

Juho Kotkaranta

**MOBIILIROBOTIN PAIKANNUS JA YMPÄRISTÖNMALLINNUS
ETÄOHJAUKSEN AVULLA**

MOBIILIROBOTIN PAIKANNUS JA YMPÄRISTÖNMALLINNUS ETÄOHJAUKSEN AVULLA

Juho Kotkaranta
Opinnäytetyö
Syksy 2018
Sähkö- ja automaatiotekniikka
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma, automaatiotekniikka

Tekijä: Juho Kotkaranta

Opinnäytetyön nimi: Mobiilirobotin paikannus ja ympäristön mallinnus etäohjauksen avulla

Työn ohjaajat: Tapio Heikkilä (VTT), Tero Hietanen (Oamk)

Työn valmistumislukukausi ja -vuosi: Syksy 2018 Sivumäärä: 38

Tämän opinnäytetyön tavoitteena oli kehittää VTT Oy:n mobiilialustaiselle robotille etäohjaus sekä SLAM-navigointimenetelmää. SLAM-navigoinnilla robotti kykenee hahmottamaan ympäristöstään kohteita, piirtämään ne karttakoordinaatistoon sekä samanaikaisesti paikoittamaan oman sijaintinsa karttaan. Työ tehtiin kokonaisuudessaan VTT:n tiloissa, ja se on osa pidempiaikaista mobiilirobottiin tehtävää kehitystyötä.

Mobiilirobottia on ohjattu robotin päällä olevista nuolinäppäimistä tai automaatiohjauksella konenäön avulla. Robotille kehitettiin etäohjaus, joka toimii WLAN-yhteyttä hyväksikäyttäen ja samalla myös robotin muu tietoliikenne siirrettiin langattomaan verkkoon. Robotinohjausohjelmat kirjoitettiin C++-kielellä käyttäen Qt Creator -ohjelmointiympäristöä.

Lopputuloksena saatiin mobiilirobottiin toimiva etäohjaus, jonka avulla käyttäjän ei tarvitse olla robotin välittömässä läheisyydessä ohjaamassa robotin liikkeitä. Robotti kykenee hahmottelemaan ympäristöstään kohteita, joita se piirtää karttakoordinaatistoon. Mobiilirobotti kykenee seuraamaan jo hahmottelemansa reitin uudelleen.

Asiasanat: navigointi, robotiikka, konenäkö, SLAM

ABSTRACT

Oulu University of Applied Sciences
Degree programme of Electrical and Automation Engineering, Automation

Author: Juho Kotkaranta

Title of thesis: Localization of a Mobile Robot and Mapping of the Environment using Remote Control

Supervisors: Tapio Heikkilä (VTT), Tero Hietanen (Oamk)

Term and year when the thesis was submitted: Fall 2018

Number of pages: 38

The goal of this thesis was to develop remote control and the SLAM navigation method for VTT mobile robot platform. With the SLAM navigation the robot would be able to perceive objects around it draw them into a map and simultaneously locate its own position in the map. The entirety of the work was done in the premises of VTT and it is a part of a long-term development of the mobile robot. This mobile robot platform was developed by Probot Oy.

The mobile robot is controlled with arrow keys by the keyboard of the main computer or by automatic driving mode using machine vision. Remote control was developed for the mobile robot, which utilizes a WLAN connection. All other data communications were also transferred to the wireless network. Robot control programs were written in C++ language using the Qt Creator development environment.

The final product of this thesis is a remote-control function for the mobile robot, that doesn't need to be in immediate vicinity of the robot to guide robot movements. The robot is able to notice walls that it draws into a map coordinate system. The mobile robot will be able to follow the route it has already outlined.

Keywords: navigation, robotics, machine vision, SLAM

ALKULAUSE

Tämä opinnäytetyöaihe on tuonut minulle mahdollisuuden päästä työskentelemään robotiikan parissa. Haluan kiittää Oamk:n tutkintovastaavaa Tero Hietasta haastavasta ja mielenkiintoisesta aiheesta.

Suuret kiitokset VTT:n kaikille työntekijöille, jotka ovat olleet apuna tässä työssä. Erityisesti kiitokset kuuluvat VTT:n johtava tutkija Tapio Heikkilälle sekä insinööri Jarkko Kotaniemelle. Heidän asiantuntemuksensa ja osaamisensa toi ison avun tähän työhön.

Oulussa 19.11.2018

Juho Kotkaranta

SANASTO

CAN	automaatioväylä, jota käytetään mittaus- ja ohjaustietojen välittämiseen muun muassa autoissa
C++	ohjelmointikieli, joka on kehitetty C-kielestä
IMU	Inertial Measurement Unit on yksikkö, jonka antureilla on mahdollista mitata kiihtyvyyksiä
odometria	arvioi robotin paikkaa ja orientaatiota
Qt	Qt on alustariippumaton ohjelmistojen ja graafisten käyttöliittymien kehitysympäristö ja lausutaan kuten sana "cute"
SLAM	simultaneous localization and mapping; tarkoittaa yhtäaikaista paikantamista ja ympäristön mallintamista
wlan	lähiverkkotekniikka, joka usein tarkoittaa IEEE 802.11 -standardia, jonka avulla voidaan yhdistää erilaisia verkkolaitteita ilman kaapeleita

SISÄLLYS

ALKULAUSE	5
SANASTO.....	6
1 JOHDANTO.....	8
2 SLAM JA ETÄOHJAUS	9
2.1 Etäohjaus.....	9
2.2 SLAM	9
2.3 SLAM historia	10
2.4 Matemaattinen määritelmä SLAM:lle	11
3 QT JA MOBIILIROBOTTI.....	13
3.1 Qt.....	13
3.2 Mobiilirobottialusta	13
4 TYÖN SUORITUS.....	16
4.1 Robotin ohjelmistoon tutustuminen ja etäohjauksen suunnittelu.....	16
4.2 Etäohjaus.....	17
4.3 IMU-anturin tutkiminen	18
4.3.1 Nopeuden laskeminen datasta	19
4.3.2 Paikan laskeminen datasta.....	22
4.3.3 Nopeuden laskeminen ohjausarvoista	23
4.4 Akseleiden korjaaminen kameralta robottiin	25
4.5 Robotin akseliston korjaus karttaan.....	27
4.6 Seinäpintojen mittaus.....	28
4.7 Robotin käyttöliittymään piirrettävä kartta.....	30
5 NAVIGOINTI AJETUN REITIN AVULLA.....	32
5.1 Kohteiden rakeistaminen	32
5.2 Itsenäinen navigointi.....	33
5.3 Käyttöliittymän päivittäminen	33
6 TULOKSET JA POHDINTA.....	35
7 YHTEENVETO.....	37
LÄHTEET	38

1 JOHDANTO

Työn tarkoituksena on kehittää etäohjaus Probot Oy:n valmistamalle mobiilirobottialustalle sekä kehittää alustalle SLAM-navigointimenetelmää. Työ on kokonaisuudessaan tehty VTT:n tiloissa. Teknologian tutkimuskeskus VTT Oy on kansallisella statuksella toimiva yksi Euroopan johtavista tutkimus- ja teknologiaorganisaatioista (1).

Mobiilirobottialustalle on tehty aikaisemmin koordinaatiohjaus (2), mobiilirobotin navigointimenetelmät (3) sekä mobiilirobotin navigointimenetelmän parantaminen odometrian avulla (4). Koordinaatiohjauksen on toteuttanut Jussi Pulkkinen ja mobiilirobotin navigointimenetelmät on tehnyt Taavi Oksanen. Viimeisimmät muutokset ovat Jarkko Kotaniemen tekemiä ja liittyvät navigointiin odometrian avulla.

Robottia on voitu ajaa joko nuolinäppäimillä tietokoneelta tai automaattiajolla kokenäön avulla. Robotin hallitseminen on ollut hankalaa, sillä nuolinäppäimillä ajettaessa on käyttäjän pitänyt kulkea robotin perässä. Automaattiohjauksella robotti on kyennyt kiertämään keiloja tai seuraamaan suoria seinäpintoja.

Robottiin aikaisemmin tehtyjen töiden pohjalta lähdettiin kehittämään SLAM-navigointimenetelmää. Tällä navigointimenetelmällä robotin tulisi mallintaa karttaan ympäristöään ja paikoittamaan itsensä siihen.

Tässä työssä mobiilirobottialustalle kehitettiin Qt Creatorilla ohjelma, jolla mobiilirobotin mukana kulkeva päätietokone vastaanottaa TCP/IP-yhteydellä ohjauskomentoja langattoman verkon kautta. SLAM-navigointimenetelmällä robotti kykenee hahmottamaan kuljetulta reitiltä seinäpintoja ja seuraamaan ajettua reittiä itsenäisesti.

2 SLAM JA ETÄOHJAUS

2.1 Etäohjaus

Etäohjain on usein pieni, käsikäyttöinen elektroninen laite, jolla ohjataan toista laiteita, kuten esimerkiksi televisiota, radiota tai nosturia. Etäohjauslaitteet käyttävät usein tiedonsiirtoon infrapuna-aaltoja, mutta tiedonsiirto voi tapahtua myös radioaaltojen välityksellä. Etäohjauksen tiedonsiirto voidaan myös toteuttaa langallisena, jolloin ohjattava laite kytketään kaapelilla ohjaavaan laitteeseen.

Tyypillisin etäohjauksessa käytetty tiedonsiirtotekniikka perustuu infrapunasäteisiin. Kun näppäintä kauko-ohjaimen näppäintä painetaan, se lähettää järjestelmällisen valopulssisarjan. Yleensä lyhyt valopulssi vastaa binäärilukua 1 ja kun valopulssia ei havaita, se vastaa binäärilukua 0. Infrapunasäteillä toimivan kauko-ohjaimen toimintasäde on lyhyt ja tiedonsiirto heikkenee merkittävästi, jos ohjattavan ja ohjaavan laitteen välissä on este. (5.)

Kun lähetettävä tietomäärä on suuri tai etäisyydet pitkiä, käytetään radio-ohjausta. Radio-ohjauslaitteet toimivat radioaalloilla, ja niillä on olemassa kahden tyyppisiä ohjaustapoja. Monikanavaisella radio-ohjauksella pystytään lähettämään monipuolisia viestejä ja ohjaamaan useita toimilaitteita yhdenaikaisesti. Yksikanavainen radio-ohjaus kykenee ohjaamaan yhtä laitetta kerrallaan. Monet tunnetut tekniikat kuten WLAN ja RFID lähettävät informaation radiotaajuuksilla. (5.)

2.2 SLAM

Onko mahdollista, että robotti sijoitetaan tuntemattomalle paikalle, tuntemattomassa ympäristössä ja robotti kykenisi luomaan ympäröivästä maailmasta kartan sekä paikoittamaan itsensä siihen? Tämä ongelma on ollut itsenäisesti liikkuvien robottien ongelma jo pidemmän aikaa ja ongelman täydellinen ratkaiseminen tekisi roboteista todella itsenäisiä. Tätä ongelmaa kutsutaan SLAM-ongelmaksi. (6, s. 1.)

SLAM-ongelman ”ratkaisu” on ollut yksi merkittävimmistä menestyksistä robotihtiyeisöjen keskuudessa viime vuosikymmenet. SLAM on muotoiltu ja ratkaistu teoreettisena ongelmana useissa eri muodoissa. Joten SLAM voidaan pitää ratkaistuna ongelmana teoreettisella ja käsitteellisellä tasolla. (6, s. 1.)

2.3 SLAM historia

Todennäköisesti SLAM-ongelma syntyi 1986 San Franciscossa pidetyssä IEEE robotiikka- ja automaatiokonferenssissa. Näinä aikoina robotiikan ja keinoitekoisen älyn kehittäminen oli vasta alkutekijöissä. Muutamia tutkijaa olivat kehitelleet arvio-teoreettisia tapoja kartoitus- ja paikannusongelmaan. Konferenssissa keskustelu johti SLAM-ongelman alkutekijöihin. Pitkien keskustelujen ja luonnosten jälkeen, lopputuloksena oli, että määrätietoinen navigointi on perustavanlaatuisen ongelma robotiikassa. Lisäksi määrätietoiseen kartoitukseen huomattiin liittyvän merkittäviä käsitteellisiä kuin myös matemaattisia ongelmia. (6, s. 1.)

Useita vuosia kestäneen työn jälkeen julkaisi joukko tutkijoita tilastollisen perustan maamerkkien välisten suhteiden ja geometrisen epävarmuuden välille. Tämän tutkimuksen pääsanoma oli, että kartassa kahden eri maamerkin paikka ja korrelaatio kasvavat perättäisissä havainnoissa. (6, s. 1.)

Kaksi eri tutkijaryhmää työskenteli samanaikaisesti navigoinnin parissa, joista toinen työskenteli visuaalisen navigoinnin ja toinen kaikuluotaukseen perustuvan mobiilirobotin parissa. Näillä kahdella tutkimusalueella oli yhteistä se, että myöhemmin näistä tutkimuksista muuntui yksi yhteinen ongelma. Tämä kulki hetken aikaa yhdistettynä kartoitus- ja paikannusongelmana. Tärkein huomio oli, että maamerkkien välinen korrelaatio, mitä monet tutkijat yrittivät minimoida, oli kriittinen osa ongelmaa ja mitä suuremmiksi korrelaatiot kasvoivat, sitä paremmat tulokset saatiin. SLAM-ongelman rakenne, tulokset sekä kirjainlyhenne SLAM esiteltiin ensimmäisen kerran kansainvälisesti 1995 kansainvälisessä robotiikan-tutkimuskonferenssissa. (6, s. 2.)

2.4 Matemaattinen määritelmä SLAM:lle

SLAM -ongelma on määritelty seuraavasti. Mobiilirobotti saapuu tai käynnistetään tuntemattomassa ympäristössä, näin ollen tiedetään, mikä on robotin sen hetkinen sijainti. Käynnistystilanteessa sen hetkinen sijainti valitaan kartan luomiseen alkupisteeksi ja robotti aloittaa paikallistamisen tunnetusta koordinaattipisteestä. Liikkuessaan robotti tekee ympäristöstään havaintoja, joiden avulla luodaan mallia ympäristöstä. Robotin rakentaessa karttaa ympäristöstään havaittujen kohteiden avulla, muodostuu ongelma, sillä robotin pitäisi kyetä paikoittamaan oma sijaintinsa karttaan. (7, s. 872.)

Merkitään aikaa t ja robotin sijainti X_t . X_t on yleensä kolmiulotteinen vektori, joka koostuu sen kaksiulotteisista koordinaateista sekä pyörimisarvoa kuvaavaa vasta suunnasta. Sijainti tai kuljetun reitin sekvenssi on kaava 1. (7, s. 872.)

$$X_T = \{x_0, x_1, x_2, \dots, x_T\} \quad \text{KAAVA 1}$$

Tästä lukujonosta voidaan päätellä vain, että x_0 on aloituspiste ja että x_T on loppuspiste, mikä voi olla ääretön. Näiden kahden pisteen väliltä ei tiedetä mitään. Matkamittaus kertoo kahden perättäisen pisteen etäisyyden toisistaan. Merkitään odometriaa kirjaimilla U_T , missä t kertoo liikkumisen ajanhetkien $t-1$ ja t välillä. Robotin suhteellista liikettä kuvaava sekvenssi on kaava 2. (7, s. 872.)

$$U_T = \{u_1, u_2, u_3, \dots, u_T\} \quad \text{KAAVA 2}$$

Häiriöttömässä liikkeessä, U_T voisi olla riittävä palauttamaan robotin takaisin lähtöpisteeseen x_0 . Kuitenkin matkamittauksen suureet ovat häiriöllisiä ja integraalitekniikat matkan suhteen erkaannuttavat sen todellisuudesta. (7, s. 872.)

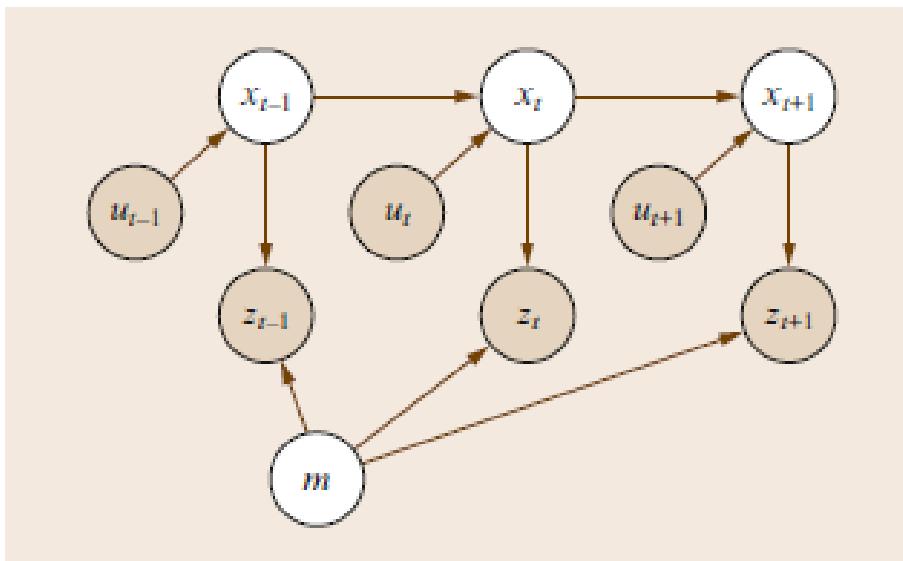
Robotti havaitsee kohteen sen ympäristössä. Ympäristö voi muodostua yksinkertaisista kohteista, kuten seinäpinnoista tai maamerkeistä. Merkitään näitä kohteita kirjaimella m , jolloin ne edustavat todellisia pisteitä robotin maailmassa. (7, s. 872.)

Havainnosta seuraa, että robotin on kyettävä mittamaan kohteen m etäisyys sen omaan sijaintiin X_t nähden. Oletetaan, että robotti ottaa täsmällisesti yhden mittauksen jokaisessa ajanhetkessä, saadaan kaavan 3 osoittama mittausselvennsi. (7, s. 872.)

$$Z_T = \{z_0, z_1, z_2, \dots, z_T\}$$

KAAVA 3

Muuttujat ovat sitoutumisen toisiinsa SLAM -ongelmassa. Graafinen malli (kuva 1) auttaa ymmärtämään muuttujien riippuvuudet SLAM-ongelmassa. Se näyttää hetkellisen syy-yhteyden sijainti- ja anturimittausselvennsien välillä. Mikäli yksikin mittausselvennsi harhautuisi, todellinen paikka kohteen ja robotin välillä piirtyisi väärin ja robotin navigointikyky heikentyisi. (7, s. 872.)



KUVA 1. SLAM-navigoinnin muuttujien vaikutus toisiinsa (7, KUVA 37.1, S. 827.)

3 QT JA MOBIILIROBOTTI

Robotilla tarkoitetaan konetta, joka kykenee suorittamaan monimutkaisia tehtäviä automaattisesti (8). Tässä työssä on käytetty mobiilirobottia, joka liikkuu renkaiden avulla. Mobiilirobotti voi liikkua myös muilla keinoin, esimerkiksi jalkojen avulla.

3.1 Qt

Qt on alustariippumaton ohjelmistojen ja graafisten käyttöliittymien kehitysympäristö. Qt sisältää hyödyllisiä laajennuksia kuten signaalit ja slotit, mikä helpottaa tapahtumien käsittelyä. Qt käyttää pääosin C++-kieltä.

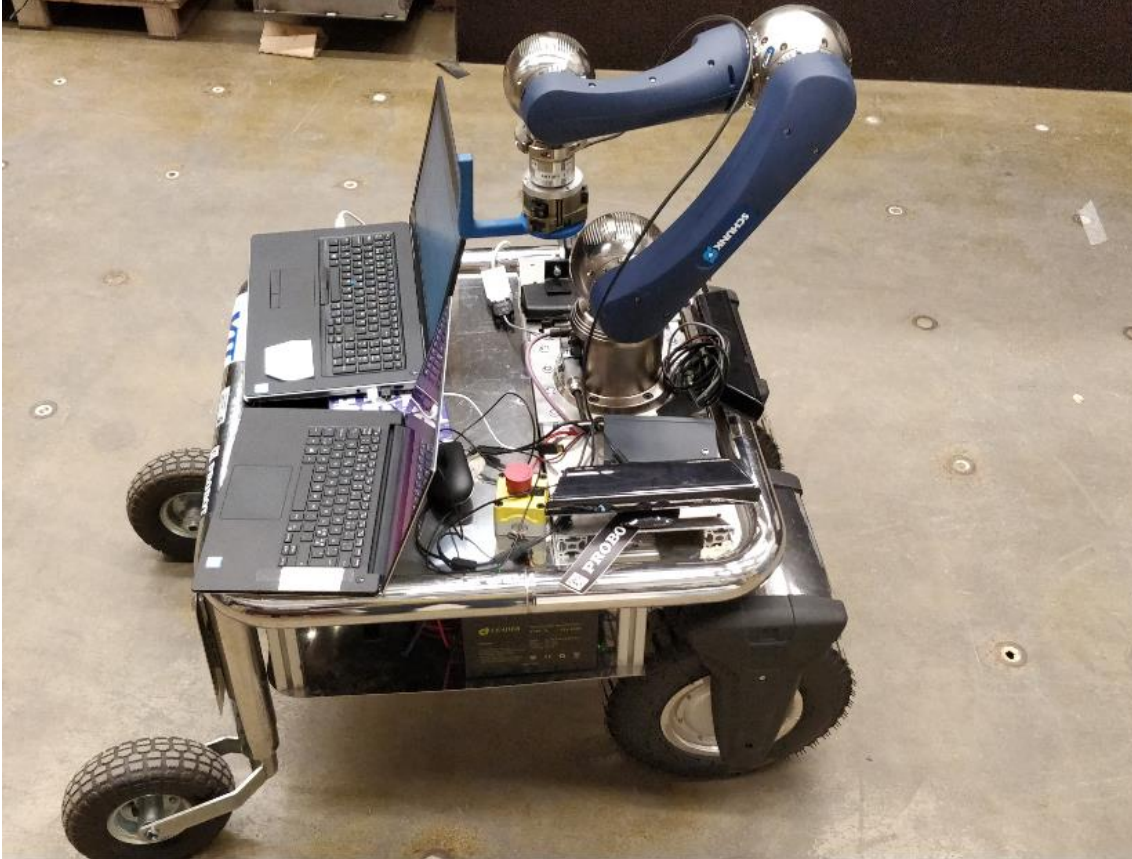
Qt Creator on ohjelmistoympäristö ja käyttää hyväksi Qt:a. Qt Creator sisältää koodieditorin, graafisen käyttöliittymien suunnittelutyökalun ja tukee useita eri käyttöliittymiä, debuggereita ja versionhallintaohjelmistoja. Pääsääntöisesti Qt Creator on suunniteltu C++-kielelle mutta myös muita ohjelmointikieliä on saatavilla erinäisten kirjastojen avulla. (9.)

Tässä työssä käytettiin hyväksi Qt:n sisältämiä hyödyllisiä sarja- ja verkkoyhteys ominaisuuksia. Verkkoyhteydellä mobiilirobotin ohjaustietokoneelle voidaan lähettää ohjausviestejä ja sarjayhteyttä käytetään muun muassa IMU-anturin ja Kinect-kameroiden lukemiseen.

3.2 Mobiilirobottialusta

Probot Oy:n valmistama mobiilirobottialusta (kuva 2) on differentiaaliajainen mobiilirobotti. Robotin moottoreina toimivat molemmissa renkaissa olevat Mobility Module -yksiköt. Mobility Moduleita ohjataan PEAK:n CAN-muuntimella. Robotissa takarenkaat pyörivät vapaasti ja ovat kasterimallin renkaita. Takarenkaat kääntyvät helposti, mutta joissain tilanteissa takarenkaat vaikeuttavat robotin liikumista. Näitä tilanteita on esimerkiksi peruuttaessa, kun takarenkaat kääntyvät

robotin alle. Mobiilirobotin moottorit ovat momenttiohjattuja, jolloin edellä mainitussa tilanteessa moottorien vääntömomentti kohdistuu renkaiden kääntämiseen eikä robotin kiihdyttämiseen.



KUVA 2. Mobiilirobotti.

Mobiilirobotin CAN-muuntimeen luodaan yhteys tietokoneen sarjaliikenneportin kautta. Yhteys luodaan Qt Creatorilla tehdyssä pääohjelmistossa. Tämän jälkeen ohjelmisto lähettää käskyjä CAN-muuntimeen, joka muuntaa käskyt sopiviksi moottoriyksiköille. Moottoreille lähetetään kahta eri tietoa, joista toinen on nimetty "speed" ja tarkoittaa vääntömomenttia. Toinen lähetettävä arvo on "steer", joka tarkoittaa moottorien vääntömomenttiero. Moottorien vääntömomenttien erot mahdollistavat robotin ajamisen vasemmalle ja oikealle.

IMU (Inertial Measurement Unit) on laite, joka mittaa kiihtyvyyksiä ja kulmanopeuksia, joskus suuntakulmaa kiihtyvyyksimittareiden, gyroskooppien ja magnetometrien avulla. Mittausdatasta voidaan laskea navigointiin tarvittavia arvoja, kuten suuntakulma.

Navigointikäytössä IMU-anturi tarvitsee jonkinlaisen korjausjärjestelmän, sillä anturi aiheuttaa kerääntyvää virhettä. Koska paikan muutos lasketaan integroimalla kiihtyvyyttä ja kulmanopeutta, kasvaa virhe jatkuvasti. Näin todellisen sijainnin ja laskennallisen sijainnin ero kasvaa jokaisella laskentakerralla. (6, s. 484.)

Mobiilirobotissa oleva IMU-yksikkö on VTT:n Little Node -yksikkö. Tämän yksikön anturina toimii LSM9DS0-inertiamoduuli. Yksikkö mittaa kolmiulotteisesti kiihtyvyyden, kulmanopeuden ja magneettikentän voimakkuuden (10). IMU-anturi mittaa jatkuvasti, mutta lähettää mittaustulokset vain silloin, kun sille lähetetään ohjaustietokoneelta "1".

Mobiilirobotti kykenee havainnoimaan ympäristöstään erilaisia kohteita kahden Kinect-kameran avulla. Robotti kykenee tunnistamaan maamerkkejä ja seinäpintoja kameroiden avulla, jotka lähettävät tietokoneelle pistepilven, mistä ROS-pohjainen ohjelmisto tunnistaa kohteita. Kameroiden ohjausohjelma toimii erillisellä tietokoneella, josta tiedot lähetetään UDP-yhteydellä mobiilirobotia ohjaavalle tietokoneelle. Tässä työssä Kinect-kameroiden kuvaa käytetään hyväksi havainnoimaan ympäristöstä kohteita.

Robotin Kinect-kameroihin liittyy pieni ongelma, sillä kameroiden näkökenttä on noin 60° vaakasuunnassa ja syvyysnäön maksimietäisyys noin 4,5 metriä. Todellisuudessa kameran kohteiden tunnistaminen onnistuu 0,8–2 metrin etäisyydeltä, näköetäisyyden ollessaan maksimissaan 3 metriä. (4, s. 15.)

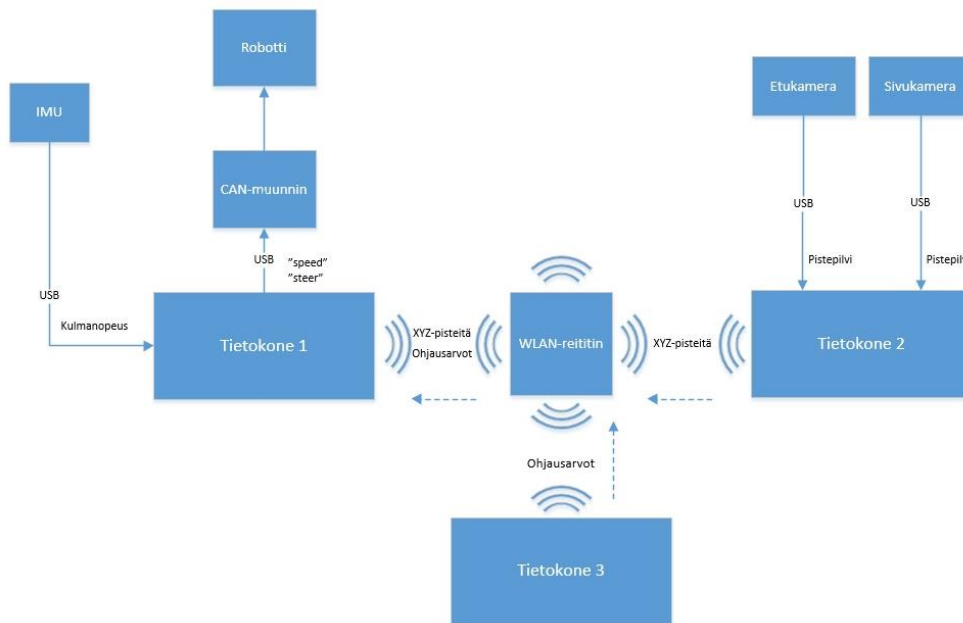
4 TYÖN SUORITUS

Työn toteutus alkoi tutustumalla mobiilirobotissa olevaan ohjelmistoon ja siihen liittyviin dokumentaatioihin. Ohjaustietokoneen ohjelmistot on kirjoitettu C++-kielillä käyttäen Qt Creator -editoria. Ohjelmistorakenteeseen täytyi tutustua tarkemmin, sillä olemassa oleva ohjelmisto oli hyvin laaja. Qt Creator oli tekijälle täysin uusi ohjelmointiympäristö, mutta C++-kielen perustaidot auttoivat pääsemään alkuun. Robotin ohjaustietokoneen lisäksi tarvittiin toinen tietokone ja siihen sopiva ohjelma, jolla pystytään lähettämään mobiilirobottiin haluttuja etäohjausviestejä. Robotin Kinect-kameroiden toiminta täytyi selvittää ja robotin IMU-yksikkö jouduttiin koteloimaan ja kiinnittämään robotin runkoon.

4.1 Robotin ohjelmistoon tutustuminen ja etäohjauksen suunnittelu

Mobiilirobotin laitteistoon kuuluu kolme tietokonetta, IMU-anturi, CAN-muunnin sekä etu- ja sivukamera. Kuvassa 3 näkyvä tietokone 1 on robotin päätietokone. Tällä koneella suoritetaan mobiilirobottia ohjaavaa pääohjelmaa, mikä pyytää IMU-anturilta mittaustuloksia ja laskennan jälkeen lähettää CAN-muuntimelle ohjauspyyntöjä.

Tietokone 2 ottaa etu- ja sivukameroilta kuvia. ROS-pohjainen ohjelma käsittelee kuvat ja segmentoi siitä erilaisia kohteita kuten seinäpintoja. Kun kuva on saatu käsiteltyä, lähetetään päätietokoneelle UDP-yhteydellä XYZ-pistepilvi.



KUVA 3. Robotin laitteistokaavio.

Tietokone 3 kuvassa 3 on tämän työn aikana lisätty etäohjausta varten. Tältä tietokoneelta voi lähettää ohjaukskäskyjä TCP/IP-yhteydellä päätietokoneelle. Täähän on valittu TCP/IP-yhteys, koska tarkoituksena on tulevaisuudessa lähettää dataa päätietokoneelta takaisin tietokoneelle 3.

Päätietokoneelta käyttäjän on valittava käyttöliittymästä käyttäjäohjaus. Kun paikallisohtaus ei käytetä, voi robottia ohjata myös automaattiohtauksella. Automaattiohtauksella robotti lähtee liikkeelle, mikäli havaitsee seinäpinnan tai maamerkin. Automaattiohtaus on valittava erikseen käyttöliittymästä.

4.2 Etäohjaus

Mobiilirobotin pääohjelma sisältää jo valmiiksi tehdyn paikallisen nuolinäppäimillä tapahtuvan ohjauksen. Tätä ohjaustapaa on hyödynnetty myös etäohjauskoneella.

Näppäimistön yksittäinen näppäin voi olla joko painettuna tai painamatta. Näin ollen arvoiksi voi muodostua joko 1, eli painettuna tai 0, jolloin näppäintä ei ole

painettu. Nuolinäppäinten asentotietojen lähettämiseen tarvitaan siis 4 bittiä, joista jokainen nuolinäppäin edustaa yhtä bittiä (kuva 4). Ohjelma lukee vain nuolinäppäimiä, joten muilla näppäimistön näppäimillä ei ole merkitystä.



KUVA 4. Nuolinäppäimien bitit.

Näppäimistä muodostuu neljän bitin jono. Ohjelma muodostaa näistä kokonaisluvun 0–15, sen mukaan, mitä näppäimiä on painettu. Kokonaisluku pakataan ja lähetetään TCP/IP-yhteydellä päätietokoneelle. Päätietokoneella kokonaisluku puretaan jälleen biteiksi ja lasketaan, mitä näppäimiä on painettu. Etäohjauksen myötä myös kuvia käsittelevä tietokone lähettää langattomalla yhteydellä havaittuja kohteita päätietokoneelle.

4.3 IMU-anturin tutkiminen

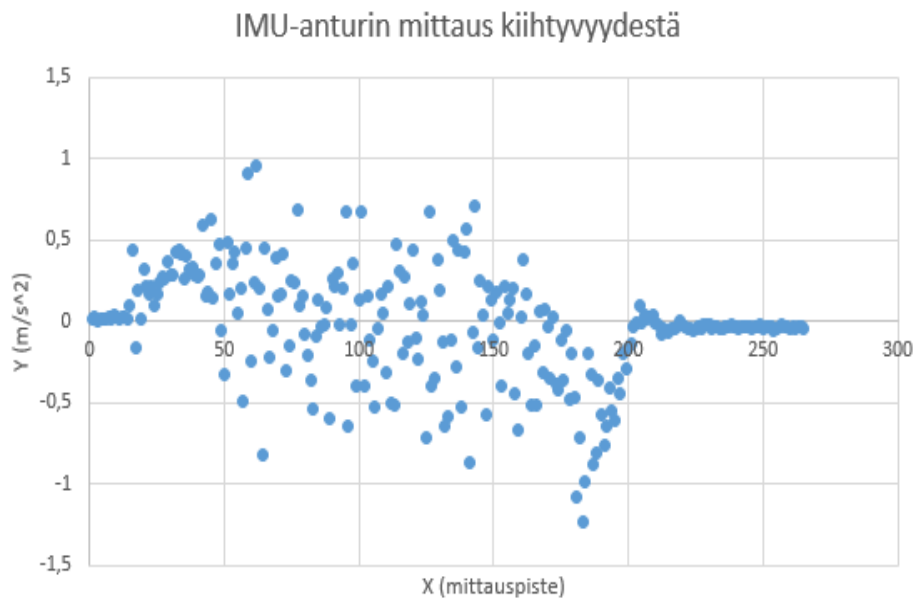
Ensimmäinen askel oli mitata IMU-anturin ominaisuuksia. Mittausjärjestelyt koostuivat tietokoneesta, johon oli tehty IMU-anturia lukeva ohjelma. Anturi on kiinnitetty robottiin ja robotilla ajetaan reitti. Reitti tiedetään, joten arvoja pystytään vertaamaan ja todentamaan oikeiksi. Ennen mittauksia IMU-anturi koteloidiin (kuva 5) ja asennettiin kiinteäksi osaksi robottia.



KUVA 5. IMU-anturi koteloituna robotin runkoon.

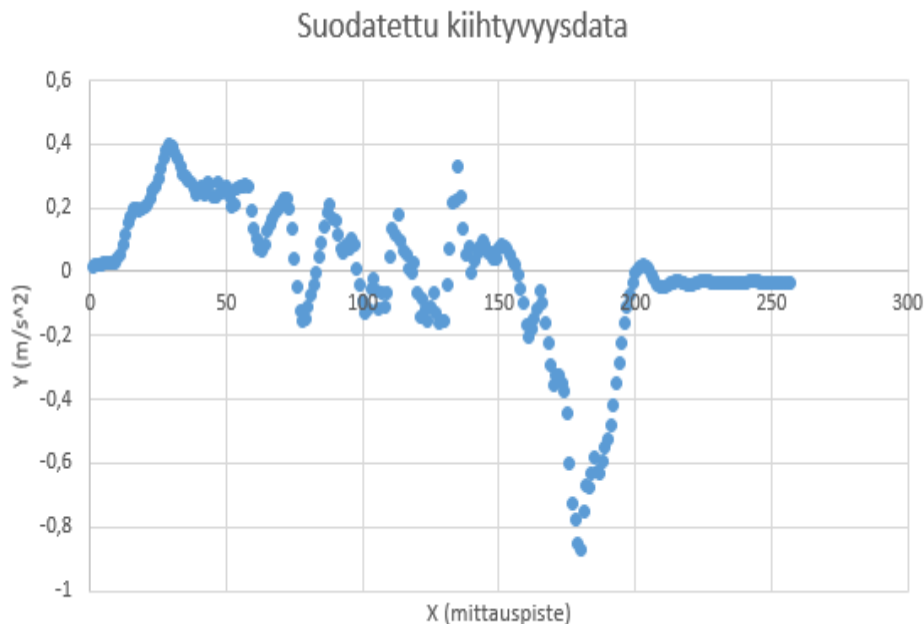
4.3.1 Nopeuden laskeminen datasta

IMU-anturilta tuleva data on hyvin häiriöllistä, joten sitä joudutaan suodattamaan. Kuvasta nähdään, että robotin liike on pysähtynyt mittauksen alussa ja lopussa, sillä kiihtyvyyden on lähellä nollaa.



KUVA 6. IMU-anturin antamaa suodattamatonta dataa, missä y-akselilla on kiihtyvyyden ja x-akselilla mittauspisteet.

Dataa joudutaan käsittelemään, jotta voidaan tehdä päätelmiä. Kuvassa 7 dataa on suodatettu ja pisteitä voi jo verrata todellisuudessa tapahtuneisiin kiihtyvyyksiin. Suodattaminen tapahtuu keskiarvoistamalla saatuja mittauspisteitä.



KUVA 7. IMU-anturin datat suodatettuna.

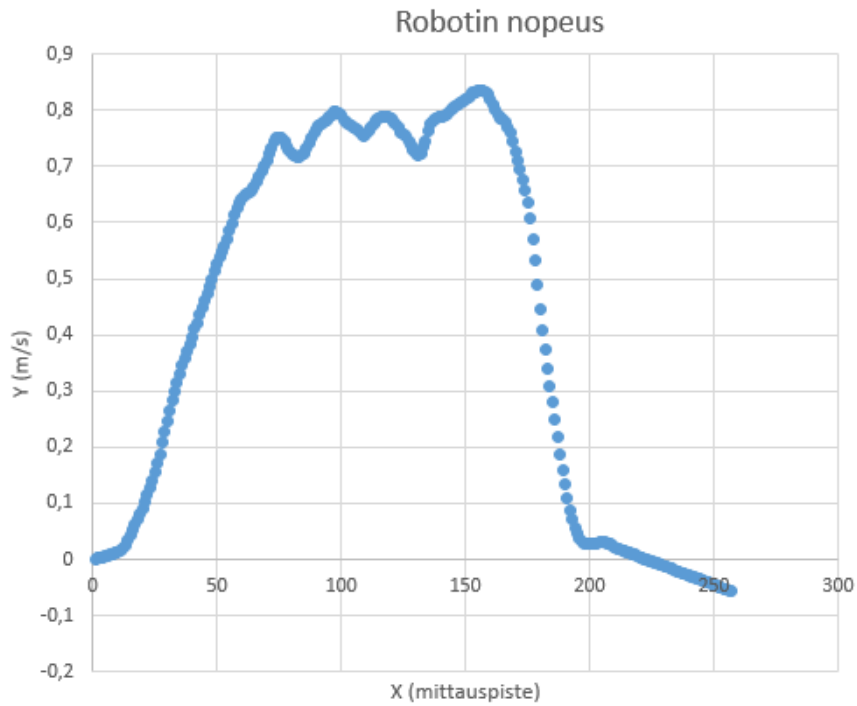
Kiihtyvyydestä saadaan nopeus laskettua kaavalla 4. Tässä vaiheessa voidaan nähdä IMU-antureihin liittyvä ongelma. Alussa nopeus on nolla, mutta loppupisteessä näyttää, että robotti jatkaisi liikkumista taaksepäin. Näin ei kuitenkaan ole, vaan robotti on pysähdyksissä.

$$v = \sum a_i * \Delta t,$$

KAAVA 4

missä v on nopeus, a on kiihtyvyys ja t aika.

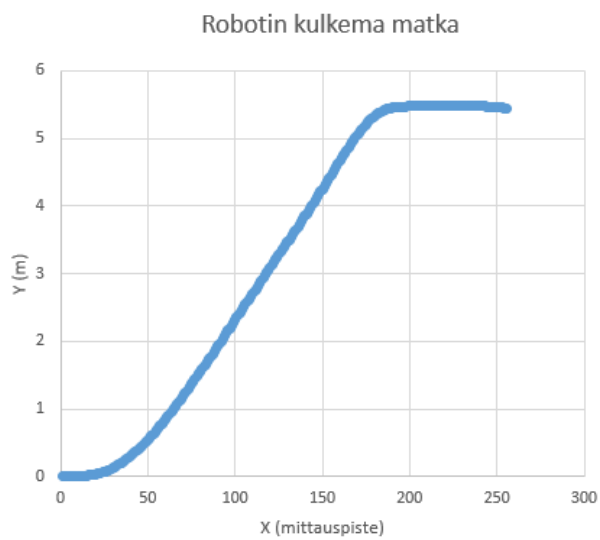
Kun tiedetään nopeus (kuva 8) ja aika, on yksinkertaista laskea siitä matka (kaava 5). Kuvassa 9 nähdään robotin kulkema matka suhteessa mittauspisteisiin. Kaavassa 5 v_i on hetkellinen nopeus ja Δt mittausaikaväli.



KUVA 8. Robotin nopeus (m/s) suhteessa mittauspisteisiin.

$$s = \sum v_i * \Delta t,$$

KAAVA 5



KUVA 9. Robotin kulkema matka (m) suhteessa mittauspisteisiin.

Mittauksen tarkoituksena oli mitata robotin IMU-anturin luotettavuutta nopeuden suhteen. Todellisuudessa robotti ei liiku, mutta anturilta saatava kiihtyvyyden integroinnista johtuva virhe muuttaa sen paikkaa laskennallisesti. Jos integrointi-
virhe sallitaan, robotin navigointi ei ole luotettavaa.

4.3.2 Paikan laskeminen datasta

Navigointi tarvitsee toimiakseen robotin kulkusuunnan, paikan ja myös nopeuden. Näiden mittausten pysyessä tarkkoina voidaan laskennallisesti pitää navigointia luotettavana. Tiedetään robotin hetkellinen nopeus sekä suuntakulma, joista voidaan laskea paikka.

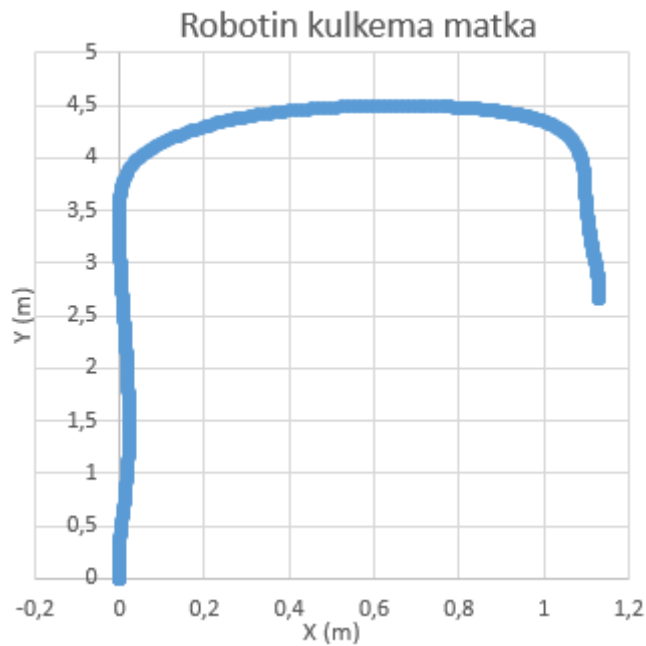
Suoritetaan mittaus, missä tarkkaillaan erityisesti robotin orientaatiota. Merkitään lattiaan teipillä robotin lähtöpiste, kohta missä tehdään ensimmäinen käänös, sekä toisen käänöksen kohta. Tässä mittauksessa nopeus viritetään jälkeensä kohdalleen. Määritetään datasta robotin kulkema reitti ja verrataan sitä todellisuudessa kuljettuun reittiin.

Datasta voidaan laskea suuntakulma θ käyttäen kaavaa 6. Tämän jälkeen voidaan laskea x- ja y-koordinaatit käyttäen hyväksi suuntakulmaa θ . Kaavassa 4 oleva ω_i tarkoittaa hetkellistä kulmanopeutta. Kaavoissa 7 sekä 8 esiintyvä v_i on hetkellinen nopeus ja Δt mittausaikaväli.

$$\theta = \sum \omega_i * \Delta t \quad \text{KAAVA 6}$$

$$x = \sum v_i * \Delta t * \cos \theta \quad \text{KAAVA 7}$$

$$y = \sum v_i * \Delta t * \sin \theta \quad \text{KAAVA 8}$$



KUVA 10. Robotin kulkema reitti koordinaatistokartassa.

Kuvasta 10 nähdään, että robotti alkoi tehdä loivaa käännoästä sen jälkeen, kun se oli ajanut suoraan noin 3,7 metriä. Loivan käännoäksen aikana robotti kulki lähtöpisteeseen nähden X -akselin suunnassa noin 80 senttimetriä ja teki toisen loivan käännoäksen. Tämän jälkeen robotti pysäytettiin.

Mittauksen tarkoituksena oli mitata robotin IMU-anturin kykyä mitata kulmanopeutta. Nopeus viritettiin jälkeinpäin kohdalleen. Kun mittaustuloksia verrattiin todellisuudessa kuljettuun reittiin, tullaan tulokseen, että tämä tarkkuus riittää robotin navigointiin.

4.3.3 Nopeuden laskeminen ohjausarvoista

Aikaisemmat mittaukset osoittivat, ettei IMU-anturilla mitattu nopeus pysy luotettavana kiihtyvyyden integrointivirheestä johtuen. Vaihtoehtoinen tapa olisi mitata nopeus esimerkiksi renkaiden pyörimisnopeudesta. Mobiilirobotin moottorit eivät kuitenkaan sisällä tähän mahdollisuutta, joten päädyttiin laskemaan jokaista eri ohjausarvoa vastaava nopeus.

Suoritetaan mittaus, missä jokaista ohjausarvoa vastaava robotin kulkunopeus määritetään. Robotti ajaa niin sanotun rullaavan lähdön, eli robotti ensin kiihdytetään kulkemaan ohjausarvoa vastaavaa nopeutta. Kun tavoitenopeus on saavutettu ja saavutaan alkupisteelle, käynnistetään sekuntikello ja loppupisteessä se pysäytetään. Tämä suoritetaan jokaiselle robotin käyttämälle ohjausarvolle kaksi kertaa. Ohjausarvoa vastaavista nopeuksista lasketaan keskiarvot. Matkasta ja ajasta saadaan laskettua nopeus (kaava 9).

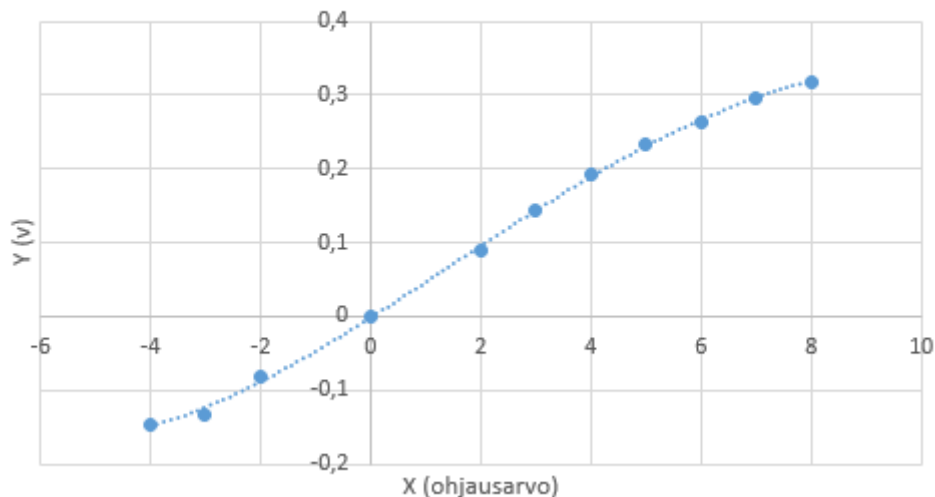
$$v = \frac{s}{t},$$

KAAVA 9

missä v on nopeus, s on matka ja t on aika.

Nopeudet asetetaan taulukkoon ja saadaan nopeutta kuvaavia pisteitä koordinaatistossa. Näiden pisteiden välille sovitetaan yhtälö, mistä saadaan laskettua nopeus millä vain ohjausarvolla. Huomattavaa on, että robotin moottoreiden vääntömomentit ovat liian pieniä liikuttamaan robottia, kun ohjausarvot olivat välillä $(-1 < x < 1)$.

Robotin nopeuden määrittäminen ohjausarvoista



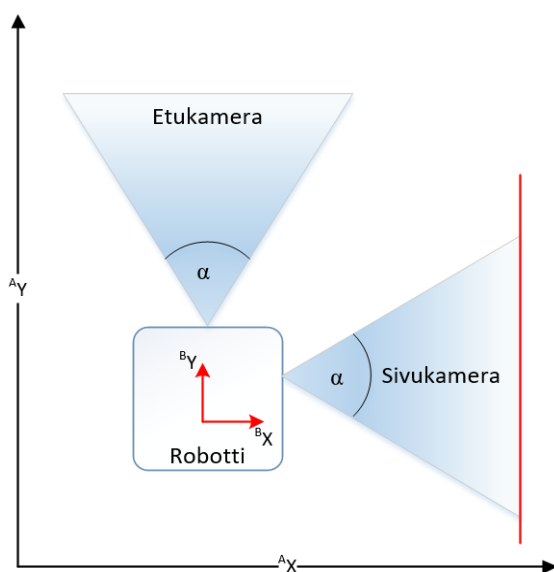
KUVA 11. Robotin ohjausarvoja kuvaava funktio.

Robotille annetaan ohjelmaan ohjausarvoista määritelty funktio (kuva 11). Funktiolla robotti kykenee laskemaan nopeutensa riittävällä tarkkuudella. Robotti käyttää navigoinnissa hyväkseen IMU-anturilta tulevaa kulmanopeusmittausta ja laskee sen avulla suuntakulmansa.

4.4 Akseleiden korjaaminen kameralta robottiin

Tässä osiossa perehdytään robotin akselistoihin sekä kartan piirtämiseen. Jotta robotti pysyisi laskennallisesti mukana sen hetkisestä paikastaan, on tiedettävä robotin nopeus sekä kulkukulma. Näillä arvoilla pystytään piirtämään karttaa, missä näkyy robotin kulkema reitti. Aikaisemmassa vaiheessa määriteltiin IMU-anturin ja ohjausarvofunktion avulla robotin paikkaa. Karttaan on tarkoitus piirtää robotin kulkema reitti ja reitillä nähtyjen kohteiden paikka.

Mobiilirobotin Kinect-kamerat kuvaavat robotin oikealle puolelle sekä kulkusuuntaan. Seinien määrittelyyn käytetään oikealle puolelle kuvaavaa kameraa (kuva 12). Kuvassa näkyy kameroiden sijoittelu. Huomioon otettava asia on, että kamerat eivät sijaitse robotin keskipisteessä, kuten IMU-anturi. Maailmankoordinaatisto on merkitty $\{A\}$ sekä robotin koordinaatisto $\{B\}$. Kuvassa 12 α on Kinect-kameran näkökentän kulma vaakatasossa.



KUVA 12. Robotin kameroiden asettelu.

Kuvassa 12 robotin oikealle kuvaava Kinect-kamera näkee oikealla puolella seinän. Seinä on kameran näkökulmasta kaksiulotteinen XY-pinta, joka on matkan Z päässä. Kameran näkemät pisteet ovat robotin näkökulmasta väärit, sillä robotin ajosuunta (Y-akseli) on kamerassa X-akselina. Ja robotin X-akseli on kameran Z-akselina ja edelleen robotin Z-akseli on kamerassa Y-akselina.

Jos kameran akseleita ei käännetä, tulevat kameran näkemät objektit väärään paikkaan kartalla. Joudutaan käyttämään rotaatiomatriisia, että saadaan akselit täsmäämään robotin akseleiden kanssa ja objektit piirrettyä robotin näkökulmasta oikeaan paikkaan. Kuvasta 13 nähdään havainnollistettuna robotin ja kameran akselistot. Rotaatiomatriisin (kaava 10) avulla saadaan korjattua kameran segmentoimat pisteet suhteessa robottiin.

$${}^A R_Z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{KAAVA 10}$$

missä R on rotaatiomatriisi koordinaatistojen {A} ja {B} välillä Z-akselin suhteen, sekä θ on koordinaatistojen välinen kulma.

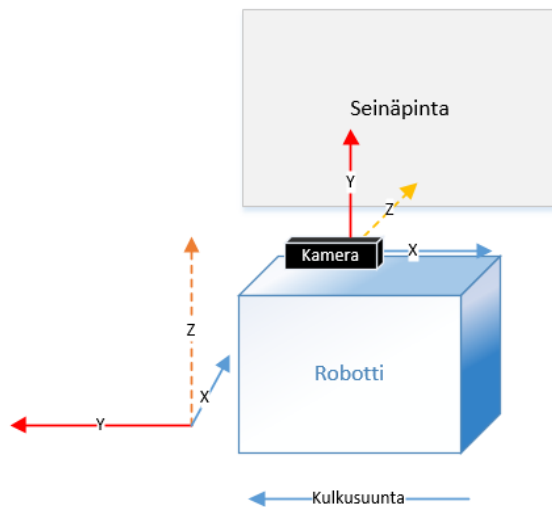
Kameran ja robotin koordinaatistojen välillä on kiertymää ja kameran sekä robotin koordinaatistojen keskipisteet eivät ole samassa pisteessä. Kameran havaitsemat pisteet eivät ole robotin koordinaatistossa ja käyttämällä kaavaa 12 saadaan kameran havaitsemat pisteet robotin koordinaatistoon. Kaava 11 kuvaa ainoastaan koordinaatistojen kiertymää.

$${}^A P = {}^A R * {}^B P, \quad \text{KAAVA 11}$$

missä piste P koordinaatistossa A saadaan, kun kerrotaan pistettä P koordinaatistossa B rotaatiomatriisilla R.

$${}^A P = {}^A R * {}^B P + P_{BORG}, \quad \text{KAAVA 12}$$

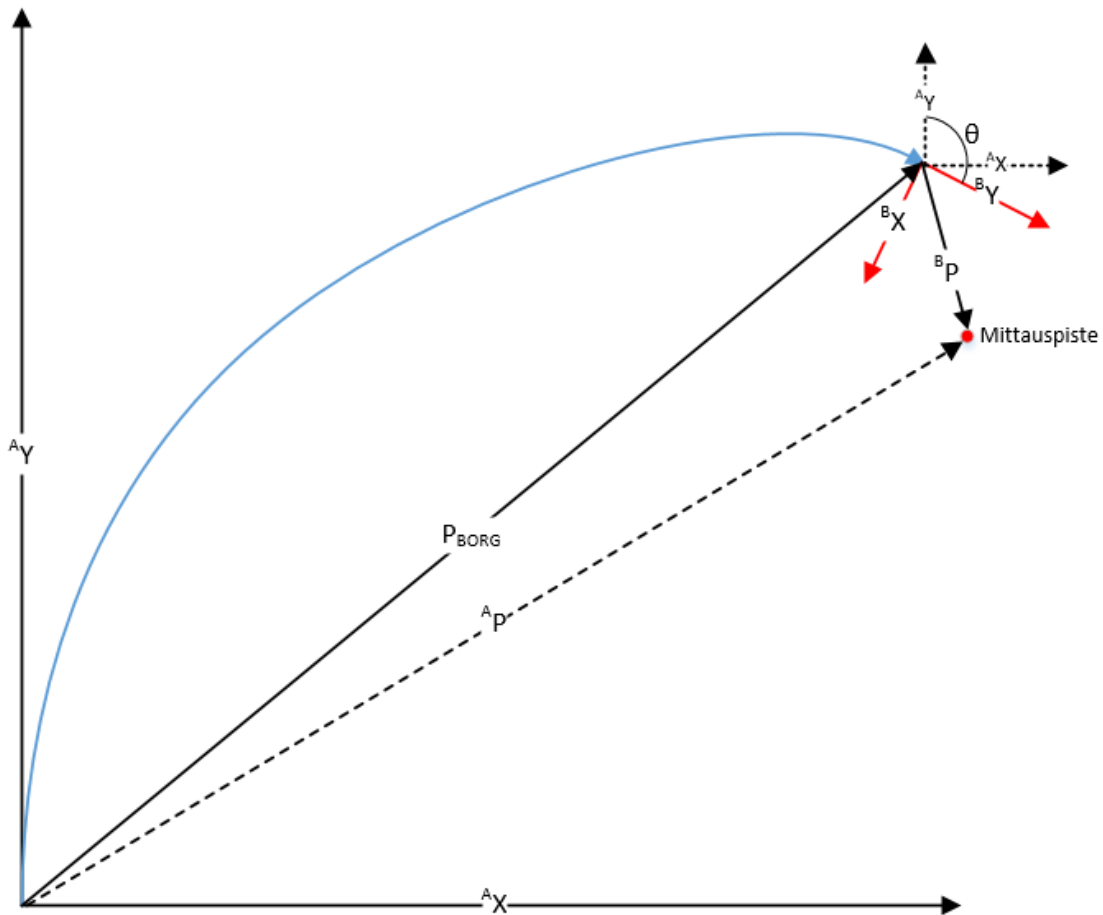
missä P_{BORG} on koordinaatistojen välinen vektori.



KUVA 13. Robotin ja kameran akselistot.

4.5 Robotin akseliston korjaus karttaan

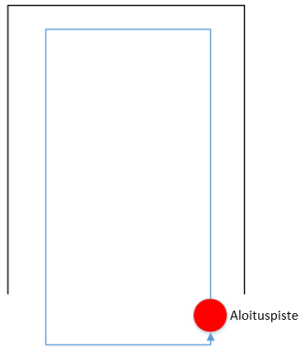
Kuvassa 14 on esitelty robotin akselistoa suhteessa todellisen maailman akselistöön. Robotti aloittaa kulkemisen $\{A\}$ -koordinaatiston origosta. Tässä tilanteessa robotin akselistö on suhteessa todellisen maailman akselistöön oikein. Robotin kulkema reitti on kaareva. Kuljettuaan kohtaan, missä $\{B\}$ -koordinaatisto sijaitsee kuvassa 14, on robotin akselistö kiertynyt ja siirtynyt todellisen maailman akselistöön nähden. Robotin näkökulmasta se kulkee aina y-akselinsa suuntaan. Tämän korjaamiseksi käytetään kaavaa 12, jolloin mittauspiste saadaan maailmankoordinaatistossa.



KUVA 14. Robotin akseliston käyttäytyminen suhteessa maailman koordinaatistoon.

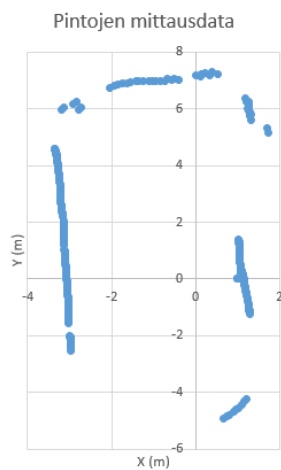
4.6 Seinäpintojen mittaus

Suoritetaan testi, missä mitataan vain konenäöllä nähtyjen seinien sijaintia. Mittausalue on suorakaiteen muotoinen tila (kuva 15). Robotin tulisi merkitä tekstitiedostoon havaittujen seinien paikat. Kuvassa 15 robotti aloittaa punaiselta aloituspisteeltä, kiertää sinisellä merkatun reitin ja kirjoittaa muistiin mustalla merkityt seinät. Ajettu reitti voi poiketa suunnitellusta, sillä robotin pitäisi osata ottaa huomioon seinäpinnan etäisyys.



KUVA 15. Mittausjärjestelyt.

Tuloksena saatiin seinäpisteitä, jotka sijoitettiin koordinaatistoon. Kuvasta 16 nähdään, että seinien segmentointi ei ole kokonaan onnistunut. Osittain tämä voi johtua siitä, että robotti on kulkenut liian läheltä mitattavaa pintaa. Mittaustuloksissa (kuva 16) näyttää, että vasemmanpuoleinen pitkä seinäpinta olisi hieman vinossa kuin myös päätyseinä ylhäällä. Vääntyminen johtuu IMU-anturin kalibroinnista.

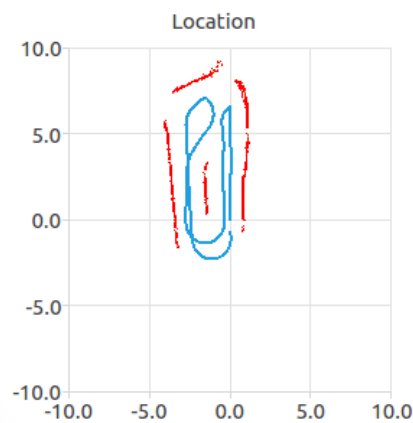


KUVA 16. Pintojen mittausdatat. Sinisellä on merkitty havaittuja kohteita.

4.7 Robotin käyttöliittymään piirrettävä kartta

Tässä vaiheessa otetaan robotin kulkema reitti mukaan karttaan. Käyttöliittymään piirretään koordinaatistokarttaan sinisellä robotin kulkema reitti ja punaisella seinäpintojen sijainnit. Koeajo suoritetaan samanlaisella mittausjärjestelyllä kuin aiemmassa kohdassa (kuva 15), mutta alkupisteelle lisätään seinäelementti, joka kierretään molemmilta puolilta. Tarkoituksena tässä on tutkia, siirtyykö seinä kartalla.

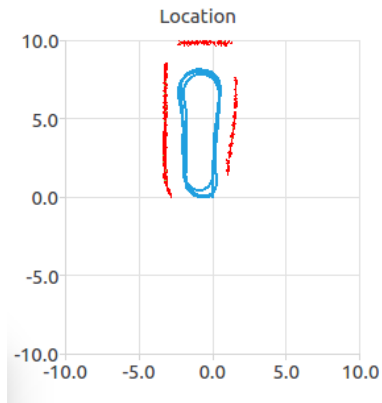
Mittaustuloksista huomaa (kuva 17), että seinät ovat koordinaatistokarttaan nähdessä selvästi vinossa. Seinät ovat suhteessa toisiinsa kuitenkin suorassa. Tämä virhe johtuu IMU-anturin kalibroinnista. Reitti kulkee molemmilla puolilla keskellä sijaitsevaa seinää. Mittaustuloksista nähdään, ettei seinän paikka vaihdu, mutta ongelmana on seinien taipuminen reitin mukana. Tämä johtuu kuvien käsittelystä koneesta. Kuvien käsittelyyn menee enemmän aikaa samalla kun IMU-anturi ehtii tuottaa useamman mittauspisteen. Virhe tulee, kun uusimpaan paikannustietoon liitetään menneisyydessä nähdyt segmentit.



KUVA 17. Robotin koordinaatistokartta käyttöliittymässä.

Seinien pyöristyminen johtuu paikannuksen ja mallinnuksen välisestä ajallisesta erosta. Ohjelmassa tehdään synkronointi nähtyjen segmenttien ja paikannusdatan välille. IMU-anturi kalibroidaan uudestaan ja suoritetaan samanlainen mittaus kuin aiemmassa kohdassa (kuva 15) on esitetty. Uusi mittaus osoittaa (kuva 18), että robotti kykenee hahmottamaan ympäristönsä erinomaisella tarkkuudella.

Mittauksessa kierrettiin kolme kierrosta rataa ympäri ja seinien paikka ei ole siirtynyt. IMU-anturin ja nopeuden laskennan avulla robotti pitää reaaliaikaisen sijaintinsa tarkkana, sillä aloitus- ja lopetuspaikat ovat samassa paikassa kartalla. Täältä pohjalta voidaan lähteä kehittämään navigointia.



KUVA 18. Robotti on kiertänyt kolme kierrosta.

5 NAVIGOINTI AJETUN REITIN AVULLA

Ongelmat on ratkaistu robotin akseleiden kanssa sekä paikan ja orientaation määrittelemisessä. Käsitellään navigointi omana osuutena. Tavoitteena on piirtää reaaliaikaista karttaa, missä näkyisi robotin kulkema reitti ja reitillä nähdyt seinäpinnat. Tässä työssä kohteet ovat pelkästään seinäpintoja. Seinäpinnoista robotin pitää luoda malli, missä otetaan huomioon etäisyys, miltä robotti havaitsi kohteen ja missä se sijaitsee robotin näkökulmasta.

Robotti ei tarvitsisi koordinaatistokarttaa toimiakseen, mutta käyttäjän on helppompaa seurata robotin liikkeitä ja havaittujen kohteiden sijaintia kartalta. Kartan piirtäminen on yksinkertainen asia, sijoitetaan reittipisteet koordinaatistoon ja samaan koordinaatistoon seinäpintojen pisteet. Kuvassa 16 suoritetussa koeajossa ilmeni ongelmia, sillä seinäpinnat taipuvat robotin liikkeiden mukaan. Seinien täytyisi pysyä paikallaan, sillä niiden avulla päivitetään robotin paikkaa x- ja y-akseleiden suunnassa.

5.1 Kohteiden rakeistaminen

Robotti näkee erilaisia kohteita kulkemallaan reitillä ja näistä kohteista muodostuu jo lyhyellä matkalla suuri joukko pisteitä. Vaikka nopealla taajuudella otettu mittaaminen olisi tarkempi tapa esittää seinien sijainteja sekä robotin paikkaa, joudutaan pisteitä käsittelemään. Mobiilirobotin keräämistä pisteitä seulotaan laskeamalla pisteiden välisiä etäisyyksiä. Näin voidaan tietää, ovatko nämä kaksi pistettä samalla alueella. Mikäli pisteet ovat samalla alueella, sulautetaan piste suurempaan kokonaisuuteen. Näin seinäpintojen pistejoukko saadaan rakeistettua. Robotti mittaa myös kulkemastaan matkasta navigointiin tarvittavia reittipisteitä. Nämä reittipisteet käsitellään samalla tavalla ja rakeistetaan lähimmät pisteet toisiinsa.

5.2 Itsenäinen navigointi

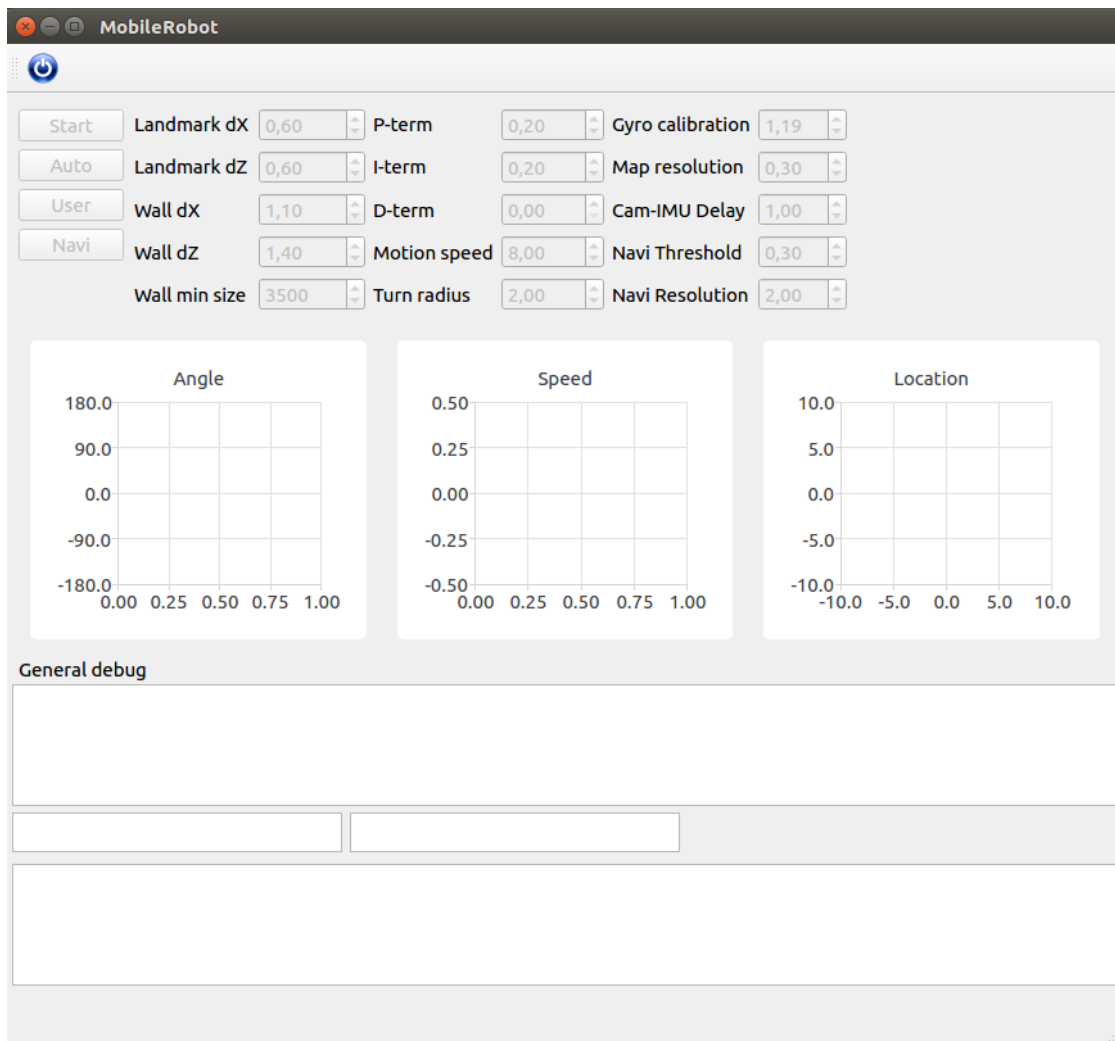
Robotin navigointi perustuu reittipisteitä sekä seinäpintoja sisältäviin karttoihin. Robotin pääsuunnistus tapahtuu reittipisteiden pohjalta. Seinäpintoja sisältävällä kartalla robotin tulisi kyetä päivittämään omaa sijaintiaan ja näin ollen tarkentamaan paikkaansa.

Tässä työssä seinien tutkimiseen käytetään vain oikealle kuvaavaa kameraa. Eteenpäin kuvaava kamera toimii törmäyksen estona. Kamera kuvaa robotin y-akselin suunnassa, joten yhdellä havaitulla kohteella voidaan päivittää sijaintia yleensä x-akselin suunnassa. Mikäli robotti kohtaisi 90° kulman, pystyttäisiin päivittämään sijaintia ensin x-akselin suunnassa ja käännöksen jälkeen y-akselin suunnassa maailman koordinaatteihin nähden.

Robotti saa reittipisteitä sisältävältä kartalta x- ja y-koordinaatin. Nämä koordinaatit käännetään maailman koordinaateista robotin koordinaatistoon, missä y-akseli osoittaa aina kulkusuuntaan. Kun tiedetään robotin sijainti sekä seuraava reittipiste, voidaan laskea suuntakulma, mihin robotin täytyy lähteä etenemään. Jos robotti ajaa pisteen ohi, voi robotti alkaa kiertää ratapistettä ympäri yrittäessään tavoittaa sitä. Tätä varten joudutaan antamaan alue, minkä sisällä voidaan sanoa, että reittipiste on saavutettu ja päästään käsittelemään seuraavaa reittipistettä.

5.3 Käyttöliittymän päivittäminen

Käyttöliittymä sisältää paljon hyödyllistä tietoa robotin toiminnasta. Tiedot ovat kuitenkin jäänteitä aiemmasta työstä. Käyttöliittymästä poistetaan suurin osa tähän työhön tarpeetonta tietoa ja tilalle lisätään viritysparametreja sekä kulmaa, nopeutta ja paikka ilmaisevia kuvaajia.



KUVA 19. Mobiilirobotin päivitetty käyttöliittymä.

6 TULOKSET JA POHDINTA

Testi ajetaan kuvan 15 osoittamalla mittausjärjestelyllä ja robottia ohjataan manuaalisesti ensimmäinen kierros. Reaaliaikaisen sijainnin laskemisesta robotti suoriutuu erinomaisesti kuten myös seinäpintojen paikoittamisesta kartalle. Robotti laskee kuljetun reitin perusteella ratapisteet ja navigoi itsenäisesti reitin uudelleen.

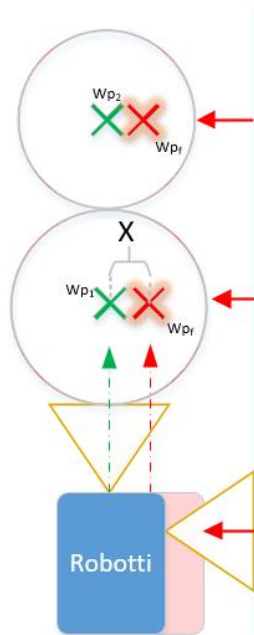
Itsenäinen navigointi perustuu tunnettuun reittiin ja tämä reitti sisältää laskettuja ratapisteitä W_p (kuva 20). Robotti navigoi laskemalla oman sijaintinsa ja ratapisteen välisen kulman ja lähtee navigoimaan ratapistettä kohti. Kun robotti saavuttaa ratapisteen, se valitsee seuraavan ratapisteen.

Robotin sijainnin päivittäminen seinäpintojen avulla olisi tarkoitus ottaa käyttöön tulevaisuudessa. Robotin anturointi on jatkuvaa, ja seinän mittaussignaalissa esiintyy mittauskohinaa. Mittauskohinan takia robotin paikanlaskenta heittelee, sillä robotin reaaliaikainen paikanlaskenta perustuu tähän mittaukseen. Kuvassa 20 näkyy, kuinka paikanlaskenta vaikuttaa ratapisteen sijaintiin: Seinän pinta-normaalin suuntainen paikanlaskennan korjaus aiheuttaa siirtymää kulkusuuntaan nähden poikittaisessa suunnassa. Mitä suurempi on mittauskohina, sitä suurempi on siirtymä. Jos reitin tavoitepiste tulee lähelle robotin paikkaa, tästä aiheutuu suuri etenemiskulman korjaus ja vastaavasti suuri suuntaa korjaava kulmanopeus. Robotin kulmanopeuden asetusarvoa lasketaan kaavalla 13 (4, s. 18). Jos siis robotti olisi y-akselin suunnassa lähellä ratapistettä ja mittauskohina aiheuttaisi x-akselin suunnassa siirtymää, aiheuttaisi tämä robotissa voimakkaita kiertoliikkeitä.

$$\omega = \frac{v \cdot \tan^{-1}(x/y)}{y},$$

KAAVA 13

missä ω on kulmanopeus ja v on nopeus.



KUVA 20. Robotin ratapisteet vääristyvät mittauskohinasta. Todellinen reittipiste on vihreä, ja vääristynyt on merkitty punaisella ristillä. Punainen ympyrä kuvaa kynnyksaluetta.

Robotin sijainnin päivittämistä seinäpintojen avulla ei käytetty näissä testeissä. Tämä ominaisuus vähentäisi robotin harhautumista, sillä omaa sijaintia voidaan korjata nähdyn seinäpinnan avulla. Robotin seinäpintoja kuvaava Kinect-kamera näkee robotin oikealle puolelle. Näin ollen robotin sijaintitietoa voidaan päivittää x-akselin suunnassa nähdyn seinän perusteella.

7 YHTEENVETO

Tämän opinnäytetyön tavoitteena oli kehittää Probot Oy:n valmistamalle mobiili-robotialustalle SLAM-navigointimenetelmää ja etäohjaus. Työssä käytettiin hyväksi konenäköä havaitsemaan ympäristöstä kohteita. IMU-anturin mittauksen ja ohjausarvofunktion avulla pystyttiin laskemaan robotin sijaintia tarkasti. Nämä kaksi osaa liitettiin toisiinsa. Lopputuloksena saatiin navigointijärjestelmä, jolla roboti kykenee seuraamaan reittipisteitä muististaan sekä piirtämään koordinaatitokarttaan reittinsä ja havaitut seinäpinnat. Etäohjaus on rakennettu WLAN-verkkoa hyväksikäyttäen ja robotin koko tietoliikenne siirrettiin langattomaan verkkoon.

Työ antoi tekijälle erinomaisen mahdollisuuden päästä syvällisemmin käsiksi tutkimaan robotiikkaa ja sen kehitystä. Tekijälle kertyi kokemusta Qt Creatorin käytöstä ja C++-kielen ohjelmoinnista.

Mobiilialustaiselle robotille on tehty pidemmän aikaa kehitystyötä, ja tämä työ on osa sitä. Työ tulee jatkumaan tulevaisuudessa. Robotin etäohjaus toimii nuolinäppäimillä, ja tämä tullaan korvaamaan Novint Falconin valmistamalla 3D-ohjaimella. Samaan ohjaukseen otetaan käyttöön myös robotin päällä oleva robottikäsi.

LÄHTEET

1. Tietoa meistä. 2018. VTT. Saatavissa: <https://www.vtt.fi/tietoa-meist%C3%A4>. Hakupäivä 19.11.2018.
2. Pulkkinen, Jussi 2016. Mobiilirobotin koordinaattiohjaus. Saatavissa: <http://urn.fi/URN:NBN:fi:amk-2016110315720>. Hakupäivä 19.11.2018
3. Oksanen, Taavi 2017. Mobiilirobotin navigointimenetelmä. Saatavissa: <http://urn.fi/URN:NBN:fi:amk-2017052910821>. Hakupäivä 19.11.2018
4. Kotaniemi, Jarkko 2018. Mobiilirobotin navigointimenetelmän parantaminen odometrian avulla. Saatavissa: <http://urn.fi/URN:NBN:fi:amk-201802122312>. Hakupäivä 19.11.2018.
5. Woodford, Chris 2018. Remote control. Saatavissa: <https://www.explain-thatstuff.com/remotecomtrol.html>. Hakupäivä 28.11.2018
6. Durrant-Whyte, Hugh – Bailey, Tim 2006. Simultaneous localization and mapping: part 1. Saatavissa: <https://ieeexplore.ieee.org/document/1638022/authors#authors>. Hakupäivä 19.11.2018.
7. Siciliano, Bruno – Khatib, Oussama 2008. Springer Handbook of Robotics. Berlin Heidelberg: Springer-Verlag.
8. Definition of 'robot'. Oxford English Dictionary
9. About Qt. 2018. Qt Wiki. Saatavissa: https://wiki.qt.io/About_Qt. Hakupäivä 19.11.2018.
10. LSM9DS0. 2013. St Datasheet. Saatavissa: <https://cdn-shop.adafruit.com/datasheets/LSM9DS0.pdf>. Hakupäivä 19.11.2018.