



Vaasan yliopisto
UNIVERSITY OF VAASA

Joonas Pihlaja

Katsaus navigaation ja liikkeen integraatioon kävelevissä mobiiliroboteissa

Tekniikan ja innovaatiojohtamisen akateeminen yksikkö
Kandidaatin tutkinto
Automaatio- ja tietotekniikka

Vaasa 2025

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen akateeminen yksikkö**

Tekijä:	Joonas Pihlaja		
Tutkielman nimi:	Katsaus navigaation ja liikkeen integraatioon kävelevissä mobiiliroboteissa		
Tutkinto:	Tekniikan kandidaatti		
Oppiaine:	Automaatio- ja tietotekniikka		
Työn ohjaaja:	Timo Mantere		
Valmistumisvuosi:	2025	Sivumäärä:	32

TIIVISTELMÄ:

Kävelevien robottien tuotanto alkoi 1900-luvun puolivälissä, mutta niiden rakenne perustuu vielä nykyäänkin sitä aikaisemmin tuotettuun tutkimukseen eläinten liikkeestä. Kävelevien robottien arkkitehtuurilla on parempi liikkuvuus ja toimintakyky vaikeakulkuisilla alueilla, kuten kivikoilla tai portaikoissa, joissa renkaita käyttävät robotit voisivat jumiutua. Kävelevät robotit kärsivät kuitenkin niiden rakenteen ja tasapainon ylläpitämisen aiheuttamista ongelmista, kuten systeemin monimutkaisuudesta ja energiankulutuksesta. Robottien suosio on selvästi kasvussa ja niiden teknologia kehittyy nopeasti.

Robotit havainnoivat ympäristöään yleensä laserkeilauksen, kameroiden ja vääntömomenttia tai kosketusta havaitsevien sensoreiden avulla. Havainnoinnin tuottama data syötetään samanaikaisesti SLAM-algoritmille, joka paikantaa robotin ja luo sen ympäristöstä digitaalisen mallinnuksen navigaatiota varten. Havainnointia voidaan edesauttaa yhdistämällä siihen konenäön menetelmiä, joiden avulla voidaan tunnistaa esineitä tai pintojen ominaisuuksia robotin liikkussa.

Ohjaussysteemit vastaavat robotin toimintojen koordinaatiosta ja tasapainottamisesta. Reitinlaskenta- ja liikkeenlaskentajärjestelmät vastaavat robotin kulkureitin ja tarkkojen liikkeiden suunnittelusta. Käytettävät menetelmät jaetaan yleensä klassisiin metodeihin, evoluutioalgoritmeihin ja koneoppimiseen. Koneoppimista voidaan haluttaessa käyttää robotin ohjaukseen, navigaation- tai liikkeenlaskennan metodien korvaamiseen.

AVAINSANAT: havainnointi, humanoidirobotit, itseohjautuvat robotit, konenäkö, kävely, navigointi

Sisällys

1	Johdanto	5
2	Taustaa kävelevistä roboteista	6
2.1	Kävelevän robotin määritelmä	6
2.2	Historia ja nykyinen tilanne	7
2.3	Haasteet ja edut kävelevässä arkkitehtuurissa	8
3	Havainnointi ja lokalisaatio	11
3.1	Havainnoinnin perusteet	11
3.2	Ympäristön mallintaminen ja lokalisaatio	12
3.3	Pintojen ja esineiden tunnistus	16
4	Ohjaussysteemit, liike ja navigaatio	18
4.1	Liike ja navigaatio yleisesti	18
4.2	Ohjauksen ja tasapainotuksen metodeja	21
4.3	Reitinlaskennan ja liikkeenlaskennan metodeja	23
4.4	Koneoppiminen mobiiliroboteissa	25
5	Yhteenveto	28
	Lähteet	30

Kuvat

Kuva 1.	Eri liikemethodien energiatehokkuus tasaisella maalla.	9
Kuva 2.	Laserkeilauksen muodostama kuva.	12
Kuva 3.	Semanttista kartoitusta käyttävän systeemin yhtäaikaiset prosessit.	17
Kuva 4.	Atlaksen virtuaalinen mallinnus ympäristöstään.	19
Kuva 5.	Koneoppimista käyttävän yhdistellyn strategian prosessit.	26

Taulukot

Taulukko 1.	SLAM-algoritmien vertailu.	15
Taulukko 2.	Koneoppimisalgoritmien käyttötarkoituksia	27

1 Johdanto

Katsojat ovat saaneet hämmästellä mobiilirobottien toimintaa jo hyvän tovin. Kävelevät robotit, kuten Boston Dynamicsin tuottama Atlas, ovat varmasti yllättäneet monia niiden navigointikyvyllä ja itsenäisellä toiminnalla viimeisen vuosikymmenen aikana. Monien tutkijoiden ja yritysten tavoitteena onkin jo pitkään ollut täysin itsenäisten robottien tuottaminen, mutta tätä on oikeasti lähestytty vasta yleisen teknologian kehityksen ansiosta. Atlaksen kaltaisten robottien olemassaolo luo paljon uusia mahdollisuuksia useille eri aloille, mutta varmasti herättää myös paljon kysymyksiä niiden toiminnasta. Miten tällaiset robotit tekevät päätöksiä? Miten robotti näkee ympäristönsä ja navigoi siinä? Minkälaisia fyysisiä- ja ohjelmistoratkaisuja vaaditaan tämän toiminnan saavuttamiseksi? Katsauksia kävelevien mobiilirobottien toimintaan kokonaisuutena on huomattavan vähän, joten tämä työ pyrkii vastaamaan yleisiin kysymyksiin ja antamaan lukijalle käsityksen niiden toiminnasta. Tutkielman lopussa lukijan pitäisi ymmärtää mitä kaikkea vaaditaan, että kävelevä robotti saadaan itsenäisesti liikkumaan paikasta A paikkaan B tuntemattomassa ympäristössä.

Työ on suunnattu robottiteknologiasta kiinnostuneille, joilla voi olla jo aikaisempaa kokemusta joistain niiden yksittäisistä osa-alueista, kuten konenäöstä, reitinlaskennasta tai liikkeenlaskennan algoritmeista. Joitain keskeisiä tutkimuskysymyksiä työlle ovat miksi kävelevä arkkitehtuuri valitaan roboteille, miten kävelevät robotit ylläpitävät tasapainonsa, mitä menetelmiä käytetään robotin reitin ja liikkeiden valitsemiseen ja miten tekoälyä hyödynnetään kävelevien robottien systeemeissä. Työ toteutetaan tutkimalla mobiilirobottien toimintaan liittyviä aineistoja ja tulee esittelemään joitain niistä löytyviä yleisiä ratkaisuja. Tämä katsaus voi siten samalla soveltua lähtökohdaksi tarkemmalle lisätutkimukselle kävelevien robottien osiin ja toimintaan.

2 Taustaa kävelevistä roboteista

Työtä varten on hyödyllistä ensin antaa määritelmä autonomiselle kävelevälle robotille. Sen jälkeen voimme lyhyesti tarkastella kävelevien robottien historiaa ja syitä kävelevän arkkitehtuurin valitsemiselle.

2.1 Kävelevän robotin määritelmä

Voimme antaa kävelevälle mobiilirobotille tarkan määritelmän kahdessa osassa: Tarkentamalla ensin autonomisen mobiilirobotin määritelmän ja määrittämällä sitten lyhyesti mitä laskemme käveleväksi robotiksi.

Rubion ja muiden (2019, Introduction) mukaan autonomisella mobiilirobotilla tarkoitetaan konetta, joka pystyy toimimaan itsenäisesti ja liikkumaan ympäristössään vapaasti. Käytännössä tämä tarkoittaa, että robotilla on kyky havainnoida ympäristöään jonkin tiedonlähteen kautta ja kyky tehdä autonomisesti reaktionomaisia päätöksiä tulkitsemalla tuota tiedonlähdettä. Robotin ei siis tarvitse päättää mitä tehtävää se suorittaa, sen täytyy vain onnistua määrittelemään toiminnot tehtävästä suoriutumista varten tietoa analysoimalla. Bekey (2005, kappale 1.1) määrittelee vastaavasti robottien autonomian niiden kykyä operoida realistisissa ja muuttuvissa ympäristöissä ilman suoraa ulkopuolista kontrollointia.

Kävelevien robottien vuorostaan tulee käyttää ainakin kahta tai useampaa jalan tyyppistä raajaa niiden päämuotoisena liikkeen metodina ja niillä tulee olla tarvittavat systeemit, jotka vastaavat tämän aiheuttamiin vaatimuksiin (Rubio ja muut, 2019, Introduction). Hybridirobotteja, jotka käyttävät kävelevää arkkitehtuuria, mutta yhdistävät siihen mukaan muun tyyppisiä liikkeen metodeja kuten renkaita tai teloja, ei tulla tarkastelemaan osana tätä katsausta.

Kävelevän mobiilirobotin tulee siis olla autonominen, kykenevä havainnoimaan ympäristöään ja reagoimaan sen muutoksiin havaitsemansa tiedon perusteella, sekä pystyä liikkumaan ympäristössään kahta tai useampaa jalkaa käyttäen.

2.2 Historia ja nykyinen tilanne

Kävelevät robotit rakennetaan yleensä imitoimaan ihmisten, eläinten, hyönteisten tai hämähäkkien liikettä ja suurin osa inspiraatiosta niiden rakenteelle on saatu luonnosta (Bekey, 2005). Esimerkiksi kaksijalkaiset robotit jakautuvat pääasiassa ihmisiä (kävelyliike) tai lintuja (hyppimisliike) imitoiviin malleihin (Mikolajczyk ja muut, 2022, Kappale 1). Vastaavasti Shishir (2022) kertoo, kuinka pohja nykyisten robottien kävelyliikkeelle perustuu Eadward Muybridgen tutkimukseen 1870-luvulta. Muybridge dokumentoi hevosten kävelyliikettä Kalifornian kuvernöörin toimesta ja jatkoi myöhemmin työtään muiden nisäkkäiden liikkeeseen. Myöhemmin 1960-luvulla General Electric tuotti ihmisen ohjaaman mekaanisen kävelevän ”auton”, jonka jalkojen rakenteet ovat verrattavissa nykyisiin käveleviin robotteihin (Shishir, 2022).

Ensimmäinen tämän katsauksen määritelmän täyttävä kävelevä robotti on kuitenkin vasta vuosien 1970 ja 1973 välillä valmistettu ihmisrobotti WABOT-1. WABOT-1 kykeni havaitsemaan etäisyyksiä sensoreillaan, kävelemään vapaasti ympäristössään ja pitämään yksinkertaisia esineitä käsissään (Waseda University, n.d.). Tällöin yksi robottituotannon suurimmista haasteista oli stabiilin liikkeen tuottaminen, jota varten keksittyjä menetelmiä käytetään vielä nykyäänkin laajasti (Shishir, 2022). Ihmisrobottien kehitys alkoi kunnolla kiihtymään HONDAN P2-robotin julkaisun jälkeen vuonna 1996 (Kaneko ja muut, 2019). Tämän seurauksena 2000-luvulla on nähty useita kaksijalkaisia robotteja eri valmistajilta, kuten NASAn Valkyrie, German Aerospace Centren tuottama Toro, Boston Dynamicsin valmistama ATLAS ja japanilaisen Advanced Industrial Science and Technology -instituutin HRP-5P. Viimeisen viiden vuoden aikana on myös julkistettu kiinalainen H1, joka on kirjoitushetkellä nopein ihmismäinen robotti, Boston Dynamicsin vuoden 2024 alussa julkaisema uusi Atlas-malli sekä Teslan tuottama Optimus Gen 2.

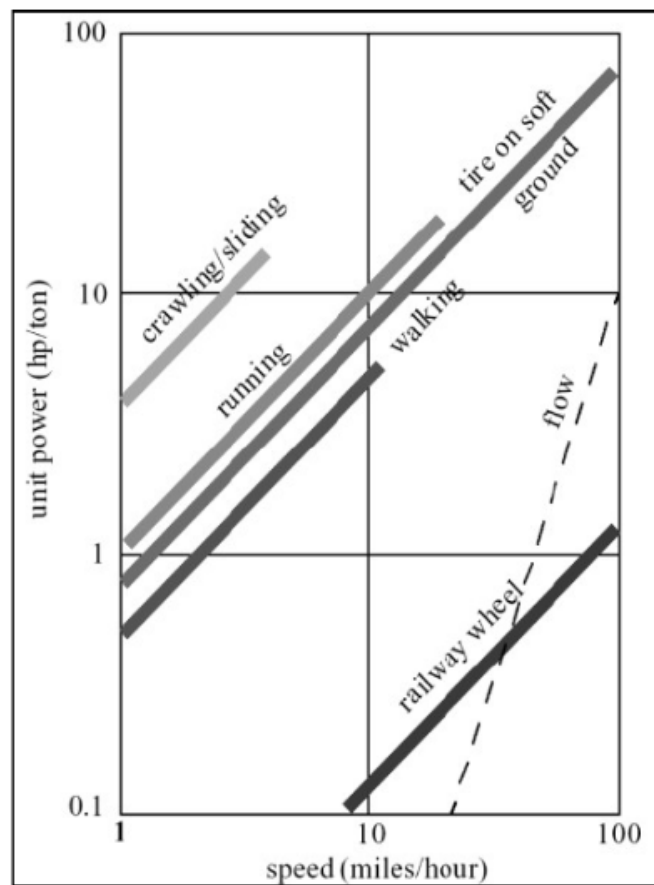
Suuri osa robottituotannosta ja tutkimuksesta tapahtuu nykyään yksityisyritysten toimesta, jonka seurauksena uusin informaatio niiden toiminnasta ei ole aina julkisesti saatavilla. Organisaatiot kuten Boston Dynamics ja Tesla eivät esimerkiksi ole julkaisseet tietoja robottiensa käyttämisestä algoritmeista tai ohjelmistoista. Mikolajczyk ja muut (2022, kappale 5) ovat kuitenkin havainneet avoimien tutkimusprojektien lisääntyneen selvästi viimeisen vuosikymmenen aikana, joka myös lisää yleistä ja markkinapohjaista kiinnostusta käveleviä robotteja kohtaan. He myös näkevät nykyisten kävelevien robottien olevan nopean kehityksen vaiheessa, joka voi hyvinkin pian johtaa ihmismäisten robottien laajempaan käyttöön ottoon, varsinkin teollisuudessa, avaruuden kartoittamisessa ja sodankäynnissä (Mikolajczyk ja muut, 2022, Kappale 5).

2.3 Haasteet ja edut kävelevässä arkkitehtuurissa

Robottien systeemit ovat nykyään huomattavasti edistyneempiä ensimmäisiin malleihin verrattuna, mutta kävelevien robottien tuotannossa on vielä useita haasteita. Edistyneiden sistemien tuotanto on haastavaa, ymmärrys kaksijalkaisen kävelemisen kaikista ominaisuuksista on puutteellista sekä kävelevien robottien huonot käyttömahdollisuudet niiden nykyisellä tasolla vähentävät niiden kustannustehokkuutta (Mikolajczyk ja muut, 2022, Kappale 5). Tuntemattomissa ja muuttuvissa ympäristöissä navigoiminen aiheuttaa myös suuria haasteita tuotannolle, koska se vaatii robottien olevan tarpeeksi kestäviä, tehokkaita ja omavaraisia selviytyäkseen tehtävistään. Esimerkiksi Boston Dynamicsin Atlasta ja Spottia on esitelty apuna pelastustyössä ja NASAn kehittämää Valkyrieta halutaan käyttää kaukaisten planeettojen tutkimiseen.

Kävelevien robottien kestävyys ja energiankäyttö nousevat monissa käyttötarkoituksissa esteeksi. Jalkojen verrannollisesti monimutkainen rakenne tekee niiden ohjaamisesta ja korjaamisesta vaikeampaa sekä johtaa usein suurempaan energiankulutukseen muihin robottityyppeihin verrattuna. Esimerkiksi renkaita käyttävät robotit ovat tasaisella maalla huomattavasti nopeampia, käyttävät paljon vähemmän energiaa ja vaativat yksinkertaisemman rakenteen liikkeen ja tasapainon ylläpitämistä varten (Sven Böttcher, 2006, Legged locomotion). Kävelevien robottien ominaisuudet voivat vaihdella suuresti

jalkojen määrästä ja rakenteesta riippuen. Yleisesti renkasiin verrattuna jalkojen energiatehokkuus voi tasaisella maalla olla jopa sata kertaa huonompi (Kuva 1), mutta pehmeillä ja epätasaisilla pinnoilla tämä ero pienenee ja niiden kulutus voi olla jopa tasavertainen (Sven Böttcher, 2006, Legged locomotion). Renkaat ja jalat yhdistävät hybridirakenteet voivat tästä syystä olla hyödyllisiä, koska ne yhdistävät renkaiden energiatehokkuuden jalkojen liikkuvuuden kanssa.



Kuva 1. Eri liikemethodien energiatehokkuus tasaisella maalla (Sven Böttcher, 2006).

Rubion ja muiden (2019, Locomotion: Walking or legged mobile robots) mukaan jalat mahdollistavat huomattavasti vapaamman navigaation monimutkaisissa ympäristöissä, helpottavat liikkumista pehmeillä tai epätasaisilla pinnoilla kuten hiekalla tai kivikoissa ja voivat suoraan välttää tilanteita, joissa renkaita käyttävä robotti jumiutuisi. Sisätiloissa kävelevä robotti kykenee vastaavasti ylittämään korkeita kynnyksiä ja kulkemaan portaita pitkin (Rubio ja muut, 2019, Locomotion: Walking or legged mobile robots). Yleinen

kiinnostus kävelevien robottien käytölle johtuukin yleensä juuri niiden kyvystä toimia samoissa ympäristöissä kuin ihmiset. Tästä huolimatta eettiset ongelmat voivat kuitenkin vielä jarruttaa robottien laajempaa käyttöönottoa ihmisten ympärillä, koska niiden vaikutuksia työturvallisuuteen, yksityisyyteen tai teknologian rooliin yhteiskunnassa ei ole vielä kunnolla käsitelty (Unlimited robotics healthcare, n.d.). Esimerkiksi Boston Dynamicsin säännöt eivät salli robottiensa aseistamista ja vaativat niiden käytön noudattavan yksityisyyteen ja työturvallisuuteen liittyviä lakeja (Boston Dynamics, n.d. - b).

Rubio ja muut (2019) ovat määrittäneet mobiilirobottien toiminnalle neljä pakollista systeemiä: havainnointi-, kognitio-, navigaatio- ja liikesysteemit (Rubio ja muut, 2019). Näiden systemien tehtäviä robotin toiminnassa tullaan tarkastelemaan tämän katsauksen aikana.

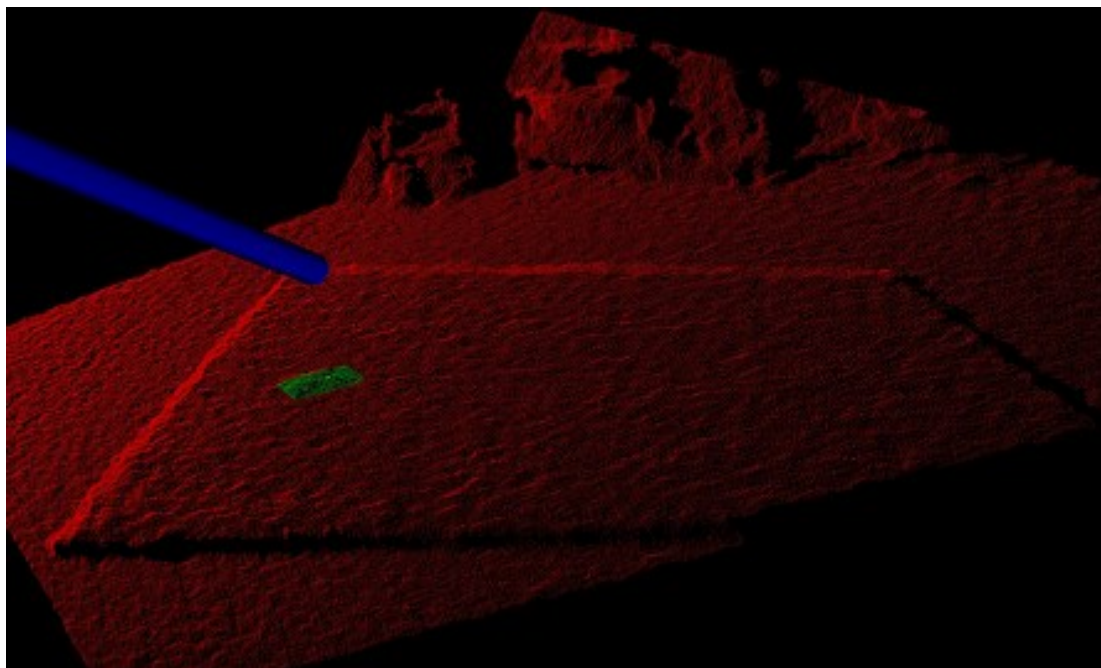
3 Havainnointi ja lokalisaatio

Tämä osio käsittelee sensorien toimintaa, SLAM-menetelmiä ja semanttista kartoitusta.

3.1 Havainnoinnin perusteet

Täyttääkseen autonomian määritelmän mobiilirobotilla tulee olla yksi tai useampi tiedonlähde, joiden pohjalta robotti voi toimia. Käytännössä tästä vastaavat robottiin asennetut sensorit. Rubio ja muiden (2019, Perception) mukaan robotin sensorit voidaan yleisesti jakaa niiden sisäisten prosessien ja toiminnan havainnoinnista vastaavien proprioseptisten sensoreiden sekä niiden ulkoisesta havainnoinnista vastaavien eksteroseptisten sensoreiden välillä. Tämän lisäksi sensorit voidaan myös jakaa passiivisiin ja aktiivisiin sensoreihin. Passiiviset sensorit mittaavat jatkuvasti ympäristöstä tulevaa tietoa, kun taas aktiiviset sensorit vapauttavat ympäristöönsä energiaa ja mittaavat sen tuottaman reaktion (Rubio ja muut, 2019, Perception). Nämä systeemit siten mahdollistavat robotin eri ominaisuuksien toiminnan, kuten sen sijainnin määrittämisen, reitinlaskennan ja esteiden välttämisen liikkussa.

Eksteroseptisiin sensoreihin kuuluvat syvyyden havainnointiin käytettävät laserkeilaus sensorit, visuaalis- ja värihavainnoinnista vastaavat kamerat sekä erilaiset kosketusanturit robotin käsissä ja jaloissa (Walas, 2014, Introduction). Proprioseptisiä sensoreita ovat erilaiset liikettä, inertiaa ja orientaatiota havaitsevat sensorit, kuten akselerometrit ja gyroskoopit, tai raajojen vääntömomenttia havaitsevat sensorit (Walas, 2014, Introduction). Visuaaliset sensorit sijoitetaan yleensä robotin päähän ihmismäisen havainnoinnin imitoimiseksi, mutta havainnoinnin tehostamiseksi esimerkiksi NASAn Valkyrie robotilla on lisäksi kameroita polvissa ja jalanpohjissaan (Rubio ja muut, 2019, Locomotion: Walking or legged mobile robots).



Kuva 2. Laserkeilauksen muodostama kuva (Walas, 2014).

Syvyyden havainnointi on roboteille erittäin tärkeä ominaisuus, sillä se muodostaa suurimman osan robotin navigaatioon käytettävästä tiedosta. Se toteutetaan yleisesti laserkeilauksella ja pistepilvi-menetelmällä (Walas, 2014, Introduction), jossa syvyyssensori lähettää ympäristöönsä lasersäteitä ja mittaa etäisyyden seiniin ja esteisiin laskemalla säteiden kimpoamiseen kuluneen ajan. Näin ympäristöstä voidaan muodostaa kolmiulotteinen pistepilvi-malli, jota robotti voi käyttää reitinlaskentaan ja ympäristön ominaisuuksien arviointiin. Laserkeilaukseen voidaan käyttää mitä tahansa siihen kykenevää sensoria ja monet julkiset kokeet ovat esimerkiksi käyttäneet Microsoft Kinect -kameraa sen helpon saatavuuden ja matalan hinnan vuoksi.

3.2 Ympäristön mallintaminen ja lokalisaatio

Rubio ja muut (2019, Navigation) määrittelevät autonomisten robottien navigaatiojärjestelmille kolme päätehtävää: robotin ympäristön mallintamisen, törmäyksiä välttävän reitinlaskennan, ja esteiden välttämisen reittiä pitkin liikkuessa. Näistä tehtävistä ensimmäinen eli ympäristön mallintaminen, toteutetaan mahdollisimman samanaikaisesti robotin havainnoinnin kanssa viiveen vähentämiseksi

(Rubio ja muut, 2019, Navigation). Mallintamisesta vastaa yleensä Simultaneous Localization And Mapping (SLAM) -algoritmit, joiden tehtävänä on muodostaa digitaalinen kartta robotin havainnoinnin tuottamasta datasta sekä vastata samalla robotin lokalisatiosta (Kiran ja muut, 2022, Introduction). Mobiiliroboteille on vaatimusten perusteella tuotettu useita optimoituja SLAM-algoritmeja, kuten GMapping SLAM-, Hector SLAM-, Cartographer SLAM- ja ORB SLAM2 -algoritmit (Li ja muut, 2021, Introduction). Nämä menetelmät voidaan jakaa niiden käyttämien havainnoinnin metodien mukaan yleensä joko LIDARia tai kameraa käyttäviksi menetelmiksi.

Mathworks-sivuston artikkelissa (n.d., Types of SLAM methods) käsitellään eri SLAM-menetelmien toimintaa. Light Detection And Ranging (LIDAR) SLAM käyttää laserkeilausta ja pistepilvi-menetelmää robotin paikan arvioimiseen. Tämä toteutetaan vertaamalla pistepilvien eroja robotin liikkuesssa, jossa käytetään valitun menetelmän, ympäristön ja systeemin suoristustehon perusteella joko 2D- tai 3D-karttaa. Visuaaliseen matkanmittaukseen perustuva Visual SLAM sen sijaan paikantaa robotin eri kulmista otettujen kuvien perusteella, jolloin robotin navigaatioon voidaan käyttää jotain keskeistä maamerkkiä. Tässä menetelmässä niin sanotut harvat metodit paikantavat robotin vertaamalla kuvia ja yhdistämällä niissä olevia piirteitä, kun taas tiheät metodit vertaavat valo- ja värieroja kuvissa (Mathworks, n.d., Types of SLAM methods). Esimerkiksi Wei ja muut tuottivat kokeen kävelevällä robotilla, jossa robotin havainnointi toteutettiin yksinkertaista harvaa metodia käyttäen. Tässä esimerkissä Hough Circles -algoritmia käytettiin havaitsemaan risteyskohtia merkkäviä ympyröitä maassa ja Canny-algoritmia havaitsemaan seinien reunoja. Robotin liikuntamahdollisuuksia pystyttiin siten arvioimaan kameran kulman ja sen havaitsemien tietojen perusteella (Wei ja muut, 2019).

Kaikki SLAM-metodit vaativat, että kuvien tai skannausten ottokohdat tiedetään, tämä vaatii lokalisatiota eli robotin paikan ja suunnan arviointia ympäristössään sensoridatan perusteella. Nykyiset GPS-systeemit eivät yksinään ole tarpeeksi tarkkoja lokalisatioon, joten robotin relatiivista paikkaa lähtökohtaansa nähden joudutaan seuraamaan jollakin

menetelmällä (Rubio ja muut, 2019, Localization and mapping). Robotin sijainnin seuraamiselle on useita vaihtoehtoja ja suurin osa systeemeistä yhdisteleekin eri menetelmiä niiden käyttötarkoituksen mukaisesti. Vaihtoehtoja ovat esimerkiksi matkamittaus, inertiaalinen navigointi, magneettiset kompassit, aktiiviset majakat, GPS-systeemit ja maamerkkien perusteella navigointi (Rubio ja muut, 2019, Localization and mapping). Eri metodien heikkouksia voidaan halutessa paikata yhdistelemällä niitä, jolloin voidaan käyttää kombinaatiota kamera-, LIDAR-, GPS- ja RADAR-systeemejä robotin paikantamiseen ja ympäristön mallintamiseen (Mathworks, n.d., Types of SLAM methods). Suuressa osassa kokeita robotin kulkema matka saatiin matkamittarin arvosta, mutta tämän arvon totuudenmukaisuus jouduttiin varmistamaan jonkin muun menetelmän avulla.

Alla olevassa taulukossa (Taulukko 1) on esitetty joitain yleisimpiä SLAM-algoritmeja ja vertailtu niiden käyttötarkoituksia Lin ja muiden (2021) vertailuun perustuen. Monet suosittu SLAM-algoritmit on suunniteltu pääasiassa sisätilojen kartoittamiseen, mutta esimerkiksi Cartographer SLAM ja ORB SLAM soveltuvat 3D-karttojensa ansiosta hyvin ulkotiloihin. GMapping-algoritmi on tällä hetkellä yleisin SLAM-algoritmi sisätiloissa toimiville roboteille, se muodostaa kartan LIDARin ja matkamittarin tietojen perusteella (Li ja muut, 2021, Principle of Mapping Algorithm). Cartographer SLAM on Googlen tuottama SLAM-algoritmi, joka on muihin algoritmeihin verrattuna huomattavasti tehokkaampi ja tarjoaa enemmän ominaisuuksia (Li ja muut, 2021, Introduction). Hector SLAM ei vaadi matkamittaria toimiakseen ja voi tuottaa samantasoisia karttoja kuin muut algoritmit, mutta robotin pitää liikkua hyvin hitaasti ja tasaisesti. (Li ja muut, 2021, Comparison of construction effect).

Taulukko 1. Yleisten SLAM-algoritmien vertailu, perustuen Lin ja muiden (2021) vertailuun.

Algoritmi	Tyyppi	Lokalisaation menetelmä	Käyttöalue	Huomatuksia
Cartographer	2D- tai 3D-LIDAR, Kamera	Pose Graph Optimisation	Sisä- ja ulkotilat	Yleisesti paras algoritmi nopeuden, tarkkuuden ja ominaisuuksien kannalta.
Hector	2D-LIDAR	Extended Kalman Filter	Sisätilat	Ei sovellu käveleville roboteille niiden epätasaisen liikkeen vuoksi.
GMapping	2D-LIDAR	Rao-Blackwellized Particle Filter	Sisätilat	Yleensä muita algoritmeja nopeampi ja tarkempi pienten huoneiden kartoittamisessa. Tarkkuus kärsii suurilla alueilla kartoittaessa.
ORB SLAM2	Kamera	Oriented FAST and Rotated BRIEF	Sisä- ja ulkotilat	Muodostaa 3D-kartan, joka soveltuu parhaiten ulkotilojen kartoittamiseen. Ei välttämättä huomaa pieniä ominaisuuksia ympäristössä.

3.3 Pintojen ja esineiden tunnistus

LIDARin muodostama syvyyskartta ei yksinään sisällä tietoja pintojen ominaisuuksista; vaikka robotti kykeneekin navigoimaan pelkästään sen avulla, sen toimintakyky on rajoitettu. Jotta robotit voisivat varmemmin ja tehokkaammin suoriutua tehtävistään niiden havainnoinnin halutaan olevan perusteellisempi. Tämä voidaan yleensä toteuttaa käyttämällä semanttista kartoitusta, jossa robotin havaitsemia pintoja ja asioita pyritään tunnistamaan ja kategorioimaan eri menetelmien avulla (Jiao ja muut, 2024, Introduction). Semanttinen kartoitus voidaan yleisellä tasolla jakaa kamera- tai laserpohjaiseksi kartoitukseksi, jossa kamerapohjaiset menetelmät arvioivat ympäristöä konenäön avulla, kun taas laserpohjaiset pyrkivät tunnistamaan ympäristössä olevia esineitä pelkästään geometrisen datan perusteella (Song ja muut, 2022, Related work). Tämän avulla robotti voi esimerkiksi kävellä havaittua jalkakäytävää pitkin tai löytää operaattorin määräämän esineen ympäristöstään (Jiao ja muut, 2024, Introduction). Nämä tekevät semanttisesta kartoituksesta varsin hyödyllisen, mikäli robotti joutuu olemaan usein vuorovaikutuksessa ihmisten kanssa.

4 Ohjaussysteemit, liike ja navigaatio

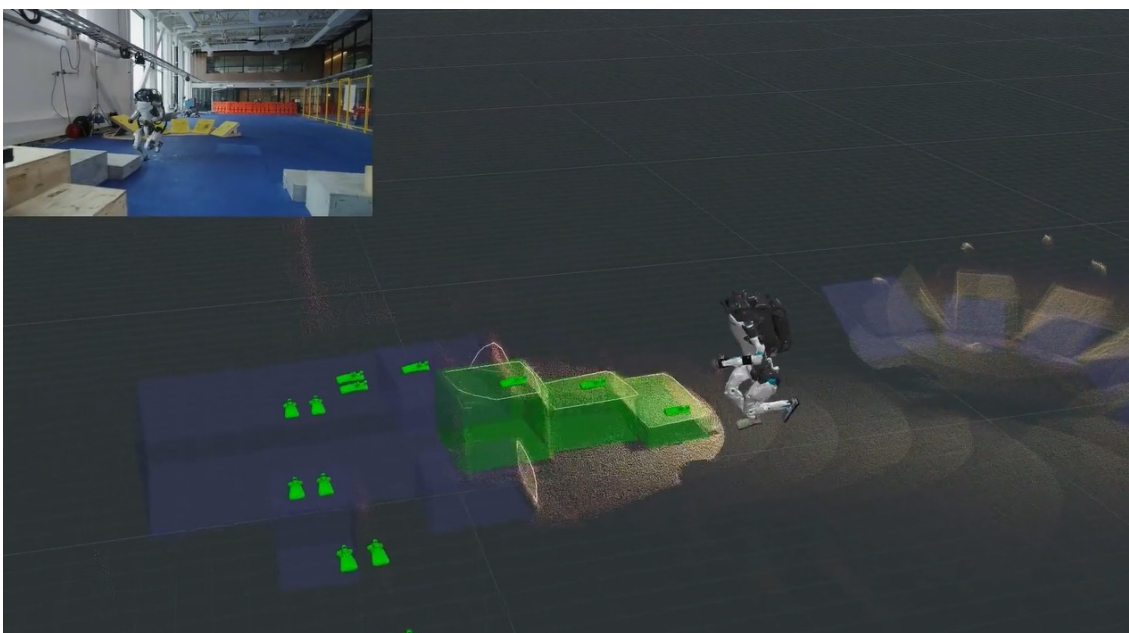
Havainnoinnin ja mallintamisen tavoin ohjaussysteemien tehtävät voidaan jakaa osiin tarkastelua varten. Ohjaussysteemien tehtäviin kuuluu törmäyksiä välttävä reitinlaskenta, esteiden välttäminen reittiä pitkin kulkiessa ja robotin liikkeiden suunnittelu. Ohjaussysteemien ominaisuuksia, reitinlaskentaa ja liikkeenlaskentaa sekä niiden toteutuksia tullaan tarkastelemaan seuraavissa kappaleissa.

4.1 Liike ja navigaatio yleisesti

Reitinlaskenta ja liikkeenlaskenta ovat robotin toiminnan kannalta haastavimpia vaiheita. Mikolajczyk ja muut (2019) kertovat kuinka niiden onnistunut toteutus vaatii tuntemusta useilta aloilta, esimerkiksi evoluutioteoriasta, energiankulutuksen minimoinnista, kasvuteoriasta ja Central Pattern Generationista. Tämän lisäksi tiedonpuute kävelemisen ominaisuuksista sekä tekniset haasteet sen toteuttamisessa, rajoittavat vielä nykyäänkin monien toteutuksien toimintaa (Mikolajczyk ja muut, 2019, Kappale 2.1). Tämän vuoksi näille systeemeille ei ole yhtä parasta toteutusta ja eri projektit käyttävätkin eroavia lähestymistapoja käytetystä laitteistosta riippuen.

Laitteistossa suurin rajoittaja ihmismäisen kävelemisen toteuttamiselle on pitkään ollut prosessointiteho (Shishir, 2022). Robotit vaativat nopeita systeemejä niiden toiminnan takaamiselle, koska niiden tulee vastata niiden käyttöympäristön, raajojen vapausasteiden, epälineaarisen liikkeen ja epästabiilin rakenteen tuottamiin haasteisiin (Mikolajczyk ja muut, 2019, Kappale 1). Fyysisiä malleja joudutaankin siksi yleensä pelkistämään prosessointinopeuden ja systeemien yksinkertaistamisen vuoksi (Mikolajczyk ja muut, 2019, Kappale 2.3). Jalkojen määrällä on myös vaikutusta käytettyihin metodeihin. Kaksijalkaisiin robotteihin verrattuna useampijalkaisilla roboteilla on parempi kantokyky ja vähemmän ongelmia tasapainon ylläpitämisessä eri ympäristöissä, mutta suurempi kompleksisuus liikesuunnittelun ja koordinaation puolella, joka yleensä lisää virrankulutusta ja vaadittua prosessointitehoa (Rubio ja muut, 2019, Walking or legged mobile robots).

Navigation voidaan yleisesti jakaa kolmeen osa-alueeseen: Reitinlaskentaan, liikkeenlaskentaan ja tasapainottamiseen (Castro, 2019), jotka voidaan toteuttaa joko online- tai offline-suunnitteluna. Online-suunnittelussa robotti reagoi dynaamisesti sen ympäristössä tapahtuviin muutoksiin ja reittiä suunnitellaan paloittain sen liikkua (Raja & Pugazhenti, 2012, On-line pathplanning algorithms). Offline-suunnittelussa robotin reitti ja liikkeet taas suunnitellaan koko tunnetulle matkalle ennen liikkeellelähtöä, jonka seurauksena robotti soveltuu vain muuttumattomiin ympäristöihin. Reitinlaskenta ja liikkeenlaskenta käyttävät usein samoja menetelmiä (Rubio ja muut, 2019, Map representation).



Kuva 4. Atlaksen virtuaalinen mallinnus ympäristöstään, jossa vihreät jalanjäljet merkkäavat suunniteltuja astumakohtia (Boston Dynamics, n.d -a).

Reitinlaskenta on ensimmäinen navigaation vaihe. Tässä vaiheessa robotille suunnitellaan reitti määränpäähensä ympäristössään ilman että fyysisiä vaikutuksia tai robotin raajojen liikkeitä huomioidaan (Rubio ja muut, 2019, Navigation: Map representation). Paras reitti arvioidaan ympäristön ominaisuuksien perusteella, kuten esteiden tiheyden tai pinnanmuotojen avulla, käyttäen yleensä joko 2D- tai 3D-

mallinnusta. Tässä vaiheessa esimerkiksi Artificial Potential Fieldiin tai hiukkasparvioptimointiin (Particle Swarm Optimization) perustuvat metodit ovat suosittuja (Raja & Pugazhenti, 2012).

Toinen vaihe on liikkeenlaskenta, jossa robotille pyritään laskemaan parhaimmat mahdolliset jalan astumakohdat ajoituksineen ja niihin perustuvat jalkojen liikkeet. Castro (2019) jakaa nämä vaiheet vielä erikseen askelsuunnitteluksi ja kuvionluonniksi. Askelsuunnittelussa lasketaan robotille toimivat askelkohdat, kun taas kuvionluonnissa robotin jaloille pyritään ensin tuottamaan yksinkertainen liikekuvio yksinkertaistetulla mallilla, jonka jälkeen jalan asennot voidaan laskea esimerkiksi käänteiskinematiikkaa käyttäen (Castro, 2019). Yksinkertaisissa malleissa keskitytään yleensä vain massakeskipisteen dynamiikkoihin (Mesesan ja muut,). Liikkeenlaskennassa pyritään halun mukaan minimoimaan reitin seuraamiseen kulutettu aika, siihen kulutettu energia tai liikkeestä aiheutuvat ”nykäykset” eli liikkeet, jotka vaikuttavat negatiivisesti robotin tasapainoon (Rubio ja muut, 2019, Navigation: Map representation). 2000-luvulla myös koneoppiminen on noussut vaihtoehdoksi robotin liikkeiden tuottamiselle tavanomaisten algoritmien lisäksi. Koneoppimiseen perustuvat robotit pystyvät selviytymään huomattavasti monimutkaisemmista tehtävistä sekä käyttämään prosessoimatonta sensoridataa hyväkseen, mutta kärsivät koneoppimisen yleisistä ongelmista kuten ylisovittamisesta ja selittämättömyydestä (Castro, 2019).

Navigaation kolmas vaihe on tasapainotus. Tasapainotus on online-suunnittelussa aktiivinen suljetun silmukan prosessi, joka käyttää hyväkseen sekä simulaatiota että robotin havainnointia (Castro, 2019), se siis tapahtuu rinnan reitin- ja liikkeenlaskennan kanssa. Sillä parannetaan liikesuunnittelussa tuotettujen liikkeiden tasapainoa (Castro, 2019) sekä vastataan odottamattomiin asennonmuutoksiin. Tasapainottamisesta vastaavia systeemejä tuottaessa joudutaan vastaamaan kolmeen ongelmaan: Miten robotin vapausasteita käsitellään ja kuinka paljon niitä yksinkertaistetaan, miten varmistetaan tuotetun liikeradan tasapaino ja kuinka tarkasti lopullisten liikkeiden onnistumista seurataan (Mikolajczyk ja muut, Kappale 4.2). Tämä aiheuttaa osittain

ristiriitaiset vaatimukset niiden tuotannolle, sillä lopullisen systeemin tulee olla tarpeeksi yksinkertainen, jotta yleisiä malleja voidaan soveltaa siihen, mutta tarpeeksi monimutkainen, jotta systeemin tarkkuus säilyy (Castro, 2019). Lopullisen mallin toimintaa voidaan yleensä arvioida virheestä robotin suunnitellun ja lopullisen paikan välillä (Rubio ja muut, 2019, Cognition and control system).

4.2 Ohjauksen ja tasapainotuksen metodeja

Ohjauksen päätehtäviä ovat robotin eri osien välinen koordinaatio, sen mekaanisen rakenteen seuraaminen ja kontrollointi sekä sen yleinen päätöksenteko (Rubio ja muut, 2019, Cognition and control systems). Ohjaussysteemit voivat käyttää tavanomaisia matemaattisia systeemejä, sumeaan logiikkaan perustuvia systeemejä tai ne voidaan korvata tekoälyllä ja koneoppimisella (Aslan ja muut, 2023, Introduction). Systeemit käyttävät ohjaukseen esim. Computed Torque Control -metodeja, Sliding Mode Control -metodeja, adaptiivisia metodeja, Linear Quadratic -ohjausta tai Model Predictive Controllia (Rubio ja muut, 2019, Cognition and control systems). Nämä systeemit ohjaavat robotin raajoja joko paikkaohjausta tai vääntömomenttiohjausta käyttäen. Paikkaohjauksessa robotin jalan paikkaa ohjataan tarkoin koordinaatein, kun taas vääntömomenttiohjauksessa raajaa siirretään haluttuun suuntaan, kunnes raajan sensorit havaitsevat tarpeeksi vastustusta liikkeelle (Mesesan ja muut, 2019, Introduction). Vääntömomenttiohjaus mahdollistaa vapaamman liikkeen, jossa robotin raajat voivat tarvittaessa poiketa liikeradasta. Tämän vuoksi varsinkin sumeaan logiikkaan perustuvat modernit metodit vaativatkin vääntömomenttiohjausta niiden toiminnalle.

Robotin tasapainon ylläpitäminen lasketaan ohjaussysteemien tehtäväksi. Matyszczak (2023) kertoo, kuinka varhaisimmat neli- ja kuusijalkaiset robotit käyttivät staattisesti stabiilia liikettä, jossa robotin massakeskipiste pidettiin aina stabiililla alueella pitämällä tarpeeksi monta jalkaa kontaktissa maan kanssa. Tämä mahdollisti robotin pysäyttämisen missä tahansa liikkeen vaiheessa, koska robotti oli aina tunnetusti stabiili. Kaikki kaksijalkaiset ja monet nelijalkaiset kävelevät robotit ovat kuitenkin dynaamisesti

stabiileja, joka tarkoittaa, että ne joutuvat liikkueessaan korjaamaan massakeskipistettään ja liikkeen yhtäkkinen pysäyttäminen johtaa kaatumiseen (Matyszcak, 2023). Tämänlainen liike on lähempänä ihmisen tasapainon toimintaa (Mesesan ja muut, 2019) sekä mahdollistaa juoksemisen ja yleisesti korkeamman liikenopeuden. Aslan ja muut (2023, Push-Recovery controllers) esittelivät kolme strategiaa tasapainotukselle: nilkka-, lonkka- ja askelstrategia. Nilkkastrategia soveltuu heikkoihin tasapainon muutoksiin, jossa tasapaino voidaan ylläpitää muuttamalla nilkkojen vääntömomenttia. Lonkkastrategiassa käytetään nilkkojen ja lonkan liikettä vastaamaan voimiin koko ruumiin tuottamalla käänteisvoimalla. Askelstrategiassa käytetään Capture Point -metodia etsimään uusi jalan astumapaikka, joka antaa robotille uuden tukipisteen (Aslan ja muut, 2023, Push-Recovery controllers).

Tasapainon ylläpitämisessä yleisiä menetelmiä ja konsepteja ovat esimerkiksi Zero Moment Point, Inverted Pendulum, Capture Points ja Model Predictive Control (Shishir, 2022).

Zero Moment Point eli nollaliikepiste, on vuonna 1969 keksitty menetelmä tasapainottamiselle. Robotin jalanpohjille lasketaan joka hetki nollaliikepiste eli paikka, jonka kohdalla robotin kehon horisontaalinen vääntömomentti on nolla (Shishir, 2022). Jos robotin jalanpohjat pysyvät aina nollaliikepisteen kohdalla, sen ei pitäisi kaatua.

Inverted Pendulum -menetelmät, kuten Linear Inverted Pendulum (LIP) ja Spring Loaded Inverted Pendulum (SLIP), mallintavat robotin yläkehon tasapainottamista jalkojen liikkeen avulla robotin liikkueessa. LIP mallintaa kävelemistä, kun taas SLIP mallintaa juoksemista.

Capture Point -menetelmä perustuu capture pointtien jatkuvaan laskemiseen robotin ympärille. Capture pointit ovat paikkoja robottia ympäröivän capture regionin sisällä, joita vasten robotti voi nojata jalallaan pysäyttääkseen liikkeensä (Pratt ja muut, 2006).

Model Predictive Control on menetelmä, jossa robotin tulevia liikkeitä yritetään ennustaa systeemin tietojen perusteella. Arvioinnin pohjalta robotin liikkeitä voidaan optimoida ja tasapainoa parantaa. Menetelmää käyttää esimerkiksi Boston Dynamicsin Atlas.

4.3 Reitinlaskennan ja liikkeenlaskennan metodeja

Reitinsuunnittelu ja liikkeensuunnittelu jakavat suuren osan metodeistaan (Rubio ja muut, 2019, Navigation). Yleisesti suunnittelumetodit voidaan jakaa kolmeen eri kategoriaan: klassisiin metodeihin, evoluutioalgoritmeihin ja koneoppimiseen (Liu ja muut, 2023, Kappale 3). Rubio ja muut (2019, Navigation) erottelevat tämän lisäksi vielä todennäköisyyspohjaiset tiekartat ja heuristiset suunnittelijat omiksi kategorioikseen (Rubio ja muut, 2019, Navigation), mutta yleisesti näitä voidaan pitää osana klassisia metodeja (Liu ja muut, 2023, Kappale 3.1).

Klassiset metodit ovat aikaisimpia robottien navigaatioon sovellettuja metodeja. Niistä käytetyimpiä ovat esimerkiksi tiekarttaan perustuvat Voronoi Diagram- ja Visibility Graph -metodit (Rubio ja muut, 2019, Navigation) sekä potentiaalifunktioihin perustuvat Artificial Potential Field- ja Cell Decomposition -metodit (Raja & Pugazhenti, 2012). Klassiset metodit ovat yksinkertaisuutensa vuoksi helppoja soveltaa, mutta aiheuttavat kompleksisissa ympäristöissä ongelmia yleensä jumiutumisen tai prosessointitehon kulutuksen takia (Rubio ja muut, 2019, Navigation).

Voronoi diagrammi. Malli robotin ympäristöstä jaetaan alueiksi eli "soluiksi" ja jokaista estettä merkataan pisteinä, jolle luodaan symmetrinen seinämä sen naapuripisteiden kanssa. Esteisiin osuvat seinämät poistetaan mallista ja jäljellejääviä seinämiä käytetään polkuina navigaatioissa.

Visibility Graph. Esteiden nurkkia kuvataan pisteinä ja jokaisesta pisteestä piirretään suora kaikkiin muihin siitä näkyviin pisteisiin, joka tuottaa robotille kuljettavia polkuja. Koska robottia kuvataan myös pisteinä avaruudessa, kaikkien esteiden kokoa joudutaan liioittelemaan.

Cell Decomposition. Robotin malli jaetaan valinnaisten akselien suhteen sektoreiksi ja esteiden sisälle jäävät alueet jätetään tyhjiksi. Robotti navigoi sektoreiden reunoja pitkin.

Artificial Potential Field. Robotin ja sitä ympäröivien esteiden simuloidaan tuottavan magneettinen voima, jolla on sama polaarisuus. Robottia ohjataan esteiden ympäri simuloimalla saman- ja eripolaarisia magneettisia voimia. Malli voi herkästi jäädä jumiin paikalliseen minimiin.

Heuristic Planners. Kaaviohakuun perustuvia heuristic plannereita käytetään joskus rinnan muiden klassisten metodien kanssa (Liu ja muut, 2023, Kappale 3.1). Niillä etsitään ympäristössä olevista poluista nopein reitti. Heuristic plannereihin kuuluvat esimerkiksi A*-, Djikstra- ja Greedy-algoritmit.

Todennäköisyyspohjaisia tiekarttoja kehitettiin korvaamaan klassisia metodeja, jotka jumiutuivat usein paikalliseen minimiin (Raja & Pugazhenti, 2012, Off-line pathplanning algorithms). Niiden toiminta perustuu yksiulotteisen todennäköisyysperäisen tiekartan käyttöön, jossa aloitus- ja loppupiste yritetään liittää etsimällä reittejä niiden välisestä tiekartasta.

Simulated Annealing perustuu numeeriseen tekniikkaan. Siinä satunnaisesti simuloitujen liikkeiden pituutta määränpäättä kohti ohjataan "lämpötilalla", jonka arvoa pienennetään optimoitua tulosta lähestyessä.

Evoluutioalgoritmien soveltaminen reitin- ja liikkeensuunnitteluun alkoi noin 2000-luvun alussa. Ne yrittävät soveltaa biologista käyttäytymistä ja parviälyä parhaan reitin löytämiseen (Liu ja muut, 2023, Kappale 3.2).

Geneettisten Algoritmien inspiraationa on geneettinen periytyminen ja luonnonvalinta. Ne pyrkivät ympäristön ominaisuuksien perusteella valitsemaan parhaan satunnaisesti luoduista vaihtoehdoista.

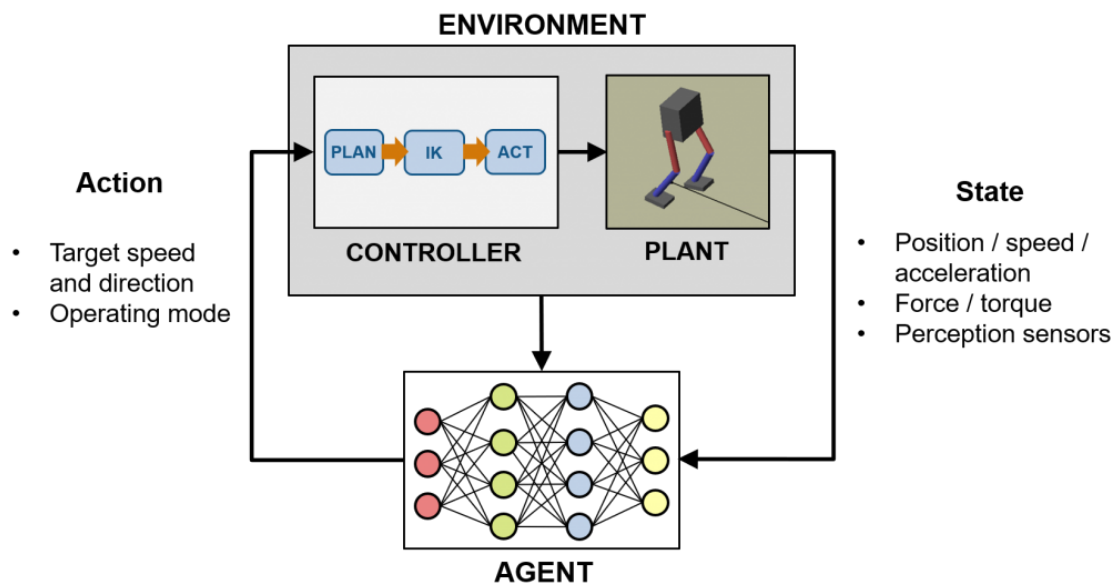
Particle Swarm Optimization tutkii iteratiivisesti tilaa käyttämällä partikkeleita, joilla on aluksi satunnaisesti arvottu liikevektori. Paras tulos valitaan iteraatioiden välillä geneettisten algoritmien tapaan.

Ant Colony Optimization perustuu muurahaisissa havaittuun reittien muodostamiseen. Simuloidut ”muurahaiset” valitsevat reittinsä pisteiden välillä satunnaisesti ja merkkäavat nopeimmiksi havaitut reitit muille. Paras reitti on se, jota käytettiin eniten.

Edellä mainittujen metodien lisäksi on olemassa joitakin törmäyksenestossa ja esteiden havainnoinnissa käytettyjä algoritmeja. Rubio ja muiden (2019, Navigation) mukaan karttaa käyttävät algoritmit voivat käyttää neljää eri metodia esteiden tunnistamiseen: Space-time volume interception, swept volume interference, multiple interference detection tai trajectory parametrization. Kartattomat algoritmit taas luottavat täysin sensorihavainnointiin ja käyttävät esim. Bug Algoritmia, Dynamic Window Approachia tai Vector Field Histogrammia (Rubio ja muut, 2019, Navigation).

4.4 Koneoppiminen mobiiliroboteissa

Suorituskehitys on mahdollistanut myös koneoppimisen käytön monissa eri mobiilirobottien tehtävissä. Reitin- ja liikkeenlaskennassa koneoppiminen ei tavanomaisiin metodeihin verrattuna käytä monia niissä yleisiä strategioita, kuten massakeskipisteen laskentaa (Shishir, 2022). Liikkeenlaskennassa koneoppiminen yrittää päästä toimivaan liikestrategiaan iteraation kautta; malli tuottaa satunnaisia liikkeitä fyysisessä simulaatiossa ja tietyt parametrit täyttäviä tuloksia suositaan. Koneoppiminen ja tavanomaiset metodit voidaan tarvittaessa kuitenkin myös yhdistää, jolloin ne voivat esimerkiksi arvioida toistensa tuottamien liikestrategioiden toimivuutta (Castro, 2019).



Kuva 5. Koneoppimista käyttävä yhdistelty liikestrategia, jossa AI-agentti tuottaa robotille liikeradan ja tavalliset metodit seuraavat sitä laskiessaan robotin jalkojen liikkeitä (Castro S., 4.4.2019).

Robottien koneoppimisessa sovelletaan yleisesti joko syvävahvistusoppimista (Deep Reinforcement learning) tai ohjattua oppimista (Supervised learning) (Castro, 2019). Koneoppimismallit voivat korvata melkein minkä tahansa robotin toiminnan vaiheen, kuten ilmenee Rybczakin ja muiden (2024) toteuttamasta katsauksesta koneoppimisen käyttöön mobiiliroboteissa, jossa he kertovat eri mallien käyttötarkoituksista. Rybczakin ja muiden mukaan ohjattu oppiminen jakautuu yleisesti regressio- (regression) ja klassifikaatio (classification) metodeiksi, jotka soveltuvat esimerkiksi esteiden havainnointiin ja välttämiseen, reitinsuunnitteluun ja liukumisen ennustamiseen (slippage prediction). Vahvistusoppiminen perustuu simuloituun agenttiin, joka yrittää iteratiivisen mallintamisen avulla löytää toimivimman strategian; parhaiten vaatimukset täyttäviä eleitä suositaan mallissa. Se jakautuu value-based-, policy-based- ja actor-critic-metodeihin. Ohjaamattomalla oppimisella etsitään yleensä yhteyksiä datassa, sitä käytetään esimerkiksi ryhmittelyyn (clustering) tai poikkeuksien etsimiseen, jonka takia sitä käytetään pääasiassa havainnoinnissa. Lopuksi Rybczak ja muut kertovat vielä osittain ohjatun oppimisen (semi-supervised learning) olevan harvinaisempaa mobiiliroboteissa muihin metodeihin verrattuna, mutta sitä voidaan soveltaa esimerkiksi

roboteille tarkoitettussa konenäössä tai puheentunnistuksessa (Rybczak ja muut, 2024). Rybczakin ja muiden (2024) vertailun pohjalta yleisten algoritmien käyttöä vertaillaan alla olevassa taulukossa (Taulukko 2).

Taulukko 2. Yleiset koneoppimismallit ja niiden käyttötarkoitukset, perustuen Rybczakin ja muiden (2024) vertailuun.

Algoritmi	Oppimismetodi	Käyttö
Neural network	Ohjattu oppiminen	Pintojen ja esteiden tunnistus, reitinlaskenta ja liikkeenlaskenta
Päätöspuu	Ohjattu oppiminen	Reitinlaskenta
Tukivektorikone	Ohjattu oppiminen	Liikkeenlaskenta
Random forest	Ohjattu oppiminen	Pintojen tunnistus
Q-Learning	Vahvistusoppiminen	Reitinlaskenta ja esteiden välttäminen
Deep Q-Network	Vahvistusoppiminen	Reitinlaskenta
Actor-critic	Vahvistusoppiminen	Liikkeenlaskenta ja esteiden välttäminen
k-means	Ohjaamaton oppiminen	Ympäristön kartoittaminen ja havainnointi
DBSCAN	Ohjaamaton oppiminen	Clustering ja havainnointi

Koneoppimisen käyttämisellä voi olla huomattavia etuja tavanomaisiin menetelmiin verrattuna. Koneoppimisalgoritmit kykenevät suoraan käyttämään sensoridataa ilman lisäprosessointia ja pystyvät helpommin jäljentämään komplekseja eleitä, joiden toteuttaminen tavanomaisin keinoin olisi hyvin vaikeaa (Castro, 2019). Koneoppimisesta on myös tuotettu huomattavasti enemmän vapaasti saatavilla olevaa tutkimusmateriaalia tavanomaisiin ohjausmetodeihin verrattuna, jonka seurauksena kynnys niiden toteuttamiselle voi olla matalampi (Shishir, 2022). Sen käyttö robottien arkkitehtuureissa on selvästi kasvussa ja se voi korvata joitain yleisesti käytetty metodeja tulevaisuudessa (Rubio ja muut, 2019).

5 Yhteenveto

Tutkielmassa selvitettiin mitkä ovat oleelliset asiat kävelevien mobiilirobottien toiminnassa. Tätä varten tarkasteltiin niiden kohtaamia haasteita sekä etsittiin niissä käytettyjä menetelmiä ja joitain niiden yleisiä toteutuksia.

Kävelevä rakenne mahdollistaa paremman liikkuvuuden ja auttaa robottia selviytymään ihmisille suunnitelluissa- tai muuten hankalakulkuisissa ympäristöissä. Tulevaisuuden kannalta keskeisimpiä ongelmia ovat niiden tuotannon huono kustannustehokkuus ja niiden rakenteesta aiheutuvat haasteet niiden kestävyydelle ja energiatehokkuudelle. Tämän lisäksi yleinen tiedonpuute robottien kävelemisen optimoinnista sekä mahdolliset eettiset haasteet niiden käytölle hankaloittavat niiden käyttöönottoa. Rajoituksista huolimatta robottituotanto kehittyy nopeasti ja yleinen kiinnostus sitä kohtaan on selvästi kasvussa. Kävelevillä roboteilla ei ole sovittuja tuotantotapoja, jonka seurauksena niissä käytettävät lähestymistavat ovat usein hyvin erilaisia. Syvyyden havainnointi LIDARin avulla tuottaa suurimman osan robotin toiminnan kannalta tärkeästä datasta. Lokalisaatioon ja ympäristön kartoittamiseen käytettävät SLAM-algoritmit vaihtelevat havainnoinnin metodien mukaan, mutta sisätiloissa niistä yleisin on LIDARIA ja matkamittaria käyttävä GMapping-algoritmi. Reitinlaskennan ja liikkeensuunnittelun käyttämät menetelmät jaetaan nykyään joko koneoppimiseen tai klassisiin ja evoluutioalgoritmipohjaisiin lähestymistapoihin. Koneoppimista voidaan myös soveltaa korvaamaan tavanomaisia menetelmiä useissa robottien osa-alueissa, sen käyttöä ovat helpottaneet ja lisänneet suoritustehon kehitys ja vapaasti saatavilla oleva tutkimusmateriaali.

Tutkielman katsaus joihinkin robottien osa-alueisiin oli rajoittunut ja jätti mahdollisuuksia lisätutkimukselle. Tutkielmää varten ei löytynyt sopivaa materiaalia ulkotiloissa käytettävistä SLAM-algoritmeista, jonka seurauksena esimerkiksi sisä- ja ulkotiloissa käytettävien SLAM-algoritmien eroihin ei perehdytty. Tekoälyn käyttöä kävelevissä roboteissa käsiteltiin myös hyvin pintapuolisesti, koska sen integraatio robotteihin on aiheena hyvin laaja ja olisi vienyt huomattavasti aikaa.

Kävelevien robottien toteuttaminen on haastavaa ja vaatii paljon ammattitaitoa ja tuntemusta eri aloilta, joka rajoittaa niiden tuotantoon kykenevien yritysten ja tutkimusryhmien määrää. Tulevaisuudessa tekoälyyn perustuva semanttinen kartoitus tulee varmasti olemaan robottien toiminnan kannalta tärkeä ominaisuus, sillä se mahdollistaa aikaisempaa monimutkaisempien toimintojen lisäämisen robotteihin. Tekoälyä ja koneoppimista käyttävien systeemien määrä roboteissa tulee varmasti myös lisääntymään tulevaisuudessa, korvaten joitain nykyisiä menettelytapoja, joka voi tekoälyn kehittyessä madaltaa myös kynnystä niiden robottituotannolle. On vaikeaa arvioida milloin kävelevien robottien laajempi käyttöönotto voi tapahtua, mutta yhteiskunnan ja yritysten kannattaisi varautua sen mahdollisuuteen mahdollisimman pian. Tätä varten jatkotutkimus esimerkiksi robottien eettiseen käyttöön ja tietoturvaan niiden kanssa työskenneltäessä olisi hyödyllistä.

Lähteet

- Aslan, E., Arserim, M. A., & Uçar, A. (2023). *Development of push-recovery control system for humanoid robots using deep reinforcement learning*. Science Direct.
<https://doi.org/10.1016/j.asej.2023.102167>
- Bartoszyk, S., Kasprzak, P., & Belter, D. (2017). *Terrain aware motion planning for a robot*. IEEE. <https://doi.org/10.1109/RoMoCo.2017.8003889>
- Bekey, G. A. (2005). *Autonomous robots: From biological inspiration to implementation and control*. The MIT Press.
- Boston Dynamics (n.d. -a) *Atlas and beyond: the world's most dynamic robot*. Noudettu 12.10.2024 osoitteesta <https://bostondynamics.com/atlas/>
- Boston Dynamics (n.d. -b) *Boston dynamics ethical principles*. Noudettu 9.1.2025 osoitteesta <https://bostondynamics.com/ethics/>
- Böttcher, S. (2006). *Principles of robot locomotion*. Noudettu 28.12.2024 osoitteesta <https://www2.cs.siu.edu/~hexmoor/classes/CS404-S09/RobotLocomotion.pdf>
- Castro, S. (2019, 24. Huhtikuuta). *Walking robot control: From PID to reinforcement learning*. Mathworks. Noudettu 26.9.2024 osoitteesta <https://blogs.mathworks.com/student-lounge/2019/04/24/walking-robot-control/>
- International Centre for Theoretical Sciences. (30.10.2022). *History of walking robots by Shishir N. Y. Kolathya*. Youtube. Noudettu 12.10.2024 osoitteesta https://www.youtube.com/watch?v=1G6cWc_KvOE
- Jiao, J., Geng, R., Li, Y., Xin, R., Yang, B., Wu, J., Wang, L., Liu, M., Fan, R., Kanoulas, D. (2024). *Real-time metric-semantic mapping for autonomous navigation in outdoor environments*. IEEE. <https://doi.org/10.1109/TASE.2024.3429280>
- Kaneko, K., Kaminaga, H., Sakaguchi, T., Kajita, S., Morisawa, M., Kumagai, I. & Kanehiro, F. (2019). *Humanoid Robot HRP-5P: An Electrically Actuated Humanoid Robot With High-Power and Wide-Range Joints*. IEEE. <https://ieeexplore-ieee-org.proxy.uwasa.fi/document/8630006>
- Kiran, B., Karthikeyan, S., Suhel Pasha, M. A., Manjunatha, K. N., Manoj Kumar, S., & Moras, S. V. (2022). *Design and Development of Autonomous Mobile Robot for Mapping and Navigation System*. IEEE. <https://doi.org/10.1109/PuneCon55413.2022.10014944>

- Li, Z. -X., Cui, G. -H., Li, C. -L., & Zhang, Z. -S. (2021) *Comparative Study of Slam Algorithms for Mobile Robots in Complex Environment*. IEEE.
<https://doi.org/10.1109/CRC52766.2021.9620122>
- Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. Science Direct.
<https://doi.org/10.1016/j.eswa.2023.120254>
- Mathworks. (n.d.). *What is SLAM?*. Noudettu 16.6.2024 osoitteesta
<https://se.mathworks.com/discovery/slam.html>
- Matyszczak, J. (2023, 5. Huhtikuuta). *Legged robots: Keeping the balance*. MAB Robotics.
Noudettu 19.10.2024 osoitteesta <https://www.mabrobotics.pl/post/legged-robots-keeping-the-balance>
- Mesanan, G., Engelsberger, J., Garofalo, G., Ott C., & Albu-Schäffer, A. (2019) *Dynamic Walking on Compliant and Uneven Terrain using DCM and Passivity-based Whole-body Control*. IEEE. <https://doi.org/10.1109/Humanoids43949.2019.9035053>
- Mikolajczyk, T., Mikołajewska, E., Al-Shuka, H.F.N., Malinowski, T., Kłodowski, A., Pimenov, D.Y., Paczkowski, T., Hu, F., Giasin, K., Mikołajewski, D. & Macko, M. (2022). *Recent Advances in Bipedal Walking Robots: Review of Gait, Drive, Sensors and Control Systems*. Sensors.
<https://doi.org/10.3390/s22124440>
- Pratt, J., Carff, J., Drakunov, S., & Goswami, A. (2006). *Capture Point: A Step toward Humanoid Push Recovery*. IEEE. <https://doi.org/10.1109/ICHR.2006.321385>
- Raja, P., & Pugazhenti, S. (2012) *Optimal path planning of mobile robots: A review*. Academic Journals. <https://doi.org/10.5897/IJPS11.1745>
- Rubio, F., Valero, F., & Llopis-Albert, C. (2019). *A review of mobile robots: Concepts, methods, theoretical framework, and applications*. Sage Journals.
<https://doi.org/10.1177/1729881419839596>
- Rybczak, M., Popowniak, N., Lazarowska, A. (2024) *A survey of machine learning approaches for mobile robot control*. MDPI. <https://doi.org/10.3390/robotics13010012>
- Song, X., Liang, X., Zhijiang, Z., & Huaidong, Z. (2022) *A Object-augmented Semantic Mapping System for Indoor Mobile Robots*. IEEE.
<https://doi.org/10.1109/SEAI55746.2022.9832075>

Unlimited Robotics (n.d.) The ethical implications of using robots in the workplace. Noudettu 6.1.2025 osoitteesta <https://www.hospital-robots.com/post/the-ethical-implications-of-using-robots-in-the-workplace>

Walas, K. (2014). Terrain classification and negotiation with a walking robot. Springer. <https://doi.org/10.1007/s10846-014-0067-0>

Waseda University. (n.d). *Development of Waseda robot*. Noudettu 24.6.2024 lähteestä https://www.humanoid.waseda.ac.jp/booklet/kato_2.html

Wei, L., Wang, Z., Jia, W., Yuan, J., Ma, S., & Li, L. (2019). Practical Vision-Based Walking Navigation for the Humanoid Robot NAO in the Maze-like Environment. IEEE. <https://doi.org/10.1109/ROBIO49542.2019.8961729>