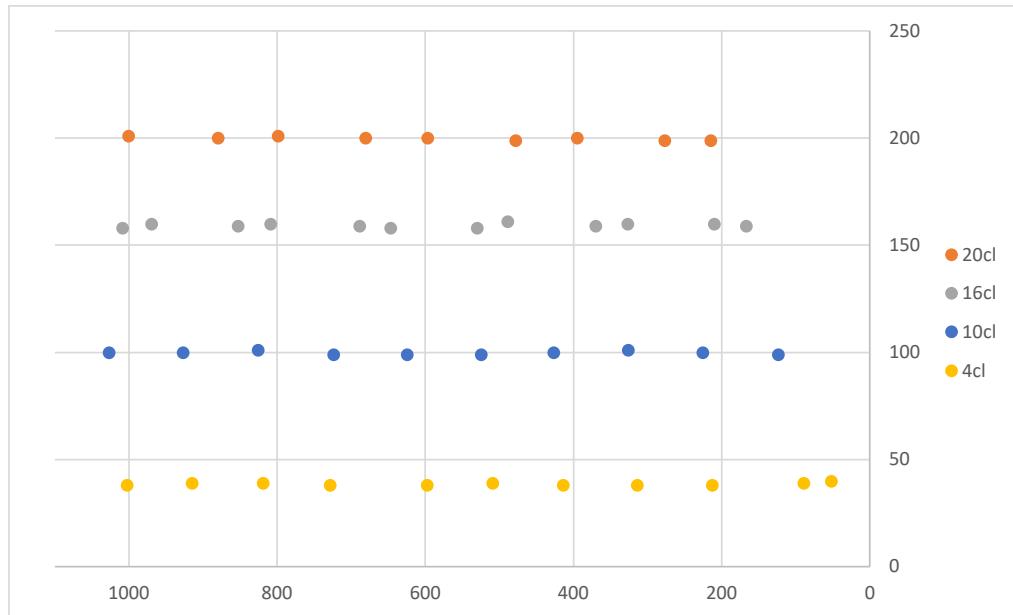


Kuva 4.1. Vanhalla kaatoratkaisulla saadut tulokset

Kuten yllä olevasta vanhan kaatoratkaisun tulosten kuvaajasta huomataan, varianssia oli kohtalaisen paljon. Saadut juomamäärit laseissa eivät olleet tasaisia, vaan joka kaadolla ne muuttuvat hieman. Esimerkiksi 4 cl tilauksessa suurin saatu juomamääriä oli noin 50 grammaa ja pienin n. 36 grammaa. 20 cl tilauksessa suurin saatu juomamääriä oli noin 255 grammaa ja pienin n. 214 grammaa. Näiden välinen ero on huomattava, varsinkin kun tarkastellaan eron prosentuaalista osuutta halutusta juomamääristä.

Kuvaajassa olevista tuloksista saadaan myös vahvistus väitteelle, että pullossa jäljellä oleva juomamääriä vaikuttaa saatuun juoman määrään. Usealla kaatomäärellä mukiin saatu määriä vaihtelee ensin satunnaisesti tietyllä välillä, mutta käyrän loppupäätä kääntyy alaspäin. Tästä voidaan päätellä, että pullostaa, jossa on vain vähän yli tarvittava määriä juomaa, virtaa sitä hitaan kaodon lopussa.

Sen lisäksi, että saadut juomamääät vaihtelevat keskenään paljon ja pullossa oleva juomamääärä vaikuttaa saadun juoman määrään, kaatomääät ovat myös keskimäärin liian suuria. Koska jokaisella kaatomäärellä juomaa on tullut liikaa suhteessa tilattuun määrään, voidaan päätellä että vanhan funktion kulmakerroin on ollut liian suuri.



Kuva 4.2. Uudella kaatoratkaisulla saadut tulokset

Kuten nähdään uudella kaatoratkaisulla saaduista tuloksista kuvassa 4.2, uusi kaatoratkaisu on huomattavasti tarkempi. Siinä kaatomääät vaihtelevat korkeintaan yhdellä grammalla eikä niihin vaikuta pullossa ennen kaatoa oleva juoman määärä.

5 YHTEENVETO

LÄHTEET

- [1] C. Heer ja S. Bieller. *IFR presents World Robotics Report 2020 - International Federation of Robotics*. 2020. URL: <https://ifr.org/ifr-press-releases/news/record-2.7-million-robots-work-in-factories-around-the-globe> (viitattu 23.01.2021).
- [2] *Palvelurobotit yleistyvät myös Suomessa — Jyväskylän yliopisto*. 2018. URL: <https://www.jyu.fi/fi/ajankohtaista/arkisto/2018/07/palvelurobotit-yleistyvat-myos-suomessa> (viitattu 23.01.2021).
- [3] T. Kelly ja A. Tomoshige. *Japanese robot could call last orders on human bartenders / Reuters*. 2020. URL: <https://www.reuters.com/article/us-japan-robot-bar/japanese-robot-could-call-last-orders-on-human-bartenders-idUSKBN1ZY17K> (viitattu 23.01.2021).
- [4] *Robot Bartenders Shake Things Up At Sea / Royal Caribbean Blog*. 2016. URL: <https://www.royalcaribbean.com/blog/robot-bartenders-shake-things-up-at-sea/> (viitattu 23.01.2021).
- [5] "You Teach Me"robot Option. Tekninen raportti. 2017. URL: www.yaskawa.eu.com.
- [6] *Drinkkirobotti 5.0 / Pullonaula ry*. 2020. URL: <https://www.pullonaula.fi/projektit/drinkkirobotti-5-0/> (viitattu 25.01.2021).
- [7] L. Kemppi. Tuotannossa tapahtuvien virheiden käsittelykeinot robottisolussa - Kirjallisuuskatsaus ja käytännön koe eri virheenkäsitteiltä voin. 2021.
- [8] F. Geiger, K. Velten ja F. J. Methner. 3D CFD simulation of bottle emptying processes. *Journal of Food Engineering* 109.3 (huhtikuu 2012), 609–618. ISSN: 02608774. DOI: 10.1016/j.jfoodeng.2011.10.008.
- [9] *Viscosity and Laminar Flow; Poiseuille's Law / Physics*. 2021. URL: <https://courses.lumenlearning.com/physics/chapter/12-4-viscosity-and-laminar-flow-poiseuilles-law/> (viitattu 24.03.2021).
- [10] *Alkoholilaki, L 28.12.2017/1102, 2017*. URL: <https://www.finlex.fi/fi/laki/alkup/2017/20171102>.
- [11] *Alcoholic spirits measure - Wikipedia*. 2020. URL: https://en.wikipedia.org/wiki/Alcoholic_spirits_measure (viitattu 22.04.2021).
- [12] *Measured Pourers — Bar Products*. URL: <https://barproducts.com/collections/measured-pourers> (viitattu 22.04.2021).
- [13] Y. Noda ja K. Terashima. *Modeling and Feedforward Flow Rate Control of Automatic Pouring System with Real Ladle*. 2007. URL: <https://www.fujipress.jp/main/>

- wp – content / themes / Fujipress / pdf %7B%5C_%7Dsubscribed . php (viitattu 24.04.2021).
- [14] Y. Noda, K. Yamamoto ja K. Terashima. Pouring control with prediction of filling weight in tilting ladle type automatic pouring system. *International Journal of Cast Metals Research*. Vol. 21. 1-4. Elokuu 2008, 287–292. DOI: 10.1179/136404608X362115. URL: <https://web-a-ebscohost-com.libproxy.tuni.fi/ehost/pdfviewer/pdfviewer?vid=1%7B%5C&%7Dsid=af16085e-d4f4-4f6f-910c-f67be81194df%7B%5C%7D40sdc-v-sessmgr03>.
 - [15] Y. Noda, K. Terashima, M. Suzuki ja H. Makino. Weight Control of Pouring Liquid by Automatic Pouring Robot. *IFAC Proceedings Volumes* 42.23 (tammikuu 2009), 185–190. ISSN: 14746670. DOI: 10.3182/20091014-3-cl-4011.00034.
 - [16] H. Liang, S. Li, X. Ma, N. Hendrich, T. Gerkmann, F. Sun ja J. Zhang. Making Sense of Audio Vibration for Liquid Height Estimation in Robotic Pouring. *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical ja Electronics Engineers Inc., marraskuu 2019, 5333–5339. ISBN: 9781728140049. DOI: 10.1109/IROS40897.2019.8968303. arXiv: 1903.00650. URL: <https://ieeexplore-ieee-org.libproxy.tuni.fi/document/8968303>.
 - [17] H. Liang, C. Zhou, S. Li, X. Ma, N. Hendrich, T. Gerkmann, F. Sun, M. Stoffel ja J. Zhang. *Robust Robotic Pouring using Audition and Haptics*. Tekninen raportti. 2020. arXiv: 2003.00342v2. URL: <https://ieeexplore-ieee-org.libproxy.tuni.fi/document/9340859>.
 - [18] C. Schenck ja D. Fox. *Visual Closed-Loop Control for Pouring Liquids*. Tekninen raportti. 2016. arXiv: 1610.02610v3. URL: <https://arxiv.org/abs/1610.02610>.
 - [19] *Collaborative Applications Masters of Robotics, Motion and Control*. Tekninen raportti. 2018. URL: https://yaskawa.co.il/wp-content/uploads/2020/08/Collaborative-Applications%7B%5C_%7DE%7B%5C_%7D10.2018.pdf.
 - [20] L. Rozo, P. Jiménez ja C. Torras. Force-based robot learning of pouring skills using parametric hidden Markov models. *9th International Workshop on Robot Motion and Control, RoMoCo 2013 - Workshop Proceedings*. IEEE Computer Society, 2013, 227–232. ISBN: 9781467355117. DOI: 10.1109/RoMoCo.2013.6614613. URL: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=%7B%5C&%7Darnumber=6614613>.
 - [21] R. W. Fox ja A. T. McDonald. *Introduction to fluid mechanics*. 2011. URL: http://ftp.demec.ufpr.br/disciplinas/TM240/Marchi/Bibliografia/Pritchard-Fox-McDonalds%7B%5C_%7D2011%7B%5C_%7D8ed%7B%5C_%7DFluid-Mechanics.pdf (viitattu 24.04.2021).