

Mika K Ahonen

**ROS - Robotiikan tutkimustyökalusta kohti kaupallista
menestystarinaa**

Tietotekniikan kandidaatintutkielma

12. huhtikuuta 2017

Jyväskylän yliopisto

Tietotekniikka

Tekijä: Mika K Ahonen

Yhteystiedot: mika.k.ahonen@gmail.com

Työn nimi: ROS - Robotiikan tutkimustyökalusta kohti kaupallista menestystarinaa

Title in English: ROS - From developer's tool towards a commercial success

Työ: Kandidaatintutkielma

Sivumäärä: 24+0

Tiivistelmä: Avoimen lähdekoodin ROS-käyttöjärjestelmä on yleisesti tutkimushankkeissa käytetty robotiikan kehitysalusta. Tutkielman tavoitteena on ollut arvioida onko ROS järjestelmä, joka tulee yleistymään myös robotiikan kaupallisissa ratkaisuisissa. ROS-pohjaisia kaupallisia tuotteita on jo olemassa ja sen käyttöä on pilotoitu runsaasti erilaisilla sovellusalueilla. Erityisesti palvelu- ja pilvirobotiikan alueilla ROSilla on kasvavaa potentiaalia. ROS on vielä kehittyvä järjestelmä ja ei nykyisellään sovellu kriittisimpiin robotiikan tehtäviin. Kuitenkin järjestelmän tuleva versio ROS 2.0 antaa lupauksia siitä, että teknologia on sovellettavissa yhä useimmilla laitealustoilla ja sovellutuksissa tulevaisuudessa.

Avainsanat: ROS, robotiikka, avoin lähdekoodi

Abstract: ROS (Robot Operating System) is a common open source framework for developing robots. The platform is currently used by many research organizations and projects and this literature review tries to estimate whether the platform is ready for commercial products. There are a few ROS-based products available and many pilots have been made in different fields. It seems that especially service and cloud robotics are growing fields where ROS has a lot of potential. ROS is a system which is still developing and it is not yet ready for critical real-time operations. However, the future version, ROS 2.0, gives promises for increased possibilities for the system to be used in different applications and on different platforms.

Keywords: ROS, robotics, open source

Sisältö

| | | |
|-----|--|----|
| 1 | JOHDANTO | 1 |
| 2 | ROBOTTIKÄYTTÖJÄRJESTELMÄ | 2 |
| 2.1 | Sanomanvälitys | 3 |
| 2.2 | Työkaluja | 4 |
| 2.3 | Käyttöönotto | 5 |
| 3 | ROBOTIIKAN SOVELLUSALUEITA | 6 |
| 3.1 | ROS teollisuusrobotiikassa | 6 |
| 3.2 | Palvelurobotiikan pilotteja | 7 |
| 3.3 | ROS maalla, vesillä ja ilmassa - Autonomisesti liikkuvat laitteet ja kulku- välineet..... | 8 |
| 3.4 | ROS pilvirobotiikassa | 9 |
| 3.5 | Robotiikan kouluttajana | 10 |
| 4 | TULEVAISUUDEN HAASTEET JA MAHDOLLISUUDET | 12 |
| 5 | YHTEENVETO..... | 14 |
| | LÄHTEET | 15 |

1 Johdanto

Avoimen lähdekoodin robottikäyttöjärjestelmä (ROS, engl. *Robot Operating System*) (Quigley ym. 2009) on tutkimushankkeissa yleisesti käytetty robotiikan kehitysalusta. Avoimen lähdekoodin ohjelmistoalustana sen käyttö on edullista, tutkimustulokset avoimesti saatavilla ja tutkimustietoa sekä kehittäjäyhteisön kontribuoituja sovellusmoduleja runsaasti tarjolla. Erinäisiä ROS-pohjaisia kaupallisia robotiikan ratkaisuja on olemassa, mutta kuitenkin erityisesti teollisuuden robotiikassa on suurelta osin käytössä toimittajakohtaisia suljettuja ohjelmistoratkaisuja ja palvelurobotiikan ensi askeleissa on havaittavissa jotain vastaavaa. Robotiikka on murrosvaiheessa, jossa siirrytään kohti älykkäämpiä ratkaisuja. Aiemmin robotin rakentaminen on ollut erittäin kallista ja aikaavievää. Kehittyneen robotiikan teknologiat, kuten tekoäly, puettava teknologia ja halventuneet sensortechnologiat mahdollistavat nyt nopeamman ja edullisemmän tavan rakentaa toteutuksia. (Fu ja Zhang 2015.)

Tutkielma esittelee yleisesti ROS-käyttöjärjestelmää ja sen nykytilaa, ajankohtaisia tutkimushankkeita ja pilotteja sekä kaupallistettavissa olevia ratkaisua mitä kirjallisuudesta on löydettävissä. Tutkielma pyrkii osaltaan silmäilemään tulevaisuuteen, onko ROS teknologia, jonka rooli vain vahvistuu tulevina vuosina robotisaatiokehityksessä ja kaupallisissa ratkaisuissa. Tutkielmassa on kartoitettu uusinta ROS tutkimustietoa, niin teollisuus- ja palvelurobotiikkaan kuin myös koulutukseen liittyen. Tämän kirjallisuuskatsauksen tarkoitus on osaltaan tuoda näkyväksi niitä kehittämispyrkimyksiä, joita ROS-järjestelmän ympärillä on tapahtumassa.

ROS on kehittyvässä tilassa oleva alusta ja aiheesta on olemassa hyvin uutta tutkimustietoa. Aiheesta löytyy kuitenkin melko vähän suomenkielistä kirjallisuutta, johon tutkielma antaa oman panoksensa. Aluksi kerromme yleisesti ROSista. Sen jälkeen tutustumme erilaisiin sovellutuksiin, kurkistamme tulevaan ja lopussa pohdimme opimme jotain.

2 Robottikäyttöjärjestelmä

ROS on kehitysalusta robotiikan ohjelmointiin. Se on avoimen lähdekoodin kokoelma työkaluja, ohjelmakirjastoja ja algoritmeja. Viimeisin ROS -distribuutio Kinetic Kame sisältää yli 3000 ohjelmapakettia, joita on kirjoittanut ja ylläpitää 2500 ihmisen yhteisö (*ROS Community Metrics July 2016*). Järjestelmä on käytettävissä yli sadassa robotissa (*ROS Wiki*) ja aiheesta on kirjoitettu reipas parisentoista akateemista julkaisua (Quiqley, Gerkey ja Smart 2015). ROS on käytössä niin teollisuuden manipulaattoreissa, autonomisissa liikkuvissa laitteissa kuin erilaisissa palvelurobotiikan laitteissa mm. humanoideissa (*ROS Wiki*).

Robotiikan kehittämisprojektit eivät ole helppo laji. Erilaiset robotit tarvitsevat erilaista laitetekniikkaa ja ohjelmistoja. Yksittäisen robottiprojektin ympärille on aiemmin yleisesti toteutettu omia kehitysympäristöjä. Tutkimushankkeissa pyrkimys on keskittyä rajattuihin tutkimusalueisiin, mutta projektit vaativat kuitenkin laajemman ohjelmisto- ja laiteympäristön tuekseen. ROS on kehittynyt tarpeesta keskittyä robotiikan tutkimiseen, tehdä robotiikan kehittämisestä helpompaa, nopeampaa ja yhteisöllisempää sekä pyrkiä standardoimaan ratkaisuja. Järjestelmän alkujuuret ovat vuoden 2006 Standfordin yliopiston STAIR (Standard AI Robot) –tutkimushankkeessa ja kehittämissyhtiö Willow Garagen PR2 (Personal Robot 2) –projektissa (*Willow Garage*). Vuodesta 2013 lähtien järjestelmän kehitystä on koordinoanut Open Source Robotics Foundation (*OSRF*).

ROS ei ole itsenäinen käyttöjärjestelmä, vaan toimii yleisimmin GNU/Linux - käyttöjärjestelmän rinnalla ja tarjoaa kommunikointikerroksen ohjelmiston eri osien välillä. Se on välitason ohjelmisto (middleware), joka hoitaa ohjelmistopakettien hallinnan sekä prosessien välistä sanomanvälistystä ja keskeisiä kriittisiä toimintoja, joita robottiohjelmistot vaativat toimiakseen. Sen hajautettu arkkitehtuuri koostuu pienistä ohjelmista, jotka voivat keskustella keskenään. Näin järjestelmän osia voidaan ajaa erilaisissa laitteissa yhtäaikaisesti, vikasietoisuus on parempi ja erillisten osien kehittäminen ja testaaminen on joustavampaa. ROS on ohjelmointikielistä riippumaton. Tällä hetkellä ohjelmointikielistä parhaiten tuettuina ovat C++ ja Python. ROS on avoimen lähdekoodin ohjelmisto. Sen kaikki koodi on julkisesti saatavilla BSD -lisenssillä (*BSD*), mikä mahdollistaa myös kaupalliset projektit. ROS sisältää monia valmiita työkaluja ja kirjastoja robotiikan erilaisiin sovellutuksiin esim. navigointiin,

reitit suunnitteluun ja ympäristön kartoitukseen. (Garber 2013; Quigley ym. 2009.)

2.1 Sanomanvälitys

ROS käyttää P2P-tyyppistä arkkitehtuuria prosessien väliseen kommunikointiin. Yksittäisiä ohjelmia/prosesseja ROSissa kutsutaan solmuiksi (node). Järjestelmän jokainen scripti tai luokka kääntyy ROS solmuksi. ROS käyttää sanomaväyliä/kanavia (topics) ja julkaisijatailaaja mallia sanomien (messages) lähettämiseen ja vastaanottoon. Julkaisija-solmu julkaisee yhden tai useamman sanoman väylällä/kanavalla ja solmut, jotka tilaavat kyseistä kanavaa vastaanottavat sen sanomat. Sanomanvälitys ei sisällä kuittauksia lähettäjälle sanoman perillemenosta. Tiedonsiirto tapahtuu käyttäen TCP/IP ja UDP -protokollia ja keskustelut sekä konfiguraatiot hoidetaan XML-RCP:llä. Solmut voivat sijaita hajautetusti verkon eri laitteissa ja niillä jokaisella on oma tehtävänsä järjestelmässä, esim. moottorin ohjaus, kamera, sensori tai jokin algoritmi. Järjestelmän Master-solmu hoitaa nimi- ja rekisteröitymispalvelut, pitää kirjaa järjestelmän solmuista, niiden julkaisemista ja tilaamista sanomakanavista ja mahdollistaa niiden keskinäisen kommunikoinnin. Vastausta vaativa kommunikointi ROSissa hoidetaan erityisillä palvelukutsuilla (services), jotka sisältävät viestiparin kyselyyn/vastaamiseen. (Castrén 2016.)

Keskeistä ROSissa on hajautettu arkkitehtuuri, missä järjestelmä koostuu pienistä keskenään kommunikoivista työkaluista, jotka suorittavat eri tehtäviä. Tämä tekee järjestelmän vikasietoisemmaksi, jolloin yksittäiset vikatilanteet eivät kaada koko järjestelmää. Arkkitehtuuri ja järjestelmän modulaarisuus mahdollistavat myös sen, että laskentatehoa vaativia toiminnallisuuksia voidaan hajauttaa ja että ajonaikainen moduulikohtainen kehittäminen on mahdollista. Perusjärjestelmä voidaan pitää ajossa ja keskittyä testaamisen kannalta oleellisiin asioihin.

ROSin tarkempaa teknologiakuvausta löytyy kirjallisuudesta runsaasti. Suomenkielellä aiheeseen voi perehtyä tarkemmin vaikka tutustumalla E. Castrénin diplomityön lukuun 2.1 (Castrén 2016).

2.2 Työkaluja

ROS työkalut mahdollistavat helpon pakettien käytön ja ne jaotellaan kolmeen: tiedostojärjestelmätyökalut (mm. rospack, rosmake), komentorivi-työkalut (roscore, rosruntime, roslaunch) sekä loki-työkalut (rosbag/roscpp). ROSissa on myös itsessään monia työkaluja datan prosessointiin, analysointiin, datan graafiseen esittämiseen (rxplot), ROS verkkorakenteen visualisointiin (rxgraph) ja 3D-visualisointiin (RViz). Näiden lisäksi ROS hyödyntää muita avoimen lähdekoodin ohjelmistoja liittyen sensoreihin, navigointiin, simulointiin, konenäköön ja algoritmeihin (OpenRave). (Qian ym. 2014.)

Robottiikan simulointiympäristöt ovat tärkeä osa robotiikan testausvaihetta. Tämä on turvallista, sekä aikaa ja resursseja säästävää. Gazebo (*Gazebo*) on robotiikan simulointiohjelmisto, joka integroituu ROS-järjestelmään. Gazebo mahdollistaa simulaatioiden rakentamisen, joilla voidaan nopeasti testata kehitettyjä algoritmeja ja toimintaa realistisissa tilanteissa. Hyvin simuloitu robotti ja sen toiminnallisuudet ovat siirrettävissä suoraan fyysiseen robotiikkiin. Vaikka Gazebo on itsenäinen ohjelmisto, voidaan sitä käyttää ROS ympäristön yhtenä solmuna. Robottien ja ympäristön mallintaminen toteutetaan URDF (Universal Robotic Description Format) XML-tiedostomuodossa. (Takaya ym. 2016.)

Autonomisen robotin tulee pystyä rakentamaan oma malli ympäristöstään ja sen on tiedettävä sijaintinsa. 3D-havaitsemisen merkitys lisääntyy koko ajan robotiikassa ja edullisten 3D-sensoreiden tulo markkinoille on lisännyt mahdollisuuksia. Microsoft Kinect-sensori on käytetty työkalu ROS-käyttöjärjestelmän kanssa. Point Cloud Library (PCL) on avoimen lähdekoodin algoritmi-kirjasto 3D-sensoreiden muodostamien pistepilvien prosessointiin ja mallintamiseen ja se integroituu osaksi ROS-järjestelmää (Rusu ja Cousins 2011).

MoveIt on joukko ohjelmistopaketteja, jotka yhdessä 3D-kameran kanssa integroituvat osaksi ROSia ja on suunniteltu erityisesti käsivarsirobotiikan kehittyneempiin ratkaisuihin liikkuvissa kappaleenkäsittely- ja reitinsuunnittelutehtävissä (*MoveIt*). ROS integroituu myös saumattomasti avoimen lähdekoodin OpenCV-konenäkökirjaston kanssa (*ROS Wiki*). Työkaluista on helposti löydettävissä syventävää tietoa niin kirjallisuudesta kuin ROS wiki-sivustolta.

2.3 Käyttöönotto

Tässä tutkielmassa emme syvenny järjestelmän asentamiseen ja käyttöön. Aiheesta löytyy kattavasti ohjeita ROS wiki-sivustolta (*ROS Wiki*) sekä kirjoja ja julkaisuja. Distribuutioita on useita ja järjestelmän kehityssykli nopeaa, dokumentaation osalta on osin todettavissa open source -maailmalle tyypillinen tiedon pirstaloituneisuus. Järjestelmään on mahdollista tutustua Gazebo -simulointiympäristössä, johon on olemassa monien ROS-pohjaisten robottien valmiita simulointimalleja. Järjestelmän testailuun ja opiskeluun on olemassa muutamia laiteympäristöjä mm. Turtlebot, Create 2. Turtlebot -robotista on tulossa keväällä 2017 päivitetty Turtlebot3, joka konseptina ja kohtuullisesti hinnoiteltuna vaikuttaisi selkeältä parannukselta ja hyvin laajempaankin opetuskäyttöön soveltuvalta (*Turtlebot3*). ROSiin on saatavilla myös ajurit Lego NXT robottiin.

3 Robotiikan sovellusalueita

ROS-käyttäjärjestelmä on alkujaan kehitetty palvelurobotiikan kehittämisprojekteissa. Vaajaassa vuosikymmenessä se on kuitenkin laajentunut useille sovellusalueille. Vaikka paljolti vielä ollaankin tuotekehitys- ja pilotointivaiheissa, niin kaupallisiakin sovellutuksia jo löytyy.

3.1 ROS teollisuusrobotiikassa

Nykyiset teollisuusrobotiikan järjestelmät suorittavat suurelta osin niille etukäteen ohjelmoituja tehtäviä rajatuissa ja suojaetuissa tiloissa. Toimialalla ei ole syntynyt suuria innovaatioita viime vuosikymmeninä, johtuen osin frakmentoituneista markkinoista, standardoinnin ja yhteentoimivuuden puutteista sekä pirstaloituneesta sovelluskehityksestä kehittyneissä teknologioissa (Bruyninckx 2008). Teollisuusroboteissa kehitys on kulkemassa turvallisemman cobot-yhteistyörobotiikan (Peshkin ym. 2001) suuntaan, joissa viime vuosina on syntynyt suurinta kasvua (*World robotics report*). Kehittyneemmälle yhteistoimintaan ja mukautuviin työtehtäviin sopeutuvalla ja joustavasti opetettavissa olevalle robotiikalle on lisääntyvää kysyntää, erilaisissa liikkuvissakin työtehtävissä. ROS-pohjaisella Intera5 -käyttäjärjestelmällä varustettu teollisuusrobotti Baxter ja tämän seuraaja Sawyer ovat saaneet osansa tästä huomiosta (*Rethink Robotics*).

ROS-Industrial on ROS-käyttäjärjestelmä teollisuusrobotiikan tarpeisiin. Tämä kehitystyö käynnistyi vuonna 2011 Southwest Research Institutin (SwRi), Willow Garagen ja Yaskawa Motoman Roboticsin yhteistyönä. Tavoitteena on ollut kehittää ja ylläpitää teollisuusrobotiikan tarpeisiin ohjelmistoa, jossa keskitytään ohjelmistokoodin laatuun ja saatavilla olevaan tukeen ja dokumentointiin. Pyrkimys on ollut myös kaventaa kuilua robotiikkaa valmistavan teollisuuden ja tutkimuksen välillä sekä lisätä toimialan yhteistyötä ja ymmärrystä avoimen lähdekoodin liiketoimintamalleista. Toimijat ovat perustaneet yhteisen ROS-Industrial konsortiumin, jossa on mukana useita isoja valmistusteollisuuden ja robotiikan yrityksiä. (*ROS-Industrial*.)

Erityistä kiinnostusta ROS-Industrial on herättänyt kehittyneempien materiaalinkäsittelyteh-

tävien sekä käsivarsirobotiikan ja mobiliteetin yhdistämiseen liittyen, joista on löydettävissä monia tutkimuksia (Tavares ja Sousa 2015; Venator, Lee ja Newman 2013). Robottiohjelmistojen helppokäyttöisyys on noussut opetettavien cobot-robottien osalta yhä tärkeämmäksi, ROS-Industrialiin tällaista kehitystyötä on tekemässä SERC Industrial Robotics Project (Yan ym. 2015).

Vaikuttaisi siltä, että vielä on jonkin verran matkaa siihen, että teollisuuden standardit ja laatuvaatimukset kaikilta osin ROS-käyttöjärjestelmässä täytyisivät. ROS-Industrial on kuitenkin saatavilla moniin teollisuusroboteihin ja on kiinnostava ajatus, että ohjelmistoa vaihtamalla ja tuomalla sen rinnalle uutta sensortechnologiaa, voidaan vanhalle ehkä vajaakäytössäkin olevalle robottikannalle antaa uusi elämä. Euroopassa vuoden 2017 alussa käynnistynyt laajahko ROSIN hankekokonaisuus vaikuttaisi isoimmalta kehityshankkeelta aiheen ympärillä, jossa on myös muutamia yliopistoja mukana (*ROSIN*).

Lisäksi monissa ihmisille vaikeakulkuisissa teollisissa ympäristöissä ja pelastustehtävissä etäohjattaville roboteille löytyy käyttöä. Tästäkin aihepiiristä on löydettävissä ROSin hyödyntämisestä tutkimustietoa (Peppoloni ym. 2015; Belzunce, Li ja Handroos 2016; Chen ym. 2016). Tällainen etäläsnäölorobotiikka soveltuu myös hyvin erilaisiin palvelurobotiikan tilanteisiin (Do ym. 2013).

3.2 Palvelurobotiikan pilotteja

Robotteja jaotellaan tällä hetkellä teollisiin- ja palveluroboteihin. Palvelurobotiikka nähdään teknologiana, joka suorittaa ihmiselle hyödyllisiä toimia tai joka laajentaa teollisuuden automatisaation sovellutuksia (ISO 8373). Ala on murroksessa ja jatkossa näemme myös teollisissa ympäristöissä yhä enemmän erilaisiin tehtäviin sopeutuvaa palvelurobotiikkaa. Palveluroboteille ominaista on itsenäinen liikkuminen, sijaintinsa ja toimintaympäristönsä tunnistaminen, oppiminen ja muutostilanteisiin reagoiminen. (Castrén 2016.)

Robotiikan kasvupotentiaalia nähdään olevan erityisesti palvelurobotiikassa. Yksinkertaisia palvelurobotteja mm. pölynimureita ja ruohonleikkureita on jo kuluttajilla yleisesti käytössä. Viime vuosien kasvaneet sijoitukset robotiikan startup-yrityksiin ovat kohdistuneet paljolti autonomisiin laitteisiin ja palvelurobotiikkaan (*CBInsights*). Kehittyneemmät palve-

lurobotiikan ratkaisut yleistynevät ensin todennäköisesti isoimmissa yksiköissä, hotelleissa, sairaaloissa, kaupoissa, varastoissa yms. Ensimmäinen ROS-robotti, palvelurobotiikan tutkimukseen kehitetty PR2 (Cousins 2010), on saanut jälkeensä monia seuraajia. Näistä ehkä parhaiten esillä viime aikoina ovat olleet hotellien huonepalvelutehtävissä häärivä Relay, vähittäiskaupan inventointi- ja varastohallinnan ratkaisujen Fetch ja Freight sekä Simbe. Silicon Valley Robotics palvelurobotiikan raporteista näistä tietoa kootusti (*SVR Reports*).

Hoivasektorille ja henkilökohtaiseen palveluun liittyviä robottiprojekteja ovat mm. Care-O-bot-4 (*Care-O-bot 4*) ja Toyotan Human Support Robot (HSR). Myös avustavassa robotiikassa käytetylle Jaco Kinova-robottikädelle on olemassa ROS-ohjelmistopaketti ja Secondhands-projektissa on toteutettu myös ohjelmia avustavalle robottikädelle. On myös tutkimusprojekteja, joissa on testattu monirobotiikan yhteistoimintaa ja pyörätuolin integroimista osaksi palvelurobotiikan kokonaisuutta [Li, Oskoei ja Hu 2013]. Pal robotics on toteuttanut Reem-humanoidirobotin (*Pal Robotics*) ja paljon medioissa esiintyneille Nao ja Pepper humanoideille on myös tehty ajurit ROSille.

ROS-käyttöjärjestelmän opiskeluun ja tutkimiseen rakennetut kohtuuhintaiset laiteympäristöt mm. Turtlebot ovat edesauttaneet erilaisten tutkimusten ja kokeilujen syntymistä. ROSBOT-robotti on varsin mielenkiintoinen tutkimus edullisesti toteutetusta sosiaalisen robotin kokeilusta, joka soveltuu koulutuksellisiin ja tutkimuksellisiin tilanteisiin (Fu ja Zhang 2015). Hyvin vastaavia kokeiluja ovat kyseisessä julkaisussa viitatus Nelson-robotti (Ferguson, Webb ja Strzalkowski 2011) ja Luke-valokuvaajarobotti (Zabarauskas ja Cameron 2014).

3.3 ROS maalla, vesillä ja ilmassa - Autonomisesti liikkuvat laitteet ja kulkuvälineet

Itsenäisesti liikkuvat laitteet ja kulkuvälineet ovat viime vuosina nousseet robotiikkakeskusteluissa usein otsikoihin. Clearpath Robotics on tällä hetkellä yksi vahvimmista ROS toimijoista tällä sektorilla, se on toteuttanut monia ROS-pohjaisia sisä- ja ulkokäyttöön suunniteltuja robotteja tutkimuskäyttöön sekä materiaalinkäsittelyyn (*Clearpath Robotics*).

ROSiin ja itsenäisiin robottiautoihin liittyviä tutkimusprojekteja on löydettävissä useita ja vertailua kaupallisiin yleisesti käytössä oleviin ratkaisuihin on myös tehty (Hellmund ym.

2016). Kyseisen julkaisun viitteistä on löydettävissä myös aiempia tutkimuksia. ROSin vahvuksina tällä hetkellä nähdään simuloinnin mahdollisuudet, ohjelmiston käytettävyys ja kehittäjien vahva yhteisö. Ompa ROSilla ja Gazebolla toteutettu jopa HIL -ajosimulaattori (Swanson ym. 2013). Isoista valmistajista ainakin BMW on käyttänyt ROSia tuotekehityksessä, erityisesti simuloinnissa ja testaamisessa. Kokeilut ovat olleet positiivisia, mutta teollisuusstandardien (ISO, AUTOSAR) noudattaminen on vaatimus sille, että ohjelmistoa voitaisiin käyttää ajoneuvoissa (*ROS at BMW*). Vaikuttaisi siltä että nykyinen ROS-versio ei ole teknologiana vielä valmis autonomisten ajoneuvojen käyttöjärjestelmäksi, koska se ei täytä tällä hetkellä reaaliaikaisen käyttöjärjestelmän vaatimuksia. Myös maataloussektorilla on ollut erilaisia projekteja, mm. itsenäisesti liikkuvien poimintarobottien pilotointeja (Bontsema ym. 2015).

Dronejen/UAV-laitteiden käyttö erilaisissa palvelutehtävissä on laajasti tutkittu aihe mm. video- ja valokuvaamisessa, valvonta- ja pelastustehtävissä sekä jakelupalveluissa. Myös ROS-käyttöjärjestelmällä toimivia lentäviä laitteita on markkinoilla (*Erle Robotics*) ja Parrot AR.Drone nelikopteria on käytetty työkaluna monissa tutkimuksissa. (Saarimäki 2015; Vélez, Certad ja Ruiz 2015). Löytyy myös tutkimusta, jossa on pyritty kehittämään turvallisempia ratkaisuja tuulimyllyjen automaattiseen tarkkailuun (Schäfer ym. 2016). Kaikki ROSilla käytettävissä olevat robotit löytyvät wiki-sivuilta. Clearpath Robotics on tehnyt pintavesien tutkimukseen Kingfisher -robottialuksen ja vedenalaisten etäältä operoitavien ROV-laitteiden liikkumisen/paikallaan pysymisen automatisoinnista löytyy myös ROS- tutkimusta (Lawrance ym. 2016).

3.4 ROS pilvirobotiikassa

Pilvirobotiikka on robotiikan alue, jossa robotin älykkyydestä suurin osa toteutetaan pilvessä. Toimintojen siirtäminen pilveen mahdollistaa kevyempiä, edullisempia ja kehittyneempiä ja toisiltaan oppivia ratkaisuja. Potentiaalisia hyötyjä ovat jaetut taito- ja käyttäytymistietokannat, tietämuskannat objektien tunnistamiseen ja kartoittamiseen sekä laskennallista tehoa vaativien tehtävien siirtäminen pilveen. Ohjelmiston päivittäminen, yleisten tehtävien ohjelmakirjastot, robottien tarvitseman datan tallennus, toiminta-alueiden kartat ja vikasietoisuuden paraneminen ovat muita esiteltyjä käyttöalueita. Pilvipalveluilla pienen alkavan

startup -yritysten saatavilla on modernit laskenta- ja tiedonkäsittelyresurssit. Pilvirobotiikka mahdollistaa myös uusien palvelumallien syntyminen, esimerkiksi uudenlaiset PaaS ja RaaS (Robot as a Service) -ratkaisut. Haasteina nähdään mitä toimintoja voidaan suorittaa pilvestä, ja mitkä toimintavarmuuteen liittyvät tehtävät on suoritettava paikallisesti. (Kehoe ym. 2015). Isoista toimijoista ainakin Toyota on valinnut strategiakseen avoimen lähdekoodin ratkaisuihin ja pilvirobotiikkaan panostamisen (*Goldberg, UC Berkeley*).

ROS-järjestelmään liittyviä merkittävämpiä kehitysprojekteja ovat olleet RoboEarth-projekti ja sen edeltäjät. RoboEarth -projektissa vuosina 2010-2014 toteutettiin Wikipedia tyylinen robottien hyödynnettävissä oleva tietokanta sekä tähän liittyvä Rapyuta - RoboEarth Cloud Engine (Castrén 2016; Mohanarajah ym. 2015). RoboEarth -projektista spin-off-yrityksenä syntynyt Rapyuta–Robotics kehittää tällä hetkellä pilvirobotiikkaa hyödyntäviä droneja valvontatehtäviin. Mielenkiintoinen robotiikan startup-yritys on myös Tend.ai, joka kehittää 3D-tulostamiseen automatisoitua ROS-pohjaista robottiratkaisua hyödyntäen pilvitekniologioita. ROS-pohjaista pilvirobotiikkaa on myös tutkittu TTYn Porin yksikössä OpenCRP-hankkeessa (Oksa ja Loula 2016).

3.5 Robotiikan kouluttajana

On fakta että robotiikan koulutuksen mahdollisuudet, mutta myös vaatimukset ovat nopeasti lisääntymässä. Monet koulutusorganisaatiot eri toimialoilla etsivät nyt tai lähivuosina niille parhaiten sopivaa kohtuuhintaista robotiikan ratkaisua (Touretzky 2013).

Robotiikan opiskelua vaikeuttaa aiheen kompleksisuus. Monesti myös opiskeluympäristöt ovat irrallisia todellisesta maailmasta. Autonomisen robotiikan kurseilta löytyy kokemuksia ROS -käyttöjärjestelmän käytöstä Lego Mindstorm NXT laitteissa. Kyseinen laite on ollut yleinen laite robotiikan opiskelussa ja sisältää myös oman ohjelmointiympäristön. ROS-koulutuksia on vertailtu aiempiin menetelmiin. On selvää että käyttämällä opiskelussa niitä työkaluja ja menetelmiä, joita myöhemmin työelämässä kysytään, opiskelijat saavat kokemusta työelämän todellisista kehitysympäristöistä. ROS on käytössä useissa yliopistoissa maailmalla. Vaikka ROS on selkeästi järjestelmä, joka vaatii aikaa opetteluun, on se tutkimuksen mukaan otettavissa haltuun vastaavassa ajassa mitä aiemmat perinteiset ympäristöt.

Sen käyttöä puoltaa tuettujen robottien määrä, järjestelmän monikielisyys, käytävissä olevat sensoriteknologiat ja valmiit ohjelmamodulit. On myös todettu että ROS-kursseilla ohjelmointi on organisoidumpaa ja koodi uudelleenkäytettävämpää. ROS sisältää suuren määrän valmiita robotiikan sovelluksia ja hyvän yhteensopivuuden moneen muuhun työkaluun, joita voidaan käyttää lähtökohtana omille toteutuksille. Hyviä puolia ovat myös mahdollisuus valita useiden ohjelmointikielten väliltä, suuri yhteisö, tuki ja tutoriaalit. (Michieletto ym. 2014; Alisher, Alexander ja Alexandr 2015.)

4 Tulevaisuuden haasteet ja mahdollisuudet

On hyvin ymmärrettävää, että teknologisia haasteita ilmaantuu, kun sovellusalueet lisääntyvät ja siirrytään tutkimustyökaluista valmiisiin tuotteisiin. Tällä hetkellä ROS vaatii käyttöjärjestelmän alleen ja näin sen käyttö ei ole mahdollista yksinkertaisissa sulautetuissa robotiikan ratkaisuisissa. Se ei myöskään sisällä ajonaikaisen järjestelmän vaatimuksia ja toimii vain tietyissä käyttöjärjestelmissä (Maruyama, S. ja T. 2016). ROS on alunperin suunniteltu joustavaan yhden robotin tutkimukseen. Se vaatii myös merkittävästi resursseja laitteistolta ja verkkokaistaa, eikä nykyisellään takaa riittävää vikasietoisuutta, aikarajoja ja prosessien synkronointia. Vaikka ROS nykyiselläänkin mahdollistaa monen robotin ympäristön, se ei tarjoa tähän standardia ratkaisua.

Ohjelmiston nopea kehityssykli ja yhteensopivuuden puuttuminen taaksepäin, ovat myös haasteita kun lähdetään toteuttamaan pysyvämpiä ja tuetumpia ratkaisuja. Kuten kaikissa avoimen lähdekoodin hankkeissa, myös ROSin ympärillä on paljon erilaisia ohjelmistopaketteja, joiden ylläpito, laatu ja dokumentaatio vaihtelee. Osin jo näiden löytyminen asettaa haasteita.

Robotiikan tietoturva-asiat kokonaisuutena vaativat tulevaisuudessa erityistä huomiota (Morante, Victores ja Balaguer 2015). ROS on syntynyt robotiikan tutkimuskäyttöön ja tutkimustyössä monet tietoturvaa lisäävät menetelmät hankaloittavat joustavaa kehitystyötä. Kun siirrytään myytäviin tuotteisiin, tulee tietoturvaan kiinnittää enemmän huomiota. ROSin salaamattomat sanomat, suojaamattomat TCP-portit ja kryptaamaton data ovat selkeitä haavoittuvuuksia, joita hyökkääjät voivat hyödyntää (McClean ym. 2013). ROSin teknologinen arkkitehtuuri nykymuodossaan on sellainen, että sitä tuskin voi käyttää turvallisuuskriittisissä ratkaisuisissa sellaisenaan. Asia on yleisesti tiedossa ja muutamia ehdotuksia turvallisuuden parantamiseksi on ehdotettu (Dóczi ym. 2016; Dieber ym. 2016).

Beta-asteella oleva ROS 2.0 tulee olemaan iso kehitysaskel, joka tähtää korjaamaan niitä haasteita, joita nykyisen version käytön laajentuessa on tullut vastaan. ROS 2.0 on reaaliaikainen käyttöjärjestelmä, jota voidaan käyttää pienissä sulautetuissa ratkaisuisissa. Se toimii monissa käyttöjärjestelmissä, kontrollointi tapahtuu suoraan ROS-järjestelmässä, eikä

master-solmua enää tarvita. ROS 2.0 käyttää DDS-kommukointiteknologiaa (Data Distribution Service), joka soveltuu reaaliaikaisiin sulautettuihin järjestelmiin ja on turvallinen, skaalautuva ja vikasietoinen ratkaisu. Kyseinen DDS-teknologia on jo käytössä vaativissa ympäristöissä (junat, lentokoneet) ja verifioitu NASAn toimesta. Mikä tahansa ROS 2.0 solmu osaa kommunikoida toisen solmun kanssa ilman yhteyttä teknologia-alustaan. ROS 2.0 on myös tietoturvalisempi versio kuin edeltäjänsä. Uusin tieto version kehittämisestä löydyy osoitteesta <https://github.com/ros2/ros2/wiki>. (*Why ROS 2.0*; Maruyama, Kato ja Azumi 2016.)

ROSin mahdollisuudet ovat tulevaisuuden laajentuissa robotiikan käyttöalueissa ja uusien älykkäämpien aplikaatioiden syntymisessä robotiikkaan. Teknologian potentiaalisia käyttäjiä ovat aloittavat robotiikan yritykset, joilla ei ole isojen yritysten resursseja kehittää robotiikkaa vuosia ja jotka etsivät edullista ja valmista alustaratkaisua sovelluksilleen. Myös vakiintunut teollisuus on kiinnostunut järjestelmän hyödyntämisestä. Merkittävä kysymys on miten hyvin se pystyy hyödyntämään avointa lähdekoodia omassa liiketoiminnassaan. Akateemiset tutkimusprojektit voivat myös synnyttää ROS startup-yrityksiä ja järjestelmä voi mahdollistaa robotiikkaan uusia liiketoimintamalleja esim. RaaS(Robot as a Service), pilvirobotiikan ratkaisuja ja erilaisia robotiikan koulutus- ja konsultointipalveluja. (Cousins 2012.)

Järjestelmän potentiaalista kertoo myös NASAn kansainvälisellä avaruusaluksella työskentelevään Robonaut2-robottiin liittyvä kehittämistyö (Hart, Dinh ja Hambuchen 2015). Pehmeän robotiikan (soft robotics) sovellutukset voivat osaltaan laajentaa ROSin käyttöalueita (McKenzie, Barraclough ja Stokes 2017). Myös laitepuolella tapahtuu, Erle Robotics on kehittämässä ROS 2.0 yhteensopivia H-ROS -laitteita, jotka olisivat helposti uudelleen käytettäviä ja konfiguroitavia, ja näin helpottavat robottien rakentamista (*H-ROS*).

5 Yhteenveto

Edellä kuvattujen sovellusalueiden monipuolisuus kertoo ROSin potentiaalista. Järjestelmän parissa tehdään aktiivista kehitystyötä niin tutkimus- ja koulutussektorilla kuin yrityksissä. Järjestelmän käyttö ja käyttäjämäärät ovat kasvussa. Ohjelmiston nopea kehityssykli kertoo aktiivisuudesta, mutta myös siitä, että se ei ole vielä kaikkineen valmis. Teknologisia haasteita on, mutta valmistuessaan käyttöjärjestelmän uusien versio ROS 2.0 ratkaisee näistä keskeisimpiä. ROS 2.0 tuo mahdollisuuden kehittää robotteja edullisille alustoille, joka voi olla avain järjestelmän kaupalliseen menestykseen. Uusia mahdollisuuksia on myös monirobottiympäristöissä ja pilvirobotiikan alueilla.

Kun robotiikan suurin kasvu on jatkossa erityisesti valmistusteollisuuden ulkopuolella, luo se potentiaalia räjähdysmäiseen kasvuun. Uudet markkinat vaativat robotiikalta juuri sellaista älykkyyttä mitä ROSin ympärillä tällä hetkellä tutkitaan ja pilotoidaan. Avoimen lähdekoodin hyödyntäminen on ollut vahvaa ohjelmistokehittämisessä viime vuosikymmenenä ja on oletettavaa, että nämä hyödyt nähdään jatkossa yhä vahvemmin myös robotiikassa.

ROS-käyttöjärjestelmän elinkaaren jatkumisesta ei voi tietää, mutta se on selkeästi luonut mallin robotiikkaan yhteentoimivuudella, helppokäyttöisyydellä ja avoimuudella. Se on selvästi suurin ja käytetyin avoimen lähdekoodin robottikäyttöjärjestelmä. Tulevaisuuden haasteelliset ratkaisut vaativat robotiikan ohjelmistoilta paljon. Robotiikan sovellusalueille on ehkä mahdotonta rakentaa yhtä toteutusta, mikä olisi ratkaisu kaikkeen. Kuitenkin avoimella, kehittyvällä ja laajasti levinneellä tutkijoiden ja oppilaitosten tuntemalla ja hyödyntämällä teknologialla, on varmasti lisääntyvää potentiaalia myös kaupallisissa ratkaisuissa.

Tutkielma on ollut melko pintapuolinen katsaus aiheeseen. Ehkä hienointa olisi, jos tämä kirjoitelma innostaisi jotakuta käärimään hihat ja tutustumaan ROSiin käytännössä. Jatko-tutkimusten kiinnostavia aiheita voisivat esimerkiksi olla syvempi tutustuminen ROS 2.0 teknologiaan ja ROSin erilaiset pilvirobotiikan ratkaisut.

Lähteet

Alisher, Khassanov, Krupenkin Alexander ja Borgul Alexandr. 2015. "Control of the mobile robots with ROS in robotics courses". *Procedia Engineering* 100:1475–1484.

Belzunce, A., M. Li ja H. Handroos. 2016. "Control system design of a teleoperated omnidirectional mobile robot using ROS". Teoksessa *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, 1283–1287. Kesäkuu. doi:10.1109/ICIEA.2016.7603782.

Bontsema, Jan, Jochen Hemming, Erik Pekkeriet, Wouter Saeys, Yael Edan, Amir Shapiro, Marko Hocevar ym. 2015. "CROPS : Clever Robots for Crops". *Engineering Technology Reference* 1 (1).

Bruyninckx, H. 2008. "Robotics Software: The Future Should Be Open [Position]". *IEEE Robotics Automation Magazine* 15, numero 1 (maaliskuu): 9–11. ISSN: 1070-9932. doi:10.1109/M-RA.2008.915411.

BSD. Saatavilla WWW-muodossa, <https://opensource.org/licenses/BSD-3-Clause>, viitattu 2.4.2017.

Care-O-bot 4. Saatavilla WWW-muodossa, <http://www.care-o-bot-4.de/>, viitattu 4.4.2017.

Castrén, Erkki. 2016. "OPENCRP - Avoimen lähdekoodin pilvipalvelu- ja mobiilirobottiympäristö". Tutkielma, Tampereen Teknillinen Yliopisto Porin laitos.

CBInsights. Saatavilla WWW-muodossa, <https://svrobo.org/looking-towards-service-robotics-in-2017/>, viitattu 13.2.2017.

Chen, K., M. Kamezaki, T. Katano, J. Yang, T. Ishida, M. Seki, K. Ichiryu ja S. Sugano. 2016. "Fundamental development of a virtual reality simulator for four-arm disaster rescue robot OCTOPUS". Teoksessa *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 721–726. Heinäkuu. doi:10.1109/AIM.2016.7576853.

Clearpath Robotics. Saatavilla WWW-muodossa, <https://www.clearpathrobotics.com/>, viitattu 3.4.2017.

Cousins, S. 2010. "ROS on the PR2 [ROS Topics]". *IEEE Robotics Automation Magazine* 17, numero 3 (syyskuu): 23–25. ISSN: 1070-9932. doi:10.1109/MRA.2010.938502.

———. 2012. "Is ROS Good for Robotics? [ROS Topics]". *IEEE Robotics Automation Magazine* 19, numero 2 (kesäkuu): 13–14. ISSN: 1070-9932. doi:10.1109/MRA.2012.2193935.

Dieber, B., S. Kacianka, S. Rass ja P. Schartner. 2016. "Application-level security for ROS-based applications". Teoksessa *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4477–4482. Lokakuu. doi:10.1109/IROS.2016.7759659.

Do, H. M., C. J. Mouser, Y. Gu, W. Sheng, S. Honarvar ja T. Chen. 2013. "An open platform telepresence robot with natural human interface". Teoksessa *2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, 81–86. Toukokuu. doi:10.1109/CYBER.2013.6705424.

Dóczi, R., F. Kis, B. Sütő, V. Póser, G. Kronreif, E. Jósvai ja M. Kozlovsky. 2016. "Increasing ROS 1.x communication security for medical surgery robot". Teoksessa *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 004444–004449. Lokakuu. doi:10.1109/SMC.2016.7844931.

Erle Robotics. Saatavilla WWW-muodossa, <https://erlerobotics.com>, viitattu 11.4.2017.

Ferguson, Michael, Nick Webb ja Tomek Strzalkowski. 2011. "Nelson: a low-cost social robot for research and education". Teoksessa *Proceedings of the 42nd ACM technical symposium on Computer science education*, 225–230. ACM.

Fu, Guohe., ja Xinyu. Zhang. 2015. "ROSBOT: A low-cost autonomus social robot". *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*: 1789–1794.

Garber, L. 2013. "Robot OS: A New Day for Robot Design". *Computer* 46, numero 12 (joulukuu): 16–20. ISSN: 0018-9162. doi:10.1109/MC.2013.434.

Gazebo. Saatavilla WWW-muodossa, <http://gazebo.org/>, viitattu 2.4.2017.

- Goldberg, UC Berkeley. Saatavilla WWW-muodossa, <http://goldberg.berkeley.edu/cloud-robotics/>, viitattu 4.4.2017.
- Hart, S., P. Dinh ja K. Hambuchen. 2015. "The Affordance Template ROS package for robot task programming". Teoksessa *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 6227–6234. Toukokuu. doi:10.1109/ICRA.2015.7140073.
- Hellmund, A. M., S. Wirges, Ö.S Tas, C. Bandera ja N. O. Salscheider. 2016. "Robot operating system: A modular software framework for automated driving". Teoksessa *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 1564–1570. Marraskuu. doi:10.1109/ITSC.2016.7795766.
- H-ROS. Saatavilla WWW-muodossa, <https://www.h-ros.com/>, viitattu 12.4.2017.
- Kehoe, B., S. Patil, P. Abbeel ja K. Goldberg. 2015. "A Survey of Research on Cloud Robotics and Automation". *IEEE Transactions on Automation Science and Engineering* 12, numero 2 (huhtikuu): 398–409. ISSN: 1545-5955. doi:10.1109/TASE.2014.2376492.
- Lawrance, Nicholas RJ, Thane Somers, Dylan Jones, Seth McCammon ja Geoffrey A Hollinger. 2016. "Ocean deployment and testing of a semi-autonomous underwater vehicle". Teoksessa *OCEANS 2016 MTS/IEEE Monterey*, 1–6. IEEE.
- Li, R., M. A. Oskoei ja H. Hu. 2013. "Towards ROS Based Multi-robot Architecture for Ambient Assisted Living". Teoksessa *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 3458–3463. Lokakuu. doi:10.1109/SMC.2013.590.
- Maruyama, Y., S. Kato ja T. Azumi. 2016. "WiP Abstract: Preliminary Evaluation of ROS2". Teoksessa *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, 1–1. Huhtikuu. doi:10.1109/ICCPS.2016.7479129.
- Maruyama, Y., Kato S. ja Azumi T. 2016. "Exploring the performance of ROS2". *2016 International Conference on Embedded Software (EMSOFT)*: 1–10.
- McClellan, Jarrod, Christopher Stull, Charles Farrar ja David Mascareñas. 2013. "A preliminary cyber-physical security assessment of the Robot Operating System (ROS)". Teoksessa *SPIE Defense, Security, and Sensing*, 874110–874110. International Society for Optics ja Photonics.

- McKenzie, Ross M, Thomas W Barraclough ja Adam A Stokes. 2017. “Integrating Soft Robotics with ROS-A hybrid pick and place arm”. *arXiv preprint arXiv:1702.00694*.
- Michieletto, Stefano, Stefano Ghidoni, Enrico Pagello, Michele Moro ja Emanuele Menegatti. 2014. “Why teach robotics using ROS?” *Journal of Automation Mobile Robotics and Intelligent Systems* 8.
- Mohanarajah, G., D. Hunziker, R. D’Andrea ja M. Waibel. 2015. “Rapyuta: A Cloud Robotics Platform”. *IEEE Transactions on Automation Science and Engineering* 12, numero 2 (huhtikuu): 481–493. ISSN: 1545-5955. doi:10.1109/TASE.2014.2329556.
- Morante, Santiago, Juan G Victores ja Carlos Balaguer. 2015. “Cryptobotics: Why robots need cyber safety”. *Frontiers in Robotics and AI* 2:23.
- MoveIt*. Saatavilla WWW-muodossa, <http://moveit.ros.org/>, viitattu 2.4.2017.
- Oksa, Petri, ja Pekka Loula. 2016. “Private Cloud Deployment Model in Open-Source Mobile Robots Ecosystem”. *Towards Autonomous Robotic Systems: 17th Annual Conference*: 239–248.
- OSRF*. Saatavilla WWW-muodossa, <http://www.osrfoundation.org/>, viitattu 26.3.2017.
- Pal Robotics*. Saatavilla WWW-muodossa, <http://www.pal-robotics.com>, viitattu 4.4.2017.
- Peppoloni, L., F. Brizzi, C. A. Avizzano ja E. Ruffaldi. 2015. “Immersive ROS-integrated framework for robot teleoperation”. Teoksessa *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, 177–178. Maaliskuu. doi:10.1109/3DUI.2015.7131758.
- Peshkin, M. A., J. E. Colgate, W. Wannasuphprasit, C. A. Moore, R. B. Gillespie ja P. Akella. 2001. “Cobot architecture”. *IEEE Transactions on Robotics and Automation* 17, numero 4 (elokuu): 377–390. ISSN: 1042-296X. doi:10.1109/70.954751.
- Qian, W., Z. Xia, J. Xiong, Y. Gan, Y. Guo, S. Weng, H. Deng, Y. Hu ja J. Zhang. 2014. “Manipulation task simulation using ROS and Gazebo”. Teoksessa *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2594–2598. Joulukuu. doi:10.1109/ROBIO.2014.7090732.

Quigley, M., B. Gerkey, K. Conley, J. Faust, T Foote, J. Leibs, E. Berger, Wheeler R. ja A. Ng. 2009. "ROS: an open-source robot operating system". *Open-Source Software workshop of the International Conference on Robotics and Automation*.

Quigley, Morgan, Brian Gerkey ja William D. Smart. 2015. *Programming Robots with ROS*. O'Reilly Media.

Rethink Robotics. Saatavilla WWW-muodossa, <http://www.rethinkrobotics.com>, viitattu 2.4.2017.

ROS at BMW. Saatavilla WWW-muodossa, <http://www.osrfoundation.org/michael-aeberhard-bmw-automated-driving-with-ros-at-bmw/>, viitattu 3.4.2017.

ROS Community Metrics July 2016. Saatavilla pdf-muodossa, <http://download.ros.org/downloads/metrics/metrics-report-2016-07.pdf>, viitattu 2.4.2017.

ROS Wiki. Saatavilla WWW-muodossa, <http://wiki.ros.org/>, viitattu 13.2.2017.

ROSIN. Saatavilla WWW-muodossa, <http://rosin-project.eu/>, viitattu 13.2.2017.

ROS-Industrial. Saatavilla WWW-muodossa, <http://rosindustrial.org/>, viitattu 2.3.2017.

Rusu, R. B., ja S. Cousins. 2011. "3D is here: Point Cloud Library (PCL)". Teoksessa *2011 IEEE International Conference on Robotics and Automation*, 1–4. Toukokuu. doi:10.1109/ICRA.2011.5980567.

Saarimäki, Ansse. 2015. "KAMERAPOHJAINEN PAIKANNUS". Tutkielma, Tampereen Teknillinen Yliopisto.

Schäfer, B. E., D. Picchi, T. Engelhardt ja D. Abel. 2016. "Multicopter unmanned aerial vehicle for automated inspection of wind turbines". Teoksessa *2016 24th Mediterranean Conference on Control and Automation (MED)*, 244–249. Kesäkuu. doi:10.1109/MED.2016.7536055.

Swanson, K. S., A. A. Brown, S. N. Brennan ja C. M. LaJambe. 2013. "Extending driving simulator capabilities toward Hardware-in-the-Loop testbeds and remote vehicle interfaces". Teoksessa *2013 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 115–120. Kesäkuu. doi:10.1109/IVWorkshops.2013.6615236.

SVR Reports. Saatavilla WWW-muodossa, <https://svrobo.org/project/svr-reports/>, viitattu 13.2.2017.

Takaya, K., T. Asai, V. Kroumov ja F. Smarandache. 2016. "Simulation environment for mobile robots testing using ROS and Gazebo". Teoksessa *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, 96–101. Lokakuu. doi:10.1109/ICSTCC.2016.7790647.

Tavares, P., ja A. Sousa. 2015. "Flexible pick and place architecture using ROS framework". Teoksessa *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6. Kesäkuu. doi:10.1109/CISTI.2015.7170602.

Touretzky, David S. 2013. "Robotics for computer scientists: what's the big idea?" *Computer Science Education* 23 (4): 349–367.

Turtlebot3. Saatavilla WWW-muodossa, <http://turtlebot3.readthedocs.io/en/latest/>, viitattu 11.4.2017.

Vélez, P., N. Certad ja E. Ruiz. 2015. "Trajectory Generation and Tracking Using the AR.Drone 2.0 Quadcopter UAV". Teoksessa *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, 73–78. Lokakuu. doi:10.1109/LARS-SBR.2015.33.

Venator, E., G. S. Lee ja W. Newman. 2013. "Hardware and software architecture of AB-BY: An industrial mobile manipulator". Teoksessa *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, 324–329. Elokuu. doi:10.1109/CoASE.2013.6653969.

Why ROS 2.0. Saatavilla WWW-muodossa, http://design.ros2.org/articles/why_ros2.html, viitattu 2.3.

Willow Garage. Saatavilla WWW-muodossa, <http://www.willowgarage.com/>, viitattu 2.4.2017.

World robotics report. Saatavilla WWW-muodossa, <http://www.itworldcanada.com/article/human-robot-collaboration-growing-says-world-robotics-report/387099>, viitattu 2.4.2017.

Yan, H., Y. Yang, R. Li, M. Dehghan ja M. H. Ang. 2015. "A preliminary study of the RADOE project". Teoksessa *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 199–204. Heinäkuu. doi:10.1109/ICCIS.2015.7274620.

Zabarauskas, M., ja S. Cameron. 2014. "Luke: An autonomous robot photographer". Teoksessa *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 1809–1815. Toukokuu. doi:10.1109/ICRA.2014.6907096.