

Konsta Hölttä

Liikkuvien robottien paikallinen esteenväistö ja liikkeensuunnittelu

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Kandidaatintyö
Espoo 9.12.2010

Vastuupettaja:

TkT Ilkka Seilonen

Työn ohjaaja:

TkT Jari Saarinen



Aalto-yliopisto
Teknillinen korkeakoulu

Tekijä: Konsta Hölttä

Työn nimi: Liikkuvien robottien paikallinen esteenväistö ja liikkeensuunnittelu

Päivämäärä: 9.12.2010

Kieli: Suomi

Sivumäärä: 3+24

Tutkinto-ohjelma: Automaatio- ja systeemitekniikka

Vastuupettaja: TkT Ilkka Seilonen

Ohjaaja: TkT Jari Saarinen

Liikkuviin robotteihin ja niihin liittyvään esteenväistöön kuuluu laaja-alaisuutensa vuoksi suuri joukko asioita esteiden havainnoinnista sopivan reitin ennustukseen. Tässä työssä kartoitetaan reaaliaikaisia menetelmiä liikkuvien robottien lähellä olevien esteiden kiertämiseen ja siitä aiheutuvaan lyhyen kantaman liikkeensuunnitteluun. Tämä suoritetaan mallintamalla esteet lähiympäristön karttaan, lasquemalla kartan pohjalta uusi, hetkellisesti turvallinen reitti ja suunnittelemalla tämän reitin toteuttava säätö robotin liikettä ohjaaville laitteille.

Työn pääpaino on yhtäkkiä ilmaantuvissa esteissä ja ympäristössä, josta ei juurikaan tiedetä mitään etukäteen. Ensin luodaan katsaus siihen, mitä liikkuvat robotit ovat ja mitä ne tarvitsevat rakenteeltaan ja antureiltaan väistääkseen esteitä. Käydään läpi muutamia yleisiä lokaaleja esteenväistöstrategioita ja niiden soveltuvuutta eri ympäristöihin: tarkoituksena on päästä määränpäähän mahdollisimman optimaalisella tavalla osumatta reitillä oleviin maaston haastavuuksiin ja stationaarisiiin ja liikkuviin esteisiin.

Lopuksi luodaan myös väljempi katsaus aiheen kokonaiskuvaan ja esitellään muutama useita menetelmiä yhdistävä ulkotiloissa liikkuva sovellus. Pohditaan esitettyjen menetelmien käyttökelpoisuutta ja robustisuutta eri näkökulmista ja esitetään haastavasta maastosta johtuvia käytännön ongelmia.

Avainsanat: liikkuva robotti, robotin paikannus, lokaali liikkeensuunnittelu, esteenväistö, dynaaminen ympäristö

Sisältö

Tiivistelmä	ii
Sisällysluettelo	iii
1 Johdanto	1
2 Lokaalin esteenväistön ongelma	2
2.1 Mekaniikka	2
2.2 Paikannus ja anturointi	2
2.3 Lokaali esteenväistö	3
3 Esteenväistön menetelmiä	4
3.1 Alkuolettamuksia	4
3.2 Varmuusruudukko	5
3.3 Ohjauskulman huomiointi	6
3.4 Potentiaalikenttä	7
3.5 Histogrammit	10
3.5.1 Parannuksia	12
3.6 Liikkuvat esteet	13
3.6.1 Havainnointi	14
3.6.2 Nopeuseste	14
3.7 Muuta	16
4 Ulkotilan sovelluksia	16
5 Yhteenveto	18
Viitteet	20

1 Johdanto

Esteenväistö ja liikkeensuunnittelu ovat klassisia ongelmia joihin on oleellisesti kiinnitettävä huomiota aina itsenäisesti liikkuvaa robottia suunnitellessa. Robotin on osattava navigoida itsenäisesti ja sen on päästävä perille nopeasti vahingoittamatta itseään tai ympäristöään. Robotilla on tietty konfiguraatioavaruus (configuration space), eli joukko mahdollisista sijaintien, asentojen ja nopeuksien yhdistelmistä eli periaatteessa kaikesta vapaasta tilasta; esteet ovat konfiguraatioavaruuden ulkopuolella. Lokaaliin anturipohjaiseen esteenväistöön on monia eri menetelmiä, joiden hyödyllisyys ja soveltuvuus riippuu robotin ympäristöstä ja anturien laadusta [1–3].

Ympäristöä havainnoidaan jatkuvasti robotin antureilla, ja kun esteitä aistitaan, pyritään laskemaan väistämissuunnitelma mahdollisimman nopeasti. Jotkin väistötavat soveltuvat erityisesti tiettyihin antureihin ja reaaliaikaisuuteen, ja anturien hyvät ja huonot puolet on muutenkin otettava huomioon mm. havaitsemisetaisyysden, kohinan ja epävarmuuksien takia. Toisaalta havainnoinnin ja esteiden tulkinnan väliin voi luoda abstraktiokerroksen, jonka avulla mitatut havainnoinnit esitetään [4]. Havainnointimenetelmiä ovat mm. ultraääni- ja infrapuna-anturit, lasertoimiset etäisyyskannerit ja videokamerat. Hyvillä menetelmillä anturointi onnistuu jatkuvasti; toisilla robotti joutuu välillä pysähtymään mittauksia varten. Pyritään siihen, että robotti voi ajaa jatkuvasti pysähtymättä.

Esteiden ohi selviämisen lisäksi robotin on pysyttävä reitillään ja sen on jotenkin saatava tietoa sijainnistaan ja suunnastaan, mikäli tavoitteena on päästä tavoitepisteeseen eikä harhailla sokeasti. Tämänkin toteutus riippuu robotin sijaintiantureista ja niiden sopivuudesta robotin ajoympäristöön. Yleisesti soveltuvia tapoja paikannukseen ovat odometria, maamerkit, GPS ja radiomajakat. Näkyviä maanmuotoja voidaan myös verrata karttaan robotin paikoittamiseksi [5]. Kaikki tavat ovat jossain määrin epätarkkoja, joten virhearviot on otettava huomioon ja mielellään käytettävä montaa tapaa rinnakkain. Kalman-suodin onkin suosittu apu reitin ja sijainnin tarkastelun yhteydessä [6].

Tässä työssä kartoitetaan yleisellä tasolla eri esteenväistömenetelmiä ulkotiloissa maanpinnalla liikkuville roboteille. Kiinnitetään erityisesti huomiota vaihtelevassa eli dynaamisessa ympäristössä kulkeviin robotteihin. Tässä yhteydessä dynaamisella ympäristöllä tarkoitetaan sellaista, jossa esteet ovat vaihtelevia ja voivat liikkua, ja robotilla jotakin itsenäisesti liikkuvaa ajoneuvoa, kuten vihivaunua tai automaattisesti ajavaa henkilöautoa. Kiinteästi asennettavia robotteja kuten manipulaattorikäsivarsia ei käsitellä.

Työssä sivutaan myös globaalia liikkeensuunnittelua, eli tilannetta jossa reitin suunnittelukartta on tiedossa etukäteen, ja sen yhdistämistä lokaaliin väistöön. Painopiste on kuitenkin yhtäkkisten, odottamattomien esteiden havaitsemisessa ja väistämisessä kun jonkinlainen tavoitepiste on tiedossa ennalta. Reitin ja ympäristön muutoksia voidaan myös päivittää robotille reaaliajassa ylemmältä taholta. Aina ei kuitenkaan voida saada ennakkotietoa alueista, joissa robotti kulkee; tällöin alueet on kartoitettava matkan varrella. Globaalin reitin suunnittelu oletetaan hoidetuksi; ei oteta kantaa, tuleeko reitti jostain ylempää vai lasketaanko sitäkin reaaliajassa robotilla.

2 Lokaalin esteenväistön ongelma

Robotin paikallisen navigoinnin pohjalla on hyvin paljon muitakin väistöalgoritmeihin liittymättömiä ongelmia, joita tässä tekstissä sivutaan vain sen verran kuin on välttämättä tarpeen. Seuraavassa esitellään pohjustukseksi joitain perusasioita pyörillä kulkevan robotin liikkumisesta, joihin kaikki myöhemmin esitettävä teksti pohjautuu.

2.1 Mekaniikka

Tässä työssä robotti on jokin autonomisesti liikkuva väline jolla on tietty määränpää. Ympäristö jossa robotti ajaa määrää pitkälti robotin mekaanisen rakenteen: rakenne on yksinkertaistettuna esimerkiksi pyöreäalustainen joka suuntaan kääntyvä ajoneuvo, henkilöauto, maasturi tai metsäkone. Ympäristö ja rakenne puolestaan määräävät, miten robottiin voi kiinnittää anturit havainnoimaan ympäristöä. Nämä kaikki yhdessä tuovat rajoitteet, joiden perusteella lokaali liikkeensuunnittelu ratkaistaan.

Pyörät ovat yksinkertaisen mutta toimivan tekniikkansa vuoksi ylivoimaisesti useimpien nykyrobottien liikkumisen perustana [7]. Haastavassa maastossa käytetään jotain pyöristä jalostettua tapaa, kuten telaketjuja tai renkaita, joihin on kiinnitetty piikkejä tms. [8, 9]. Rakenteen puolesta (mm. renkaiden lukumäärä ja kääntymistapa) pyörillä liikkumiseenkin on hyvin monia tapoja, joissa on kaikissa omat hyvät ja huonot puolensa. Tapoja rajoittaa enimmäkseen robotin muoto ja kohdeympäristö; jotkut anturoinnit ja väistöalgoritmit soveltuvat vain tietynlaisille kinematiikoille eli robotin rakenteelle. Erityisesti joka suuntaan ohjautuvan robotin (omnidirectional robot) käsittely eroaa merkittävästi perinteisen auton kaltaisesta kulkuvälineestä, jonka kiihtyvyyssuunnat rajoittuvat eturenkaiden mukaan [10].

Nivelletyillä jaloilla käveleviä robotteja tutkitaan nykyään yhä enemmän. Humanoidirobotit soveltuvat ihmisille tarkoitettuun ympäristöön, ja näyttäisivät ihmisille luontevilta jolloin kanssakäyminen olisi helppoa. Nivelletyt jalat saattavat sopia haastavaan, epätasaiseen maastoon pyöreitä renkaita paremmin. Tätä on kokeiltu esimerkiksi metsäkoneissa [11].

2.2 Paikannus ja anturointi

Oleellinen ongelma liikkuville roboteille on itsensä paikannus [12]. Kun robotin oletetaan tietävän kohdepiste tarkasti, pitää myös tietää missä itse robotti sijaitsee. Tähän voidaan käyttää tapauksesta riippuen esim. odometriaa [13], inertiasuunnistusta [14], radiomajakoita [5] tai maamerkkejä (kamerakuvalla tai maanpinnasta ”kosketuksella” havaittavia) [2, 15].

Maamerkkejä kannattaa suhteuttaa myös ennalta tunnettuun karttaan, mikäli sellainen on saatavilla – tällöin voidaan tarkentaa paikkatietoa löydettyistä maamerkeistä ja niiden oletetuista sijainneista; myös mikäli havaittu ympäristö ei vastaa kartasta tulkittavaa oletettua näkymää, on havaittu este [16, 17]. Sensorifuusiota Kalman-suotimella voidaan myös hyödyntää, jolloin saadaan tarkempi tulos monen

anturin yhdistetystä mittauksesta, kun kullakin mittauksella on omat vahvat puolensa – esimerkiksi odometrian antama nopeustieto on luotettavaa mutta paikkatieto kerää virhettä pikkuhiljaa; GPS:ltä taas saadaan suhteellisen tarkkoja tietoja paikasta mutta sen nopeustieto taas on onnetonta [5, 6, 13]. Usein uutta karttatietoa kerätään samalla kun antureilta saadaan dataa (simultaneous localization and mapping, SLAM) [18, 19].

Esteet on havainnoitava robotin lähistöltä jollain lailla. Infrapunaheijastuksiin ja ultraäänikaikuun perustuvat sensorit ovat yksinkertaisuutensa ja edullisuutensa ansiosta hyvin yleisesti erityisesti sisätiloissa käytettyjä, mutta myös häiriöalttiita ja epävarmoja [20, 21]. Ne toimivat tutkamaisesti lähettämällä signaalin ja seuraamalla kaikua esteestä, johon siten arvioidaan etäisyys kaiun matkaan kuluneen ajan perusteella. Näitä antureita on tyypillisesti useita robotin ympärillä osoittamassa eri suuntiin.

Laserskannerit ja kamerat ovat tarkkoja ja näkevät kauas: niitä käytetään monesti ulkotiloissa [3]. Kaksiulotteinen laserskanneri on myös tutkamainen ja kuvaa automaattisesti joka suuntaa 360 asteen keilalla; kolmiulotteinen laserskanneri kuvaa eteenpäin kahdella akselilla ja tuottaa kamerakuvan kaltaisen kuvan syvyyksistä [22, 23]. Vielä tarkempaa kuvaa saadaan videokameroilla, joiden kanssa tarvitaan laskentatehoakin huomattavasti edellä mainittuja enemmän. Konenäkö on ylivoimainen, kun tarvitaan tarkempaa tietoa edessä olevista pinnoista, kuten kuviointia tai asentoa. Kahdella oikein sijoitellulla kameralla saadaan aikaan syvyysnäkö [24].

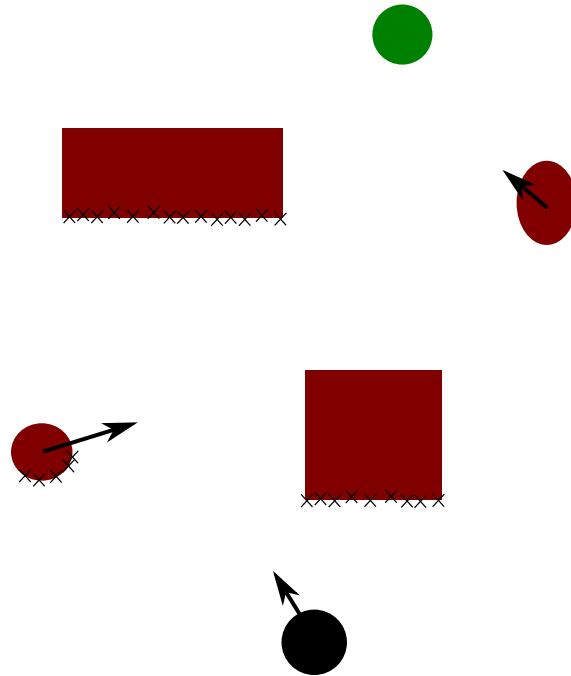
Itse esteiden väistämisen aiheuttaman reitin lokaali suunnittelu on puhtaan laskennallista, mutta pohjalla on kuitenkin oikea ulkomaailma epätarkkuuksineen ja anturivirheineen. Jotkin menetelmät perustuvat hyvinkin vahvasti tietynlaiseen anturointitapaan, kuten siihen, että ultraääneltä saadaan suhteellisen epävarmoja mittauksia [21]. Virheet ja epävarmuudet on pidettävä aina mielessä.

2.3 Lokaali esteenväistö

Kuvassa 1 on yksinkertaistettu esimerkki tyypillisestä paikallisen esteenväistön ongelmasta. Robottia kuvaa alhaalla oleva ympyrä, määränpäättä ylhäällä oleva ympyrä ja loput kuviot ovat esteitä. Robotti havainnoi edessä päin olevien esteiden reunat näkökentässään: rastit kuvaavat tutkaamalla havaittuja estepisteitä. Nuolet kuvaavat liikkuvien objektien nopeusvektoreita. Kuvan abstraktoitu tilanne voi vastata toimiston käytäviä tai kaupungin liikennettä.

Kuvan tilanteessa robotin on suotuisinta jatkaa matkaa keskellä olevan neliön ohitse, sillä se ei ehdi jarruttaa ja kiertää oikealta. Seuraavaksi täytyy hidastaa, jotta vasemmalta tuleva este pääsee ohitse (mikäli se jatkaa matkaa robotin ohi, eikä käännä suuntaansa nykyisestä). Lopulta kuljetaan nelikulmaisten esteiden välistä päätepistettä päin ja saatetaan joutua väistämään ylempänä olevaa liikkuvaa kappaletta. Tässä työssä pyritään saamaan vastaavanlainen ohjauslogiikka aikaiseksi.

Esteenväistön ensimmäinen askel on esteiden sijainnin etsiminen. Tästä eteenpäin oletetaan, että esteet ja muu lähiympäristön informaatio sekä robotin sijainti suhteessa referenssikoordinaatistoon on havaittu jollain menetelmällä, ja tietyllä varmuudella tiedetään ympäristön muoto ja mahdollisesti reitillä olevat esteet – havain-



Kuva 1: Tavanomainen esteenväistötilanne.

nointimenetelmään sen enempää ottamatta kantaa mitä siitä välttämättä tarvitsee huomioida. Liikkumismenetelmäkin oletetaan hoidetuksi sekä reitti annetuksi ylemmältä taholta. Seuraavassa käsitellään itse esteiden tulkitsemista ja ohittamista.

3 Esteenväistön menetelmiä

3.1 Alkueleettamuksia

Sisätila eroaa ulkotilasta esteenväistön kannalta merkittävästi siinä, että tilasta voidaan varmuudella tehdä tiettyjä yksinkertaistavia oletuksia. Robotti on oletettavasti aina samassa kulmassa maanpintaan nähden (tasainen lattia), esteet ovat mitä todennäköisimmin korkeahkoja robottiin nähden ja kohtisuorassa maanpintaan nähden (seinät, hyllyt, ihmiset) ja rakennuksen pohjakuvasta saataneen hyvä kartta, joka voidaan approksimoida kaksiulotteiseksi. Renkailla liikkuminen on helppoa ja yhtäkkisiin kuoppiin tai kynnyksiin ei tarvitse kiinnittää erityisemmin huomiota. Näkymää voi tutkaskannata kohtuullisen luotettavasti jo ultraäänellä. Esteet tuskin liikkuvat ihmistä nopeammin. Mitkään näistä oletuksista eivät päde ulkona tutkimattomassa ympäristössä.

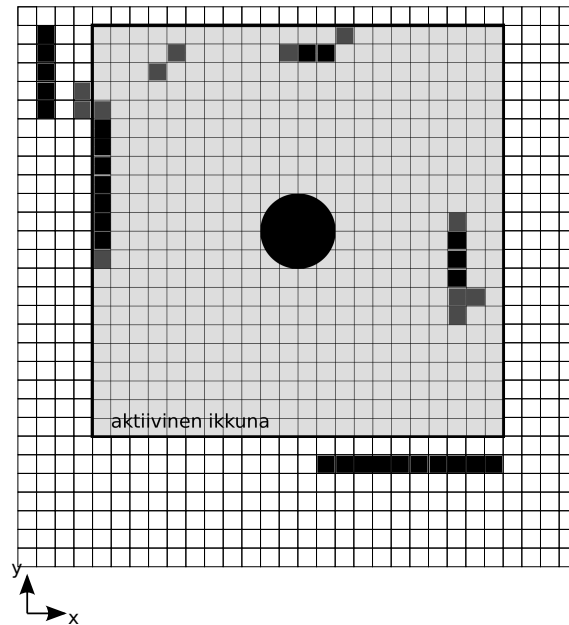
Ulkona robotin mekaniikkaan, esteenväistöön ja -havainnointiin on kiinnitettävä enemmän huomiota. Ulkotilat voidaan erotella edelleen mm. kaupunkimaastoon, työmaahan, peltoon, avolouhoksiin ja kaivoksiin, metsäkaltaiseen maastoajoon jne. Kaupunkimaasto on kompromissi sisätilojen ja erityisen haastavan ympäristön välillä, ja sen teknisiä haasteita ovat erityisesti vinot ja käyrät pinnat joissa robotti lipsuu ja joissa kiinteät anturit voivat ottaa häiriömittauksia maanpinnasta mä-

keä alas ajettaessa; ylös ajettaessa reunan yli ei nähdä ennen kuin maasto on taas tasaista. Liikkuvat esteet voivat olla autoja tai polkupyöriä, joita pitäisi väistää hyvin nopeasti ja jotka täytyisi havainnoida jo kaukaa, jotta väistöliikkeen ehtisi tehdä. Tämä on erityisen kriittistä varsinkin robotisoiduissa henkilöautoissa [25]. Täysin raaka vaikeasti kartoitettava epätasainen maasto taas on edelleen hankalimman pään erikoistapaus, kun maanpinta on avohakkuun kaltainen jossa pitää vain mönkiä eteenpäin ja väistää yhtäkkisiä maastonmuotoja.

Tasaisessa maastossa ”kaksiulotteisessa” maailmassa esteet on aina kierrettävä sivulta ja mittauksia on helppo tehdä tutka-antureilla, kun vinoja maastoja ei ole. Seuraavassa käsitellään esteenväistöä tällaisessa maailmassa, sillä se yksinkertaistaa tilannetta huomattavasti – kaikki tilanteet kannattaa approksimoida kaksiulotteiseksi mikäli vain mahdollista. Monesti ulkotilatkin voidaan vinoudestaan huolimatta käsitellä kaksiulotteisena, ellei robotti lennä tai sukella. Jos edessä on hyvin matalakin este, se kannattanee kiertää elleivät renkaat joustu erityisen paljon.

3.2 Varmuusruudukko

Varmuusruudukko tai varausruudukko (certainty grid, occupancy grid) on yksi tapa esittää ympäristön esteet ja toimii monien väistömenetelmien pohjana [4]. Sen ideana on nimensä mukaisesti rakentaa ruudukko robotin globaalista työavaruudesta, johon päivitetään anturien tietoa siitä, millä varmuudella ruudussa on este. Kun etäisyysanturi sanoo esteen olevan tietyllä etäisyydellä, voidaan tietyllä todennäköisyydellä sanoa, että siinä kohdassa on este, ja lisäksi robotin ja esteen välissä on tyhjää tilaa. Tällaista tietoa päivitetään robotin maailmankarttaan antureilta jatkuvasti.



Kuva 2: Esimerkki varmuusruudukosta aktiivisella ikkunalla.

Kuvassa 2 on kuviteltu esimerkkitilanne robotin ympäristön kuvaamisesta var-

muusruudukossa. Vasempaan alanurkkaan on merkitty globaali koordinaatisto; ruudukko on aina samassa, etukäteen sovitussa asennossa siten, että x-akseli osoittaa esim. itään tai sinne minne robotti suuntasi alkuhetkellä. Tummemmat arvot ruuduissa kuvastavat varmempaa estettä eli sellaista, jossa ruutuun on osunut paljon mittauksia. Aktiivinen ikkuna -laatikko (active window) kuvaa ”ikkunaa” koko ruudukkoalueesta johon tällä hetkellä kiinnitetään huomiota: robotin ei tarvitse ottaa huomioon koko varmuusarvoruudukon maailmaa vaan ainoastaan pieni alue robotin ympäriltä jonka sisältä laskettavat voimat ovat merkittäviä suunnan suhteen – täten laskennan haastavuus ei kasva sitä mukaa kun saadaan lisää havaintoja ympäristöstä. Ikkuna on joka hetkellä robottikeskeinen (robottia kuvaa ympyrä ikkunan keskellä) [4].

Kun robotin havaitsema maailma diskretoidaan sopivan kokoisiin ruutuihin, voi sen jatkokäsittely yksinkertaistua algoritmista riippuen hyvinkin paljon [4]. Kompromisseja tarkkuuden ja tehokkuuden välillä tehdään ruudun koossa: suurilla ruuduilla maailma approksimoituu vain muutamiksi ruuduiksi joiden läpikäynti on nopeaa, ja useammilla pienemmillä ruuduilla saadaan tarkkuutta laskentatehon kustannuksella. Ruudukko voidaan myös yhdistää globaaliin karttaan ympäristöstä, jolloin sen avulla voidaan estimoida paikkaa [26].

Usein sensorille voidaan olettaa gaussinen todennäköisyysjakauma [4]. Tällöin ruudukkoon päivitetään pistemäisestäkin mittauksesta mittauskeskeinen jakauman muotoinen kuvio sekä nollataan estetieto robotin ja mittauksen väliltä – jos tutkan kaiku selvisi tietylle etäisyydelle heijastumatta aiemmin takaisin, ei välissä ole mitään. Esteiden koostakin saatetaan tehdä jotain oletuksia, koska sensorien ilmoittamien esteiden takaa ei voida saada mitään tietoa; ulkotiloissa voidaan alustaa varmuusruudukko tietyllä epävarmuudella tyhjyyden sijaan, ja olettaa että esteet ovat pienehköjä ja niiden takana olevaa tietoa ei päivitetä mitenkään [27].

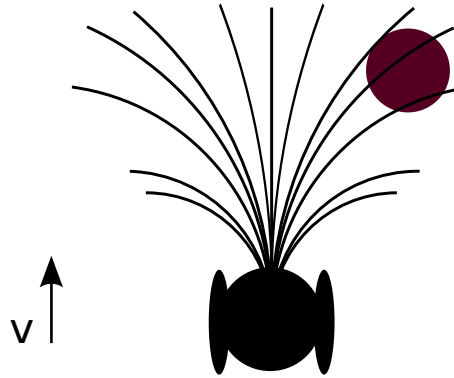
Menetelmä soveltuu erityisesti häiriöiseen ja epävarmaan mittaukseen esim. ultraääniantureille joihin liittyy lähes ominaisesti epävarmuutta, vahinkokaikuja ja vierekkäisten sensorien ristivaikutusta [28]. Kun karttaa päivitetään jatkuvasti robotin edetessä, samoista esteistä saadaan useita pistemäisiä mittauksia samoihin ruutuihin ja varmempien ruutujen todennäköisyys kasvaa. Lisäksi tyhjien ruutujen tyhjystodennäköisyys kasvaa kun niiden takaa saadaan mittauksia, mikä kompensoi liikkuvien esteiden muodostamat ”seinämät”. Jos ympäristössä on useita liikkuvia kohteita ja etenkin jos samaa karttaa käytetään tällöin globaaliin suunnitteluun, on syytä vanhentaa anturoituja mittauksia ajan kuluessa jolloin muuttuvasta ympäristöstä ei tehdä vääriä päätelmiä [29].

3.3 Ohjauskulman huomiointi

Muun muassa ohjauskulmakentäksi (steer angle field) [30] ja kaarinopeusmetodiksi (curvature-velocity method) [31] kutsuttu menetelmä ottaa huomioon robotin kinematiikasta sen, että kaarella kulkevaan robottiin – esim. differentiaaliajoa tai Ackermannin ohjausta (eli henkilöauton kinematiikkaa: kaksi kääntyvää pyörää neljästä) käyttävään [12] – ei välttämättä kannata kohdistaa kiihtyvyyttä kohtisuoraan kaarta vastaan jos haluaisi sen mahdollisimman äkkiä pois päin vieressä olevasta esteestä,

koska robotti voi vain hetkellisesti kääntyä enemmän vasemmalle tai oikealle. Kyseessä ei sinänsä ole suoranaisesti esteenväistömenetelmä, vaan tapa kartoittaa reitit joita robotti ylipäättään voi kulkea.

Kuvassa 3 esitetään eri kaaria, joita differentiaaliajoinen eli kahdella erikseen ohjattavalla renkaalla käänneltävä robotti pystyy kulkemaan, kun sillä on tietty vauhti josta ei ehditä jarruttaa tai kääntyä aivan toiseen suuntaan. Kaaria on periaatteessa ääretön määrä, joista kuvaan on yksinkertaistettu muutama. Tasaista vauhtia kulkemalla ko. robotti ajaa aina suoraan tai ympyrän kaarta pitkin. Suurempi nopeus myös rajoittaa kääntökulmaa, mikäli halutaan estää renkaiden lipsuminen keskeiskiihtyvyydestä: hitaammin ajamalla voi kääntyä tiukempia mutkia. Kuvan robotti ei esimerkiksi pysty lähtemään välittömästi suoraan vasemmalle, ja oikealle ajaakseen sen olisi tehtävä joko hyvin loiva tai jyrkkä mutka jottei pyöreään esteeseen osuttaisi.



Kuva 3: Robotin ajolinjoja. Oikealla puolella on este, vasemmassa alanurkassa nopeusvektori.

3.4 Potentiaalikenttä

Potentiaalikenttä (artificial potential function, artificial potential field, APF) on vanha mutta hyvin usein (sellaisenaan tai toisen navigointitavan lähtökohtana) käytetty väistömenetelmä [32, 33]. Kentät eivät tuota robotille oikeita fysikaalisia voimia, jolloin kenttiä sanotaan toisinaan keinotekoisiksi tai virtuaalisiksi.

Potentiaalikentät syntyvät kohteesta ja esteistä; kohde tuottaa puoleensavetävän ja esteet hylkivän potentiaalin. Käytetty potentiaalfunktio on suhteellisen yksinkertainen, esimerkiksi robotin ja esteen väliseen etäisyyteen, etäisyyden käänteisarvoon tai tämän neliöön perustuva [34]. Määränpään ja esteiden aiheuttavien potentiaalien gradienttien summasta saadaan robottiin kohdistuva ”voima” jonka mukaan sitä kiihdytetään. Potentiaalikenttiä käytetään myös robottimanipulaattoreilla, mutta niitä ei tässä käsitellä. Intuitiivisesti potentiaalikenttiä voi verrata vuoristaisen maaston korkeuskarttaan, jossa pyritään ”vierimään” mahdollisimman alas pohjalle, määränpäähän.

Formaalisti potentiaalfunktiot U_a (puoleensavetävä, attraktiivinen) ja U_r (hyl-

kivä, repulsiivinen) ovat esimerkiksi [34]

$$U_a(q) = \xi(q_m - q), \quad (1)$$

$$U_r(q) = \begin{cases} \eta(\frac{1}{|q_e - q|} - \frac{1}{\rho_0})^2 & \text{jos } |q_e - q| \leq \rho_0 \\ 0 & \text{jos } |q_e - q| > \rho_0 \end{cases} \quad (2)$$

missä q -vektori kuvaa robotin sijaintia, q_m kohteen sijaintia ja q_e yhden esteen sijaintia. ξ ja η ovat optimoitavia skaalauskerroimia ja ρ_0 on suurin käytännöllinen etäisyys, jossa este havaitaan tai jossa se tuottaa merkittävää voimaa. Potentiaalfunktion $U(q)$ aiheuttama voima F saadaan yleisesti potentiaalitentäfunktion negatiivisesta gradientista paikan suhteen

$$F(q) = -\nabla_q U(q), \quad (3)$$

oli potentiaalfunktio mikä tahansa (kuten kaava 1 tai 2, tai jokin muu), sillä halutaan päästä “korkeuskarttaa” pitkin alaspäin. Kokonaisvoima F_{kok} on kohteen ja kaikkien esteiden aiheuttamien voimien summa, kun esteitä on N kpl. F_a on laskettu kaavalla 3 esim. kaavasta 1, ja F_{ri} on järjestysluvultaan i :nnen esteen repulsiivinen voima (myös kaavalla 3):

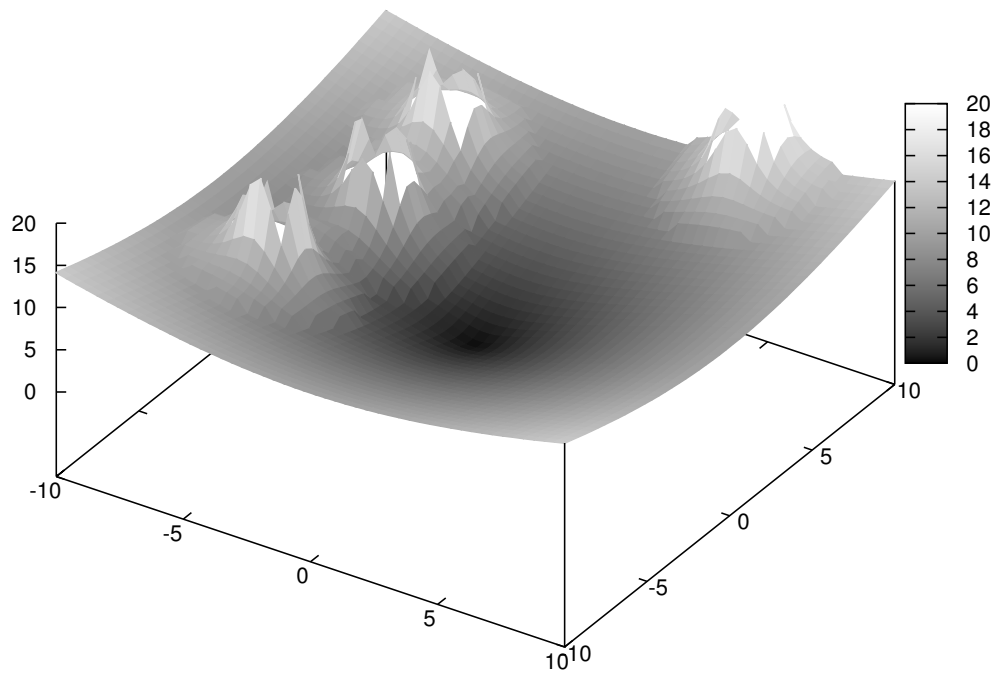
$$F_{kok}(q) = F_a(q) + \sum_{i=1}^N F_{ri}(q). \quad (4)$$

Kuvassa 4 on esimerkki potentiaalfunktion arvojen suuruuksista erittäin yksinkertaisessa ympäristössä. Kohde on origossa ja siellä on siis matalin potentiaali. Neljä huippua ovat pistemäisiä esteitä joista kolmen ryppäänä olevan takana on yksi lokaali minimi, johon voi jäädä loukkuun ilman muita menetelmiä. Käytetyt potentiaalfunktiot ovat kaavat 1 ja 2 arvoilla $\xi = 1, \eta = 10, \rho_0 = 3$. Todellisuudessa esteitä lienee joka puolella ja monen pistemäisen estemittauksen yhdistelmä tuottaa viivamaisia kuvioita seinien yms. kohdille.

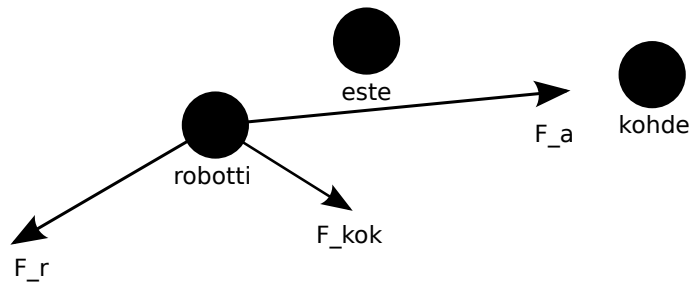
Kuvassa 5 visualisoidaan, kuinka voimat toteutuvat kohteen ja yhden estepisteen funktiona kaavalla 4. Voimavektorit F_a ja F_r lasketaan potentiaalfunktioista ja summataan yhteen, jolloin syntyy kokonaisvoima F_{kok} . Kokonaisvoima tuottaa hetkellisen kiihtyvyyden robotille kunnes seuraavat voimavektorit on laskettu seuraavalla laskukierroksella.

Koska voima voidaan laskea hetkellisesti joka pisteessä ja potentiaalfunktioiden laskeminen on yksinkertaista, tämä menetelmä soveltuu dynaamiseen ja reaaliaikaiseen tilanteeseen. Etäisyys esteeseen saadaan esim. ultraäänianturilta tai varausruudukosta, ja esteiden liikkua repulsiivinen kokonaisvoima muuttuu siten, että robotti ohjaa esteistä pois päin jatkuvasti. Menetelmä ei ota kantaa esteiden liikkumiseen vaan yksinkertaisesti määrää robottia hylkimään esteitä silloin kun ne ovat havaintosäteellä.

Eri painokertoimet ξ , η ja ρ_0 on optimoitava robotin kinematiikan ja esteiden mukaan [34]; robotin liike on jatkuvaa eikä se voi välttämättä kääntyä välittömästi



Kuva 4: Yksinkertaistettu esimerkkikuva potentiaalikartasta. Leveys- ja syvyysakselit vastaavat paikka-akseleita x ja y , ja korkeus on potentiaalifunktioiden summan suuruus.



Kuva 5: Kohdepisteen ja yhden esteen aiheuttamat voimavektorit.

mihin suuntaan tahansa. Sillä on myös massasta aiheutuva hitaus sekä jokin maksiminopeus, joista täytyy optimoida kenttien voimakkuus siten, ettei esteitä kierretä turhaan hyvin kaukaa, mutta samalla siten, ettei niihin voi törmätä. Tässäkin on tehtävä myös oletuksia esteiden liikkumisnopeudesta ja robotin reaktioajasta, jos se on relevanttia – potentiaalikentät eivät ehdi apuun nokkakolareihin jos hylkimisvoima ei ole tarpeeksi suuri. Painoparametrien optimoinniksi on esitetty etukäteen suunniteltavan hyvyysfunktion minimointia [35], mikä sopii ennalta laskettavaksi muttei kelpaa muuttuvaan ennalta tuntemattomaan ympäristöön tekemättä alkuoletuksia. Jos ympäristöä voidaan simuloida, voidaan hyvyysfunktiot kehittää esim.

laskemaan reitin pituutta tai reitin läheisyyttä esteisiin [35].

Lokaalin minimin ongelmaan on esitetty useita samaan menetelmään integroitavia ratkaisuja. Lokaali minimi on paikallinen pieni potentiaaliukuoppa, jonne jouduttuaan robotti ei pääse sieltä pois omin avuin tällä menetelmällä, koska siellä kaikki robottiin kohdistuvat voimat kumoavat toisensa ja se pysyy paikallaan. Yksi mahdollinen ratkaisu on poistumisvoima, joka otetaan käyttöön jos huomataan robotin olevan lähellä minimiä, esimerkiksi ennakkoon potentiaalikartasta tulkitsemalla [35]. Tällöin kokonaisvoimaan lisätään esteiden hylkivää vaikutusta sopivaan suuntaan.

On myös ehdotettu robotin takana samaa rataa kulkevaa virtuaalista estettä, joka aiheuttaa voiman eteenpäin ja estää nollapotentiaaliukuoppiin jumiutumisen [36]. Tässäkin on ongelmana ko. voiman suuruuden optimointi: liian pieni voima on hyödytön ja liian suuri työntää robottia turhaan väärään suuntaan ja aiheuttaa hitautta ja voi estää kohteeseen pääsemisen luullen sitä lokaaliksi minimiksi.

Koska potentiaalikentät ovat hyvin universaali ja abstrakti ohjaustapa, niihin voidaan yhdistää muita menetelmiä rinnalle; kun huomataan, että minimikuoppa on lähistöllä, globaali reitinsuunnittelija voi aiemmin anturoidun ympäristön mukaan esimerkiksi varausruudukkoa analysoimalla ohjata robottia toisen suuntaan virtuaalisella lisävoimalla tai jopa ohjata samaa polkua taaksepäin ja yrittää toista suuntaa esteiden ohi.

3.5 Histogrammit

Borenstein ja Koren [37] toteavat potentiaalikentät riittämättömiksi ja esittävät niihin ja varmuusruudukkoon pohjautuvan virtuaalisen voimakentän (virtual force field, VFF) ja siitä edelleen kehittyneen vektorikenttähistogrammin (vector field histogram, VFH) [28, 38].

Virtuaalinen voimakenttä perustuu Borensteinin ja Korenin työssä varmuusruudukkoon, jonka ruutuihin summataan arvoja robotin 24 ultraäänianturilta. Tästä ruudukosta otetaan tietyn kokoinen neliö robotin sijainnin kohdalta (kuten ikkuna kuvassa 2), jota päivitetään mittauksia saadessa ja josta lasketaan robotin suuntaa ja nopeutta ohjaava voima samankaltaisesti kun potentiaalikentissä: lähellä olevat esteet tuottavat voimakkaita hylkiviä voimia. Esteiksi lasketaan ruudut, joissa on mittauksia; mittausten määrä kuvaa esteen kokoa ja kompensoi täten myös epävarmuuksia ja häiriömittauksia. Hylkivät voimavektorit lasketaan seuraavasti [37]:

$$F_{i,j} = F_r C_{i,j} \left(\frac{W}{|c_{i,j} - r|} \right)^n (c_{i,j} - r), \quad (5)$$

missä i ja j ovat ruudun koordinaatit, $F_{i,j}$ ruudun muodostama voima, F_r repulsiivisuusvakio, $C_{i,j}$ ruudun varmuusarvo, W robotin leveys, $c_{i,j}$ ruudun paikkavektori ja r robotin paikkavektori. n on jokin kokonaisluku joka määrää hylkimisfunktion muodon. Kohdepiste vetää robottia puoleensa seuraavasti voimalla F_k kaavalla

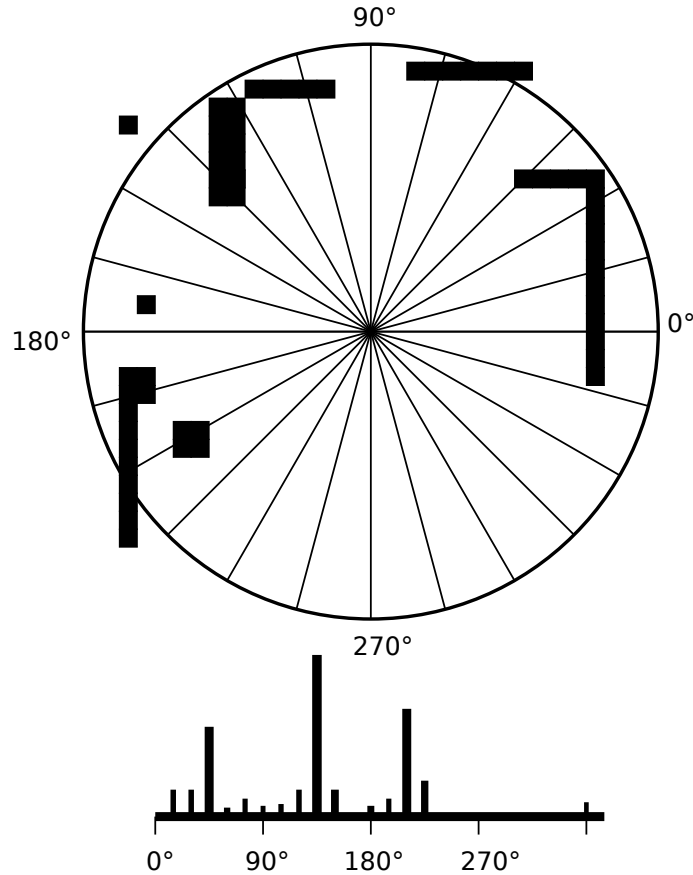
$$F_k = F_a(k - r), \quad (6)$$

missä F_a on attraktiivisuusvakio, k kohteen paikkavektori ja r robotin paikkavektori.

Robotin ohjauskulmaa säädetään seuraavasti:

$$\Omega = k(\delta - \theta), \quad (7)$$

missä Ω on ohjaukselle syötettävä kääntönopeus, k suhteuttamisvakio, δ kulma johon robottia ohjataan (saadaan voimavektorien summasta) ja θ nykyinen robotin ajokulma. Borenstein ja Koren toteavat tästä eteenpäin mm. ohjauksen värähtelemisen esteiden läheisyydessä [37].



Kuva 6: Karkea kaavakuva vektorikenttähistogrammin toiminnasta: ylempänä mielikuvituksellinen kartta ja esteet, alempana kartasta laskettu histogrammi. Ympyrän ja histogrammin kulmamerkinnot vastaavat toisiaan.

Borenstein ja Koren [28] toteavat että suoraan kaksiulotteisesta kartasta voiman tulkitseminen menettää liikaa informaatiota, ja ottavat käyttöön toisen laskentavaiheen, histogrammitaulukon. Histogrammi kattaa täyden ympyrän robotin ympärillä, ja sen joka sektorille lasketaan asteittain varmuusarvoruudukon arvot niistä ruuduista, jotka kullekin sektoreille osuvat. Histogrammiin syntyy huippuja ja laaksoja. Laaksot ilmaisevat alueita, joita on turvallista lähestyä; korkeat huiput kuvaavat paikkoja, joissa on todennäköisesti jokin lähellä oleva tai suuri este, jota on vältettävä.

Varmuusruudukon arvoista lasketaan painotetut arvot riippuen siitä, kuinka suuri arvo ruudussa on ja kuinka kaukana se on robotista [28]:

$$m_{i,j} = (c_{i,j})^2(a - b * d_{i,j}). \quad (8)$$

Kaavassa i ja j ovat ruudun koordinaatit, $m_{i,j}$ on uusi painotettu arvo, $c_{i,j}$ ruudun varmuusarvo, a ja b positiivisia skaalausvakioita ja $d_{i,j}$ etäisyys ruutuun robotin keskipisteestä. a ja b valitaan siten, että m :n arvoksi tulee 0 varmuusruudukon ikkunan kauimmaisessa ruudussa [28]. Sektorin painoarvo s_k lasketaan summana painotetuista ruutujen arvoista:

$$s_k = \sum_{i,j} m_{i,j}, \quad (9)$$

kun i, j käyvät läpi ruudut, jotka osuvat sektorille k .

Histogrammiin jaettavan sektorin epäsuotuisuus lasketaan varmuusruudukon arvojen neliöistä, jolloin varmempia mittauksia painotetaan huomattavasti enemmän, ja pienet, häiriömäiset mittaukset jäävät lähes huomiotta. Verrannollisuus $-d$:hen aiheuttaa kauempana olevien esteiden painottamisen vähemmän kuin lähellä olevien.

Iso ruudukko tuo isoja vaihteluita diskreetin luonteensa takia, jolloin tarvitaan kokeellisesti viritettävä alipäästösuodin ettei ohjaukseen tulisi suuria yhtäkkisiä vaihteluita; lopulliset painoarvot s'_k sektoreille lasketaan painotettuna keskiarvona kohdan k ympäriltä esimerkiksi seuraavasti [28]:

$$s'_k = \frac{1}{4}(s_{k-1} + 2s_k + s_{k+1}). \quad (10)$$

Kuvassa 6 havainnollistetaan karkeasti vektorikenttähistogrammin toiminta – oikeassa tilanteessa segmenttejä olisi huomattavasti enemmän; kuvan ympyrä on jaettu 15 asteen segmentteihin hahmottamissyistä. Robotti sijaitsee pistemäisenä ympyrän keskellä, ja ympyrän eri segmenttien sisältämät estemäärät lasketaan yhteen. Histogrammiin muodostuu suuria arvoja niihin kohtiin, mitä vastaavissa suunnissa on varmoja tai lähellä olevia esteitä. Ohjaus valitaan suuntaan, jossa on suurin tyhjä aukko histogrammissa eli jossa lienee varmin kuljettava reitti. [28].

3.5.1 Parannuksia

VFH ei ota huomioon robotin kinematiikkaa millään lailla, vaan olettaa että robotti on pistemäinen ja voi vaihtaa kulkusuuntaansa välittömästi. Borenstein ja Ulrich esittävät VFH+-algoritmin, joka korjaa tämän ongelman [39]. He tutkivat ihmisvoimin työnnettävää, differentiaaliajolla ohjattavaa robottia ohjaamaan sokkoa ihmistä. Tämä robotti eroaa kinematiikaltaan oleellisesti aiemmin käytetystä, jokasuuntaisesta (omnidirectional) robotista.

VFH+ tuo seuraavat muutokset: Robotti oletetaan pistemäiseksi ja esteitä vastaavasti kasvatetaan robotin koon ja muodon tuomien rajoitteiden mukaan. Histogrammikartasta tehdään ensin binäärinen kartta jolloin alipäästösuo-dinta ei tarvita [39] – suuntiin joko voi tai ei voi kulkea, riippuen siitä osuuko robotin koolla laajennettua estettä viereisistä ruuduista kyseiselle suunnalle. Tämän jälkeen karttaa

laajennetaan edelleen poistamaan ne suunnat, joihin robotti ei ehdi kääntymään kinematiikkansa takia. Nämä lasketaan tässä tilanteessa kääntösäteen avulla. Lopulta ohjaussuunnan valinnassa käytetään apuna hyvyysfunktioita, jolla voidaan painottaa erilaisia ominaisuuksia vapaista suunnista. Eri suuntia ovat eri kapeat käytävät, sekä leveistä käytävistä on valittava ajetaanko keskeltä vai läheltä määränpäättä lähempänä olevaa reunaa – voidaan painottaa joko nopeasti määränpäähän pääsemistä tai turvallista polkua. Borenstein ja Ulrich esittävät hyvyysfunktioiksi seuraavaa [39]:

$$g(c) = \mu_1 \Delta(c, k_t) + \mu_2 \Delta(c, \theta_i/\alpha) + \mu_3 \Delta(c, k_{n,i-1}), \quad (11)$$

missä c on kandidaattisuunta, $\Delta(c_1, c_2)$ on funktio kahden sektorin c_1 ja c_2 välisen kulman laskemiseen, k_t on kohteen suunta, θ_i/α kuvaa robotin renkaiden nykyistä suuntaa sekä $k_{n,i-1}$ on viimeksi valittu kulkusuunta. Kolmella painokertoimella μ_1 , μ_2 , μ_3 voidaan tehokkaasti painottaa maalihakuisuutta, pyörimisen estävää liikettä ja sileää polkua.

VFH+:sta vielä edelleen kehitetty lokaalin ja globaalin navigoinnin yhdistävä VFH* [40] käyttää A*-hakualgoritmia [41] etsimään optimaalisen valinnan saman laatuista hetkellisistä ohjaussuunnista, ja näkee täten hieman kauemmas. VFH+ itsessään on täysin lokaali, ja saattaa väistää estettä umpikujaa päin, vaikka toiseen suuntaan väistäessä ei päädyttäisi umpikujaan; VFH* laskee reittiä joka kierroksella tietyn määrän askelia eteenpäin ennalta havaitun kartan avulla eri suuntakandidaateille ja evaluoi VFH+:ssa esitetyn hyvyysfunktion reitiltä, jolloin muodostuu askelmäärän syvyinen hakupuu [39]. Puusta valitaan paras reitti saman hyvyysfunktion avulla, ja parhaaseen reittiin johtava suunta valitaan.

3.6 Liikkuvat esteet

Aiemmin esitellyt menetelmät soveltuvat liikkuvienkin esteiden väistämiseen, sillä niitä lasketaan jatkuvasti aina uudelleen uusista havainnoista. Kun esteiden sijainnit muuttuvat, myös esteenväistön tuottama ohjaus robotille muuttuu. Jos tiedetään ennalta esteiden sijaintien lisäksi niiden nopeudet, voidaan ennakoida koko ympäristön käyttäytymistä ja osittain kumota myöhemmin tapahtuvan liikettä monimutkaistavan esteenväistön tarvetta oikealla reitin valinnalla. Esimerkiksi risteystilanteessa omaa reittiä vasten kohtisuoraan sivusta tuleva este voi aiheuttaa jarruttamisen tarvetta mikäli robotti väistää sitä vasta lähelle päästyään. Ennakoiden voidaan kääntää reittiä siten, että nopeutta ei tarvitse vähentää ja päästään esteen edestä tai takaa siten, että törmäystä ei synny.

Liikkuviin esteisiin tulee kiinnittää huomiota pääasiassa kahdella eri tavalla: ensinnäkin liikkumaan kykenevän esteen ei voi olettaa olevan samassa kohdassa, mikäli kohtaan palataan myöhemmin – jos jonkinlaista karttaa ympäristöstä rakennetaan, on sensoreilta saataville karttapisteille hahmoteltava varmuusluvut jotka ikääntyvät pikkuhiljaa, jolloin epävarmat esteet unohdetaan eikä niiden pohjalta voi pitkän ajan päästä suunnitella reittiä. Toisekseen, jos robotilla on erityinen kiire tai jos haetaan esim. ihmisten lähellä hyvin toimivaa käyttäytymistä, voidaan tunnistaa liikkuvat objektit jo havainnointivaiheessa ja esim. risteystilanteessa odottaa hetki, että nopeampi sivulta päin tuleva este menisi ohi – tätä käsitellään seuraavaksi.

3.6.1 Havainnointi

Riittävän tarkoista mittauksista voidaan estimoida perättäisten mittausten eroista ympäristön muutoksia, kun robotin nopeudesta ja mittauskartasta tiedetään, millä lailla paikallaan seisovien esteiden havainnointi muuttuisi robotin liikkuesssa.

Prassler ym. [42] käyttävät varausruudukon kaltaista aikaleimakarttaa (time stamp map) havaintopisteiden varastoimiseen ja erottelevat siitä liikkuvat esteet paikallaan olevista tutkimalla kartan historiatietoa; kartan ruuduille alustetaan joka mittausta vastaavalla ajanhetkellä tieto siitä, että ruudussa oli este tällä hetkellä. Edellisten ajanhetkien ruudukoista etsitään sama ruutu kuin nykyhetken mittauksella on, ja verrataan sen lähistöllä olevia mittauksia. Riittävän lähellä oleva mittaus lasketaan samaksi stationääriseksi objektiksi; kaukana oleva taas on todennäköisimmin liikkuva kappale.

Riittää, että kartasta tunnistetaan liikkuvat esteet pistemäisinä; varausruudukon pisteet approksimoidaan tällöin ruudun kokoiseksi ympyräksi. Laskemista nopeuttaa, jos esteet tunnistetaan kokonaisina, sillä tällöin väistettäviä esteitä on vähemmän.

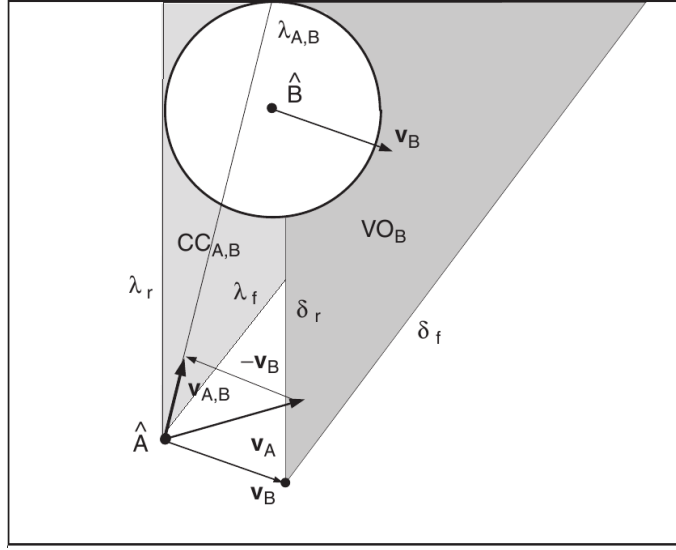
3.6.2 Nopeuseste

Chakravarthy ja Ghose [43] esittävät trigonometrisen törmäyskeilamenetelmän (collision cone) abstraktin muotoisten liikkuvien esteiden väistämiseksi. Menetelmä ottaa huomioon esteiden nopeudet ja muodostaa robotin ja esteiden nopeusvektorien keskinäisestä suhteesta ”keilan”, joka kuvaa mihin suuntaan liikkuminen aiheuttaa törmäyksen ja mihin suuntiin kulkemalla taataan turvallinen reitti.

Menetelmä ei ota kantaa esteiden havainnointitapaan vaan olettaa esteille tunnetut nopeudet, joista lasketaan suhteellinen nopeus itse robottiin nähden, oletetaan nopeudet vakioiksi ja lasketaan, millä robotin nopeusvektoreilla törmäys aiheutuisi johonkin aikaan tulevaisuudessa nykyhetkestä laskien. Robotti ei kuitenkaan kulje suoraa linjaa, eivätkä tosielämässä esteidenkään nopeudet ole välttämättä vakioita – mallia lasketaan uudelleen tihein aikavälein, jolloin ennakoidaan esteen väistäminen hyvissä ajoin.

Yleisemmin tunnettu ja sovellettu, törmäyskeilaa vastaava Fiorinin ja Shillerin [44, 45] esittämä nopeusestemalli (velocity obstacle), myös suhteellisen nopeuden mallina (relative velocity approach) tunnettu, yksinkertaistaa robotin pisteeksi ja esteet ympyröiksi ja esittää mallin hieman eri tavalla.

Kuvassa 7 on esitetty oleelliset nopeusvektorit (v_a robotin nopeus, v_b esteen nopeus), törmäyskeila ja nopeuseste. Törmäyskeilaa rajaavat B :n A :n kautta piirretyt tangentit λ_r ja λ_f ; nopeusesteelle vastaavat reunat ovat δ_r ja δ_f . Nopeusvektorin kärki ei saa olla esteen sisässä eli nopeuserotusvektorista $v_{A,B} = v_A - v_B$ jatkettu suora $\lambda_{A,B}$ ei saa leikata esteympyrää, jottei törmäystä tapahtuisi jollain ajanhetkellä tulevaisuudessa. Törmäysvapaata suuntaa valittaessa on huomioitava myös robotin maksiminopeus; v_A :lla on jokin maksimipituus [44]. Nopeusvektori ei myöskään voi muuttua mielivaltaisen nopeasti johtuen robotin massasta ja maksimikiikhtyvyydestä.



Kuva 7: Prasslerin ym. [42] nopeusestekuva. Robotti on pisteessä A , estettä kuvaa ympyrä pisteessä B . Törmäyskeila on $CC_{A,B}$ ja B :stä johtuva nopeuseste VO_B .

Matemaattisesti nopeuseste koostuu niiden nopeuksien joukosta robotille, joilla törmäys esteeseen tapahtuu jollakin tulevilla ajanhetkellä. Nopeuseste lasketaan Minkowskin summana törmäyskeilasta ja v_B :stä [44]. Geometrisesti ajateltuna törmäyskeilaa vain siirretään v_B :n verran. Formaalisti tämä määritellään seuraavasti:

$$VO_B = CC_{A,B} \oplus v_B, \quad (12)$$

missä \oplus on Minkowskin summaoperaattori:

$$A \oplus B = \{a + b | a \in A, b \in B\}, \quad (13)$$

ja koska v_B on vain vektori, tuloksena jokaista törmäyskeilan pistettä siirretään tämän suuntaan. Itse törmäyskeilan hyödyllinen matemaattinen muoto vaihtelee toteutuksesta riippuen; siihen kuuluvat kaikki pisteet, jotka jäävät reunojen λ_r ja λ_f rajaavan sektorin sisälle.

Kun näin on saatu aikaan kartta hyväksyttävistä nopeuksista, valitaan niistä suosituin. On otettava huomioon, että vain nopeusvektorin osoittaman paikan lähistöllä olevat alueet ovat kelpollisia – robotin massan ja sitä mukaa maksimikiikhtyvyyden mukaan karttaan approksimoidaan monesti suorakaide nykyhetken nopeusvektorin pään ympärille siten, että sen sisällä olevat nopeudet ovat mahdollisia, kun nykyiseen nopeusvektoriin summattava maksimi-arvo riippuu robotin maksimikiikhtyvyydestä [42, 44, 46]. Fiorini ja Shiller [44] esittävät nopeuden valintaan yksinkertaisia heuristiikkoja: valitaan tilanteen mukaan suurin mahdollinen nopeus suoraan kohdetta päin, suurin nopeus joka saattaa erota hieman optimaalisesta suunnasta siten, että nopeus maksimoituu, tai nopeus jolla erityisesti kuljetaan esteen takaa. Toimintamalli valitaan tilannekohtaisesti ja se voi vaihdella prioriteettien mukaan.

Fulgenzi ym. [27] simuloivat nopeusesteitä laajennettuna todennäköisyyksillä ja varausruudukolla ja huomioivat lisäksi epävarmojen ja tuntemattomien tilojen vai-

kutuksen: esimerkiksi nurkan taakse ei voi nähdä, jolloin nurkkaa lähestytään hitaalla nopeudella siten, että törmäys voidaan estää jarruttamalla mikäli liikkuva este yhtäkkiä havaitaan.

Van den Berg ym. [46] kehittivät nopeusesteitä edelleen pienin parannuksin ”keskinäisiksi” nopeusesteiksi, ja suurella määrällä kappaleita simuloidut tulokset esittävät luotettavaa ja sileää käytöstä kun vastaavissa tilanteissa alkuperäinen nopeusetsen menetelmä saa aikaan värähtelyä. Keskinäiset nopeusesteet olettavat, että muutkin liikkuvat esteet käyttäytyvät suurin piirtein samalla tavalla kohdatessaan esteen. Tällä tiedolla kompensoidaan ohjaus siten, että kaikkien robottien keskinäisten ohjausten tuloksena on jokaiselle törmäyksetön rata. Koska malli kuvaa läheisesti ihmisten liikettä, sillä saadaan robotti liikkumaan ihmisten lähistöllä sulavasti.

3.7 Muuta

Alunperin vihivaunut eli automaattitrukit käyttivät mekaanisia hätä-seis-kytkimiä huomaamaan törmäämiset [47]. Tällainen tunnistaminen ei kuitenkaan sovellu ennaltaehkäisevään esteenväistöön, sillä mekaanisen havaitsemisen yhteydessä esteeseen on jo törmätty. Esteenväistön rinnalla voidaan toisaalta käyttää hätä-seis-kytkimiä, jos ei voida olla varmoja siitä, voiko esim. kasvillisuuden läpi mennä vai tuleeko siellä vastaan huomaamaton este ja kun robotti liikkuu niin hitaasti, että mekaanisiin antureihin reagoiminen on riittävän nopeaa [48].

Tietynlaista yleistä heuristiikkaakin on tutkittu mm. sumean logiikan avulla [49, 50]. Sumea logiikka käyttää hyväkseen ihmisille tuttuja ohjaussääntöjä: jos edessä vasemmalla näkyy este, renkaita ohjataan oikealle; Jos este on lähellä, hidastetaan vauhtia; jos este on hyvin lähellä, hidastetaan entistäkin kovempaa.

4 Ulkotilan sovelluksia

Sisätilat ja kaupunkimaastot ovat rakennettua ympäristöä, josta voi olettaa lukuisia yksinkertaistuksia. Rakentamaton maasto eli metsät, suot, muut kasvillisuudet ja kivikkoiset alueet ovat vaikeita, sillä maa ei ole tasainen eivätkä esteet ole korkeita tai kohtisuoria suhteessa maahan. Esteiksi on luokiteltava myös vaarallisen vinot mäet sekä kuopat. Aiempana esitettyjä väistömenetelmiä voidaan soveltaa varsin suoraan myös tällaiseen maastoon – esteiden havainnointi ja tulkinta tosin eroaa oleellisesti rakennetun ympäristön havainnointitoiminnasta, mutteivät ne vaikuta itse väistämistoiminnon laskemiseen. Vain maasto tai syvät kuopat voidaan havainnoida kokenäköllä ja merkitä edelleen varausruudukkoon väistettävinä kohteina. Maastosta voidaan kerätä lisäinformaatiota kuten pinnan epätasaisuutta, joka voidaan merkitä karttaan alueiden ominaisuutena.

Stentz ja Hebert [51] esittävät Humvee-ajoneuvoon asennetun, laserskannerilla havainnoivan ja GPS:n ja odometrian yhdistelmällä paikkaa seuraavan järjestelmän. Järjestelmä yhdistää globaalin ja lokaalin navigoinnin. Se käyttää D*-algoritmia (kuten A* [41], mutta tehokkuusparannuksilla dynaamiseen ympäristöön kun reittiä joudutaan laskemaan useasti) etsimään ennakkotietoon pohjautuvasta tiedos-

ta optimaalisen globaalin reitin tavoitteeseen. Samalla sen lokaali navigointi antaa vihjeitä paikallisista esteistä laserskannauksen avulla. Globaalin ja lokaalin reitittäjän ohjaustuloksista lasketaan painotettu keskiarvo siten, että lokaalia esteenväistöä painotetaan reitinhakua enemmän, jolloin pysytään reitillä mutta mennään esteiden ohitse [51, s. 18].

Myös Kelly [52] esittää tutkimuksessaan Humveehen kytketyn navigaattorin, joka ottaa ennustavasti huomioon useita eri uhkia joita valittavalla reitillä saattaa tulla vastaan [52, s. 337]. Algoritmi ottaa huomioon törmäysten ja tavoitteen haun lisäksi mm. liian suuren kääntökulman, jarrutustehon ja muut kinematiikasta aiheutuvat rajoitteet kaarinopeusmetodin lailla.

Koska edellä mainituissa on kyseessä ”tavallinen” auto rattiohjauksella, algoritmi ehdottaa sekä ajonopeutta että eturenkaille tiettyä kääntökulmaa, joiden aiheuttamaa polkua ennustetaan tietty lyhyt matka eteenpäin [52, s. 45]. Eri kääntökulmista seuraavat polut tutkitaan kartasta, ja näille lasketaan haastavuusarvot uhkista kulmapohjaiseksi histogrammiksi VFH+:n lailla. Histogrammit summataan yhteen jolloin paras yhteinen hyvyysarvo valitsee suuntakulman. Kelly väittää menetelmän toimivan 15 km/h vauhdissa [52, s. 335].

Antureina tässä ajoneuvossa on laseretäisyysanturi ja stereokamera. Näiden tuottamista informaatiosta Kellyn algoritmi etsii adaptiivisesti jyrkkiä reunoja, joiden ympäristöä tarkastellaan maanpinnan vaihteluiden varalta pysyäkseen tasaisella maalla [52, s. 211].

Aiempia vastaavalla tavalla toimii myös Singhin ym. [29] kuvaama navigointi, joka yhdistää lokaalin ja globaalin liikkeensuunnittelun. Globaalisti käytetään D*:ä kuten aiemmin. Lokaali suunnittelija perustuu Kellyn tutkimukseen ja arvioi lähiympäristön kaaren muotoisten reittien kustannuksia. Suunnittelija ottaa lisäksi huomioon ympäristön epävarmuuden; kartan pisteillä on varmuusarvoja, jotka vanhenevat robotin liikkuessu suhteessa matkan pituuteen, ja samalla uusille mitauksille alustetaan varmempi arvo. Varmuusruudukon arvot siis ”unohtuvat” ajan kuluessa.

Michels ym. [53] käyttävät vahvistusoppimista yhden kameran konenäöllä. Järjestelmälle opetetaan ennalta kamerakuvien ja laserskannerilla mitattujen etäisyyksien korrelaatiota, minkä jälkeen kauko-ohjattava auto väistää esteitä pelkän kameran avulla. Vahvistusoppimisessa algoritmi saa ”palkkion” tekemänsä toiminnon mukaan ja korjautuu täten adaptiivisesti paremmaksi. Tässä algoritmi saa palkkiota siitä, että se pitää oikean nopeuden ja on törmäämättä esteisiin.

Schäferin ym. [54] nelipyöräinen robotti on varustettu stereonäöllä sekä kolmiulotteisella laserskannerilla, ja tuottaa ohjauksen käyttäytymispohjaisesti. Korkeisiin makiin, kuoppiin (negative obstacle) ja hyvällä tiellä pysymiseen sekä esteiden väistöön varaudutaan erikseen; eri heuristiset käyttäytymismallit näille tuottavat vihjeitä etu- ja takarenkaiden kääntökulmalle ja nopeudelle. Mallit toimivat sumean logiikan kaltaisesti: jos oikealla näkyy este, pyritään kääntymään vasemmalle. Esteet (sekä hankalat maastot että konkreettisesti läpipääsemättömät kappaleet) havaitaan pääasiassa syvyysnäön epäjatkuvuuskohtien avulla maaston geometriasta.

Toisessa tutkimuksessa Schäfer ym. [48] tutkivat nimenomaan kasvillisuuspainotteista maastoa, jossa harvojen kasviesteiden tunnistaminen on haastavampaa. Täs-

sä stereonäön ja laserskannauksen avuksi on liitetty kosketusanturi robotin eteen. Kun edessä huomataan harvaa ja mahdollisesti joustavaa kasvillisuutta, edetään silti eteenpäin mutta hidastetaan huomattavasti, jolloin kosketusanturi ehtii ilmoittaa turvallisesti mahdollisesta törmäyksestä kiinteään kappaleeseen kasvillisuuden sisällä.

5 Yhteenveto

Universaalia aivan kaikkialle sopivaa lokaalia esteenväistöä ja/tai liikkeensuunnittelua ei ole, koska mahdollinen työalue voi olla mitä vain tehdastiloista toimistotiloihin ja edelleen kaupunkimaastoon ja metsään. Toisaalta yleisiä liikkuvien robottien lähialueen esteenväistömenetelmiä ei ole kovin montaa, ja monet sovellettavat menetelmät pohjautuvat pitkään tutkittuihin hyväksi todettuihin aiheisiin. Esteenväistöön liittyy monia asioita robotin kinematiikan huomioon ottamisesta ja anturien sijoittelusta itse väistöalgoritmin toimintaan ja toimilaitteiden ohjaukseen. Robotin mekaniikka määrää pitkälti millaista väistöalgoritmia on käytettävä ja mitä on robotin ohjaamisessa otettava huomioon. Lisäksi haastavassa maastossa ja erityistilanteissa on aina rajoitteita, jotka on huomioitava erikseen.

Havainnointilaitteissa yleisimmät ovat ultraäänianturit, konenäkö ja kolmiulotteiset laserskannerit. Ultraääntä käytetään paljon sisätiloissa, mutta sen tarkkuus ja nopeus jäävät muille jälkeen niin ylivoimaisesti, ettei sille ole ulkotiloissa juuri sovelluksia. Ulkona kulkevissa roboteissa on monesti sekä kolmiulotteinen laserskanneri että kamera robotin etuosassa.

Liikkuvat esteet tunnistetaan etsimällä perättäisistä mittauksista toisiaan vastaavat estekuviot ja tutkimalla niiden liikkumisnopeuksia. Nopeusestemallia kannattaa käyttää jos robotti tulee kulkemaan ympäristössä jossa on useita muitakin vastaavalla tavalla liikkuvia asioita. Sen voi myös lisätä jonkin toisen pääasiallisesti käytettävän menetelmän päälle.

Tutkatyypisten anturien mittaukset varastoidaan usein varmuusruudukkoon tai johonkin sen johdannaiseen. Nopeassa kamerapohjaisessa anturoinnissa väistetään useimmiten sitä mikä näkyy suoraan edessä. Etenkin ulkotiloissa kameran kuvasta rakennetaan VFH:n kaltainen histogrammi josta valitaan reitti ohjauskulman rajoitteiden mukaisesti. Erityisesti VFH+:n tuomat ideat sopivat hyvin myös tilanteisiin, joissa robotti kulkee erityisen kovaa vauhtia, sillä edessä olevista esteistä erotellaan helposti eri alueet, joiden suuntaan kannattaa ohjata; potentiaalikenttiin perustuvat ohjaukset sopivat hitaammille roboteille, etenkin joka suuntaan ohjautuville joissa kinematiikka vastaa itse menetelmän toimintaideaa eli voiman kohdistamista robotiin tiettyyn suuntaan.

Erilaisista mahdollisista reiteistä voidaan valita suosituin eri kriteereillä VFH+:n käyttämän hyvyysfunktion lailla. Aina ei välttämättä kannata ajaa suoraan tavoittepistettä kohti, jos toinen reitti minimoi jarruttelun, liikkuvien esteiden läheltä kulkemisen, maaston haastavuuden tai muun reitin lyhyiden sijaan optimoitavan kriteerin. Tämä on toisaalta enemmän globaalin kuin lokaalin suunnittelun ongelma.

Kun tiedetään ennalta millaiseen ympäristöön robotti on suunniteltu, voidaan tehdä yksinkertaistavia heuristiikkoja anturidatalle ja esteille. Henkilöautoa automatisoidessa voidaan olettaa, että se ajaa autoteillä ja siten etsiä maanpinnasta kaistaviivoja (jolloin ohjaus ei tosin toimi lumella) sekä kiinnittää huomiota ennalta mm. risteystilanteisiin ja kaistanvaihtoihin. Kävelytiellä kulkeva robotti taas voisi yrittää etsiä talon seinää vierestä, joka auttaisi kulkemaan suoraan. Haastavassa maastossa on muistettava maaston kuopat ja vinot mäet sekä kasvillisuus jonka läpi saattaa kyetä ajamaan vaikka kasvit olisivatkin korkeita.

Viitteet

- [1] Y.K. Hwang and N. Ahuja. Gross motion planning—a survey. *ACM Computing Surveys (CSUR)*, 24(3):219–291, 1992.
- [2] G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.
- [3] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296, 2008.
- [4] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.
- [5] J. Borenstein, HR Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of robotic systems*, 14(4):231–249, 1997.
- [6] L. Jetto, S. Longhi, and G. Venturini. Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation*, 15(2):219–229, 1999.
- [7] R. Siegwart and I.R. Nourbakhsh. *Introduction to autonomous mobile robots*. The MIT Press, 2004.
- [8] P. Abad-Manterola, J. Burdick, I.A.D. Nesnas, and J. Cecava. Wheel design and tension analysis for the tethered axel rover on extreme terrain. In *Aerospace conference, 2009 IEEE*, pages 1–8. IEEE, 2009.
- [9] U. Saranli, M. Buehler, and D.E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616, 2001.
- [10] G. Campion, G. Bastin, and B. Dandrea-Novet. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation 2002*, 12(1):47–62, 1996.
- [11] J. Billingsley, A. Visala, and M. Dunn. Robotics in agriculture and forestry. 2008.
- [12] J. Borenstein, HR Everett, and L. Feng. Where am I? Sensors and methods for mobile robot positioning. *University of Michigan*, 119:120, 1996.
- [13] J. Borenstein and L. Feng. UMBmark: A benchmark test for measuring odometry errors in mobile robots. *Ann Arbor*, 1001.
- [14] B. Barshan and H.F. Durrant-Whyte. Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342, 1995.

- [15] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, 1994.
- [16] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.
- [17] I. Ohya, A. Kosaka, and A. Kak. Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*, 14(6):969–978, 1998.
- [18] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The International Journal of Robotics Research*, 21(8):735, 2002.
- [19] C.C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, volume 1, pages 842–849. IEEE, 2003.
- [20] G. Benet, F. Blanes, JE Simó, and P. Pérez. Using infrared sensors for distance measurement in mobile robots. *Robotics and autonomous systems*, 40(4):255–266, 2002.
- [21] J. Borenstein and Y. Koren. Obstacle avoidance with ultrasonic sensors. *IEEE Journal of Robotics and Automation*, 4(2):213–218, 1988.
- [22] J. Guivant, E. Nebot, and S. Baiker. Localization and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583, 2000.
- [23] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198, 2003.
- [24] D. Murray and C. Jennings. Stereo vision based mapping and navigation for mobile robots. In *IEEE International Conference on Robotics and Automation, 1997. Proceedings.*, volume 2, pages 1694–1699. IEEE, 1997.
- [25] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 694–711, 2006.
- [26] B. Schiele and J.L. Crowley. A comparison of position estimation techniques using occupancy grids. *Robotics and autonomous systems*, 12(3-4):163–171, 1994.

- [27] C. Fulgenzi, A. Spalanzani, and C. Laugier. Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid. In *IEEE International Conference on Robotics and Automation, 2007*, pages 1610–1616. IEEE.
- [28] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.
- [29] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja, and K. Schwehr. Recent progress in local and global traversability for planetary rovers. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00*, volume 2, pages 1194–1200. IEEE.
- [30] W. Feiten, R. Bauer, and G. Lawitzky. Robust obstacle avoidance in unknown and cramped environments. In *IEEE International Conference on Robotics and Automation, 1994. Proceedings.*, pages 2412–2417. IEEE, 1994.
- [31] R. Simmons. The curvature-velocity method for local obstacle avoidance. In *IEEE International Conference on Robotics and Automation, 1996. Proceedings.*, volume 4, pages 3375–3382. IEEE, 1996.
- [32] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on robotics and automation*, 8(5):501–518, 1992.
- [33] C.W. Warren. Global path planning using artificial potential fields. In *IEEE International Conference on Robotics and Automation, 1989. Proceedings.*, pages 316–321. IEEE, 1989.
- [34] SS Ge and YJ Cui. New potential functions for mobile robot path planning. *IEEE Transactions on Robotics and Automation*, 16(5):615–620, 2002.
- [35] P. Vadakkepat, K.C. Tan, and W. Ming-Liang. Evolutionary artificial potential fields and their application in real time robot path planning. In *Proceedings of the 2000 Congress on Evolutionary Computation*, volume 1, pages 256–263. IEEE, 2000.
- [36] C. Thureau, C. Bauckhage, and G. Sagerer. Learning human-like movement behavior for computer games. In *From animals to animats 8: proceedings of the seventh [ie eighth] International Conference on Simulation of Adaptive Behavior*, page 315. The MIT Press, 2004.
- [37] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE international conference on robotics and automation*, volume 2, pages 1398–1404. Citeseer, 1991.
- [38] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fact mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1179–1187, 1989.

- [39] I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. In *IEEE International Conference on Robotics and Automation, 1998. Proceedings.*, volume 2, pages 1572–1577. IEEE, 1998.
- [40] I. Ulrich and J. Borenstein. VFH*: local obstacle avoidance with look-ahead verification. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00.*, volume 3, pages 2505–2511. Ieee, 2000.
- [41] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [42] E. Prassler, J. Scholz, and P. Fiorini. A robotics wheelchair for crowded public environment. *Robotics & Automation Magazine, IEEE*, 8(1):38–45, 2001.
- [43] A. Chakravarthy and D. Ghose. Obstacle avoidance in a dynamic environment: A collision cone approach. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 28(5):562–574, 1998.
- [44] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760, 1998.
- [45] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using the relative velocity paradigm. In *IEEE International Conference on Robotics and Automation, 1993. Proceedings.*, pages 560–565. IEEE, 1993.
- [46] J. Van den Berg, M. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation. ICRA 2008.*, pages 1928–1935. IEEE, 2008.
- [47] R. Bostelman, T. Hong, and R. Madhavan. Experiments toward non-contact safety standards for automated industrial vehicles. In *Proceedings of the SPIE International Symposium on Defense and Security*. Citeseer, 2004.
- [48] H. Schafer, A. Hach, M. Proetzsch, and K. Berns. 3d obstacle detection and avoidance in vegetated off-road terrain. In *IEEE International Conference on Robotics and Automation. ICRA 2008*, pages 923–928. IEEE, 2008.
- [49] W. Li. Fuzzy logic-based perception-action’behavior control of a mobile robot in uncertain environments. In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, pages 1626–1631. IEEE.
- [50] D.K. Pratihar, K. Deb, and A. Ghosh. A genetic-fuzzy approach for mobile robot navigation among moving obstacles. *International Journal of Approximate Reasoning*, 20(2):145–172, 1999.

- [51] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2):127–145, 1995.
- [52] A. Kelly. *An intelligent, predictive control approach to the high-speed cross-country autonomous navigation problem*. PhD thesis, Citeseer, 1995.
- [53] J. Michels, A. Saxena, and A.Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 593–600. ACM, 2005.
- [54] H. Schäfer, M. Proetzsch, and K. Berns. Stereo-Vision-Based Obstacle Avoidance in Rough Outdoor Terrain. In *International Symposium on Motor Control and Robotics*. Citeseer, 2005.