

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Kennar Kahju
Navigeerimine välistingimustes mudelautoga
Bakalaureusetöö (9 EAP)

Juhendaja:
Tambet Matiisen, MSc

Tartu 2025

Navigeerimine välistingimustes mudelautoga

Lühikokkuvõte:

Tänapäeva autonoomsete sõidukite arendus keskendub valdavalt teedel ja tänavatel liikuvatele autodele. Bakalaureusetöö eesmärk oli arendada sõiduk, mis sõidaks iseseisvalt vabas looduses. Platvormina kasutati mudelautot Donkey Car S1, millele disainiti peale 3D-prinditud raamid koos GNSS-seadme ja LiDARiga. Täiendati Donkey Car projekti vajaliku tarkvaraga sihtpunktipõhiseks navigeerimiseks. Isejuhtimise hindamiseks viidi läbi erinevaid katseid nii asfaldil kui ka parkmetsas. Sihtpunktide vahel navigeerimine toimis, nii siis kui sihtpunkt oli ees kui ka selja taga. Nii GNSS kui ka LiDAR töötasid ootuspäraselt. GNSSi täpsuseks hinnati 2 cm LiDARi täpsus jäi 5 cm vahemikku. Takistustest eemale hoidmine ei olnud aga tõhus ning on analüüsitud võimalikke põhjuseid. Samuti selgus katsetuste käigus, et valitud raam ei sobi maastikul sõitmiseks. Töö lõpus on esitatud soovitusel edasiseks arenduseks ja vigade vältimiseks.

Võtmesõnad: isejuhtimine, autonoomsus, RTK, LiDAR, Donkey Car

CERCS:P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Outdoor Navigation with a Model Car

Abstract:

The development of autonomous vehicles is primarily focused on cars operating on roads. The goal of this bachelor's thesis was to develop a vehicle to navigate in off-road environments. The Donkey Car S1 platform was used as a base, to which custom-designed and 3D-printed frames were added to mount a GNSS device and a LiDAR sensor. Necessary functionality was developed to the Donkey Car project to enable waypoint-based navigation. To evaluate the self-driving capability, various tests were conducted both on asphalt and in a park environment. Navigation between waypoints was successful, even when the waypoint was behind the car. Both GNSS and LiDAR worked as expected, with the GNSS evaluated to the accuracy of 2 cm and LiDAR achieving a measurement accuracy within 5 cm. However, obstacle avoidance proved to be ineffective and the possible reasons for this were analyzed. It was also found that the frame of the chosen platform was not suitable for off-road driving. This thesis concludes with recommendations for further development and suggestions for avoiding similar issues in future implementations.

Keywords: self-driving, autonomy, RTK, LiDAR, Donkey Car

CERCS: P170 Computer science, numerical analysis, systems, control

Sisukord

1.	Sissejuhatus.....	5
2.	Mõisted ja lühendid.....	6
3.	Taust.....	7
3.1	Autonoomseks liikumiseks vajalikud osad.....	7
3.2	Ülevaade GNSS süsteemist.....	8
3.2.1	GNSS andmete täpsustamistehnikad	8
3.3	Ülevaade LiDAR tehnikast.....	9
3.3.1	Võrdlus LiDAR ja kaamera vahel.....	10
3.3.2	Takistuse leidmine LiDARiga	11
4.	Arhitektuuriline ülevaade.....	13
4.1	Riistvara lühikirjeldus.....	13
4.2	Olemasolevad komponendid.....	13
4.2.1	Auto raam.....	13
4.2.2	Juhtseadmed.....	13
4.2.3	Kiirusregulaator Hobbywing WP-1625 Brushed.....	13
4.2.4	Mikroarvuti Raspberry Pi 4B.....	13
4.2.5	Vooluallikad.....	14
4.3	Juurde lisatud komponendid	14
4.3.1	GNSS positsioneerimisseade simpleRTK2B Fusion.....	15
4.3.2	Antenn koos kaitseplaadiga	16
4.3.3	2D LiDAR-seade RPLIDAR A1M8.....	17
4.3.4	Kinnitusvahendid	18
4.4	Seadmete vahelised ühendused.....	18
4.5	Tarkvara kirjeldus	20
4.5.1	Juurde lisatud tarkvara	22
4.5.2	Kasutajakogemus	23
4.5.3	Juhtmoodulite tööpõhimõtted	24
5.	Autoga päris maailmas sõitmine	26
5.1	GNSS-seadme täpsuse hindamine	26
5.2	LiDAR-seadme täpsuse mõõtmine	27
5.3	Autonoomsuse hindamine.....	29
5.3.1	Sõitmine ainult GNSS-andmetega	29

5.3.2	Sõitmine koos LiDARiga.....	30
5.3.3	Sõitmine maastikul.....	32
6.	Arutelu	33
	Kokkuvõte.....	35
	Viidatud kirjandus.....	36
	Litsents.....	39

1. Sissejuhatus

Meie igapäevased liikumisvahendid muutuvad üha autonoomsemaks, mis vabastab inimeste ajakulu sõidukite juhtimiselt muude tegevuste tarbeks. Valdav osa toodetest keskenduvad aga tänaval sõitmisele. 2023. aasta statistika kohaselt hinnati autonoomsete sõidukitega tegelevate firmade koguturuväärtuseks 208 miljardit dollarit, ning aastaks 2031 on see prognoositud tõusma 2874 miljardile dollarile [1]. Kui enamik turust keskendub tänaval sõitmisele, siis maastikul sõitmiseks leidub sarnaseid arendusi üsna vähe. Maastikul sõitmise teevad keerukaks sobiva pinnasekatte puudumine, ettearvamatu keskkond ning ilmastikuolud, mis võivad muuta maastiku raskesti läbitavaks [2].

Eestis asutatud firma Starship Technologies väikeroboteid leiab nüüdseks mitmes riigis, hetkel on need robotid liikumas rohkem kui 100 erinevas linnas nii Ameerikas, Ühendkuningriikides, Saksamaal, Soomes kui ka Eestis [3]. Need suudavad liikuda vahemaid ilma ühegi kõrvalise sekkumiseta, luues endale teekonna käsitsi koostatud kaartide põhjal ning kasutades andureid, et vältida ootamatuid takistusi [4]. Kuid nende põhiline liikumistrajektor jääb pinnatud teedele ehk kõnniteed ning ülekäigurajad, ka mõningad kruusateed [5].

Käesoleva töö eesmärk on luua sõiduk, mis suudaks navigeerida sihtkohta keskkonnas, kus ei ole sobilikku infrastruktuuri - eelkõige vabas looduses liikumiseks. Töö kasutab aluseks avatud lähtekoodiga mudelauto projekti Donkey Car ning täiendab seda, luues võimaluse navigeerida autol määratud sihtpunkti automaatselt GPS- ja LiDAR-seadmetega. Loodav auto pannakse tulemuste hindamiseks erinevates keskkondades sõitma ning seejärel analüüsitakse, kas antud lahendus on sobiv autonoomseks liikumiseks tundmatus keskkonnas.

Töö on jagatud kuueks peatükiks: kolmandas peatükis tutvustatakse lugejale autonoomse liikumise teoreetilist tausta ja täpsemalt kahte tehnikat. Neljandas peatükis kirjeldatakse autode loomiseks vajalikke komponente ja kirjeldatakse, kuidas loodud tarkvara toimib. Viiendas peatükis tuuakse välja katsed sõita autoga ning hinnatakse, kuidas auto suudab maastikul navigeerida. Kuuendas peatükis arutletakse saadud tulemuste üle ja hinnatakse, kas need vastavad ootustele.

2. Mõisted ja lühendid

GNSS - *Global Navigation Satellite System*, ülemaailmne satelliitnavigatsioonisüsteem

LiDAR - *Light Detection And Ranging*, laserskanneerimistehnika

RTK - *Real-Time Kinematics*, tehnika GNSS-i täpsuse suurendamiseks

NTRIP - *Networked Transport of RTCM via Internet Protocol*, protokoll RTK parandite saatmiseks üle võrgu

RTN - *Real-Time Network*, RTK tugijaamade võrgustik

IMU - *Inertial Measurement Unit*, inertsiaalandur objekti kiirenduse mõõtmiseks

kinematic bicycle model - lihtsustatud sõiduki mudel liikumistrajektoori arvutamiseks

NMEA - *National Marine Electronics Association*, meresõiduelektronika protokolle haldav organisatsioon

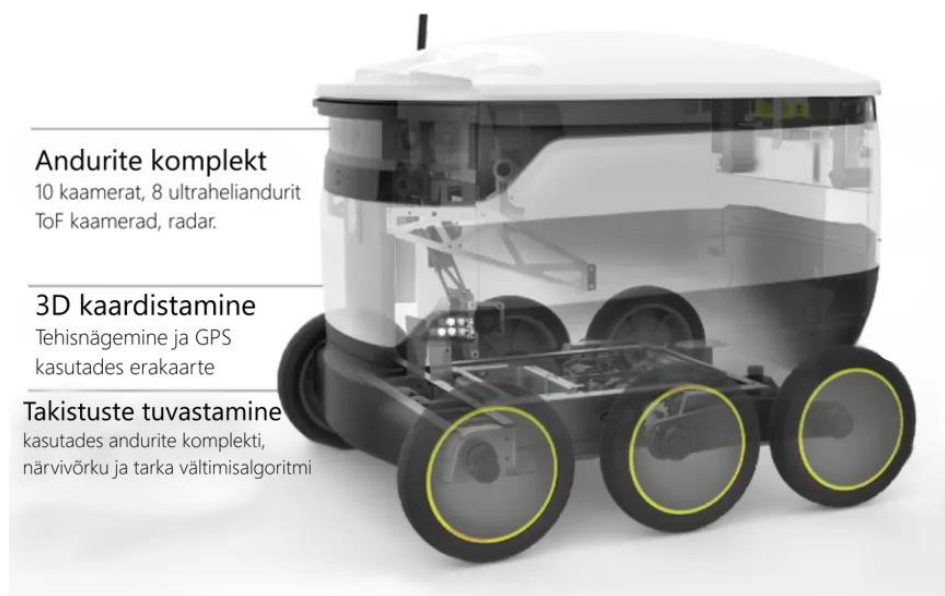
3. Taust

Selles peatükis on välja toodud ülevaade isejuhtivast liikumisest, lisaks on toodud ülevaated selleks vajalikest tehnoloogiatest GNSS (*Global Navigation Satellite System*) ja LiDAR (*Light Detection and Ranging*) ning nendega seotud lahendustest andmete täpsustamiseks.

3.1 Autonoomseks liikumiseks vajalikud osad

Kõik tänapäeva autonoomse liikumise vahendid kasutavad otsuste tegemiseks andureid. Nägemiseks on olulised radar, kaamera, LiDAR ja asukoha tuvastamiseks GNSSi vastuvõtja [6]. Igal anduritüübil on oma tugevused ja nõrkused ning need sõltuvad järgmistest teguritest: ilmastikutingimused, valgustus, päikese asukoht anduri suhtes ja liikumiskiirus [7].

Usaldusväärsete andmete kogumiseks liidetakse tavaliselt kokku mitmest erinevast andurist saadud andmed (*sensor fusion*). See tagab, et ühest allikast vigaste mõõtmistulemuste saamine mõjutab üldist olukorda vähem, kui eraldi mõõtes. Samuti, paigutades andurid üksteisest eraldi, on võimalik saavutada laiem vaatepilt väliskeskkonnast. Jooniselt 1 on näha, et Starship robotid kasutavad 10 kaamerat, 8 ultraheli andurit ning radarit [8].



Joonis 1. Starship Technologies roboti integreeritud seadmed ja andurid [8].

3.2 Ülevaade GNSS süsteemist

GNSS on satelliitidest koosnev süsteem, millega on võimalik oma asukohta Maal määrata. Tänapäeval toimub asukoha arvutamine enamasti ainult satelliitsignaalide ehk õhus levivate raadiolainete kaudu. Kolmemõõtmelises ruumis on nendega asukoha saamiseks vaja vähemalt nelja erinevat signaaliallikat - kolm asukoha jaoks, ning üks aja sünkroniseerimiseks [9]. Laineid emiteerivate allikate asukohti teades on võimalik mõõta kaugused nendest ning määrata asukoht, mis on kõikidest allikatest mõõdetud kaugustel.

GNSS on koondnimetus tänapäeval kasutatavate navigatsioonivõrkude jaoks, milleks on GPS (USA), GLONASS (Venemaa), Galileo (Euroopa Liit ja Euroopa Kosmoseagentuur) ning BeiDou (Hiina). Levinuim neist on GPS.

Andmete lugemiseks GNSS seadmetest on enim levinud kommunikatsioonistandard NMEA 0183. Algselt loodud 1983. aastal laevaelektronika kasutamiseks, on selle kasutus nüüdseks laienenud pea kõikidele GNSS-seadmetele [10]. Standardiprotokoll defineerib suhtluskanalid paljudel teemadel, kuid asukohaandmete jaoks sobivad neist vähesed, populaarseim on teema \$GNRMC, mille kaudu saadetakse minimaalselt vajalikud andmed asukoha määramiseks [11]. See sõnum koosneb järgnevalt: ajamärke, andmete õigsuse märke, seadme koordinaadid, liikumiskiirus sõlmedes, liikumissuund, süsteemi režiimi indikaatorid, ning sõnumi kontrollsumma [12].

3.2.1 GNSS andmete täpsustamistehnikad

Artu Ellmanni jt järgi [13] peavad raadiosignaalid läbima satelliitide kaugusest tingituna maakera ionosfääri kihti, kus levivad vabad ioonid, mis võivad takistada signaalide levikut. Samuti takistavad signaale troposfääri kihis leviv õhuniiskus ja pilved, ka halva kvaliteediga vastuvõtja võib tuua andmetesse kadusid [14]. Selle parandamiseks kasutatakse vähemalt ühe baasjaama suhtes kinemaatilist mõõtmist, ehk RTK (*Real-Time Kinematic*) mõõtmismeetodit. Välja töötatud 1990. aastatel, võimaldab see baasjaamast kuni 20 km kauguse ulatuses saavutada 1-5 sentimeetrise asukohatäpsuse reaajas. See nõuab, et üks GNSS-vastuvõtja (baasjaam) on fikseeritud koordinaatidega punktis, milleks sobib mõne hoone katus või kindel punkt maapinnal. Reaajas andmetöötlusega arvutatakse välja parandid satelliitsignaalidest arvutatud positsiooni ja tegeliku positsiooni vahel, ning edastatakse raadioühenduse või interneti abil NTRIP- teenusega (*Networked Transport of RTCM via Internet Protocol*) teisele

vastuvõtjale. Kui vastuvõtja on baasjaama läheduses, on satelliitide asetus ning vigade allikad mõlemal sarnased ning signaalivigade parandustes leidub vähem vigu [13].

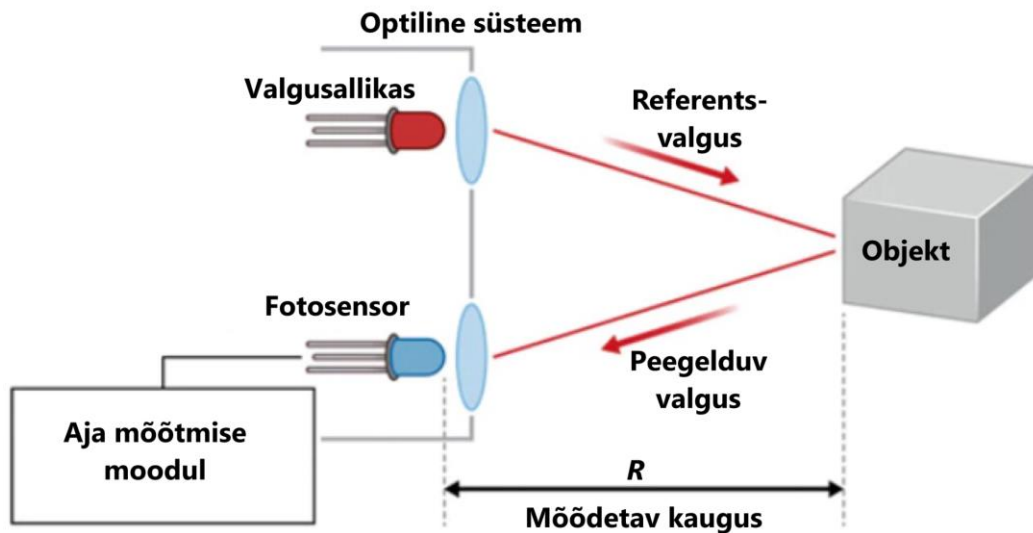
Kasutades olemasolevaid GNSS-tugijaamu, saab luua suurema katvusega RTK-võrke ehk RTN (*Real-Time Networks*). Eestis on see saadaval alates 2015. aastast Maa-ameti hallatava ESTPOS¹ teenuse kaudu. Seega täpse asukoha jaoks on vaja vaid ühte vastuvõtjat ning head internetiühendust. Türgis läbi viidud katsetuste järgi mõõdeti samade satelliitide, baasjaama ja mõõtmistehnikaga RTK täpsuse mediaanerinevuseks x- ja y- teljel vastavalt 1.1 ja 1.6 sentimeetrit, ning kõrgus z-teljel 1.7 sentimeetrit [15]. See on piisavalt täpne, et mõõta tegelikku asukohta maapinnal.

Tavakasutuses on vastuvõtjad sageli liikuvad, näiteks autodes, ning seetõttu võib signaali levikuga esineda katkemisi, kuna ümbritsev keskkond ja ehitised peegeldavad ja blokeerivad laineid. Katkeva signaali leevendamiseks on võimalik kasutada lisaks inertsiaalandureid (*inertial measurement unit*, IMU), mille abil saab hinnata umbkaudset asukohta teatud algpunktist (siinkohal signaali kadumisest) alates. Kui võrrelda IMU kaasamist tavapärase ainult GNSS andmete mõõtmisega, siis see aitab linnakeskkonnas hoida läbitud trajektoori ühtlasemana signaaliaukudest tulenevate vigade tõttu [16]. Seega kasutades RTK kõrval veel lisainfot andvaid andureid, saab luua usaldusväärsema asukohatäpsuse.

3.3 Ülevaade LiDAR tehnikast

LiDARi tööpõhimõte seisneb valguskiire väljasaatmises ning kiire peegelduse tagasi jõudmiseks kulunud aja mõõtmises (vaata Joonist 2) [17]. Teades valguse liikumiskiirust, on võimalik arvutada valguskiire läbitud vahemaa ehk kaugus anduri ja selle ees oleva takistuse vahel.

¹ ESTPOS kodulehekülj. <https://geoportaal.maaamet.ee/est/ruumiandmed/estpos-riiklik-gnss-satelliitandmete-keskus-p838.html>



Joonis 2. LiDARi tööpõhimõtte kujutatud visuaalselt [7].

Seadme ehituse poolest on võimalik jagada need kaheks: 2D ja 3D LiDAR seadmed. Mõlemal on samad tööpõhimõtted, kuid 2D LiDAR mõõdab kaugusi ühel joonel ning mõõdetav tulemus on kujutatav kahedimensionaalselt [18]. Need sobivad eelkõige lihtsate ülesannete täitmiseks, kas robotolmuimeja peale või muudes kohtades, kus oluline on vaid takistuste märkamine [19]. Samas 3D seadmed suudavad mõõta ka kolmemõõtmelises ruumis, ning tulemus on palju detailsem [18]. Neid kasutatakse rohkem autode peal detailse olukorra mõistmiseks, samuti ka täpsete 3D mudelite valmistamiseks [18]. Suur erinevus 2D ja 3D vahel on ka hind. Kui 2D variandid jäävad paarisaja euro hinnavaheemikku ja kõrgema kvaliteediga andurid mõne tuhande euro juurde, siis 3D hinnad algavad alles mitmest tuhandest eurost [20].

3.3.1 Võrdlus LiDAR ja kaamera vahel

Kui võrrelda LiDARi ja kaamera hindasid, võib tunduda, et kaamera on odavama hinna tõttu parem valik. Kuid nii kaameral kui ka LiDARil on eelistatud tingimused, milles need töötavad paremini. Kui kasutuskeskkonnaks on avar väli, on valgustus ühtlane ning maapinna detailid kergemini eristatavad. Sellisel juhul on kaamera eelistatum valik, sama pildikvaliteeti on võimalik saavutada odavama hinna eest. Metsas või pargis liigeldes võivad varjud ja ühtlase tekstuuriga aluspind (rohttaimed) häirida kaamera tööd kindlate takistuste piiride leidmises närvivõrkudega. Sellises keskkonnas, kus ei eristu taimestikust eriti palju, võib kaamerapilti töötlev närvivõrk teha palju vigu. LiDAR suudab keskkonnast sõltumata leida kõik kaugused andurist, ning oma olemuse poolest ei ole vaja saadud andmeid koheselt töödelda [21]. Kaamera üks suurim eelis on võime näha üldpilti värvides ning sellega hinnata tuvastatavaid takistusi, kas need on läbitavad või mitte [22].

Arvestades käesoleva töö eesmärki (navigeerimine struktureerimata keskkonnas), on tabelis 1 toodud võrdluse järgi eelistatum valik LiDARi kasutamine, kuna see on takistuste tuvastamises parem ning töötab kindlamini halvemate oludega või pimedamas keskkonnas. Samuti sobivad LiDARi andmed koheseks kasutamiseks, kui kaamerapilti peab enne töötlemata ressursikuluka närvivõrguga.

Tabel 1. Mõõdetavate tulemuste võrdlus LiDARi ja kaamera vahel [23].

	Lidar	Kaamera
Takistuse tuvastamine	Hea	Rahuldav
Kauguse hindamine	Hea	Rahuldav
Piirjoonte tuvastamine	Hea	Hea
Nähtavusulatus	Rahuldav	Rahuldav
Halvas ilmas töötamine	Rahuldav	Halb
Halbades valgustingimustes navigeerimine	Hea	Rahuldav

3.3.2 Takistuse leidmine LiDARiga

Liikudes sirgjooneliselt, on vaja takistuse nägemiseks vaadata vaid otsesuunas. Manööverdamise korral ei ole võimalik seda nii lihtsasti teha, kuna auto trajektoori manööverdamisel ei ole enam sirgjooneline (vaata Joonist 3). Lihtsaim viis teekonna simuleerimiseks on lihtsustatud kinemaatilise jalgratta mudeli abil. Üldistades auto telgi jalgrattana, saab joonise 3 abil arvutada välja ringjoone, mida mööda tagaratas liigub, ning sealt kiiruse ja ajaga auto asukohta [24].

Auto asukoht arvutatakse jalgratta mudeliga järgnevalt [24]:

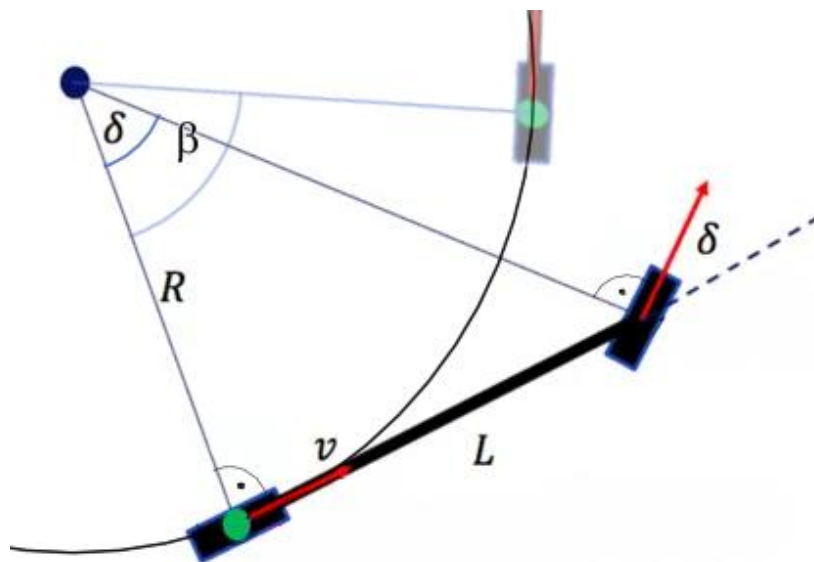
1. esmalt leitakse ringi raadius: $R = \frac{L}{\tan(\delta)}$, kus δ on rataste keeramise nurk ning L on teljevahe meetrites,
2. arvutatakse läbitava ringjoone keskpunkt: $x_k = x - (\sin(\theta) \cdot R)$, $y_k = y + (\cos(\theta) \cdot R)$ kus x, y on praegune asukoht, θ on liikumissuund x -telje suhtes ning R on eelnevalt arvutatud raadius,

- ringjoonel läbitud kaare kesknurga leidmine: $\beta = \frac{v \cdot t}{R}$, kus v on liikumiskiirus, t on kuluv aeg ning R on eelnevalt arvatud raadius,
- arvutatakse uus asukoht: $x_{uus} = x_k + \sin(\theta + \beta) \cdot R$, $y_{uus} = y_k - \cos(\theta + \beta) \cdot R$, kus x_k, y_k on eelnevalt arvatud ringi keskpunkt, β on eelnevalt arvatud kesknurk ning R eelnevalt arvatud raadius.

Kuna LiDAR fikseeritakse auto külge, ei ole vaja liikumise hinnangul arvestada auto asetust muu keskkonna suhtes ning selle saab lugeda nulliks. Võttes auto asukoha koordinaattelgede nullpunktiks, saab valemeid lihtsustada järgnevalt:

- ringi raadiuse leidmine: $R = \frac{L}{\tan(\delta)}$,
- ringjoonel läbitud kaare kesknurga leidmine: $\beta = \frac{v \cdot t}{R}$,
- uue asukoha leidmine algasukoha suhtes: $x = R \cdot \sin(\beta)$, $y = R \cdot (1 - \cos(\beta))$.

Jalgratta mudeli rakendamine vähendab teekonna arvutamise keerukust nii, et olulised andmed jääksid alles: saab arvutada auto asukohta, võttes sisendiks rataste keeramisnurka, teljevahe pikkuse ja liikumiskiiruse. Leides auto asukoha aja suhtes, on võimalik lugeda LiDARI andmeid teatud asukohas.



Joonis 3. Kinemaatilise jalgrattamudeli näide. Auto tagaratas liigub mööda ringjoont [25].

4. Arhitektuuriline ülevaade

4.1 Riistvara lühikirjeldus

Auto on üles ehitatud vabavaralise platvormi Donkey Car² põhjal. Juhtimine toimub läbi Raspberry Pi mikroarvuti, mis edastab signaali digitaalsele mootorikontrollerile edasi liikumiseks ning servomootorile manööverdamiseks. Töö käigus lisati autole GNSS-seade, antenn ja LiDAR ning nende kinnitamiseks disainiti ja prinditi sobiv raam. Auto raamil on olemas postid, mille külge on võimalik paigutada lisaseadmeid. Postide otsa on kinnitatud 3D-prinditud alused, kuhu peale saab fikseerida eelnevalt nimetatud seadmed.

4.2 Olemasolevad komponendid

4.2.1 Auto raam

Töö jaoks on valitud mudelauto Donkey Car S1 (Joonis 4). Igal rattal on eraldi vedrustus pehmendava amordiga. Auto juhtskeem on 4x4, st neljast rattast kõik on vedavad rattad. Lisaks on telgedele lisatud diferentsiaaljuhtimine, mis lubab autoga manöövreid tehes ühe telje mõlemal rattal veereda üksteisest sõltumatu kiirusega. Esitelg on juhitud.

4.2.2 Juhtseadmed

Auto liikumisjõu allikaks on harjastega elektrimootor ning esitelge keerab servomootor E6001, mille maksimaalne rakendatav jõud on 6 kg. Mootorid on eelpaigaldatud auto peale.

4.2.3 Kiirusregulaator Hobbywing WP-1625 Brushed³

Veekindel komponent, mis reguleerib auto mootori kiirust. Võimaldab juhtmootorit nii edaspidi kui ka tagurpidi liigutada. Seadme nimivoolutugevus on 25 amprit ning maksimaalne lühiajaline voolutugevus 100 amprit. Sisendiks sobib pinge vahemikus 6-12 volti. Juhtmetega on eraldatud ülejäänud seadmest lüliti. Regulaator on eelpaigaldatud auto peale.

4.2.4 Mikroarvuti Raspberry Pi 4B

Seadmele on paigaldatud ametlik Raspberry Pi operatsioonisüsteem ning Pythoni käituskeskkond, milles töötab projekti tarkvara. Seadmele on lisatud Donkey Hat⁴, mis koondab kõik vajalikud Pi programmeerimisviigud sobivatesse gruppidesse, mida on lihtne

² Põhjalik dokumentatsioon projektist: <https://docs.donkeycar.com/>

³ Regulaatori infoleht: <https://www.hobbywing.com/en/products/quicrun-wp-1625-brushed53>

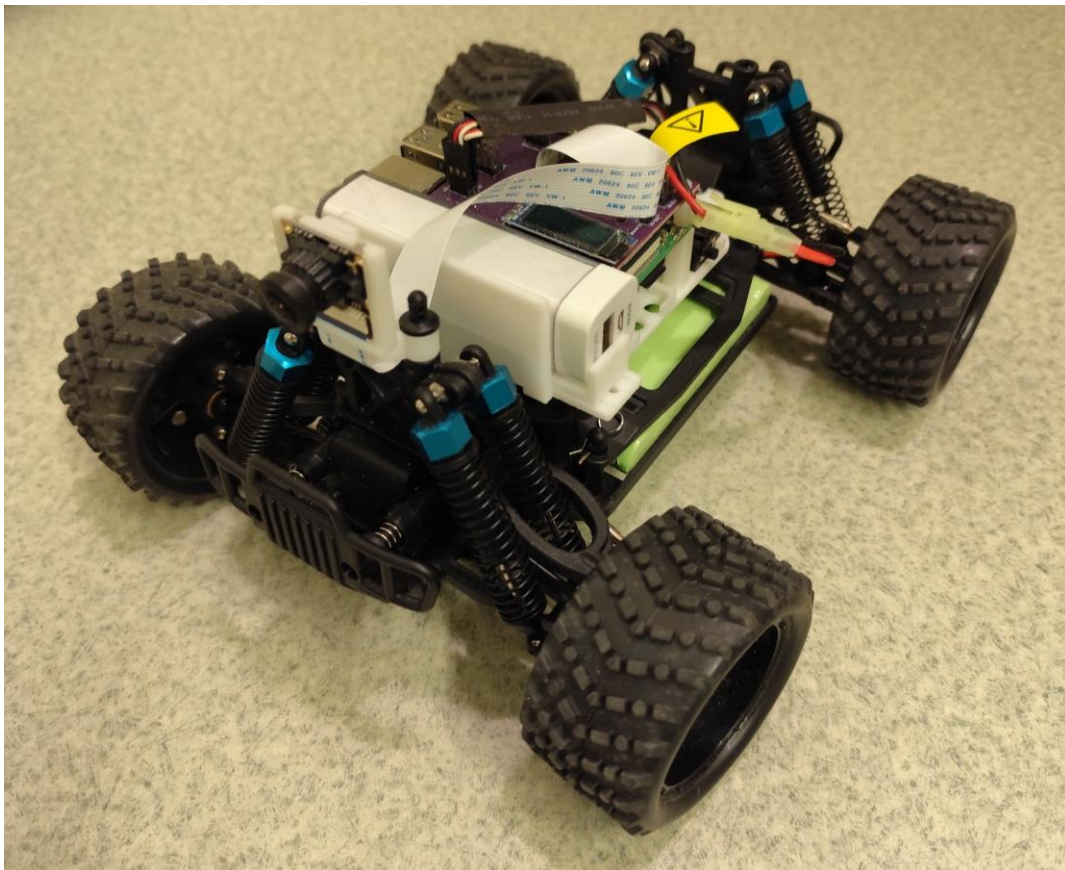
⁴ Dokumentatsioon komponendi kohta: https://docs.donkeycar.com/parts/rc_hat/

ühendada kiirusregulaatori ja servomootoriga. Samuti on Donkey Hat varustatud LCD (*Liquid Crystal Display*) paneeliga, mis näitab töötamise ajal tarvilikku teavet: ühendatud Wi-Fi võrku, seadme nime võrgus, töösoleku aega. Arvuti toide on 5 volti läbi USB juhtme. Seade on eelpaigaldatud auto peale.

Arvutiga on ühendatud ka kaamera, kuid pilditöötluse jaoks suure arvutusvõimsuse nõude tõttu ei saa seda arvuti enda peal teha, seega ei kasutata seda teekonnaarvutuses.

4.2.5 Vooluallikad

Autol on kaks eraldiseisvat vooluallikat. Esimene neist on väike akupank USB juhtme ühendusega. Pinge välja on 5 volti, see sobib Raspberry Pi toiteks. Teine on kuuest nikkel-metall hüdriid (NiMH) elemendist koosnev aku, mille küljes on klemm-pistikühendus. Aku väljundpinge jääb vahemikku 6.2-8.4 volti, mis sobib kiirusregulaatori toiteks.



Joonis 4. Donkey Car projekti näidisauto algseisundis.

4.3 Juurde lisatud komponendid

Selles peatükis on välja toodud eelnevas peatükis nimetatud osadele lisatavad komponendid. Tabelis 2 on märgitud lisatud komponentide hinnad.

Tabel 2. Lisatud toodete hinnad.

Toote nimetus	Hind
SimpleRTK2B Fusion	279.00 € ⁵
NTRIP moodul	66.00 € ⁶
Karp	49.00 € ⁷
Antenn	99.00 € ⁸
Kaitseplaat	8.20 € ⁹
RPLIDAR A1M8	109.80 € ¹⁰
KOKKU	611.00 €

4.3.1 GNSS positsioneerimisseade simpleRTK2B Fusion

Seadme tootja andmelehe järgi [26] põhineb positsioneerimisseade (Joonis 5) u-blox ZED-F9R moodulil, ning on loodud töötama mitme erineva seadmega, sealhulgas Raspberry Pi. Seadmel on indikaatorid, mis näitavad kasutusel oleva GNSS signaali täpsust (RTK olemasolu).

Seadmele on lisatud peale Wi-Fi-moodul⁶, mis ühendub ESTPOS teenusega RTK parandite saamiseks. Mõlemad on ümbritsetud karbiga⁷, mis kaitseb neid ilmastikuolude eest ning teeb seadmete paigaldamise lihtsamaks.

ZED-F9R on seadistatud u-center¹¹ tarkvara abil väljastama NMEA sõnumeid sagedusel 6 Hz teemadel GNRMC (sisaldab andmeid asukoha kohta) ja GNVTG (sisaldab andmeid suuna ja kiiruse kohta).

⁵ SimpleRTK2B Fusion infoleht: <https://www.ardusimple.com/product/simplertk2b-fusion/>

⁶ NTRIP mooduli infoleht: <https://www.ardusimple.com/product/wifi-ntrip-master/>

⁷ Karbi infoleht: <https://www.ardusimple.com/product/plastic-case-simplertk2b/>

⁸ Antenni infoleht: <https://www.ardusimple.com/product/helical-antenna/>

⁹ Maandusplaadi infoleht: <https://www.ardusimple.com/product/ground-plate-for-gnss-antenna/>

¹⁰ LiDARi hind pärineb siit: <https://eu.robotshop.com/products/rplidar-a1m8-360-degree-laser-scanner-development-kit>

¹¹ Lisateave u-center rakenduse kohta: <https://www.u-blox.com/en/product/u-center>



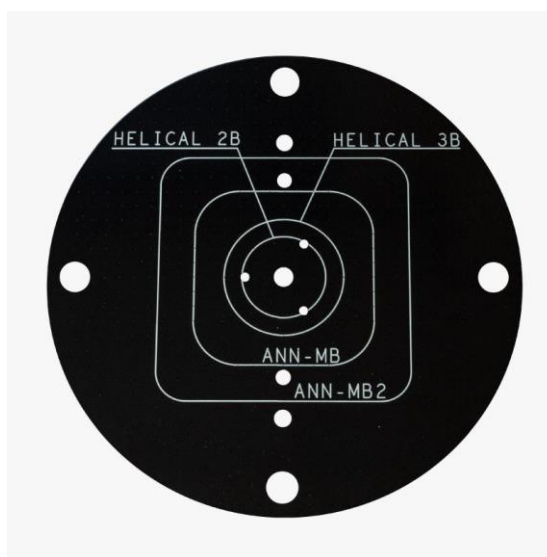
Joonis 5. simpleRTK2B Fusion [27].

4.3.2 Antenn koos kaitseplaadiga

Eelnevalt nimetatud seadmele on juurde ühendatud spiraalantenn⁸ (Joonis 6), mis on disainitud vastu võtma kõiki tüüpe GNSS-signaale igast suunast. Selle alla on paigaldatud maandusplaat⁹ (Joonis 7), mis võimendab kättesaadavat signaali ning kaitseb antenni võimaliku signaalimüra eest, mis võib tulla juuresolevatelt elektroonikaplaatidelt.



Joonis 6. Antenn [28].



Joonis 7. Antenni maandusplaat [29].

4.3.3 2D LiDAR-seade RPLIDAR A1M8

Seade kasutab ainult ühte laserkiirt ning selle peegelduse vastuvõtjat, ning mõõdetav ala saavutatakse kiire saatja ja vastuvõtja mootoriga pöörlemisel ümber enda telje (vaata Joonist 8). See tähendab, et see suudab mõõta kaugusi kahedimensionaalselt, ning mõõtetasand tuleneb LiDAR-seadme asendist. Mõõtmistulemuseks on ringikujuline punktipilv, kus punkti kaugus keskpunktist tähistab kaugust lähima objektini anduri kõrguselt.

Tootja andmete järgi [28] suudab see mõõta vahemaid andurist esimese takistuseni vahemikus 15 sentimeetrit kuni 12 meetrit. Selle mõõtesagedus on alates 2 kuni 10 korda sekundis. Pöörlev laserkiir suudab mõõta kaugust kuni 8000 korda sekundis. Tabelis 3 on välja toodud mõõtmistäpsused sõltuvalt mõõdetavast kaugusest. Lisaks on toodud suurimal kaugusel esinev mõõteviga.

Tabel 3. Tootja poolt antud täpsusvahemikud seadmele RPLIDAR A1M8 [28].

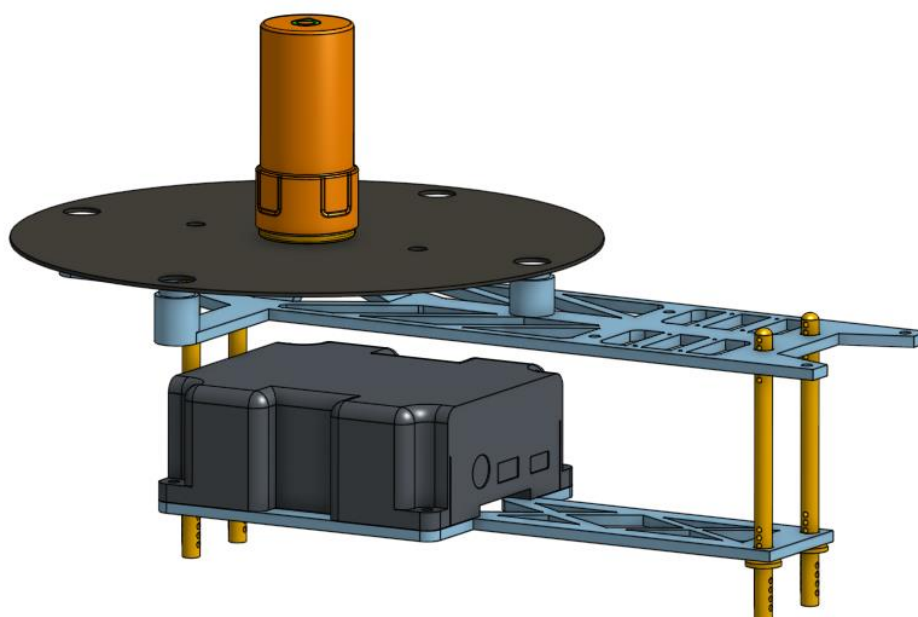
Mõõtekaugus	Mõõtmistäpsus	Mõõteviga suurimal kaugusel
kuni 3 meetrit	1% kaugusest	3 cm
3-5 meetrit	2% kaugusest	10 cm
5-25 meetrit	2,5% kaugusest	62,5 cm



Joonis 8. Kasutatud LiDAR-seade [29].

4.3.4 Kinnitusvahendid

Kõikide eelnimetatud osade kinnitamiseks on töö käigus disainitud ja prinditud kahekorruseline lisaraamistik, kuhu saab paigutada seadmed. Auto raamil on neli auku postide paigutamiseks, originaalpostide asemele on prinditud pikendatud postid. Mudelid on disainitud keskkonnas Onshape ning prinditud printeritega WEEDO Tina2¹², FLSun Super Racer¹³ ja Anycubic Kobra 2 Max¹⁴. Joonisel 9 oleva visuaalse paigutuse nägemiseks on keskkonda imporditud ka teiste komponentide mudelid [30,31].



Joonis 9. Visuaalne paigutus prinditud raami ja olemasolevate komponentidega. Prinditud osad on sinised platvormid ja kuldsed postid. Alumisele korrusele on paigutatud GNSS-seadme karbi mudel ning ülemisele korrusele oranž antenni mudel koos kaitseplaadiga.

4.4 Seadmete vahelised ühendused

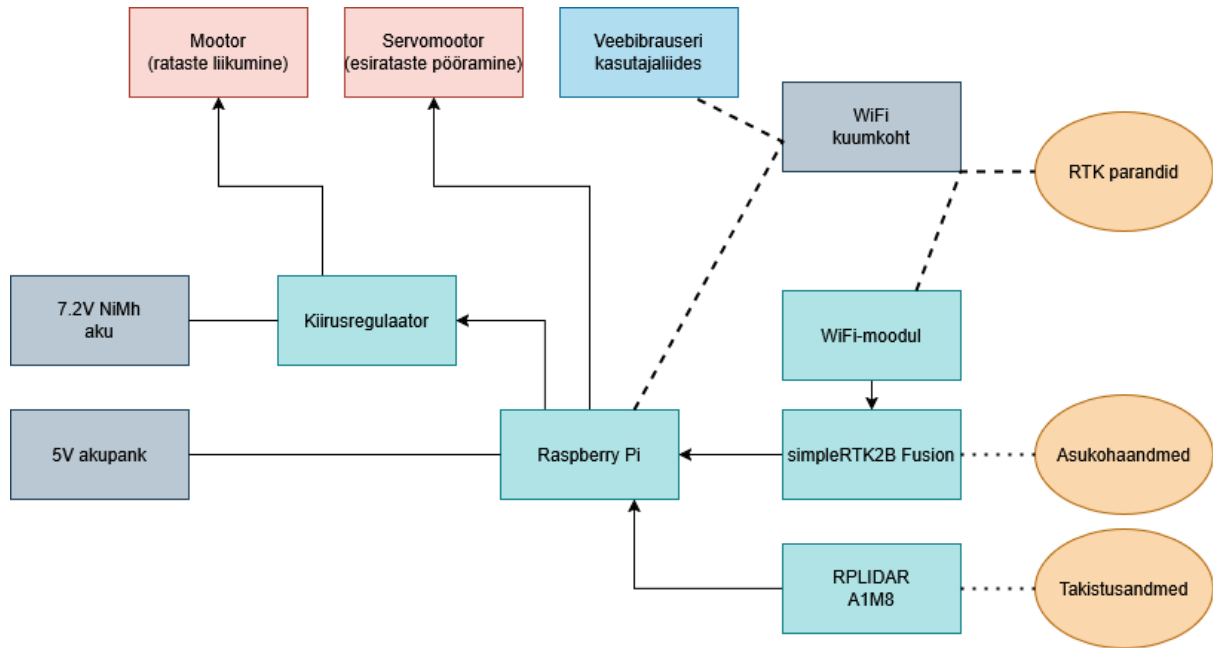
Auto kaks vooluallikat on ühendatud kahe seadmega: 5 volti väljastav akupank on ühendatud mikroarvutiga Raspberry Pi, ning teine NiMH aku on ühendatud kiirusregulaatoriga. GNSS positsioneerimisseade on ühendatud USB-kaabli kaudu Raspberry Pi-ga, ning nende vahel

¹² Printeri WEEDO Tina2 infoleht: <https://www.weedo3d.com/products/weedo-tina>

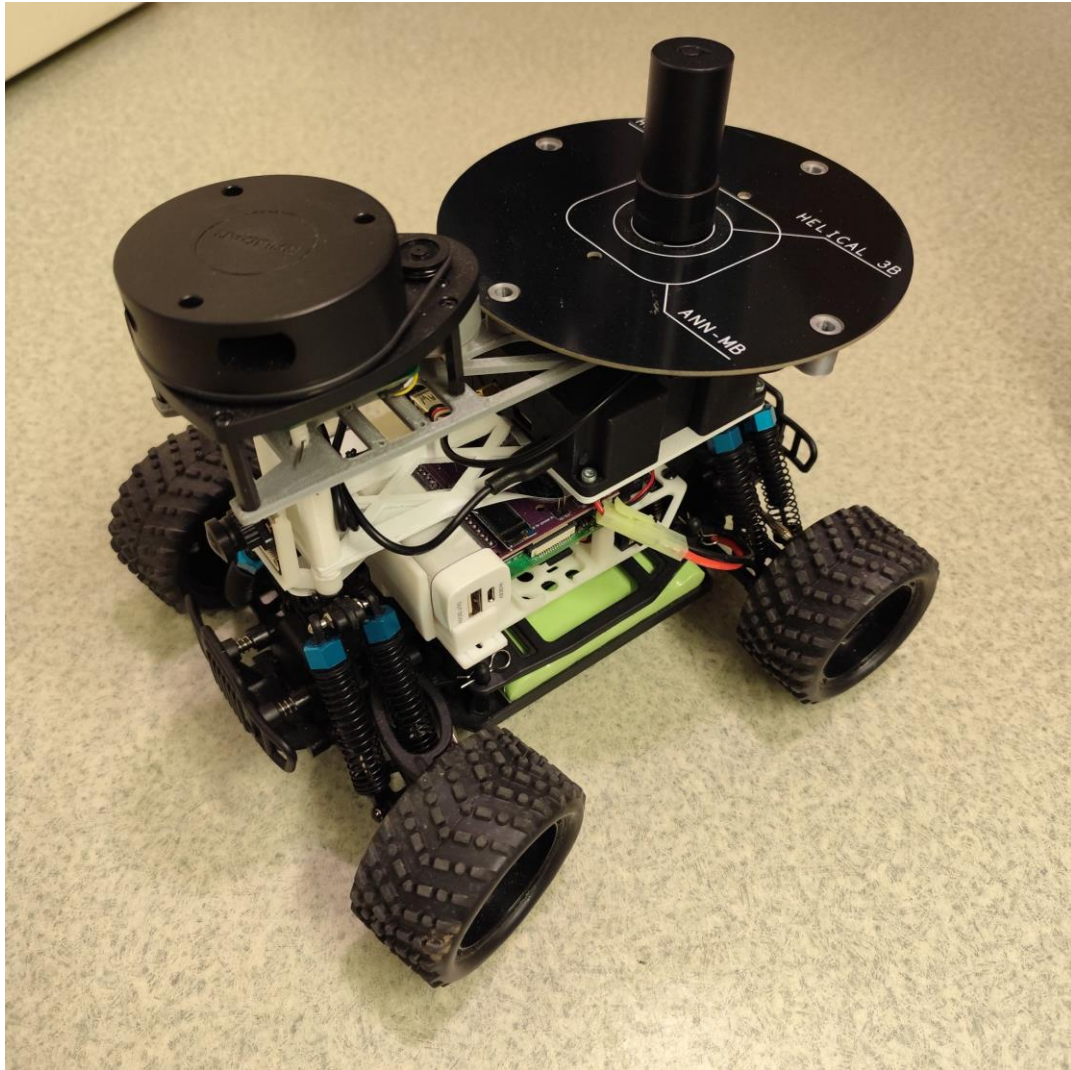
¹³ Printeri FLSun Super Racer infoleht: <https://flsun3d.com/pages/sr>

¹⁴ Printeri Anycubic Kobra 2 Max infoleht: <https://store.anycubic.com/collections/fdm-printer/products/kobra-2-max>

toimub andmeülekanne jadaihenduse kaudu. Antenn on ühendatud GNSS-seadmega. Kiirusregulaator ja Pi on ühendatud selle programmeerimisviikude kaudu, mille leidmine on lihtsamaks tehtud Donkey Hat-i kasutades. Pi ja GNSS-seadmel olev Wi-Fi-moodul on ühendatud telefonist loodud Wi-Fi-kuumkohaga. Võrgu kaudu on võimalik kasutada teisel seadmel kasutajaliidest veebibrauseriga auto juhtimiseks ning eelnimetatud Wi-Fi-moodul ühendub internetiga RTK parandite lugemiseks. Joonisel 10 on näha seadete vahelised ühendused, ning joonisel 11 on auto koos kõikide paigaldatud osadega.



Joonis 10. Visuaalne kujutus kirjeldatud ühendustest. Täpilsed jooned tähistavad andmete kogumist, ning kriipsudega joon tähistab juhtmevaba ühendust. Nooled tähistavad füüsilist ühendust, ja nende vahelist infovoo suunda.



Joonis 11. Auto koos kõikide paigaldatud lisaseadmetega.

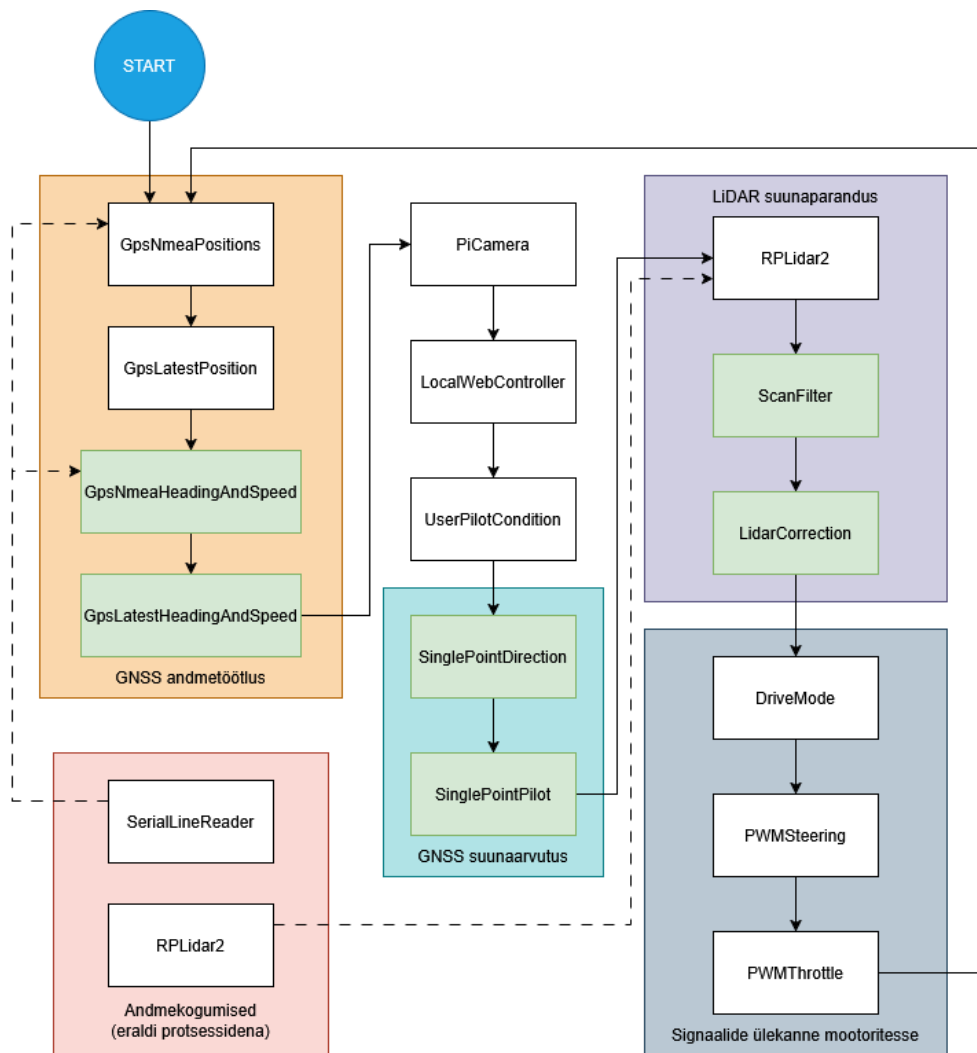
4.5 Tarkvara kirjeldus

Kogu loodav tarkvara on kirjutatud programmeerimiskeeles Python ning töötab mikroarvutil Raspberry Pi 4B. Donkey Car projektil on mahukas baas olemas¹⁵ ning käesolev töö täiendab seda. Algtarkvara lihtsamaks paigaldamiseks on kasutatud operatsioonisüsteemi kettatõmmist¹⁶, mis sisaldab kõik vajaliku auto liikuma saamiseks mängupuldi juhtimisega, ning lisafunktsioone auto ekraanil info näitamiseks, millest olulisemad on ühendatud Wi-Fi võrgu nimi, seadme IP aadress ja seadme parool. Kettatõmmis kasutab Donkey Car tarkvara versiooni 5.0-dev3.

¹⁵ GitHub repositoorium: <https://github.com/autorope/donkeycar>

¹⁶ GitHub repositoorium, mille kirjelduses on alla laetavad kettatõmmised: <https://github.com/robocarstore/donkeycar-images>

Olemasolev projekt on üles ehitatud modulaarselt: kõik tarkvarakomponendid on kirjutatud eraldi komponentidena (*parts*) ning iga auto konfiguratsioon kasutab neid vastavalt vajadusele ja selles järjekorras, kuidas need kirja pandud on. Igal komponendil on oma peafunktsioon ja see tagastab vajalikud andmed. Peafailis on määratud nimetused igale komponendi sisendile ja väljundile, nii on võimalik lihtsasti liigutada vajalikke andmeid nende vahel. Töö ajal käivitatakse järjest iga komponent eraldi läbi (vaata Joonist 12) sellele määratud sisendkanalite andmetega ning selle tagastatud andmed salvestatakse määratud väljundkanalitesse.



Joonis 12. Loodud programmi loogiline ülesehitus. Nool tähistab komponentide tööjärjekorda. Rohelised komponendid on töö eesmärgi täitmiseks juurde kirjutatud.

4.5.1 Juurde lisatud tarkvara

Käesoleva töö raames on projektile juurde kirjutatud nimetatud failidesse järgnev funktsionaalsus¹⁷:

1. **parts/gps.py**: klassid **GpsNmeaHeadingAndSpeed** ja **GpsLatestHeadingAndSpeed** - GNSS andmete ümber töötlemine liikumissuunaks ja kiiruseks,
2. **parts/lidar.py**: klassid **ScanFilter**, **LidarValidator** ja **LidarVisualizer** - LiDAR-andmete ümber töötlemine ja ühtlustamine vigade vältimiseks ja mugavamaks hilisemaks töötluks, teine aitab hinnata LiDARi täpsust ja viimane näitab LiDARi väärtusi reaajas konsoolis,
3. **parts/path.py**: klass **SinglePointDirection** - õige liikumissuuna leidmiseks auto praegusest asukohast,
4. **parts/path.py**: klassid **SinglePointPilot** ja **LidarCorrection** - esimene arvutab PID-kontrolleri abil vajaliku rataste pööramisnurka, ning teine leiab parima suuna pööramise suunas, mis väldiks eesolevaid takistusi,
5. **parts/kinematics.py**: klass **SimpleBicycle** - võimaldab hinnata rataste keeramise ja kiiruse järgi hinnata, kui kaugel ja mis suunas on auto 1 sekundi pärast, et eelnevalt nimetatud klass **LidarCorrection** saaks hinnata takistuste olemasolu,
6. **parts/web_controller/web.py**: klass **SetDestination** - loob olemasolevale veebiserverile juurde uue POST-veebiotspunkti, kuhu saab saata uue sihtkoha koordinaadid
7. **parts/web_controller/templates/vehicle.html** ja **templates/static/main.js**: muudetud kasutajaliidest, et eelneva **SetDestination** klassile mugavalt saata uusi koordinaate
8. **templates/drive_to_destination.py** ja **cfg_drive_to_destination.py**: esimene on peafail, mis defineerib soovitud järjekorras valitud komponendid, laeb need sisse ja haldab andmete voogu nende vahel. Komponentide parameetrid on salvestatud teises failis.

Auto liikumise loogika:

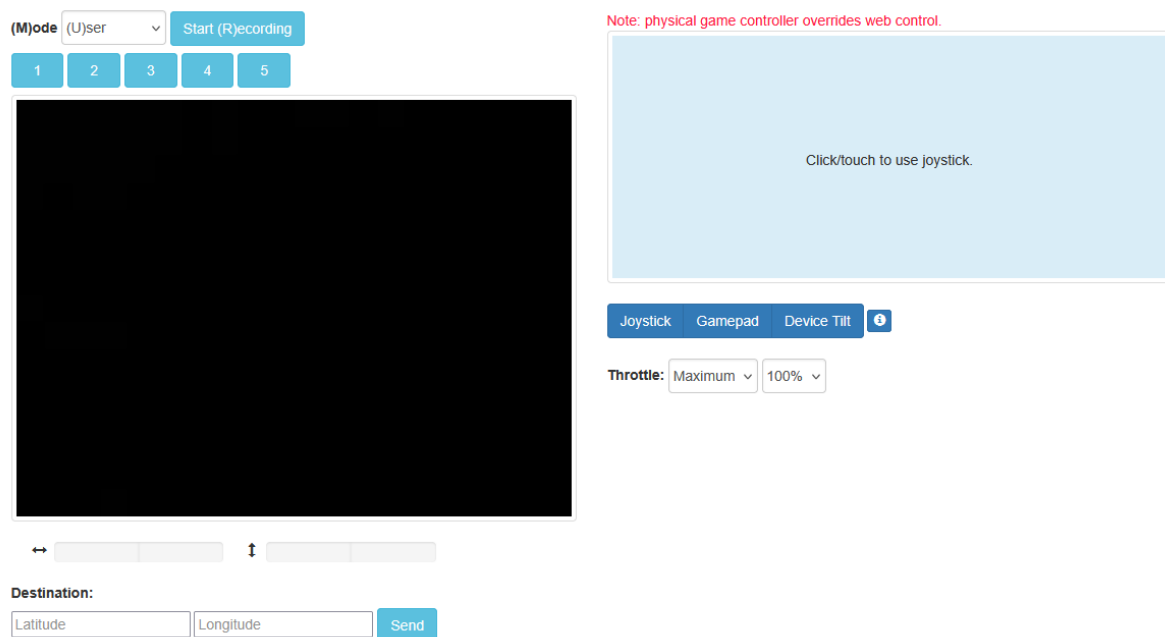
- Kasutades klassi **SinglePointPilot**, arvutab see esirataste asendi määratud sihtpunkti suunas.

¹⁷ Projekti haru GitHub repositoorium: <https://github.com/kennark/donkeycar>

- Seejärel arvutab LiDAR-andmete abil lähima “ohutu” teekonna, kuhu ei jää ühtegi takistust ette ning on ka lähim eelmises punktis arvutatud suunale.
- Saadud tulemus saata edasi esirattaid keeravale klassile.

4.5.2 Kasutajakogemus

Sisseehitatud veebiserver avab kasutajaliidese (vaata Joonist 13) auto IP aadressile pordiga 8887, kust saab juhtida autot seadmesse ühendatud mängupuldiga või kasutades liidesel olevat virtuaalset juhtpulti, samuti saab näha auto kaamerapilti, seada sihtkohaks uued koordinaadid ning muuta auto juhtrežiimi.



Joonis 13. Veebibrauseri kasutajaliides. Musta ruudu asemel kujutatakse sõidu ajal kaamerapilti. Kasutajaliidesele on lisatud väli Destination, millega saab autole uue sihtkoha koordinaate seada.

Auto võimalikud juhtrežiimid on:

- “user”, kus kasutaja saab autot juhtida manuaalselt,
- “full auto”, millega sõidab auto täielikult ise. Selles režiimis ise juhtides muudab automaatselt “user” peale tagasi,
- “auto steer”, millega on määratud ainult rataste keeramissuund ning kiirust ei rakendata. Sellega saab eelkõige kontrollida, kas auto rattad on sihtkoha poole suunatud.

Üldine kasutajakogemus pärast tarkvara käivitamist: kasutaja avab veebiliidese, sisestab sinna koordinaadid, ning seadistab auto režiimi “full auto” iseseisvaks juhtimiseks. Liikumise

alustamiseks peab autot hetkeks lükkama, et kinnitada liikumissuund, ning seejärel sõidab see ise kuni jõuab sihtkoordinaadile eelmääratud meetrite lähedusse. Sõitu on võimalik katkestada, muutes auto juhtrežiimi “user” peale.

4.5.3 Juhtmoodulite tööpõhimõtted

Donkey Car projektiga on võimalik teha palju erinevaid konfiguratsioone ning selleks on loodud “template” kaust, kuhu on loodud töö eesmärgi täitmiseks uus fail `drive_to_destination.py`. Faili kasutamiseks tuleb sellest aga luua uus autoprojekt, ehk konsoolist käivitada käsk `donkey createcar -template=drive_to_destination`¹⁸. Seejärel saab genereeritud kaustast käivitada auto käsuga `python manage.py drive`.

Töötamise ajal toimub taustal eraldi protsessina LiDAR-andmete ja GNSS-andmete sisselugemine. Kuna LiDAR on pöörleva teljega, siis mõõdetavad kraadid ei ole täisarvud, vaid varieeruvad kahe komakoha täpsusega. ScanFilter võtab algsed andmed ja ühtlustab need ehk ümardab kõik kraadiväärtused täisarvudeks formaadis {suunakraad : kaugus}. Seejärel suurendab see takistuse ulatust eelmääratud arvu kraadide võrra nii paremale kui vasakule. Niimoodi vähendab see võimalikke kohti, kus andmetes võib olla auke, samuti loob takistustele puhvriala ohutumaks teekonnaarvutuseks. GpsNmeaPositions tõlgib ümber märgendiga \$GNRMC sõnumeid ning GpsLatestPosition tagastab kõige viimase asukohateabe. Analoogselt ka GpsNmeaDirection tõlgib ümber märgendiga \$GNVTG sõnumeid ning GpsLatestHeadingAndSpeed tagastab kõige viimase suuna- ja kiiruseteabe.

Juhtimisel on peamised klassid SinglePointDirection, SinglePointPilot ja LidarCorrection. SinglePointDirection võtab sisendiks kasutajaliideses sisestatud koordinaadid ja praegused koordinaadid, teeb nendest punktidest vektori, ning arvutab välja vektori suuna ehk liikumissuuna kraadides, ning vektori pikkuse ehk kauguse sihtkohast meetrites. SinglePointPilot arvutab välja vea praeguse ja nõutava liikumissuuna vahel, ning edastab selle PID-kontrollerile, mis tagastab sobiva rataste keeramise suuna. Sõltuvalt kontrollerile antud parameetritest võib see tagastada liiga suuri väärtusi, seega on keeramise väärtus piiratud vahemikku [-1, 1]. Kiirus on fikseeritud eelmääratud arvuga vahemikus [0, 1]. Kui kaugus sihtkohast on vähem, kui eelmääratud arv meetreid, on kiirus 0, ehk auto lõpetab liikumise.

¹⁸ Täpsem selgitus käsust: <https://docs.donkeycar.com/utility/donkey/#create-car>

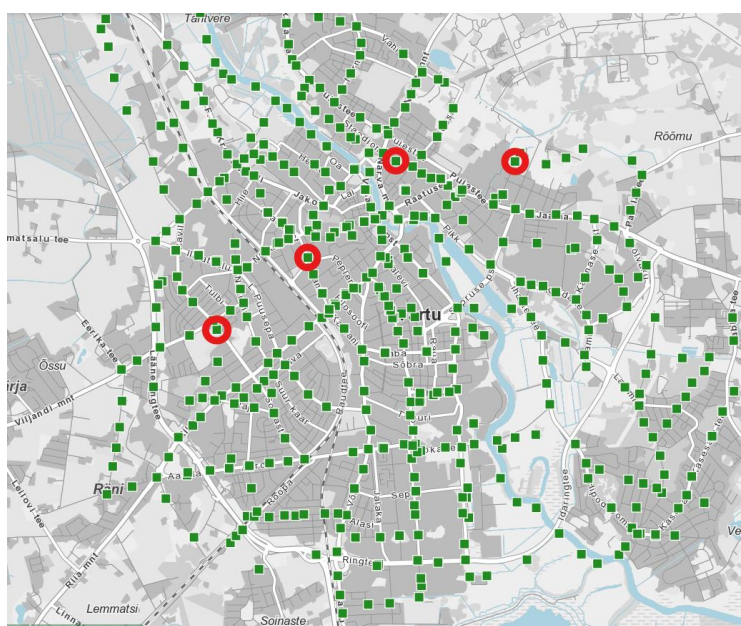
Viimasena võtab LidarCorrection üle saadud rataste keeramisväärtuse, ning arvutab jalgrattamudeli abil auto asukoha 1 sekundi pärast algasendi suhtes. Kui saadud asukohas asub takistus ees, otsib see kõikidest võimalikest auto keeramisväärtustest (eelmääratud parameetriga max_steering_angle) parima teekonna, kus takistusi pole. Kui ühtegi ohutut teekonda ei ole võimalik võtta (igal pool on ees takistus), siis auto jääb seisma ning ootab, kuni kasutaja tõstab auto ümber avaramasse kohta.

Andmete puudumisel saadavad puudutatud komponendid hoiatusteate ning sellisel juhul auto ei liigu edasi kuni uued andmed saabuvad.

5. Autoga päris maailmas sõitmine

5.1 GNSS-seadme täpsuse hindamine

Seadme täpsuse hindamiseks on kasutatud Maa-ameti hallatavaid kohalikke geodeetilisi punkte¹⁹, mille asukoht koordinaatides on fikseeritud ning avalikult kättesaadav. Neid leidub üle Eesti kokku 65 asulas. Maa-ameti kaardirakendusest²⁰ leiab kõikide punktide kohta kirjeldusest nii täpsed koordinaadid kui ka sõnalised kirjeldused punkti kergemaks leidmiseks. Käesoleva töö raames on valitud neli punkti Tartu linna ümbruses (asukohad märgitud joonisel 14) seadme täpsuse hindamiseks.



Joonis 14. Kuvatõmmis Maa-ameti kaardirakendusest [32]. Rohelised täpid on Tartu linna kohalikud geodeetilised punktid. Seadme täpsuse hindamised on läbi viidud punase ringiga tähistatud kohtades.

Hindamised viidi läbi, hoides seadme antenni käega punkti tsentrit tähistava märgi kohal ning seejärel pannes kirja saadud koordinaadid. Tabel 4 näitab punktides läbi viidud mõõtmiste tulemusi, ning mõõdetud koordinaatide erinevust tegelikust sentimeetrites.

¹⁹ Rohkem infot kasutatud geodeetiliste punktide kohta:

<https://geoportaal.maaamet.ee/est/ruumiandmed/geodeetilised-andmed/geodeetilised-vorgud/kohalik-geodeetiline-vork-p284.html>

²⁰ Kaardirakendus: <https://xgis.maaamet.ee/xgis2/page/app/gpa>

Rakenduse avades märkida külgriba pealt “Kohalik geodeetiline võrk”.

Tabel 4. Geodeetiliste punktidega läbi viidud asukohamõõtmised.

Punkti number	Oodatav laiuskraad	Mõõdetud laiuskraad	Oodatav pikkuskraad	Mõõdetud pikkuskraad	Punktide erinevus (cm)
11433	58.36919097	58.36919100	26.69121841	26.69121850	0.6219
14	58.37665091	58.37665083	26.71072182	26.71072217	2.041
1761	58.38678357	58.38678350	26.72983046	26.72983100	3.242
10670	58.38621183	58.38621183	26.75457614	26.75457650	2.098

Keskmise erinevusega umbes 2.00 sentimeetrit on seade üsna täpne. Kui auto on 25 cm pikk ja 23 cm lai, siis 2 cm erinevus ei ole oluline pikema vahemaa läbimiseks. Arvestades, et käsitsi läbi viidud mõõtmistel on suur mõõtemääramatus, siis saadud seadme täpsus on piisav käesoleva töö eesmärgi täitmiseks.

5.2 LiDAR-seadme täpsuse mõõtmine

LiDARi lahendus on projektis üles ehitatud niimoodi, et protsessi lõpetamisel väljastab see info mõõdetud andmete hulga kohta: mitu mõõtmist kokku tehtud, mitu tiiru sekundis toimus mõõtmine ja mitu mõõtmist 1 sekundi jooksul tehti. Need arvud varieeruvad üsna suurel määral. Korrates katset 10 korda, saab keskmise mõõtesageduse.

Pärast kümnet korda protsessi käivitamist ning saadud logidest keskmise välja arvutamist on keskmine mõõtesagedus 4.08 Hz. See ei ole seadme suurim võimalik mõõtesagedus, milleks on kuni 10 Hz. Kui aga eemaldada tööjärjekorrast GNSS-seade, mis kasutab ka enda andmete lugemiseks teist protsessi, tõuseb mõõtesagedus 6.5 Hz peale. See näitab, et tõenäoliselt on kas protsesside käitus halvasti optimeeritud või arvutil ei ole piisavalt arvutusjõudu. Sellegipoolest, väiksematel kiirustel on 4 Hz piisav sagedus, et reaajas trajektoori otsuseid teha. Kuid liikudes kiirusega 2 meetrit sekundis, tähendab see andmete uuendust iga 50 sentimeetrise teekonna läbimise järel.

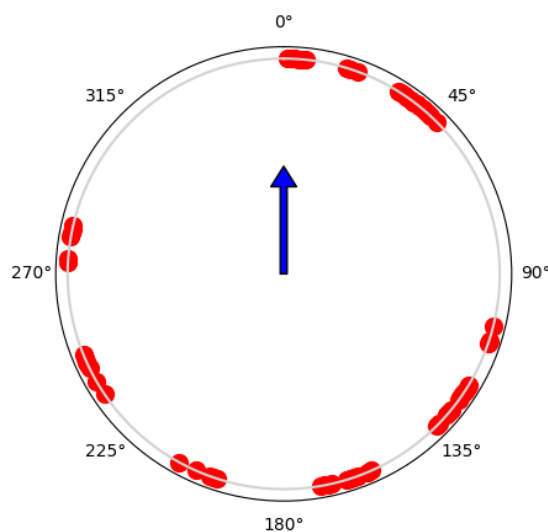
Kuna kasutatud LiDAR on enda telje ümber pöörlev (vaata Joonist 8), ei ole nurgakraadid kindlad ega täisarvulised; mõõdetav nurk varieerub 2 komakoha täpsusega. ScanFilter klass muudab need loetavaks, ümardades kõik kraadid täisarvudeks. Seadme hindamiseks on loodud klass LidarValidator, mis vaatab:

1. kas andmed on terviklikud, st ei esine lünki ka pärast andmete eeltöötlust,

2. kas andmed on muutumatud, st sama koha pealt uuesti mõõtmise toob sama tulemuse, mis varem.

Klass väljastab andmed töö käigus konsoolile formaadis kraad, andmetes esinenud lünkade arv, lähim mõõdetud kaugus, suurim mõõdetud kaugus. Katse läbiviimiseks on auto paigutatud kinnisesse siseruumi, ning autot ei liigutata, seega kaugused takistustest ei muutu. Andmete kogumine kestis 2 minutit ning kogumisest on välja jäetud antenni poolt blokeeritud alad anduri taga.

Katse kestvuse jooksul leidsid püsivad augud 64 erineval suunakraadil (vaata Joonist 15), kogu aja jooksul ei loetud nendele kraadidele ühtegi kaugusväärtust. Ülejäänud kraadidel andmete puudumise sagedus oli 36% katses tehtud mõõdetest. Mõõtetulemused varieerusid maksimaalselt 5 sentimeetri võrra. Mõõdetud kaugus oli suurima variatsiooniga asukohas 1,3 meetrit ning määratud täpsusvahemik sellele on 2,6 sentimeetrit. Eeldades, et tegelik kaugus jääb mõõtevea vahemiku keskele, on tulemuseks 2,5 sentimeetrine viga, mis vastab tootja poolt määratud täpsusvahemikule. Keskmise mõõteerinevus oli aga 8 millimeetrit, mis on väga täpne ning jääb kõikidel kaugustel tootja poolt määratud täpsusvahemikku.



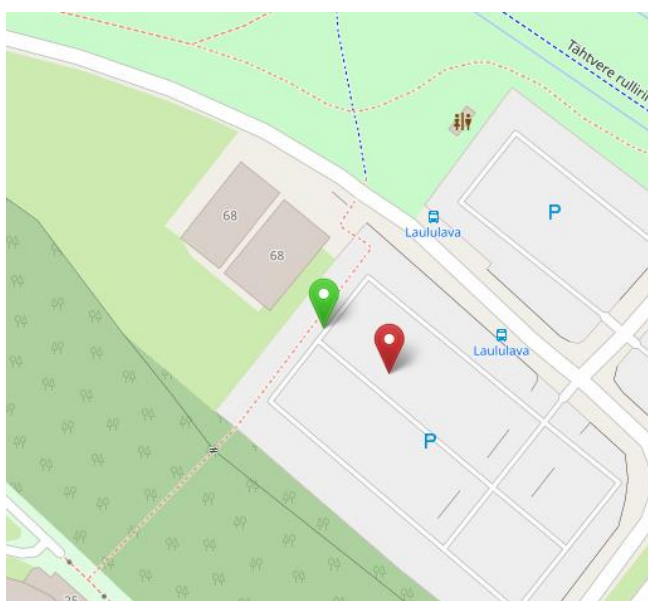
Joonis 15. Kohad, kus LiDAR ei mõõtnud ühtegi kaugust 2 minuti jooksul. Punased alad on nn “pimedad kohad”. Auto liikumissuund on tähistatud noolega.

Aukude vältimiseks on sisse ehitatud ScanFilter klassi takistuste ulatuse laiendamine, et tekitada nn “ohuala” nende ümber ning vähendada joonisel 15 tekkivat olukorda. Erinevate väärtuste katsetamise käigus saavutati parim tulemus 8-kraadise laiendusega. See lahendus kannatab kaugusväärtuste ühtluse poolelt, kuid andmetes esineb vähem puudujääke.

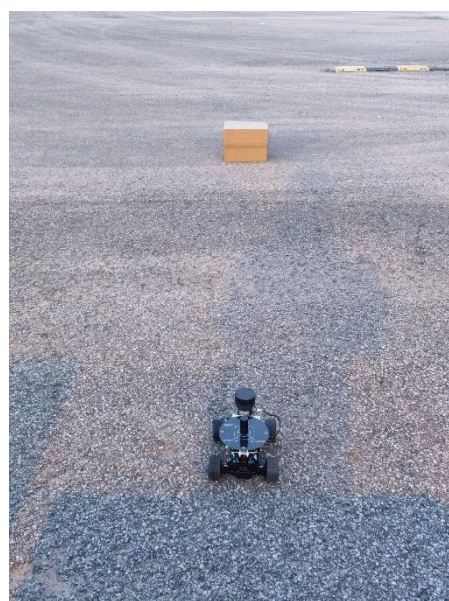
saavutati 5 km/h ehk 1.39 m/s. Kuid kuna autol ei ole võimalik hinnata eesolevaid takistusi, sõidab see esimese takistuse sisse. Seega, ainult GNSS ei ole piisav autonoomseks sõitmiseks.

5.3.2 Sõitmine koos LiDARiga

Selles alapeatükis on eelkõige toodud välja auto oskus takistustest hoiduda. Esimene valitud teekond on sirgejooneline (vaata Joonist 18) ning selle peale on asetatud 2 meetri kaugusele takistuseks 30 sentimeetrit kõrge pappkast nii, et see jääks LiDARi andurile ette (vaata Joonist 19). Valitud on pappkast, et vähendada auto purunemise ohtu. Teine teekond selgitab välja, kas LiDARiga on võimalik mööduda suurematest takistustest, nagu ümber maja nurga sõitmine. Sihtkoht on jäetud samaks, aga auto on liigutatud 1 meetri kaugemale, lillepeenra kasti taha ning auto ise keeratud 90 kraadi.



Joonis 18. Valitud teekond, mis sisaldas ühte takistust [33].



Joonis 19. Auto sirgejooneline teekond, mille ette on paigutatud pappkast.

LiDARi sisse lülitades sõidab auto takistusteta teekonda sarnaselt eelnevale katsele sõita ilma LiDARita. Videotest²² on näha, et otse sõitmiseks ei esine probleeme, kui takistust ei ole ees. Kui sihtpunkt on auto taga ning peab ümber keerama, on näha, et auto ei reageeri suuna muutusele samamoodi nagu enne, esineb ülejuhitavus ning auto peab rohkem korrektsioone hiljem läbi viima, et jõuda sihtpunkti. See võib tähendada, et andmeid ei värskendata piisavalt

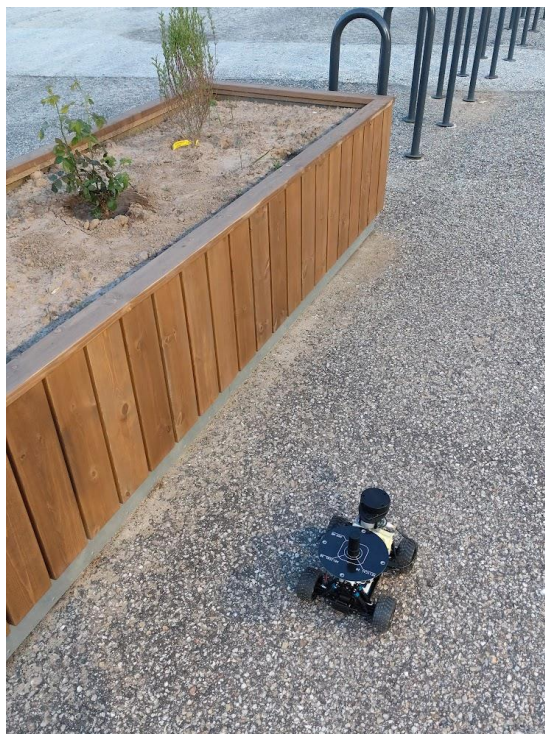
²² Videod LiDARiga läbi viidud katsetest:

https://drive.google.com/drive/folders/101pqsD_QpNV3rEc_ePSPkK6lxr8zAj-K

tihedalt. Sama ülesande trajektoor ilma LiDARita eelmises peatükis oli palju sirgjoonelisem ning ei esinenud teise suunda manööverdamist.

Kui aga tekib takistus ette (antud katsetes pappkast), siis enamiku katsetest ignoreeris auto takistust ehk sõitis otse sisse. Läbi viidud kuue katse korral ei suutnud auto kordagi täielikult eemale hoida kastist. Ühel korral pööras auto järsult kastist eemale, kahel korral sõitis ühe küljega vastu ning kolmel korral sõitis auto täielikult vastu kasti.

Asetades auto lillepeenra kasti taha (vaata Joonist 20) eesmärgiga sõita ümber selle, ei saanud see kahel läbi viidud katsel ülesandega hakkama. Mõlemal korral eiras see LiDARist saadud andmeid ning sõitis otse takistuse sisse, nagu seda ei oleks seal.

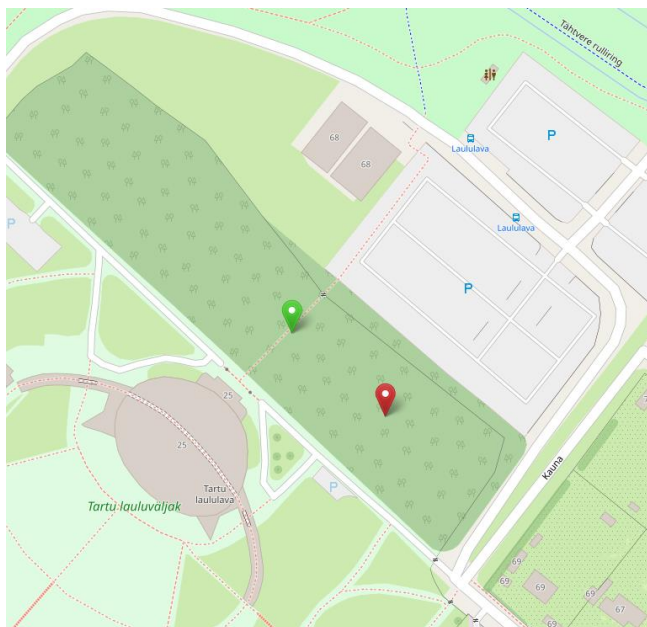


Joonis 20. Auto paigutatud peenrakasti taha. Sihtkoht on määratud teisele poole kasti ning otsesuunas teekond oleks läbi kasti.

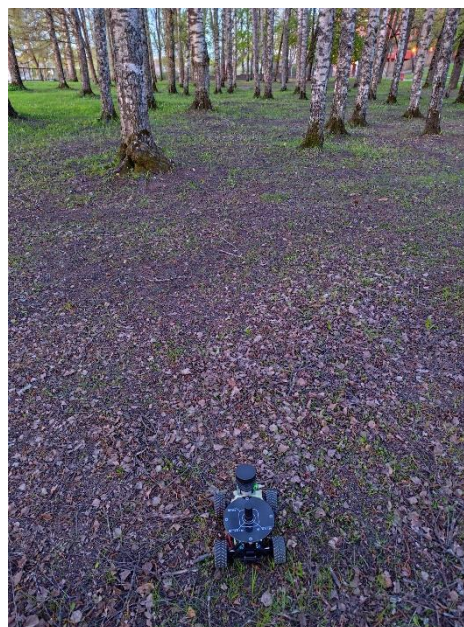
Tõenäoliselt peitub viga programmi loogikas, kuna peatükkides 4.1 ja 4.2 on andmekvaliteet hinnatud piisavaks autoga sõitmisel. Samuti esines ilma takistuseta sõitmisel aeglast keeramisinfo uuendamist. Vaadates LiDARi andmeid sõitmise ajal, on näha, et takistus leitakse, arvutatakse alternatiivne liikumissuund, kuid seda suunaväärtust ei rakendata rataste keeramisena, vaid teekond jätkub otse sihtkoha suunas.

5.3.3 Sõitmine maastikul

Sõitmiseks valiti 40 meetri pikkune sirgjooneline teekond puude vahel (vaata Joonist 21, Joonist 22). Maastik on üldiselt sile, teekonnal esinevad takistused on puud, maha langenud oksad ja üksikud puukännud. Sõitmisel kasutati fikseeritud kiirust 100%.



Joonis 21. Valitud maastikuteekond kaardil [33].



Joonis 22. Avar metsasalu maastikul läbimiseks.

Auto asetades maastikule on näha, et lisatud seadmed on liiga rasked ning vaba ruum auto ja maapinna vahel on 1,7 sentimeetrit (vaata Joonist 23), mis on liiga madal. Videotest²³ on näha, et sirgjooneliselt sõites jääb auto kergesti maapinna kumeruste ja puuokste taha kinni ning edasi liikumiseks peab eemaldama takistuse alt või autot lükkama. Samuti kolmandal katsel jäi see puukännu taha kinni, kuna LiDAR on paigutatud selliste takistuste tuvastamiseks liiga kõrgele. Kõikidel katsel jäi auto seisma.

²³ Videod maastikul sõitmisest:

https://drive.google.com/drive/folders/1Mfy_yfBTkwkOlasDsJQSS78tBymLdvc

6. Arutelu

Valitud komponentide jaoks ei ole Donkey Car S1 kõige sobivam auto, kuna see on disainitud sõitma kiiresti. Kuigi auto on ehitatud sõitma nelja vedava rattaga ning vedrustus aitab kaasa heale haarduvusele, siis oluline on silmas pidada ka muid omadusi, nagu näiteks kõrgus. Auto algkaalule 1,204 kilogrammi on lisatud komponendid kogukaaluga 465 grammi ning auto kõrgus maapinnast vajub poole võrra (vaata Joonist 23). Künklikuma pinna peal on suurem tõenäosus jääda autol põhja peale seisma. Põhjuseks on nõrgad vedrud, neid vahetades tugevamate vastu on võimalik auto uuesti ülespoole tõsta ning sõit sujuvamaks teha. Samuti ei ole mootor mõeldud aeglaseks sõiduks, lisatud raskustega suudab auto liikuda minimaalselt 60% kiirendusega. Asfaldikatsetel kasutati fikseeritud kiirust 80% ning maastikukatsetel tõsteti kiirus 100% peale.



Joonis 23. Auto põhja kõrgus maapinnal olles.

Takistuste vältimiseks on auto esiotsale disainitud koht LiDARi jaoks. Kuid see on paigutatud liiga kõrgele, mis tekitab olukorra, kus takistused, mis jäävad andurist madalamale, jäävad märkamata ning auto sõidab, nagu neid ei oleks seal. Sobivad lahendused on kas raami ümber disainida, et LiDAR oleks võimalikult maadligi, või asendada see mõne kolmemõõtmelise ruumi anduriga, et koguda rohkem andmeid nii alt kui ülevalt.

Kõik katsed on tehtud asfaltpinna peal, sest ebatasasel maastikul ei suutnud auto liikuda iseseisvalt pikka maad. Suuremate tõusude ületamine ei ole võimalik sellise raskuse ja mootoriga.

Arvestades, et peatükis 4.2 toodud hindamiste järgi suureneb GNSS-seadme tööjärjekorrast eemaldamisel LiDARi mõõtesagedus 6.5 Hz-le ning peatükis 4.3.2 sõites sisse lülitatud LiDARiga ei olnud ümberkeeramise teekond sirgjooneline, võib aimata, et Raspberry Pi ei jõua arvutada teekonda piisavalt kiiresti, st jõudlust jääb puudu, et kõike reaajas arvutada. Kuna

töö kasutab aluseks Donkey Car projekti, ei ole töö mahtu arvestades võimalik optimeerida seda piisavalt väiksel suurusel, et nii LiDAR ja GNSS saaksid koos töötada maksimaalsel kiirusel.

Kui tulevikus ehitada auto uuesti, on mõistlikum valida raam, millele on võimalik paigaldada suuremad rehvid. Need teevad auto teekonna stabiilsemaks ning sõites on vähem tunda väikseid ebatasasusi. Samuti aitab paremale tasakaalule kaasa suurem teljevahe. See vähendab maastiku konarlikkusest tulenevat auto kõikumist ning sellega on LiDARiga saadavad andmed ühtlasemad. Maastikul liikumiseks sobib rohkem mootor, mis suudab madalal kiirusel sõita. Muidugi on nimetatud soovitustel ka halvemad küljed, kõige suurem neist on halvenenud juhitevus ja suurem pöörderaadius.

Kokkuvõte

Käesolevas töös loodi sõiduk, mis suudaks sõita vabas looduses. Mudelautole Donkey Car S1 on disainitud ning printitud juurde lisaraam, kuhu peale on kinnitatud GNSS-seade SimpleRTK2B Fusion koos antenniga ning LiDAR-seade RPLIDAR A1M8. Töö käigus täiendati olemasolevat Donkey Car projekti vajalike funktsioonidega, et auto suudaks sõita iseseisvalt ette antud sihtkohta, hoides eemale teekonnal olevatest takistustest.

Autole lisatud komponentide hindamiseks viidi läbi erinevaid katseid. GNSS-seadme täpsust kontrolliti Maa-ameti kohalike geodeetiliste punktidega. Tulemused osutusid piisavaks töö eesmärkide saavutamiseks, võttes arvesse kasutatud hindamismetoodikat. LiDARiga hinnati mõõdetud kauguste täpsust ning andmetes aukude olemasolu. Hoides seadet töös kaks minutit ning salvestades kõik suurimad ja vähimad mõõdetud kaugused iga nurgakraadi kohta, hinnati seadme mõõteväärtused vastavaks tootja andmelehtedega. Leiti, et sama tööaja jooksul leidis täielikke andmeauke mitmes kohas üle kogu mõõteala, kuid laiendades kõrvalolevaid andmeid, on võimalik enamik auke ära eemaldada.

Sõiduuskuste hindamiseks valiti kolm erinevat meetodit: esimene ainult GNSS-andmetega sõitmine kõvakattega teel, teine koos LiDARi andmetega sõitmine samal teel ning kolmas samade andmetega, kuid maastikul. Kõikidel meetoditel seati eesmärgiks sõita läbi 20-30 meetri pikkune testlõik. Kõik katsed dokumenteeriti videolõikudena. Esimese meetodiga sõitis auto edukalt seatud sihtpunkti, kuid kuna auto sõidab ilma takistuste tuvastamiseta, siis sõidab see kõikide ette tulevate takistuste vastu. Mootori võimsusel 80% saavutati kiirus 5 km/h. Teisel meetodil sõitis auto ilma takistusteta samamoodi, nagu esimesel meetodil, kuid asetades takistusena pappkasti ette, sõitis auto kõikidel kordustel sellele vastu. Esines üks katse, kus auto keeras järsult enne takistust, kuid ei suutnud sellest täielikult eemale hoida. Maastikul ei suutnud auto ilma kõrvalise abita sõita, kuna auto jäi loodusliku pinna peale põhjaga kinni.

Auto kogumass suurenes koos kõikide lisatud komponentidega 1600 grammini. Valitud raami jaoks on see raske ning auto põhi vajuks liiga madalale maastikul sõitmiseks, samuti ei olnud võimalik sõita aeglasel kiirusel. Videote alusel on toodud väited, et tarkvara komponendid ei ole piisavalt optimeeritud sellise töö jaoks. Tulevikus edasi arendamiseks on välja toodud soovitusi auto füüsilise poole parandamiseks.

Viidatud kirjandus

- [1] Pangarkar T. Autonomous Vehicles Statistics and Facts (2025). 2025. <https://www.news.market.us/autonomous-vehicles-statistics/> (07.05.2025)
- [2] Carruth D. W., Walden C. T., Goodin C., Fuller S. C. Challenges in Low Infrastructure and Off-Road Automated Driving. 2022 *Fifth International Conference on Connected and Autonomous Driving (MetroCAD)*, 2022. <https://doi.org/10.1109/MetroCAD56305.2022.00008> (07.05.2025)
- [3] Operations - Starship Technologies. <https://www.starship.xyz/operations/> (01.05.2025)
- [4] Lääne J. How Starship Delivery Robots know where they are going. *Medium*, 2024. <https://medium.com/starshiptechnologies/how-starship-delivery-robots-know-where-they-are-going-c97d385a1015> (02.05.2025)
- [5] Our Robots - Starship Technologies. <https://www.starship.xyz/our-robots/> (01.05.2025)
- [6] Campbell S., O'Mahony N., Krpalcova L., Riordan D., Walsh J., Murphy A., et al. Sensor Technology in Autonomous Vehicles : A review. 2018 *29th Irish Signals and Systems Conference (ISSC)*, 2018. <https://doi.org/10.1109/ISSC.2018.8585340> (03.05.2025)
- [7] Vargas J., Alsweiss S., Toker O., Razdan R., Santos J. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. *Sensors*, 2021. <https://doi.org/10.3390/s21165397> (03.05.2025)
- [8] Kosonen P. Autonomous robots out in the wild — a software engineering challenge. *Medium*, 2021. <https://medium.com/starshiptechnologies/running-autonomous-robots-on-city-streets-is-very-much-a-software-engineering-challenge-66927869090a> (02.05.2025)
- [9] Horn Y. How many satellites are needed for GNSS? 2025. <https://www.ncesc.com/geographic-faq/how-many-satellites-are-needed-for-gnss/> (07.05.2025)
- [10] GPS NMEA 0183 Messaging Protocol 101 | Arduino Documentation. <https://docs.arduino.cc/learn/communication/gps-nmea-data-101/> (08.05.2025)
- [11] The NMEA 0183 Protocol. 2001. <https://www.tronico.fi/OH6NT/docs/NMEA0183.pdf> (14.05.2025)
- [12] GPS NMEA sentence structure. 2022. <https://tavotech.com/gps-nmea-sentence-structure/> (14.05.2025)
- [13] Ellmann A., Kall T., Kollo K., Liibus A. GNSS - globaalne satelliitnavigatsioonisüsteem geodeetilise täpsusega asukohamääranguks. Tartu: Eesti Maailikool. 2022.
- [14] GNSS error sources. <https://novatel.com/an-introduction-to-gnss/gnss-error-sources> (15.05.2025)

- [15] Mekik C., Arslanoglu M. Investigation on Accuracies of Real Time Kinematic GPS for GIS Applications. *Remote Sensing*, 2009. <https://doi.org/10.3390/rs1010022> (08.12.2024)
- [16] Ning F.-S., Meng X., Wang Y.-T. Low-Cost Receiver and Network Real-Time Kinematic Positioning for use in Connected and Autonomous Vehicles. *The Journal of Navigation*, 2019. <https://doi.org/10.1017/S037346331800111X> (08.12.2024)
- [17] Lidar 101: An Introduction to Lidar Technology, Data, and Applications. <https://coast.noaa.gov/digitalcoast/training/lidar-101.html> (01.05.2025)
- [18] Ali D. What is the difference between LiDAR and 3D LiDAR? 2025. <https://www.ncesc.com/geographic-faq/what-is-the-difference-between-lidar-and-3d-lidar/> (01.05.2025)
- [19] The Value of 2D LiDAR in Manufacturing Applications. 2024. <https://sickconnect.com/the-value-of-2d-lidar-in-manufacturing-applications/> (01.05.2025)
- [20] Mazzari V. How to select the right LiDAR? *Génér. Robots - Blog*, 2019. <https://www.generationrobots.com/blog/en/how-to-select-the-right-lidar/> (01.05.2025)
- [21] Moses M. LIDAR Versus Camera. 2021. <https://lidarandradar.com/lidar-versus-camera/> (14.05.2025)
- [22] Dubois A.-S. A detailed comparison of LiDAR, Radar and Camera Technology. 2023. <https://insights.outsight.ai/how-does-lidar-compares-to-cameras-and-radars/> (15.05.2025)
- [23] Wang P. Research on Comparison of LiDAR and Camera in Autonomous Driving. *Journal of Physics: Conference Series*, 2021. <https://doi.org/10.1088/1742-6596/2093/1/012032> (02.05.2025)
- [24] Kinematic Bicycle Model: 101 - OMSCS Notes. <https://www.omscs-notes.com/robotics-ai-techniques/kinematic-bicycle-model/> (07.05.2025)
- [25] Ding Y. Simple Understanding of Kinematic Bicycle Model. *Medium*, 2021. <https://dingyan89.medium.com/simple-understanding-of-kinematic-bicycle-model-81cac6420357> (08.05.2025)
- [26] simpleRTK2B Fusion Datasheet. https://www.ardusimple.com/wp-admin/admin-post.php?action=generate_product_pdf&product_id=17302 (02.05.2025)
- [27] simpleRTK2B Fusion. <https://www.ardusimple.com/product/simplertk2b-fusion/> (02.05.2025)
- [28] RPLIDAR-A1 Laser Range Scanner Parameters | SLAMTEC. <https://www.slamtec.com/en/Lidar/A1Spec> (28.04.2025)
- [29] RPLiDAR A1M8-R6 360 Degree Laser Scanner Kit. <https://www.seeedstudio.com/RPLiDAR-A1M8-R6-360-Degree-Laser-Scanner-Kit-12M-Range-p-4785.html> (15.05.2025)
- [30] Plastic Case for simpleRTK boards. <https://www.ardusimple.com/product/plastic-case-simplertk2b/> (28.04.2025)

[31] Lightweight helical antenna for multiband GNSS (IP67).
<https://www.ardusimple.com/product/helical-antenna/> (28.04.2025)

[32] X-GIS 2.0 [gpa]. <https://xgis.maaamet.ee/xgis2/page/app/gpa> (05.05.2025)

[33] OpenStreetMap. <https://www.openstreetmap.org> (15.05.2025)

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Kennar Kahju ,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Navigeerimine välitingimustes mudelautoga

(*lõputöö pealkiri*)

mille juhendaja on Tambet Matiisen ,
(*juhendaja nimi*)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;
4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Kennar Kahju

15.05.2025