

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Markkus Koddala
Tartu Ülikooli isejuhtivate sõidukite labori
väikesemõõtmelise auto kaugjuhtimise arendus
Bakalaureusetöö (9 EAP)

Juhendaja: Ardi Tampuu, PhD

Tartu 2025

Tartu Ülikooli isejuhtivate sõidukite labori väikesemõõtmelise auto kaugjuhtimise arendus

Lühikokkuvõte:

Töö eesmärk on arendada väiksemõõtmelise mänguauto kaugjuhtimissüsteem, mis toimib Tartu Ülikooli Delta õppehoone Wi-Fi-võrgus ja rakendab katkestuste korral autonoomset juhtimist. Lahendus põhineb DonkeyCar platvormil ning koosneb veebipõhisest kasutajaliidesest ja Raspberry Pi-l töötavast sõidukipoolsest rakendusest. Sõiduki ja kasutaja vahel toimub reaajas andmevahetus WebSocket-i protokolliga kaudu, võimaldades juhtimiskäskude ja videopildi sujuvat edastamist. Testid viidi läbi Delta õppehoone III korrusel, kus mõõdeti süsteemi töökindlust mitme pääsupunkti tingimustes. Tulemused näitasid, et loodud süsteem suudab katkestuste korral töökindlalt lülituda autonoomsele režiimile ning sobib hästi teleoperatsiooni ja autonoomia uurimiseks kontrollitud keskkonnas.

Võtmesõnad: kaugjuhtimine, autonoomne sõit, Wi-Fi, latentsus, WebSocket, DonkeyCar

CERCS: T125 Automatiseerimine, robotika, control engineering

Development of a Remote Control System for a Small-Scale Car in the University of Tartu Autonomous Driving Lab

Abstract:

The aim of this thesis is to develop a remote control system for a small-scale model car that operates within the Wi-Fi network of the University of Tartu's Delta building and switches to autonomous control in case of connection interruptions. The solution is based on the DonkeyCar platform and consists of a web-based user interface and a vehicle-side application running on a Raspberry Pi. Real-time data exchange between the vehicle and the user is achieved via the WebSocket protocol, enabling smooth transmission of control commands and video feed. Tests were conducted on the third floor of the Delta building to evaluate the system's reliability under multiple access point conditions. The results demonstrated that the developed system can reliably switch to autonomous mode during interruptions and is well-suited for exploring teleoperation and autonomy in a controlled environment.

Keywords: remote control, autonomous driving, Wi-Fi, latency, WebSocket, DonkeyCar

CERCS: T125 Automation, robotics, control engineering

Sisukord

Sissejuhatus.....	5
1. Teoreetiline taust.....	6
1.1 Platvormi valik ja riistvaraline ülesehitus.....	6
1.2 Autonoomsed ja kaugjuhitavad süsteemid.....	7
1.2.1 Kaugjuhitavad ja autonoomsed süsteemid.....	7
1.2.2 Kaugjuhtimine ja teleoperatsioon.....	10
1.3 Andmeedastus kaugjuhtimises.....	12
1.3.1 Wi-Fi-põhise ühenduse eripärad kaugjuhtimises.....	12
1.3.2 Latentsus kaugjuhtimises.....	13
1.3.3 Andmeedastusprotokollid kaugjuhtimises.....	14
1.3.4 Videoedastuse tehnoloogiad.....	15
1.4 Tehisintellekti roll teleoperatsioonis.....	17
1.4.1 Tehisintellekt ja teleoperatsioon.....	17
1.4.2 Poolautonoomne juhtimine ja takistustest hoidumine.....	18
2. Meetodid.....	20
2.1 Süsteemi ülesehitus ja arhitektuur.....	20
2.2 Kasutajaliides.....	21
2.3 Sõidukipoolne rakendus.....	22
2.4 Autonoomse režiimi otsustusloogika.....	24
2.5 Testraja kirjeldus.....	26
3. Tulemused.....	27
3.1 Kasutajaliides.....	27
3.2 Süsteemi jõudlus.....	29
3.2.1 Kaadrisagedus.....	29
3.2.2 Süsteemiressursid.....	29
3.3 Testringi tulemused.....	30
3.3.1 Pääsupunkti vahetuste sagedus.....	30
3.3.2 Videopildi katkestused.....	31
3.3.3 Režiimivahetuste töökindlus ja õiguspärasus.....	33
4. Tulevikusuunad ja rakendusvaldkonnad.....	35
4.1 Edasised arengusuunad.....	35
4.2 Võimalikud rakendused.....	36

Kokkuvõte.....	37
Viited.....	38
Lisad.....	42
I. Väiksemõõtmeline kaugjuhitav sõiduk.....	42
II. Testraja asukoht Delta õppehoone kolmandal korrusel	43
III. Kasutajaliidese paigaldamise juhend	44
IV. Sõidukipoolse rakenduse paigaldamise juhend	45
V. Lahenduse demontsratsioon katseringil	46
VI. Litsents.....	47

Sissejuhatus

Autonoomsete sõidukite arendus on kiiresti arenev valdkond, kuid täismõõdus isejuhtivate sõidukite testimine on kulukas ja potentsiaalselt ohtlik. Seetõttu kasutatakse üha enam väiksemõõtmelisi autosid uurimisplatvormidena, mis pakuvad soodsamat ja turvalisemat viisi uute lahenduste katsetamiseks [1]. Uuringud on kinnitanud, et sellised miniatuursed platvormid võimaldavad ohutult imiteerida erinevaid liiklussituatsioone kontrollitud keskkonnas, ilma et seaks ohtu inimesi või kallist varustust [1].

Samal ajal on autonoomsete sõidukite valdkonnas kasvanud huvi kaugjuhtimise vastu, mis lubavad operaatoril juhtida platvorme pika vahemaa tagant ning toetavad järk-järgulist üleminekut täisautonoomsetele lahendustele [2]. Käesoleva töö ajendiks on nende kahe suundumuse, miniatuursete uurimisplatvormide ja kaugjuhtimise, ühendamise vajadus.

Töö keskne eesmärk on arendada väiksemõõtmelise mänguauto kaugjuhtimise süsteem, mis võimaldab Tartu Ülikooli Delta õppehoone avalikus Wi-Fi-võrgus platvormi turvaliselt distantsilt juhtida. Süsteem on kavandatud töötama reaajas ning tagama juhtimiskäsu ja videopildi stabiilse edastuse läbi muutuvate võrgutingimuste. Olulise lisavõimekusena rakendub katkestuste korral autonoomne juhtimisrežiim, mis võimaldab sõidukil jätkata liikumist ka siis, kui visuaalne või juhtimisside ajutiselt katkeb. Tegemist on labori jaoks uudse rakendusega, mis ühendab teleoperatsiooni ja autonoomia ning loob eeldused kaugjuhitavate süsteemide testimiseks realistlikes keskkonnatingimustes.

Bakalaureusetöö on koosneb neljast peatükist. Esimeses peatükis antakse ülevaade autonoomsete ja kaugjuhitavate sõidukite põhimõtetest, varasematest uurimustest ning kasutatavatest tehnoloogiatest. Teises peatükis kirjeldatakse valminud lahenduse arhitektuuri, sealhulgas kasutatud riist- ja tarkvaralahendusi. Kolmandas peatükis esitatakse süsteemi testimise tulemused ning hinnatakse selle toimivust seatud eesmärkide valguses. Neljandas peatükis tehakse kokkuvõtte töö olulisematest tulemustest ning pakutakse välja edasised arengusuunad.

Käesoleva töö keelelise toimetamise ja sõnastuse täpsustamise käigus kasutati abivahendina generatiivsel tehisintellektil põhinevat juturobotit (ChatGPT), mille abil lihtsustati keerukamaid lauseid ning parandati teksti üldist loetavust. Kõik tehisaru soovitused vaadati autori poolt kriitiliselt üle ning lõplikud muudatused tehti käsitsi.

1. Teoreetiline taust

Selles peatükis antakse ülevaade kaugjuhitavate ja autonoomsete süsteemide teoreetilisest taustast. Alustuseks selgitatakse erinevaid autonoomia tasemeid ning võrreldakse kaugjuhtimise, teleoperatsiooni ja autonoomse juhtimise põhimõtteid. Seejärel keskendutakse andmeside tehnilisele korraldusele: käsitletakse Wi-Fi kasutamise eripärasid, latentsuse mõju juhtimistäpsusele ning erinevaid andmeedastusprotokolle. Edasi tutvustatakse reaajas videopildi edastamise tehnoloogiaid ja nende rolli visuaalse tagasiside pakkumisel operaatorile. Peatüki lõpus vaadeldakse tehisintellekti rakendusi teleoperatsioonis, sealhulgas otsustusabi ja poolautonoomset juhtimist toetavaid lahendusi.

1.1 Platvormi valik ja riistvaraline ülesehitus

Käesolevas töös uuritakse väiksemõõtmelise kaugjuhitava sõiduki juhtimissüsteemi töökindlust ja reageerimisvõimet muutuvates võrgutingimustes. Selleks oli vaja kasutada platvormi, mis võimaldaks turvaliselt katsetada nii juhtimiskäskude ja videopildi edastust kui ka automaatseid režiimivahetusi sidekatkestuste korral. Valitud lahendus pidi toetama käsitsi juhtimist, kuid võimaldama sujuvalt lülituda autonoomsele juhtimisele olukordades, kus ühendus ajutiselt katkeb.

Valitud platvormiks on DonkeyCar, avatud lähtekoodiga raamistik, mida kasutatakse autonoomse sõidu ja kaugjuhtimislahenduste arendamiseks väiksemõõtmelistel sõidukitel [3]. DonkeyCar-i platvorm on kirjutatud Pythoni programmeerimiskeeles ning selle eesmärgiks on pakkuda haridusasutustele, entusiastidele ja teadlastele ligipääs autonoomse sõidu tehnoloogiate uurimiseks ja rakendamiseks. Platvormi iseloomustab kasutajasõbralik arhitektuur, modulaarsus ning aktiivne kogukonna tugi.

DonkeyCar on osutunud sobivaks platvormiks autonoomse sõidu ja kaugjuhtimise uurimiseks väiksemõõtmeliste mänguautodega. Petrovići magistritöö [1] tulemused kinnitasid, et platvorm võimaldab testida autonoomse sõidu erinevaid komponente sarnasel viisil täismõõtmeliste süsteemidega. Töö näitas, et lokaliseerimise ja kaardistamise osas saavutab DonkeyCar tulemusi, mis on võrreldavad täismõõtmeliste sõidukitega. Petrovići töö rõhutas ka platvormi sobivust kaugjuhtimissüsteemide ja latentsuse uurimiseks, mis on käesoleva töö keskmes.

Käesolevas töös arendatakse kaugjuhtimislahendust väikesemõõtmelisele autole, kasutades DonkeyCar platvormi. Kasutatav auto on 1:10 skaalas, millele on lisatud järgmised komponendid (Lisa I):

- Raspberry Pi 4 Model B¹ – arvuti, millel on neljatuumaline ARM Cortex-A72 protsessor sagedusega 1,5 GHz ning 2 GB muutmälu. Raspberry Pi töötab keskse arvutina, millel jookseb DonkeyCar raamistik ja server.
- Raspberry Pi Camera Module v2² – 8-megapiksline kaamera, mis võimaldab jäädvustada kuni 1080p resolutsiooniga videot sagedusel kuni 30 kaadrit/s.
- PIS-1685 lainurkobjektiiv – plug-and-play 160° lainurkobjektiiv Raspberry Pi Camera Module V2-le, mis võimaldab jäädvustada 3280 × 2464 resolutsiooniga fotosid ja 1080p@30 H.264-videoid.
- RoboHAT MM1 Robotics Controller Board³ – PWM-signaale genereeriv juhtmoodul, mis võimaldab reguleerida auto mootori kiirust ja roolisüsteemi tööd.
- Servo E6001 – servomootor, mis pöörab roolivarda täpselt soovitud nurga alla (0–180°), võimaldades autol sujuvalt ja täpselt keerata.

Selline riistvaraline kooslus valiti eesmärgiga saavutada optimaalne tasakaal sõiduki kompaktsuse ja süsteemi arvutusliku võimekuse vahel. Käesolev konfiguratsioon võimaldab uurida, kuidas Raspberry Pi toime tuleb reaajas juhtimise ja video voogedastusega, ning millised väljakutsed tekivad Wi-Fi-põhise kaugjuhtimisega avalikes ja mitme pääsupunktiga võrkudes.

1.2 Autonoomsed ja kaugjuhitavad süsteemid

Enne kaugjuhtimislahenduste tehniliste detailide käsitlemist on oluline mõista, millised on erinevused kaugjuhitavate, poolautonoomsete ja autonoomsete süsteemide vahel. See alapeatükk selgitab erinevate juhtimismudelite põhimõtteid ning autonoomia tasemeid, asetades käesolevas töös arendatava lahenduse laiemasse konteksti.

1.2.1 Kaugjuhitavad ja autonoomsed süsteemid

Sõidukeid ja muid mehitamata süsteeme saab klassifitseerida nende autonoomsuse taseme järgi – alates täielikult inimese juhitavatest kuni täiesti iseseisvate, autonoomsete süsteemideni.

¹ <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf>

² <https://www.raspberrypi.com/documentation/accessories/camera.html>

³ <https://robohatmm1-docs.readthedocs.io/en/latest/>

Ameerika Ühendriikide Riiklik Standardi- ja Tehnoloogiainstituut on kirjeldanud oma dokumendis [4] kolme terminit: autonoomne sõiduk, poolautonoomne sõiduk ning kaugjuhitav sõiduk. Dokumendis on selgitatud, et autonoomne sõiduk on sõiduk, mis suudab iseseisvalt sõita ja teha otsuseid ilma inimese otsese sekkumiseta kasutades andureid ümbritseva keskkonna tajumiseks ning tarkvara juhtimisülesannete täitmiseks. Poolautonoomne sõiduk (ehk osalise autonoomiaga) aga viitab süsteemile, kus osa sõiduki juhtimisest on automatiseeritud, kuid inimene peab sõidukit jälgima ja teatud tingimustel sekkuma. Kaugjuhitav sõiduk on mehitamata vahend, mida inimene juhib eemalt reaalajas; see tähendab, et juhil puudub silmside sõidukiga ja juhtimisotsused tehakse distantsilt.

Peamine erinevus seisneb otsuste tegijas: autonoomne süsteem teeb need iseseisvalt, poolautonoomne jagab otsustus- ja juhtimisülesandeid inimese ja masina vahel ning kaugjuhitav sõiduk sõltub distantsilt operaatori otsustest. Karaböcek jt [2] toovad esile, et autonoomsed lahendused ühendavad andurid ja tehisintellekti, et ümbritsevat keskkonda tajuda ja navigeerida. Kaugjuhitavad süsteemid eeldavad usaldusväärset sidekanalit ja pidevat tagasiside operaatorilt, kasutades sensoreid reaalajas visuaalse ja muu tagasiside edastamiseks. Poolautonoomsed lahendused asuvad täisautonoomsuse ja kaugjuhtimise vahel kasutades autonoomsust näiteks roolimise või kiiruse kontrollimisel, kuid nõudes, et juht oleks valmis kontrolli üle võtma olukordades, kus autonoomne süsteem ületab talle seatud kriteeriumid [5].

Rahvusvaheliselt kasutatakse sõidukite automatiseerituse kirjeldamiseks SAE International-i standardit J3016 [6, 7]. SAE J3016 standard eristab kuut taset (joonis 1), nullist (puudub automaatika) kuni viieni (täielik autonoomia), kirjeldades iga taseme puhul, milliseid sõiduki juhtimise ülesandeid teostab masin ja milliseid inimene. Tasemete lühikirjeldused on järgmised:

1. Tase 0 (ingl k *No Driving Automation*) – sõidukis puudub automaatjuhtimine; inimene teeb kõik juhtimisotsused ja -toimingud.
2. Tase 1 (ingl k *Driver Assistance*) – sõidukil on juhi abistamiseks mõni süsteem, mis suudab kontrollida ühte juhtimise aspekti korraga (kas roolimist või kiirendust/pidurdust) teatud tingimustel.
3. Tase 2 (ingl k *Partial Driving Automation*) – sõiduk suudab juhtida nii rooli kui ka kiirendust/pidurdamist, kuid kogu vastutus, sealhulgas ümbritseva liikluse jälgimine ja sekkumise otsustamine, jääb juhile. Süsteem ei saa aru, kui ta on olukorras, kus ta ei oska sõita.

4. Tase 3 (ingl k *Conditional Driving Automation*) – sõiduk suudab teatud tingimustes ise sõita ja ümbritsevat jälgida. Kui olukord muutub liiga keeruliseks, peab juht olema valmis juhtimise üle võtma – süsteem annab sellest ise märku.
5. Tase 4 (ingl k *High Driving Automation*) – sõiduk on võimeline iseseisvalt sõitma kindlas piirkonnas ja tingimustel (näiteks kindlas linnas või kindla ilma korral). Kui tingimused ei sobi, ei alusta süsteem sõitu või peatub ohutult – inimese sekkumine ei ole eeldatud.
6. Tase 5 (ingl k *Full Driving Automation*) – täielik autonoomia: sõiduk suudab ise toime tulla kõikides tingimustes ja olukordades, kus ka inimene sõidaks – sealhulgas keeruline linnaliiklus, halb ilm, tundmatu maastik.



SAE J3016™ LEVELS OF DRIVING AUTOMATION™

Learn more here: sae.org/standards/content/j3016_202104

Copyright © 2021 SAE International. The summary table may be freely copied and distributed AS-IS provided that SAE International is acknowledged as the source of the content.

	SAE LEVEL 0™	SAE LEVEL 1™	SAE LEVEL 2™	SAE LEVEL 3™	SAE LEVEL 4™	SAE LEVEL 5™
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in “the driver’s seat”		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	

Copyright © 2021 SAE International.

	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features	<ul style="list-style-type: none"> • automatic emergency braking • blind spot warning • lane departure warning 	<ul style="list-style-type: none"> • lane centering OR • adaptive cruise control 	<ul style="list-style-type: none"> • lane centering AND • adaptive cruise control at the same time 	<ul style="list-style-type: none"> • traffic jam chauffeur 	<ul style="list-style-type: none"> • local driverless taxi • pedals/steering wheel may or may not be installed 	<ul style="list-style-type: none"> • same as level 4, but feature can drive everywhere in all conditions

Joonis 1. SAE J3016 standardi järgi määratletud sõidukite autonoomsuse tasemed [7].

2025. aastal ei ole olemas sõidukeid, mis vastaksid 5. taseme autonoomiale. Kõige arenenumad isesõitvad autod on parimal juhul 4. taseme juures ning peavad keerulistes olukordades turvaliselt peatuma [8].

1.2.2 Kaugjuhtimine ja teleoperatsioon

Teleoperatsioon mängib autonoomsete sõidukite arenduses olulist toetavat rolli. Täielikult usaldusväärse isejuhtiva tehisintellekti arendamine on osutunud tehniliselt keerukaks. Kuna autonoomsed sõidukid ei suuda sageli toime tulla äärmuslike või ootamatute olukordadega, kasutatakse kaugjuhtimist kui tagavarasüsteemi nende lünkade täitmiseks [9].

Mõisted kaugjuhtimine ja teleoperatsioon on sisult sarnased – mõlemad viitavad juhi ja masina vahelisele interaktsioonile distantsilt [4]. Huang kirjeldab kaugjuhtimist kui olukorda, kus operaator juhib sõidukit distantsilt, nähes seda, kuid ilma otsese sensor- või videopõhise tagasisideta. Teleoperatsiooni puhul saab aga süsteemi juhtiv kasutaja otsest tagasisidet videopildi või sensori andmete näol, mis võimaldab teha täpsemaid ja kontekstitundlikumaid juhtimisotsuseid. Käesolevas töös kasutatakse mõistet „kaugjuhtimine“ just viimases, Huang'i poolt teleoperatsioonina kirjeldatud tähenduses.

Lu jt [10] on jaganud kaugjuhtimise kolmeks erinevaks juhtimistasemeks, mille rakendamine sõltub konkreetsetest vajadustest ja olukordadest:

- Kaugseire (ingl k *remote monitoring*) – operaator jälgib sõiduki liikumist reaalajas ilma otsese sekkumiseta, võimaldades analüüsida autonoomsete süsteemide toimimist ning koguda andmeid autonoomsuse parandamiseks.
- Kaugabi (ingl k *remote assistance*) – operaator sekkub vaid keerulistes olukordades, andes üldiseid juhiseid (näiteks marsruudi või trajektoori valik), samal ajal kui sõiduk täidab detailseid juhtimisülesandeid autonoomselt.
- Otsene kaugjuhtimine (ingl k *remote control*) – operaator võtab sõiduki juhtimise täielikult üle, kontrollides reaalajas rooli, pidureid ja kiirendust olukordades, kus autonoomne süsteem ei tule toime või olukord liiga keerukas.

Kõigi eespool kirjeldatud tasemete edukaks rakendamiseks tuleb tagada vajalikud tehnoloogilised eeldused. Elinav selgitab [11], et teleoperatsiooni toimimine sõltub erinevate tehnoloogiate ja süsteemide integratsioonist. Oluliseks tehniliseks eelduseks on usaldusväärne ja minimaalse viiteajaga andmeside, mis võimaldab reaalajas edastada kõrglahutusega videopilti ja andurite andmeid – näiteks lidari⁴ punkt pilvi, radari objektituvastusi ning objektide kaugushinnanguid [10]. Lu rõhutab, et kogutud andmed tuleb teleoperaatorile esitada

⁴ <https://et.wikipedia.org/wiki/Lidar>

selgelt ja intuiitiivselt, mis eeldab hästi läbimõeldud kasutajaliideseid ja juhtimist toetavat tarkvara.

Lu jt [10] jagavad oma artiklis teleoperatsiooni uurimissuunad nelja suuremasse valdkonda: operaatori situatsiooniteadlikkuse arendamine, kasutajaliideste arendus, juhtimissüsteemide optimeerimine ning sideühenduse latentsuse vähendamine ja kompenseerimine. Neis valdkondades toimub aktiivne uurimistöö – rakendatakse nii intelligentseid nõustussüsteeme kui ka servaarvutust⁵ (ingl k *edge computing*), mis vähendavad operaatori töökoormust ja võrgu latentsust. Samuti rõhutatakse vajadust kaugjuhtimise autonoomsuse tõstmiseks, et sõiduk saaks ilma operaatori pideva sekkumiseta teatud olukorrad iseseisvalt lahendada [10]. Nende arendustega taotletakse kaugjuhtimise efektiivsuse ja ohutuse tõstmist – vähendades operaatori sekkumiste arvu ning võimaldades tal keskenduda pigem strateegilistele ja laiahaardelistele otsustele.

Kaugjuhtimise arenduse üheks keskseks väljakutseks on tagada, et üks operaator suudaks samaaegselt turvaliselt hallata mitut autonoomset sõidukit. Selle lähenemise majanduslikku potentsiaali illustreerib Goodalli [12] analüüs. Kogu Ameerika Ühendriikide masinapark võiks tema hinnangul olla hallatav 4000 – 40 000 teleoperaatori poolt, kui kõik sõidukid oleksid täielikult autonoomsed ja vajaksid inimsekkumist harva – näiteks nagu Waymo sõidukid 2018. aastal. See tähendaks üle 99% tööjõukulude vähenemist võrreldes 2018. aastaga, kus Ühendriikides töötas ligi 4,4 miljonit elukutselist juhti. Sama meetodika järgi saab Uberi globaalse autopargi, mille jaoks töötas aastal 2018 ligi 3,9 miljonit juhti, teoreetiliselt asendada vähem kui 400 teleoperaatoriga [12]. See rõhutab kaugjuhtimise võimet ümber kujundada kogu transpordivaldkonna töökorralduse eeldusel, et autonoomsete süsteemide töökindlus on tagatud ja inimsekkumiste arv jääb minimaalseks.

Mitme autonoomse sõiduki kaugjuhtimine eeldab süsteemi suutlikkust samaaegselt reageerida mitme sõiduki sekkumisvajadusele. Grabbe jt [13] rõhutavad, et sellistes olukordades tuleb:

- prioriseerida ülesandeid, et kriitilisemad juhtumid saaksid viivitamata tähelepanu;
- määratleda selged rollid operaatorite vahel, vältimaks vastutuse kattumist;
- rakendada dünaamilisi töökorraldust, näiteks hoida üks operaator „valvel“, et tippaegadel kiiresti abistada.

⁵ <https://et.wikipedia.org/wiki/Servaarvutused>

Ilma nende mehhanismideta kuhjuvad sekkumisvajadused, tekivad viivitused ning langevad nii süsteemi usaldusväärsus kui ka liiklusohutus. Tõhus töökorraldus vähendab sõidukite seisakuid ja hoiab operaatori koormuse optimaalsel tasemel.

Teleoperatsioon on mitmekülgne ja kiiresti arenev valdkond, kus arendatakse sensoreid ja visualiseerimisvahendeid, uuritakse erinevaid juhtimisstrateegiaid ning otsitakse lahendusi samaaegselt mitme sõiduki haldamiseks. Edu teleoperatsioonis sõltub eelkõige usaldusväärsest, madala latentsusega sideühendusest, ergonoomiliselt disainitud kasutajaliidestest ning tehisintellekti ja automatiseerimise tasakaalustatud integreerimisest, mis vähendab operaatori koormust, tõstab ohutust ja parandab tööjõu efektiivsus.

1.3 Andmeedastus kaugjuhtimises

Käesolevas peatükis käsitletakse nelja võtmeaspekti: Wi-Fi kasutamise eripärasid, latentsuse mõju juhtimise täpsusele, erinevaid suhtlusprotokolle ning reaalajas videopildi edastamise tehnoloogiaid. Need teemad aitavad mõista, kuidas andmeside mõjutab süsteemi reageerimisvõimet ja kaugjuhi töökindlust, ning milliseid tehnilisi valikuid tuleb teha, et tagada sujuv ja ohutu teleoperatsioon.

1.3.1 Wi-Fi-põhise ühenduse eripärad kaugjuhtimises

Wi-Fi⁶ on laialt levinud lühimaa traadita võrgutehnoloogia, mida kasutatakse ka teleoperatsiooni rakendustes [14]. Kamtam tõi oma artiklis [15] välja, et Wi-Fi-põhine ühendus on eelistatud paljudes olukordades, kus kommunikatsioon süsteemide vahel toimub lühikesel distantil, kontrollitud keskkonnas (näiteks lennujaamad, kaubanduskeskused). Artiklis toodi välja, et Wi-Fi on laialdaselt kättesaadav ja soodne ning pakub väikestel vahemaadel suurt andmeedastuskiirust. Näiteks on siseruumides või piiratud territooriumil lihtsam töötavaid kaugjuhitavaid süsteeme ühendada olemasoleva Wi-Fi võrguga kui tugineda satelliitühendusele.

Wi-Fi-põhisel teleoperatsioonil esineb mitmeid väljakutseid, millest üks on leviala piiratus [16]. Tüüpilistes siseruumides jääb Wi-Fi leviulatus mõnekümne meetri piiridesse, mistõttu laiematel või avatud aladel võib ühendus kiiresti katkeda, viies kaugjuhitava süsteemi võrgu levipiirkonnast välja ning põhjustades juhtimise katkemise [17]. Selliste katkestuste vältimiseks paigaldatakse mitmeid pääsupunkte⁷ (AP), et katta kogu ala ühtlase signaaliga.

⁶ <https://en.wikipedia.org/wiki/Wi-Fi>

⁷ https://en.wikipedia.org/wiki/Wireless_access_point

Kuid olukorras, kus kaugjuhitav sõiduk liigub ühe AP levialast teise omasse, peab sõiduk AP sujuvalt ümber lülitama — seda protsessi nimetatakse AP-vahetuseks (ingl k *handover*) [18]. Sagedased pääsupunktide vahetused ning koormatud võrk võivad tekitada häireid ja katkestusi, mõjutades nii ühenduse kvaliteeti kui ka süsteemi üldist jõudlust [19].

Wi-Fi pääsupunkti (AP) vahetused võivad põhjustada lühiajalisi katkestusi või viivitusi juhtimissignaali edastuses, mis mõjutavad otseselt kaugjuhtimise töökindlust. Käesolevas töös rakendatakse selle probleemi leevendamiseks autonoomset juhtimisrežiimi: kui süsteem tuvastab ühenduse katkemise või viivituse AP vahetuse ajal, lülitub sõiduk automaatselt eelnevalt treenitud mudelil põhinevale autonoomsele juhtimisele [20]. Sellisel juhul jätkab masin liikumist turvalise, konstantse eelseadistatud kiirusega ja navigeerib vastavalt õpitud mustrile.

1.3.2 Latentsus kaugjuhtimises

Latentsus⁸ on viivitus, mis kulub andmete (näiteks käskluse või videokaadri) edastamiseks ühest süsteemi osast teise. Kaugjuhtimises hõlmab latentsus nii signaali liikumist võrgus kui ka signaalitöötlust seadmetes. Kamtani jt [15] rõhutavad, et latentsuse tekkimist mõjutavad peamiselt füüsilised piirangud (raadiolainete levikiirus ja kaugus), võrguseadmete töötlemis- ja puhverdusajad ning teiste seadmete sisemine viide. Wi-Fi latentsust võivad suurendada ka võrgu koormus ja andmepakettide kadumine, mis tekitavad täiendavat ooteaega. Seetõttu on teleoperatsioonis viide paratamatu nähtus, mille suurus sõltub nii kasutatavatest tehnoloogiatest kui ka võrgutingimustest.

Viivituse esinemine ja ulatus vähendavad kaugjuhtimise täpsust ning suurendavad operaatori töökoormust. Uuringus [15] toodi välja, et inimesed suudavad tajuda juba väga väikest, umbes 10–20 millisekundi (ms) suurust peiteaega. Teadustöös kirjeldati, et viivitused 300–500 millisekundit muudavad teleoperatsiooni tõsiseks väljakutseks - operaatoril tuleb oma käitumist kohandada vastavalt olukorrale. Latentsuse kasv sekundini ja üle selle, muudab kaugjuhtimise reaalselt võimatuks. See tähendab, et operaator ei ole enam võimeline reageerima ning kaotab kontrolli olukorra üle.

Latentsuse negatiivse mõju leevendamiseks on välja töötatud mitmeid meetodeid. Üks potentsiaalne lähenemine on sujuvama võrguühenduse tagamine – see hõlmab kiiremate võrgutehnoloogiate kasutamist, näiteks 5G mobiilsid, ning võrgu optimeerimist, et vähendada

⁸ <https://et.wikipedia.org/wiki/Latentsusaeg>

ja stabiliseerida latentsust [21, 22]. Teine lahendus on latentsuse kompenseerimine. Lu jt [10] pakkusid oma artiklis välja kaks lahendust: *predictive display* ja *parallel virtual vehicle control*. Esimese lahenduse puhul ennustatakse, milline võiks olla ümbritsev keskkond mingis ajahetkes tulevikus, ning kuvatakse see ennustatud pilt kaugjuhile. Teise variandi puhul kuvatakse operaatorile virtuaalne sõiduk, mis reageerib operaatori käskudele viivitusega. Tegelik sõiduk järgib virtuaalset mudelit - nii saab operaator tunnetuslikult kohese tagasiside, kuigi reaalsuses jõuavad käsud sõidukini viivitusega.

1.3.3 Andmeedastusprotokollid kaugjuhtimises

Kaugjuhtimises kasutatav protokoll⁹ määrab, kuidas juhtimiskäsud, andmed ja videopilt liiguvad operaatori ja seadme vahel. Andmeedastusprotokolli valimisel tuleb arvestada kaugjuhtimise spetsiifiliste omadustega - suhtlus peab toimuma reaalajas, minimaalse viitega ning kommunikatsioon peab olema kahepoolne. Operaator saadab süsteemile käsklusi ja süsteem vastab samal ajal muude andmetega (näiteks videovoog). Allpool võrreldakse levinumaid protokolle ja standardeid - WebSocket, HTTP, MQTT ja UDP - ning analüüsitakse, milline lahendus on sobivaim lõputöö praktilisele osale.

Hüpertexti edastusprotokoll¹⁰ (ingl k *Hypertext Transfer Protocol* ehk HTTP) veebisuhtluse protokoll, mis toimib päring-vastus põhimõttel – klient saadab päringu ja server vastab. HTTP protokoll on ühesuunaline, seega kahe-suunaliseks suhtluseks tuleb leida arhitektuuriline lahendus, mis võimaldaks saata tihedalt suurel hulgal päringuid mõlemalt osapoolelt. Kuigi on olemas meetodeid nagu *long polling*¹¹ või *HTTP streaming*¹², siis need lahendused, nagu ka võimalik arhitektuuriline lahendus, toovad juurde keeruksut ja latentsust, mida teleoperatsioonis üritatakse vältida [23].

WebSocket¹³ on andmevahetuseprotokoll, mis võimaldab kahe-suunalist andmevahetust kliendi ja serveri vahel üle [23] edastusohje protokoll¹⁴ (ingl k *Transmission Control Protocol* ehk TCP). WebSocket'i eelis on see, et pärast ühenduse püsti seadmist, lülitub andmevahetus ümber WebSocketi protokollile, mis võimaldab järjepidevat andmevoogu minimaalse lisakoormusega [23]. Srinivasan jt [24] võrdlesid teleoperatsioonis WebSocketi ja HTTP

⁹ [https://et.wikipedia.org/wiki/Protokoll_\(andmeside\)](https://et.wikipedia.org/wiki/Protokoll_(andmeside))

¹⁰ https://et.wikipedia.org/wiki/H%C3%BCpertexti_edastusprotokoll

¹¹ <https://www.pubnub.com/guides/long-polling/>

¹² <https://www.pubnub.com/guides/http-streaming/>

¹³ <https://et.wikipedia.org/wiki/WebSocket>

¹⁴ https://et.wikipedia.org/wiki/Edastusohje_protokoll

protokolli. Nad leidsid, et WebSocket parandas kaadrisagedust ja vähendas latentsust ligikaudu kümnekordselt (57 ms vs 500 ms). Ribalaiuse kasutus alanes peaaegu kolm korda (190 kbps vs 450 kbps) ning operaatorite juhtimiskiirus ja efektiivsus paranesid oluliselt. Seetõttu on WebSocket reaktiivsem kui päringupõhine HTTP ning paljudes veebipõhistes teleoperatsioonirakendustes eelistatud.

MQTT (ingl k *Message Queuing Telemetry Transport*) on lihtsustatud, publitseerijat ja tellijat sisaldava mudeliga protokoll, mida sageli kasutatakse nutistu seadmetes. Sõnumid edastatakse läbi keskses vahendaja, mistõttu saatjal ja vastuvõtjal ei pea olema omavahelist otses ühendust. MQTT eelised on lihtsus, skaleeritavus ning toimimine ebastabiilses internetivõrgus, kuid ei ole toetatud veebibrauserites [25]. Autor toob välja, et teleoperatsioonis, kus kasutatakse üks-ühele ühendust (üks operaator ja üks süsteem), võib MQTT lisada üleliigset keerukust, kui puudub vajadus keerukama suhtluse järele. Lihtsam on kasutada otseühendust WebSocketiga.

Kasutajadatagrammi protokoll (ingl k *User Datagram Protocol* ehk UDP) on transpordikihi protokoll, mis erineb eelpool mainitust sellepolest, et on ebausaldusväärne: pakette saadetakse ilma kättesaamist kinnitamata, suvalises järjekorras ning krüpteerimata. Nende funktsionaalsuste puudumine toob aga esile UDP protokoll eelised: väike latentsus ja vähem lisakoormust võrgule. Artiklis [26] rõhutatakse, et UDP sobib internetipõhiste kaugjuhtimisrakenduste jaoks paremini kui TCP, kuna see pakub väiksemat viivitust ja stabiilsemat andmeedastust. Seega on UDP lihtne ja efektiivne, kuid kasutades tuleb arvestada, et kõik andmed ei pruugi teisele osapoolale kohale jõuda.

Kokkuvõtvalt, HTTP sobib lihtsaks päringupõhiseks suhtluseks, kuid teleoperatsioonis eelistatakse reaajas kahepoolset suhtlust. WebSocket on sobiv üks-ühele andmevahetuseks tänu püsivale ja usaldusväärsele ühendusele. MQTT sobib paremini olukordadesse, kus on rohkem osapooli või ebastabiilne võrk. Kui eesmärgiks on minimaalne latentsus, on kõige tõhusam UDP, mis võimaldab kiiret edastust, ohverdades seejuures andmete terviklikkuse.

1.3.4 Videoedastuse tehnoloogiad

MJPEG¹⁵ (ingl k *Motion JPEG*) on videotihenduse meetod¹⁶, kus iga kaader pakitakse eraldi JPEG-pildina. TechNexioni andmetel on MJPEG-i peamiseks eeliseks lihtne kodeerimine ja dekodeerimine, mis vähendab protsessori koormust ning võimaldab madalat latentsust. Kuna

¹⁵ <https://et.wikipedia.org/wiki/MJPEG>

¹⁶ https://en.wikipedia.org/wiki/Video_coding_format

kaadrid on pakitud üksteisest eraldi, nõuab kodeerimine ja dekodeerimine vähem protsessori tööaega, mis muudab meetodi sobivaks piiratud arvutusvõimsusega süsteemides. Samas toob artikkel välja, et MJPEG kasutab ebaefektiivset tihendusvormingut, mis tähendab, et võrreldes kaasaegsemate vormingutega, näiteks H.264-ga, nõuab MJPEG märkimisväärselt suuremat ribalaiust. TechNexion rõhutab, et sama kvaliteediga video võib MJPEG kujul võtta mitu korda rohkem andmemahutu, mistõttu on see vähem sobiv madala ribalaiusega võrkudes. Seetõttu soovitatakse MJPEG-i kasutada eelkõige lokaalsetes süsteemides, kus andmeside maht ei ole piirav tegur.

H.264 (AVC) ja H.265 (HEVC) on videotihenduse meetod, mis on loodud pakkuma kvaliteetset ja efektiivset videoedastust, sobides piiratud ressursidega süsteemidesse. H.264 suudab saavutada hea pildikvaliteedi madalama bitikiiruse juures võrreldes MJPEG-ga, kuna salvestab efektiivselt järjestikuste kaadrite vahelisi erinevusi ja väldib üleliigsete andmete dubleerimist kaadrist kaadrisse [27]. H.265 on videotihenduse meetod sarnaselt H.264-le, pakkudes samaväärse kvaliteedi juures kuni 50% paremat tihendust võrreldes H.264-ga [28]. See tähendab, et piiratud ribalaiusega sidekanalis, näiteks mobiilivõrk, on H.265 suuteline edastama kõrge resolutsiooniga videot seal, kus H.264 või MJPEG seda ei suudaks. Mõlemad meetodid, eriti aga H.265, nõuavad oma algoritmide keerukuse tõttu rohkem riistvaralist ressursi, mis tähendab vajadust võimsama protsessori või riistvaralise kiirenduse järele.

WebRTC¹⁷ (ingl k *Web Real-Time Communication*) on avatud lähtekoodiga tehnoloogia, mis võimaldab reaajas heli- ja videoedastust minimaalse viitega. See on loodud otspunktidevaheliseks¹⁸ (P2P) suhtluseks, kasutades UDP-põhist andmeedastust ning sisaldades mehhanisme nagu STUN/TURN, et tõhusalt lahendada NAT¹⁹ läbivusega seotud probleeme [29]. Gregersen tõi välja, et tehnoloogia tugevusteks on seadmetevaheline otseühendus, sisseehitatud turvalisus ning laialdane tugi brauserites ja erinevatel platvormidel. Samas kirjeldati artiklis, et WebRTC rakendamine võib olla keeruline, sest P2P ühenduse loomine nõuab signaleerimist ning vanemates süsteemides või piiratud riistvaraga seadmetes võib esineda ühilduvusprobleeme.

Teleoperatsioonis videovoo tehnoloogia valikul tuleb lähtuda konkreetse rakenduse vajadustest, otspunktide arvutusvõimsusest ja võrgu piirangutest. Kui lõpp-punktidel on

¹⁷ <https://en.wikipedia.org/wiki/WebRTC>

¹⁸ <https://et.wikipedia.org/wiki/P2P-v%C3%B5rgustik>

¹⁹ https://et.wikipedia.org/wiki/V%C3%B5rguaadresside_teisendus

piiratud arvutusvõimsus, kuid võrgu ribalaius on suur, on mõistlik eelistada lihtsamat kodeeringut (näiteks MJPEG), sest kooderinguaeg jääb madalale ja see kompenseerib suuremat andmemahutu. Vastupidisel juhul, kui arvutusvõimsus on kõrge, kuid ribalaius piiratud, on otstarbekas kasutada efektiivsemaid vorminguid nagu H.264 või H.265, kuna need pakivad video väiksema bitikiirusega. Kui nii arvutusvõimsus kui ribalaius on piiratud, tuleb läbi viia konkreetseid katseid ja analüüse, et määrata, kas peamine pudelikael on kodeerimisel või võrguedastuses.

1.4 Tehisintellekti roll teleoperatsioonis

Tehisintellekti (AI) kasutamine teleoperatsioonis võimaldab parandada kaugjuhitavate süsteemide täpsust, töökindlust ja ohutust. AI-põhised algoritmid suudavad töödelda sensorite andmeid, pakkuda abi otsuste tegemisel ja kohandada reaajas juhtimisparameetreid. Need võimed vähendavad operaatori töökoormust ja suurendavad süsteemi autonoomsust, säilitades samas inimese lõpliku kontrolli.

1.4.1 Tehisintellekt ja teleoperatsioon

Teleoperatsiooni üks peamisi väljakutseid on piiratud taju – operaator sõltub kaamerapildist ja anduritest, mis ei pruugi anda piisavat ruumilist tunnetust. Masinnägemise ja sügavõppe meetodid võimaldavad süsteemidel reaajas analüüsida videopilti, tuvastada objekte ning tõlgendada keskkonda, parandades nii ohutust kui ka täpsust.

Alljärgnev loetelu toob välja peamised tehisintellekti rakendusvaldkonnad teleoperatsioonis:

1. Keskkonna tajumine objektituvastuse kaudu

Sügavõppemudelid nagu YOLOv3 ja Mask R-CNN tuvastavad reaajas videopildilt nii kirurgilised instrumendid kui ka patsiendi anatoomiat, andes kirurgile täpse visuaalse tagasiside instrumentide asendi ja liikumise kohta [30]. See vähendab operaatori kognitiivset koormust ja parandab operatsiooni täpsust.

2. Juhtimisparameetrite automaatne kohandamine

AI-põhised algoritmid optimeerivad juhtimisabi parameetreid reaajas, võttes arvesse nii võrgukatkestuste tõttu tekkivaid viiteaegu kui ka operaatori otsustusprotsessi dünaamikat. Corredor jt [31] kasutavad selles kontekstis Drift-Diffusion Model'i (DDM), mis simuleerib operaatori otsustusmudelit ja hindab tema reaktsiooniaegu. Mudeli alusel

kohandatakse abi taset reaajas varasemat sooritust arvestades. Samuti prognoositakse operaatori reageerimist, et tagada sujuvam kaugjuhtimiskogemus.

3. Situatsioonituvastus ja automatiseeritud ülesannete täitmine

Masinõppemudelid tuvastavad korduvaid töövoos etappe (nt koe lõikamine, õmblemine) ning täidavad need iseseisvalt, vähendades operaatori töömahtu ja parandades protseduuri kiirust ning täpsust [32]. Selline poolautonoomne lähenemine võimaldab inimesel keskenduda keerukamatele otsustele.

4. Poolautonoomne juhtimine

Inimene ja masin teevad koostööd: AI haldab rutiinseid või täpsust nõudvaid manöövreid, operaator jälgib ja sekkub vajadusel. Li jt [30] kirjeldavad CNN-põhist algoritmi, mis kaardistab otse sensoripildid juhtkäskudeks ilma vahepealsete töötlemisetappideta. See lähenemine ühendab inimese kohanemisvõime ja AI kiiruse, pakkudes stabiilset ning turvalist teleoperatsiooni.

Kõik neli valdkonda toetavad omavahel: parem objektituvastus loob aluse adaptatiivsele juhtimisele, mis omakorda võimaldab sujuvat situatsioonituvastust ning poolautonoomset režiimi. Eriti oluline on poolautonoomne juhtimine, sest see kombineerib inimese otsustusvõime ning AI automatiseerimise eelised.

1.4.2 Poolautonoomne juhtimine ja takistustest hoidumine

Poolautonoomsus ehk jagatud juhtimine tähendab, et kontroll on jaotatud inimese ja masina vahel. Selliste süsteemide arendamisel on keskseks küsimuseks tasakaalu leidmine autonoomse süsteemi sekkumise ja inimese otsustusvabaduse vahel – süsteem peab tagama ohutuse ja ülesande edukuse, kuid samal ajal säilitama operaatorile tunnetuse, et protsessi juhtimine on tema enda käes [31].

Üks levinud meetod sellise tasakaalu saavutamiseks on intelligentne trajektoori parandus, kus masinõppe ja juhtimisalgoritmide abil korrigeeritakse inimese antud käsklusi ainult siis, kui need võivad viia kokkupõrke või ohtliku olukorrani. See lähenemine põhineb põhimõttel “tee, mida ma mõtlen, mitte seda, mida ma teen” (*do what I intend, not what I do*), mis rõhutab, et süsteem ei tohiks täita käsked sõna-sõnalt, vaid peaks püüdma mõista nende taga olevat kavatsust [33]. Näiteks kui operaator suunab roboti otse takistuse poole, kuid tema tegelik eesmärk on jõuda sihtpunkti, peaks intelligentne süsteem korrigeerima trajektoori nii, et see

oleks ohutu, kuid kooskõlas inimese kavatsusega. Bohren jt [33] toovad välja, et just selline lähenemine on kriitilise tähtsusega keskkondades, kus eksimused võivad põhjustada tõsiseid tagajärgi. See aitab arendada usaldusväärsemaid ja operaatorile intuitiivsemaid poolautonoomseid süsteeme.

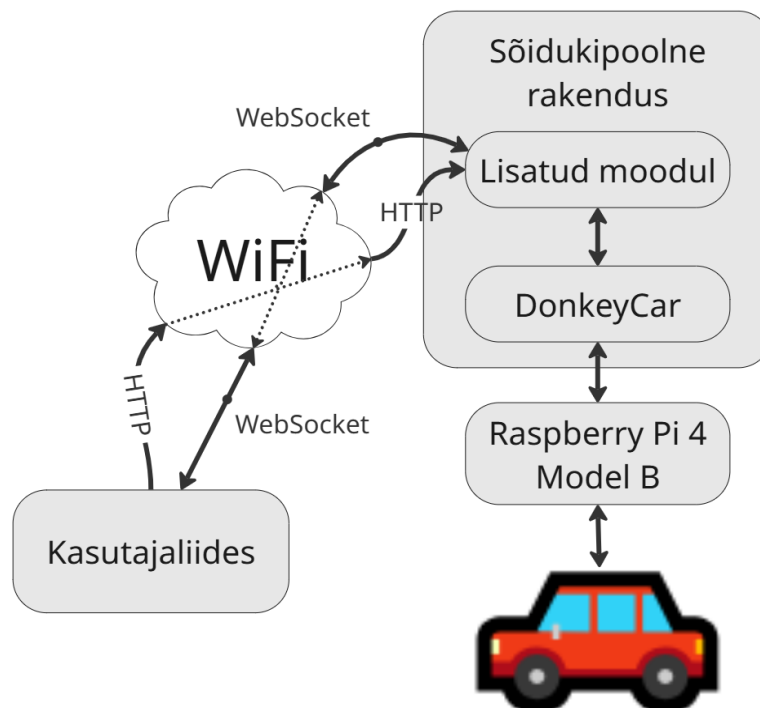
Takistuste vältimine on teleoperatsioonis hädavajalik, et tagada sujuv ja ohutu liikumine, eriti keerulistes või muutuvates keskkondades. Masinõppe abil saab süsteem reaajas analüüsida andurite (näiteks kaamerate või lidarite) andmeid, et tuvastada takistusi, korrigeerida trajektoori või hoiatada operaatorit. Rubagotti jt [34] tutvustasid poolautonoomset teleoperatsiooni raamistikku, kus UR5 robot kasutas MPC algoritmi, et järgida inimese käe liikumist, samal ajal takistusi vältides. Süsteem kasutas referentssignaale inimese liikumise kohta ning planeeris nende põhjal roboti trajektoori reaajas, ennetades võimalikke kokkupõrkeid. Katsed näitasid, et algoritm suutis uue liikumisteede arvutada kiiremini, kui teleoperatsiooni juhtimistsükkel nõudis, mis kinnitab, et masinõppe ja optimeeriva juhtimise kombinatsioon võimaldab ohutut ja viiteaja piires toimivat takistuste vältimist teleoperatsioonis.

2. Meetodid

Selles peatükis kirjeldatakse praktilise osa arendatud süsteemi arhitektuuri: veebipõhist kasutajaliidest, Raspberry Pi-1 töötavat sõidukipoolset rakendust ning autonoomse režiimi otsustusloogikat. Lisaks tutvustatakse katsesõitude testringi ja hindamismetoodikat, mille abil süsteemi toimivust valideeriti. Iga alapeatükk käsitleb süsteemi rakenduslikke aspekte ning põhjendab tehtud valikuid, tuginedes nii praktilistele nõuetele kui ka asjakohastele teaduskirjandusallikatele.

2.1 Süsteemi ülesehitus ja arhitektuur

Süsteemi arhitektuur on üles ehitatud kolmest omavahel seotud kihist (joonis 2). Esimese kihina toimib veebipõhine kasutajaliides, mis käideldakse veebibrauseris. See võimaldab operaatoril jälgida reaajas videopilti ning annab samal ajal võimaluse edastada juhtimiskäsklusi ja hallata süsteemi tööd. Teiseks kihiks on sõidukipoolne rakendus, mis jookseb Raspberry Pi 4 platvormil DonkeyCar raamistikku ja lisatud mooduli toel ning tõlgendab juhtkäsklusi ja juhib sõiduki riistvara. Kolmas kiht on Wi-Fi, mis võimaldab madala latentsusega kahepoolset andmeedastust: brauserist pärinevad juhtkäsklused jõuavad sõidukini ja sealt edastatakse kaamerapilti ning telemeetria tagasi kasutajaliidesesse.



Joonis 2. Süsteemi üldine arhitektuur.

Kasutajaliidese suhtlus sõidukipoolse rakendusega toimub peamiselt WebSocket-protokolli kaudu, mis võimaldab kahesuunalist reaalaja andmevahetust minimaalse latentsusega. Teatud juhtudel, näiteks konfiguratsiooni või salvestuse sisse- ja väljalülitamisel, kasutatakse ka HTTP-päringuid, kuna nendes olukordades ei ole vaja pidevat andmevoogu.

Andmevahetuskiht koordineerib brauseri kaudu saadetavaid juhtimiskäskke ning auto pardalt saabuvat videopilti ja telemeetriat. Juhtkäskudena edastatakse roolinurga ja mootori kiiruse väärtused, millele vastuseks saadetakse reaalajas telemeetria ja kaamerapilt. Videopilt saadetakse MJPEG formaadis ja edastatakse WebSocketi kaudu, mis hoiab latentsuse madalana ning vähendab protsessori koormust. See võimaldab kaadrite töötlemist ja edastamist märgatavalt efektiivsemalt.

Väikesemõõtmelise auto tarkvara töötab Raspberry Pi 4 Model B platvormil, kus DonkeyCari raamistik ja autori arendatud täiendav moodul moodustavad sõidukipoolse rakenduse (vaata peatükk 2.3). DonkeyCar kasutab modulaarset pipeline-struktuuri, kus põhikomponendid (näiteks roolimise ja kiiruse juhtimine) ning lisamoodulid (parts) täidavad oma ülesandeid järjestatult [3]. Sõidukipoolne rakendus ühendab DonkeyCari torupõhise arhitektuuri ning autori loodud mooduli, mis tegeleb võrguühenduse halduse, režiimide vahetuse ning andmeedastuse loogikaga.

2.2 Kasutajaliides

Kasutajaliides on veebirakendus, mis on välja töötatud Svelte raamistikuga. Svelte'i eeliseks on see, et erinevalt enamikust JavaScripti raamistikest ei kasuta see virtuaalset dokumendi objektimudelit²⁰ (ingl k *virtual* DOM – kergekaaluline koopia brauseri tegelikust DOM-puust, mida kasutatakse muudatuste tuvastamiseks) [35]. Samas artiklis selgitatakse, et Svelte kompileerib komponendid otse optimeeritud JavaScriptiks, mis vähendab töömahtu brauseris. Selline arhitektuur võimaldab vähendada latentsust ja suurendab sujuvust, mis on kriitilise tähtsusega just kaugjuhtimise kasutajaliidese puhul, kus reageerimiskiirusel on otsene mõju süsteemi kasutatavusele ja ohutusele.

Liidese kujundus järgib minimaalset ja funktsionaalset lähenemist. Keskseks visuaalseks elemendiks on reaalajas videovoog auto esikaamerast, mille ümber paiknevad

²⁰ https://en.wikipedia.org/wiki/Virtual_DOM

telemeetriaandmed ja juhtimisnupud (vaata peatükk 3.1). Selline paigutus toetab kiiret reageerimist ja aitab vältida liigselt koormavat kasutuskogemust.

Operaator saab liidese seadete vaates valida juhtimisseadme (näiteks pult või roolikontrolleri), sisse ja välja lülitada autonoomse režiimi ning alustada või lõpetada andmete salvestamise. Kasutajaliidese kaudu saab operaator määrata gaasi ja roolinurga käskudele skaleerimistegurid; süsteem korrutab iga käsu vastava teguriga, piirates maksimaalset väärtust ja tagades sujuvama reageerimise. Kõik need valikud on koondatud seadete vaatesse, millele pääseb ligi liidese vasakul üleval servas asuva menüünupu kaudu.

Kasutajaliides ei piirdu üksnes videopildiga, vaid kuvab ka telemeetrilisi andmeid, mis iseloomustavad võrgu kvaliteeti ja süsteemi üldseisundit. Ekraanil esitatakse reaajas järgmine teave:

- signaali tugevus (*Received Signal Strength Indicator* ehk RSSI);
- aktiivse pääsupunkti MAC²¹ aadress;
- parajasti rakenduv juhtimisrežiim (autonoomne või käsitsi);
- kasutajaliidese ja serveri vaheline ühenduse olek.

Selline ülevaade võimaldab operaatoril hinnata süsteemi stabiilsust ja reageerida kiiresti sõiduki käitumise muutustele. Kasutajaliidese paigaldamise ja konfigureerimise täpsemad juhised on esitatud lisas (Lisa 3).

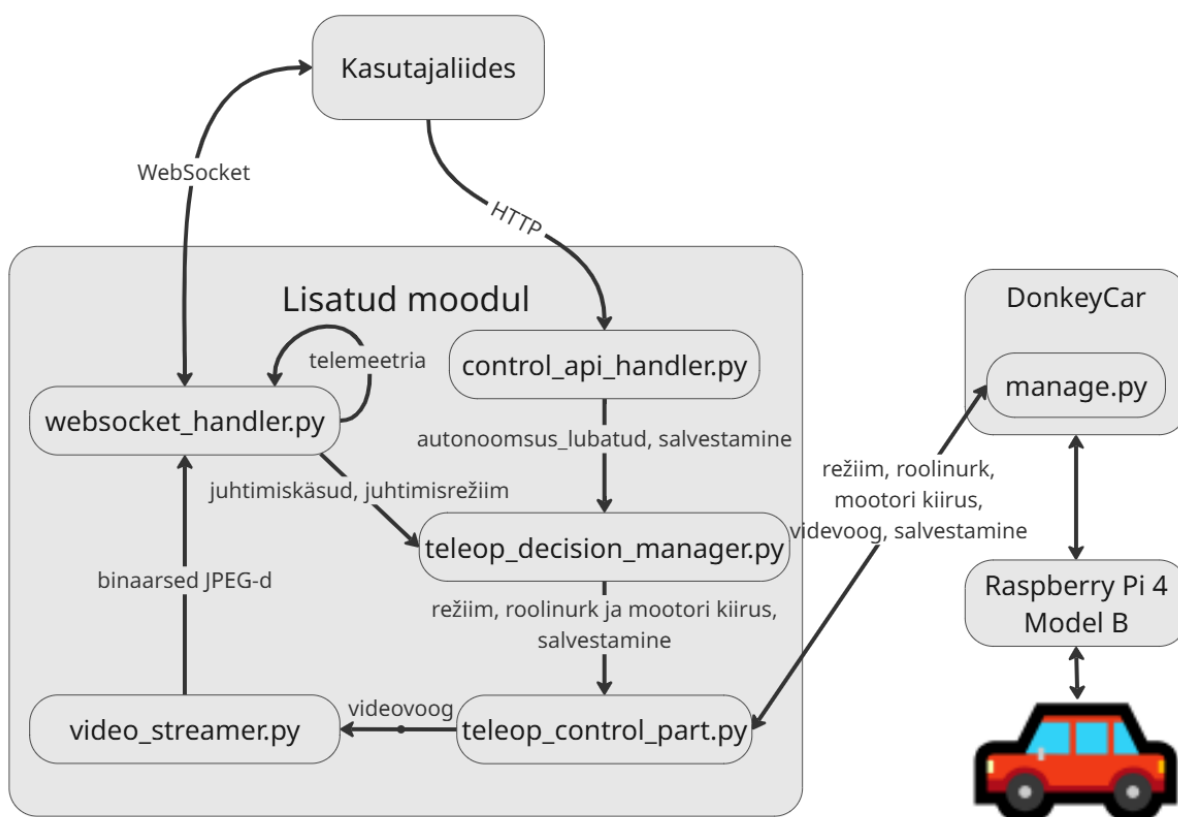
2.3 Sõidukipoolne rakendus

Sõidukipoolne rakendus koosneb kahest integreeritud komponendist (joonis 3): DonkeyCari avatud lähtekoodiga raamistikust ja käesoleva töö raames arendatud täiendavast moodulist. Rakendus töötab Raspberry Pi 4 Model B platvormil ning täidab kogu juhtimisloogika, võrgujälgimise ja andmevahetuse funktsionaalsuse.

DonkeyCar toimib platvormi selgroona. Raamistik käivitab järjestikku töötavad moodulid mille kaudu loetakse sisendeid (nt kaamerapilt või operaatori käsud), arvutatakse abstraktsel skaalal $[-1,1]$ vahemik juhtimisotsustest täpsed käsud servomootoritele, mis kontrollivad esirataste nurka ning rataste pöörlemist, ning saadetakse need edasi sõiduki komponentidele. Kui kasutatakse autonoomset režiimi, rakendab DonkeyCar treenitud masinõppemudelit, et

²¹ <https://et.wikipedia.org/wiki/MAC-aadress>

otsustada kaamerapildi põhjal automaatselt, millised juhtimiskäskud on sobilikud.



Joonis 3. Täiendava mooduli ülesehitus.

Töö käigus loodud täiendav moodul integreeritakse DonkeyCar-iga ning vastutab teleoperatsiooniks vajalike funktsioonide täitmise eest. Selle ülesanded jagunevad mitmeks valdkonnaks:

- Võrguliidesed: reaaliajas andmevahetus WebSocketi kaudu („*websocket_handler.py*“) ning seadistuste haldamine ja muude mitte-ajakriitiliste käskude vastuvõtt HTTP-põhiselt („*control_api_handler.py*“).
- ArUco-markeri tuvastus: kaameravoo reaaliajas analüüs markerite tuvastamiseks; iga tuvastatud marker autonoomses režiimis käivitab peatumisloogika, tagades platvormi kohese peatuse ohuolukorras („*teleop_decision_manager.py*“).
- Režiimi otsustusloogika: käsitsi ja autonoomse juhtimisrežiimi vahel lülitumine vastavalt võrguühenduse olekule ja operaatori eelistusele („*teleop_decision_manager.py*“).

- Videovoog töötlemine: kaamerapildi kodeerimine JPEG-kaadriteks („video_streamer.py“) ja nende edastamine kasutajaliidesesse („websocket_handler.py“).
- Juhtimiskäskude edastamine: reaaliajase roolinurga ja mootori kiiruse väärtuste saatmine DonkeyCari sisendisse („control_manager.py“).
- Telemeetria edastamine: süsteemi hetkeseisu ja info (näiteks režiim, salvestusolek, RSSI) edastamine kasutajaliidesesse („websocket_handler.py“).

Mooduli aktiveerimiseks lisatakse DonkeyCari „manage.py“ faili eraldi seadistuslipp, mis võimaldab mooduli sisse või välja lülitada, säilitades samas kogu platvormi paindlikkuse ja ühilduvuse algse arhitektuuriga. Rakenduse paigaldamise juhend ja mooduli aktiveerimise sammud on toodud lisa (Lisa 4).

2.4 Autonoomse režiimi otsustusloogika

Autonoomse režiimi valik põhineb hierarhilisel mehhanismil, kus iga tase järgneb eelnevale ja võtab arvesse esmalt operaatori eelistust ning seejärel võrgu ja videovoogude seisundit. Režiimivahetust juhitakse spetsiaalse WebSocketi sõnumiga (*autonomy = true* või *false*), mida kasutajaliides edastab 10 Hz sagedusega ning mille pardal töötav tarkvara vastu võtab ja rakendab. Alljärgnevalt on tasandid ja nende loogika lahti seletatud (joonis 4):

1. Operaatori valik

Kõrgeima prioriteediga on operaatori otsus – kui kasutajaliideses pole autonoomset režiimi lubatud (serveris hoitakse eraldi muutujat, mis on muudetav HTTP põhise API kaudu), jääb sõiduk alati käsitsi juhtimisele, olenemata võrgu või videovoogude seisundist.

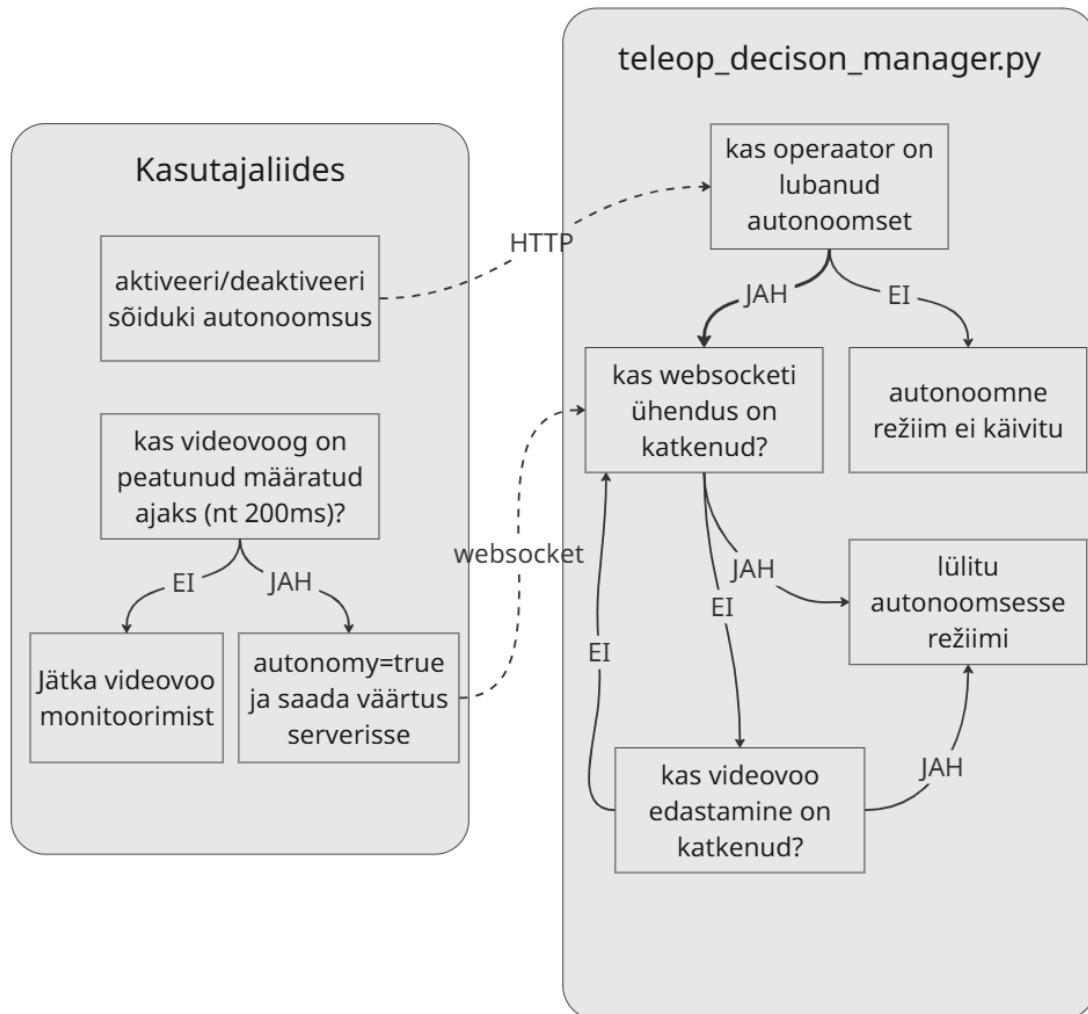
2. Videovoogude monitooring

Kui autonoomia on lubatud, jälgib süsteem pidevalt kaadrite saabumist. Kui kasutajaliideses katkeb videoedastus määratud ajaperioodiks (näiteks 200 ms), saadab see WebSocket i kaudu *autonomy = true*, et lülitada sisse autonoomne juhtimisloogika. See tagab, et sõiduk jätkab ohutult liiklemist isegi videopildi puudumisel.

3. Täielik ühenduse katkestus

Kui WebSocketi kaudu ei laeku režiimi sõnumeid (*autonomy = true/false*) määratud ajavahemiku jooksul (näiteks 200 ms), käsitletakse seda täieliku katkestusena. Kui

operaator on liideses autonoomse režiimi lubanud, lülitub süsteem sel juhul automaatselt autonoomsele juhtimisele.



Joonis 4. Autonoomsele režiimile lülitumise otsustusloogika katkestuse korral.

Autonoomse režiimi sisselülitamisel edastab server DonkeyCar'i raamistikule käsu rakendada eelnevalt treenitud närvivõrgu mudelit. Seejärel sõidab platvorm eel konfigureeritud püsikiirusega, samal ajal kui iga kaadri põhjal arvutatakse mudelis hetke sobiv roolinurk ja saadetakse kontrolleriile. Kui autonoomses režiimis tuvastatakse ArUco marker, käivitub peatumise loogika ja sõiduki kiirus määratakse nulliks.

Pärast video või ühenduse taastumist jääb sõiduk autonoomsesse režiimi seni kuni operaator annab esimese käskluse (roolinurga või kiiruse muutus). Nii välditakse kiireid režiimivahetusi lühiajaliste katkestuste ajal ja tagatakse sujuv, ettearvatav üleminek tagasi manuaalsele

juhtivale režiimile. Juht ei pea ühenduse taastumisel hetkega konteksti tabama ning ise juhtima hakkama vaid võib autonoomset sõitu jälgida enne kui juhtimiskäske andma asub.

2.5 Testraja kirjeldus

Testraja eesmärk oli hinnata, kui kiiresti ja usaldusväärset sõiduk lülitub käsitsi juhtimiselt autonoomsele režiimile (ja tagasi) olukordades, kus visuaalne tagasiside katkeb lühiajaliselt või side kaob Wi-Fi pääsupunkti vahetuse tõttu. Testrada asus Tartu Ülikooli Delta õppehoone III korruse kaarsilla-poolses tiivas.

Testrada algas 3. korruse aatriumi-poolsest koridorist (punase punktiirjoonega märgitud trajektoori, Lisa II) ja kulges mööda esimest koridori kuni Vaba Lava ette. Seal pöördus sõiduk teise, paralleelse koridori kaudu tagasi aatriumipoolsesse koridori, moodustades ringikujulise trajektoori.

Rada kujundati nii, et sõidukil oleks kogu trajektoori vältel pidev visuaalne referentspunkt paremal pool, mis lihtsustas autonoomse juhtimise mudeli treenimist trajektoori iseseisvaks järgimiseks. Seetõttu paiknes enamikus ringi osades paremal pool sein või muu referentspunkt.

Testirada sisaldas järjestikku paiknevaid Wi-Fi pääsupunkte, mis võimaldasid AP vahetuste loogika testimist. Eelnevatest mõõtmistest selgus, et ühe ringi läbimisel toimub keskmiselt 3–4 AP vahetust. Iga vahetuse juures logiti RSSI väärtus, ühenduse katkestuse kestus ja taastumise aeg, et hinnata võrgu stabiilsust ja selle mõju reaalses juhtimisele.

Demoringi käigus kogutud andmete põhjal arvutati järgmised kvantitatiivsed näitajad, mis esitatakse töö tulemuste osas:

- keskmine AP vahetuste arv ringi kohta;
- videopildi katkestuste sagedus ja kestus;
- ühenduse taastumise aeg pärast AP vahetust;
- režiimivahetuste töökindlus ja õiguspärasus.

Kõik need mõõdikud võimaldavad hinnata nii süsteemi töökindlust keerukates võrguoludes kui ka autonoomse juhtimisloogika täpsust ja reageerimisvõimet. Lahenduse praktilist toimimist katseringil on võimalik näha lisas V esitatud videos.

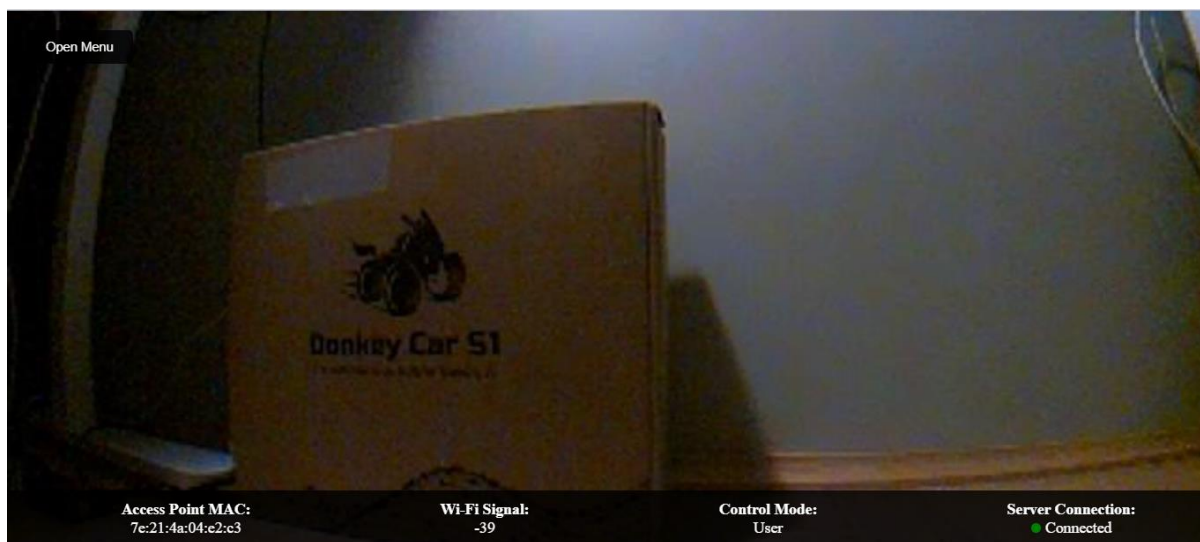
3. Tulemused

Käesolevas peatükis esitletakse välja töötatud süsteemi katsetulemusi kolmest vaatenurgast. Esiteks antakse ülevaade kasutajaliidese põhivaatest ja seadistuspaneelist koos olulisemate infotahvlitega; teiseks tuuakse välja Raspberry Pi 4 platvormil mõõdetud reaalajas jõudlusnäitajad; kolmandaks käsitletakse sõiduki töökindlust testiringil muutuvates Wi-Fi-tingimustes.

3.1 Kasutajaliides

Jooniselt 5 on näha kasutajaliidese peavaade, kus kogu ekraani täidab reaalajas videopilt sõiduki esikaamerast. Vasakus ülanurgas paikneb nupp “Open Menu”, mille aktiveerimine avab seadistuspaneeli (joonis 6). Ekraani allservas jookseb musta taustaga olekuriba, mis kuvab kriitilisi andmeid ja mõõdikuid:

- „Access Point MAC“ – ühendatud pääsupunkti MAC-aadress;
- „Wi-Fi Signal“ – signaali tugevus (RSSI);
- „Control Mode“ – hetkel valitud juhtimisrežiim (näites „Autonomous“);
- „Server Connection“ – serveriühenduse olek, mis kasutab värvilist indikaatorit (punane – ühendus puudub, roheline – ühendus olemas)

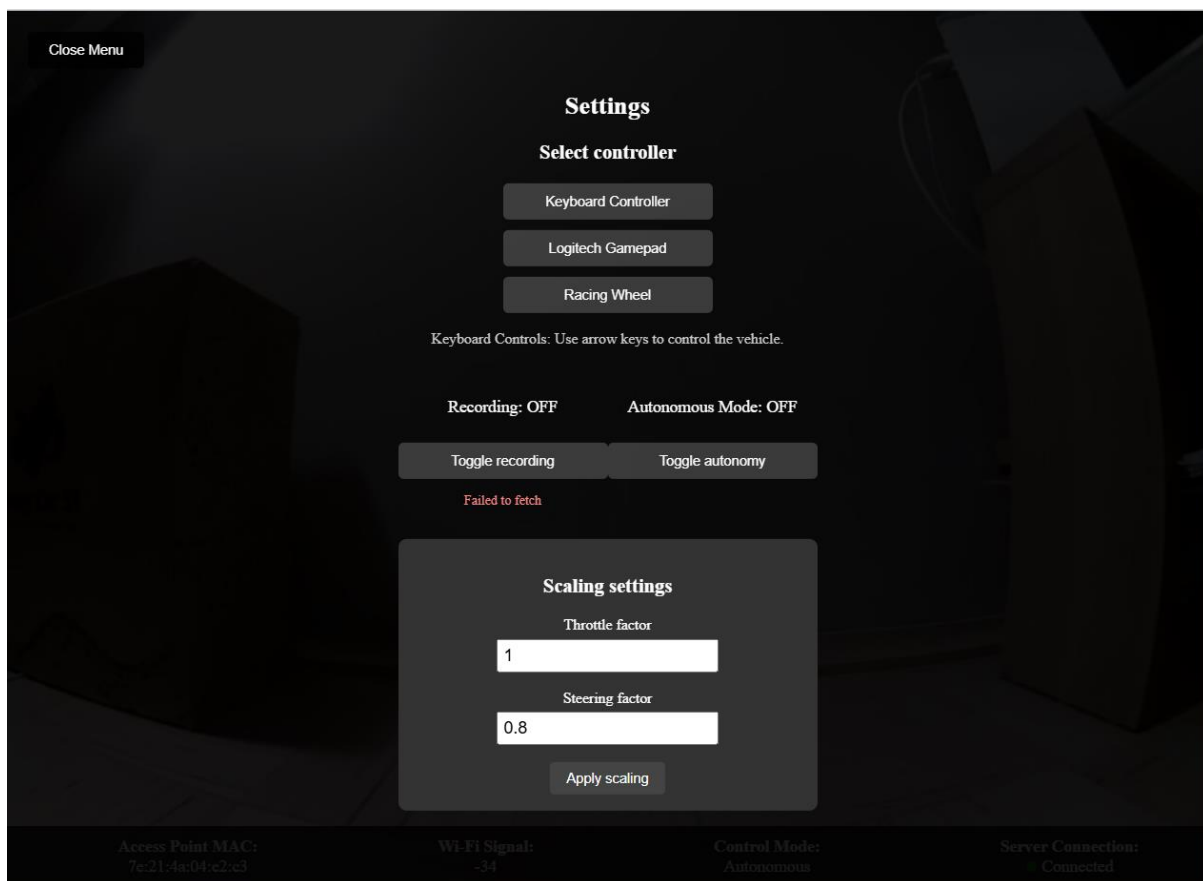


Joonis 5. Kasutajaliidese põhivaade.

Selline minimalistlik vaade koondab olulise võrgustatistika ja süsteemi-oleku kasutaja vaatevälja, jättes põhiosa ekraanist videopildile, mis on kriitiline sõiduki juhtimiseks.

Joonis 6 kuvab seadete vaadet, mille saab sulgeda vasakus ülanurgas asuva nupuga „Close Menu”. Paneeli sisu on jaotatud kolmeks alamosaks:

- „Select controller“ – kasutaja valib seadme, millega kontrollib sõidukit üle kasutajaliidese. Pärast valikut näidatakse ploki all kontrolleri põhinuppe lühikirjeldusega, kuni valikut pole tehtud, jääb nähtavale teade „Select a controller to see its scheme“.
- „Operating modes“ – tekstilised indikaatorid „Recording” ja „Autonomous Mode” kajastavad režiimi hetkeolekut; nupud „Toggle recording” ja „Toggle autonomy” lubavad režiime sisse või välja lülitada.
- „Scaling settings“ – jaotises määratakse gaasi ja roolinurga käskudele skaleerimistegurid („Throttle factor“, „Steering factor“); sisestatud väärtused jõustuvad nupu „Apply scaling“ aktiveerimisel.



Joonis 6. Kasutajaliidese seadistuspaneel juhtimis- ja režiimivalikutega.

3.2 Süsteemi jõudlus

Peatükis keskendutakse süsteemi võimele töödelda reaajas videovoogu ja säilitada stabiilne jõudlus riistvararessursside piires. Jõudlustestid viidi läbi peatükis 2.5 kirjeldatud rajal, mille sõiduk läbis kokku kümme ringi.

3.2.1 Kaadrisagedus

Videopildi sujuvuse hindamiseks mõõdeti kasutajaliideses Raspberry Pi poolt saadatud kaadrite reaalsel saabumise sagedust kaadrites sekundis (fps). Iga kaadri saabumisel kutsuti välja *handleCameraFrame* funktsioon, mis loendas WebSocketi kaudu jõudnud videokaadreid. Tulemused logiti 5-sekundilise intervalliga. Logidesse kirjutati need perf tüüpi sündmustena.

Analüüsi tulemused näitasid:

- Keskmise kaadrisagedus: 15,6 fps
- Maksimaalne mõõdetud kaadrisagedus: 20 fps
- Minimaalne mõõdetud kaadrisagedus: 0 fps

Kaadrisagedus püsis katse jooksul üldiselt stabiilne, kuid esines episoodilisi langusi, mis viitavad ajutistele takistustele andmevoos. Arvestades, et süsteemi ressursikasutus jäi mõõduka koormuse piiridesse (vaata peatükk 3.2.2), võib eeldada, et edaspidi võiks katsetada videovoogu, mille kaadrisagedus on seatud 30 fps peale. See aitaks saavutada visuaalselt sujuvamat kuvapilti, säilitades samas süsteemi töökindluse.

3.2.2 Süsteemiresursid

Süsteemi töökindluse ja jätkusuutlikkuse hindamiseks viidi läbi Raspberry Pi ressursikasutuse monitooring, mille käigus koguti andmeid protsessori (CPU) koormuse ja muutmälu (RAM) kasutuse kohta. Logimine toimus üks kord sekundis ning iga kirje sisaldas ajatempiliga mõõtmisi protsessori kasutusmäärast protsentides ja kasutusel olevast muutmälumahust megabaitides. Andmed salvestati sys-tüüpi logikirjetena.

Tulemuste analüüs näitas, et keskmine CPU-koormus mõõteperioodi vältel oli 46,7%, ulatudes maksimaalselt kuni 67,0%. Keskmine mälukasutus oli 463 MB, samas kui maksimaalne väärtus ulatus 566 MB-ni. Need näitajad viitavad sellele, et süsteem ei opereeri riistvara maksimaalse arvutusvõimekuse lähedal.

Erinevate režiimide võrdlus näitas, et autonoomsel režiimil esinesid märgatavalt kõrgemad CPU- ja mälukasutuse väärtused. See on ootuspärane, kuna autonoomne juhtimine nõuab

suuremat arvutusvõimsust. Lisaks tavapärasele videosignaali töötlemisele aktiveerub selles režiimis ka masinõppe mudel, mis teeb jooksvalt otsuseid videopildi põhjal. Seega koormatakse süsteemi paralleelselt nii visuaalse sisendi dekodeerimise, otsustusloogika käivitamise kui ka juhtsignaalide väljastamisega.

Arvestades, et süsteem töötas neljatuumalise Raspberry Pi 4 peal, millel oli mõõtmiste hetkel piisavalt vaba mälumahtu (üle 1 GB), võib järeldada, et rakendus ei ammendanud saadaval olevaid riistvaralisi ressursse. Nii CPU kui RAM-i osas jäi piisavalt töövaru, mis võiks võimaldada lisakoormuse korral (näiteks kõrgema resolutsiooniga videovoo töötlemisel või paralleelse videokodeerimise lisamisel) säilitada süsteemi stabiilsuse.

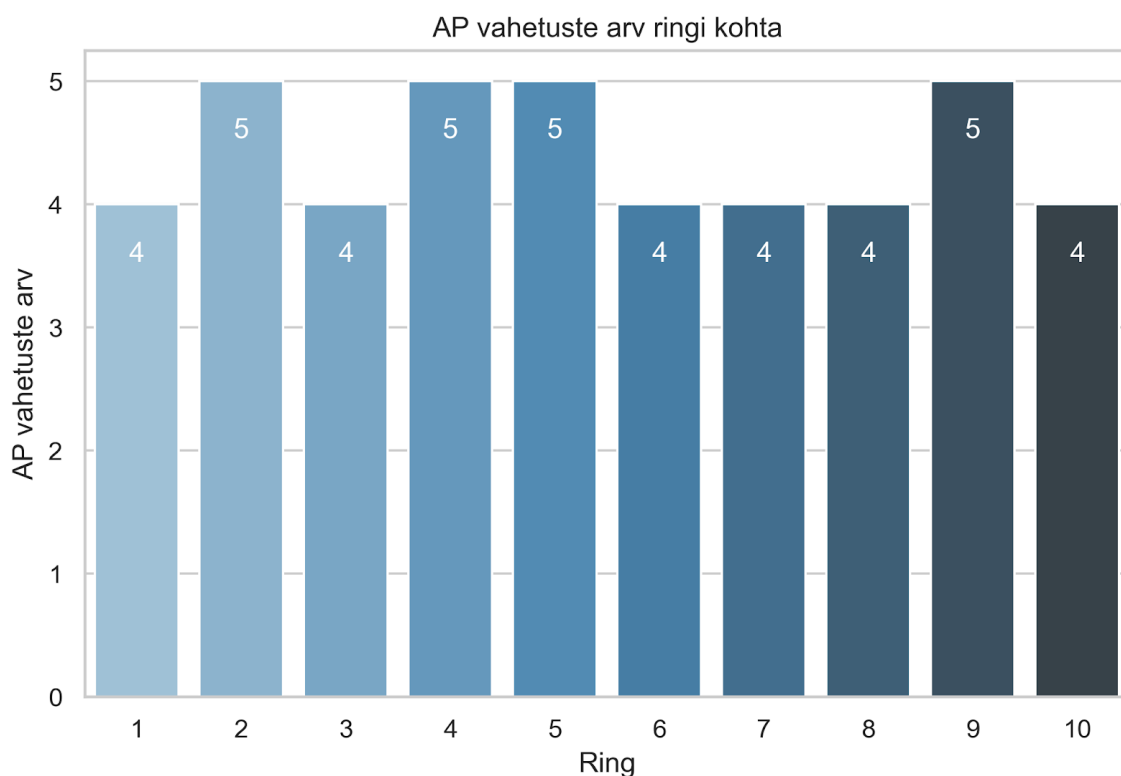
Oluline on märkida, et vaatamata videovoo töötlemisele ja juhtsignaalide paralleelsele haldamisele ei esinenud süsteemis ressursside ülekasutust, mis võiks põhjustada töökatkestusi või viivitusi. See näitab, et kasutusel olev tarkvaraarhitektuur ning valitud andmesideprotokollid on piisavalt optimeeritud ja ressursisäästlikud, et toetada töökindlat kaugjuhtimist ka piiratud riistvaravõimekusega seadmel.

3.3 Testringi tulemused

Käesolevas peatükis analüüsitakse süsteemi töökindlust ja võrguühenduse kvaliteeti, tuginedes kümne testiringi jooksul kogutud logiandmetele. Jõudlustestid viidi läbi peatükis 2.5 kirjeldatud rajal, kus sõiduk läbis kokku kümme ringi. Mõõdetud näitajateks olid pääsupunktide (AP) vahetuste sagedus, videopildi katkestuste esinemine ja nende kestus, ühenduse taastumise kiirus ning režiimivahetuste töökindlus. Kõik näitajad on arvutatud ringipõhiste keskmistena, et anda usaldusväärne ülevaade süsteemi tüüpilisest käitumisest reaalajas kaugjuhtimise olukorras.

3.3.1 Pääsupunkti vahetuste sagedus

AP-vahetused registreeriti logiandmetes sündmustüübina *ap_switch*. Joonisel 7 on kujutatud AP-vahetuste arv igas ringis. Analüüsi tulemuste põhjal toimus ühe ringi jooksul keskmiselt 4,4 pääsupunkti vahetust. Kõige sagedasemad vahetused registreeriti 2., 4., 5. ja 9. ringil, kus igas ringis leidis aset viis vahetust. Seevastu 1., 3., 6.–8. ja 10. ringil piirdus vahetuste arv neljaga.



Joonis 7. Pääsupunktide (AP) vahetuste arv ühe katseringi kohta.

Iga AP-vahetus põhjustas lühiajalise sidekatkestuse kasutajaliidese ja sõiduki vahel, mille tõttu aktiveerus automaatselt autonoomne juhtimisrežiim. Enamasti vastas üks autonoomne episood ühele AP-vahetusele, kuid 4. ja 5. ringil esines olukord, kus lühikese ajavahemiku jooksul toimus järjest mitu pääsupunkti vahetust. See viitab võimalikule võrgu katvuse ebahühtlusele nendes lõikudes, mis võib tekitada lühiajalisi ühenduse ebastabiilsusi ja mitmekordseid režiimimuutusi.

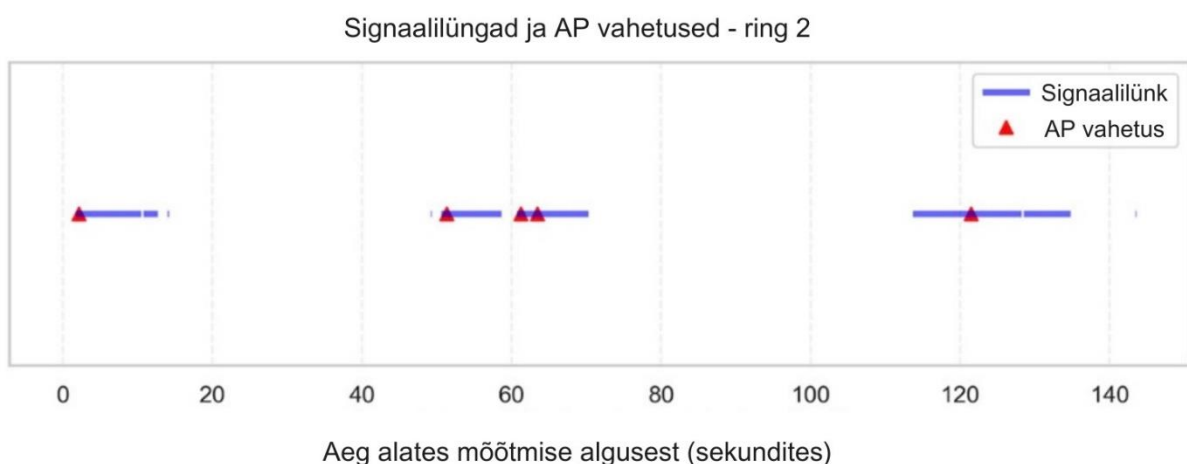
3.3.2 Videopildi katkestused

Videopildi katkestus (signaalilünk) tähistas ajavahemikku, millal süsteem ei suutnud ühtegi uut videokaadrit edastada kasutajaliidesele vähemalt 200 millisekundi jooksul. Iga selline katkestus logiti koos algus- ja lõpuajatempli (*start* ja *end*) väärtustega, võimaldades määrata katkestuse kestuse ja esinemissageduse. Käesolevas alapeatükis keskendutakse katkestustele, mille kestus ületas ühe sekundi. Sellised katkestused kajastusid kasutajaliidese kriitilise videopildi kadumisena ning põhjustasid süsteemi automaatse lülitumise autonoomsele juhtimisrežiimile.

Analüüsi kohaselt esines keskmiselt 5 signaalilünka ühe ringi kohta. Keskmine katkestuse kestus oli 7,1 sekundit ning maksimaalne tuvastatud katkestus ulatus 20,79 sekundini. Kuigi videovoog püsis suurema osa ajast stabiilsena, viitavad tulemused olulistele ajutistele ühendushäiretele, mis võivad mõjutada reaajas juhtimise usaldusväärsust.

Lisaks analüüsiti, kui kiiresti videopilt pärast pääsupunkti vahetust taastub. Selleks võrreldi *ap_switch* sündmuste ajatemplite ümbruses toimunud signaalilünki, mille lõpp langes ± 500 millisekundi aknasse AP-vahetuse ümber. Tuvastatud juhtumite põhjal oli keskmine videopildi taastumise aeg pärast pääsupunkti vahetust 7.37 sekundit. Tulemused näitavad, et kuigi ühendus taastub tavaliselt kiiresti, võib isegi lühiajaline katkestus kriitilistes juhtimissituatsioonides vähendada süsteemi kasutatavust ja ohustada operatiivset usaldusväärsust.

Joonisel 8 on visualiseeritud signaalilünkade ja pääsupunkti vahetuste ajastused. Punased kolmnurgad tähistavad AP-vahetusi ning sinised horisontaalsed jooned signaalilünki. Graafikust nähtub, et enamik videopildi katkestusi järgnesid vahetult pääsupunkti vahetusele, kusjuures mõnel juhul toimus järjestikku mitu AP-vahetust. Nähtav on selge korrelatsioon: enamik katkestusi toimus vahetult pärast pääsupunkti vahetust. Samuti tuvastati üksikuid katkestusi, mis ei olnud seotud AP sündmustega, viidates üldise võrgu ebastabiilsusele või süsteemi ajutisele ülekoormusele.

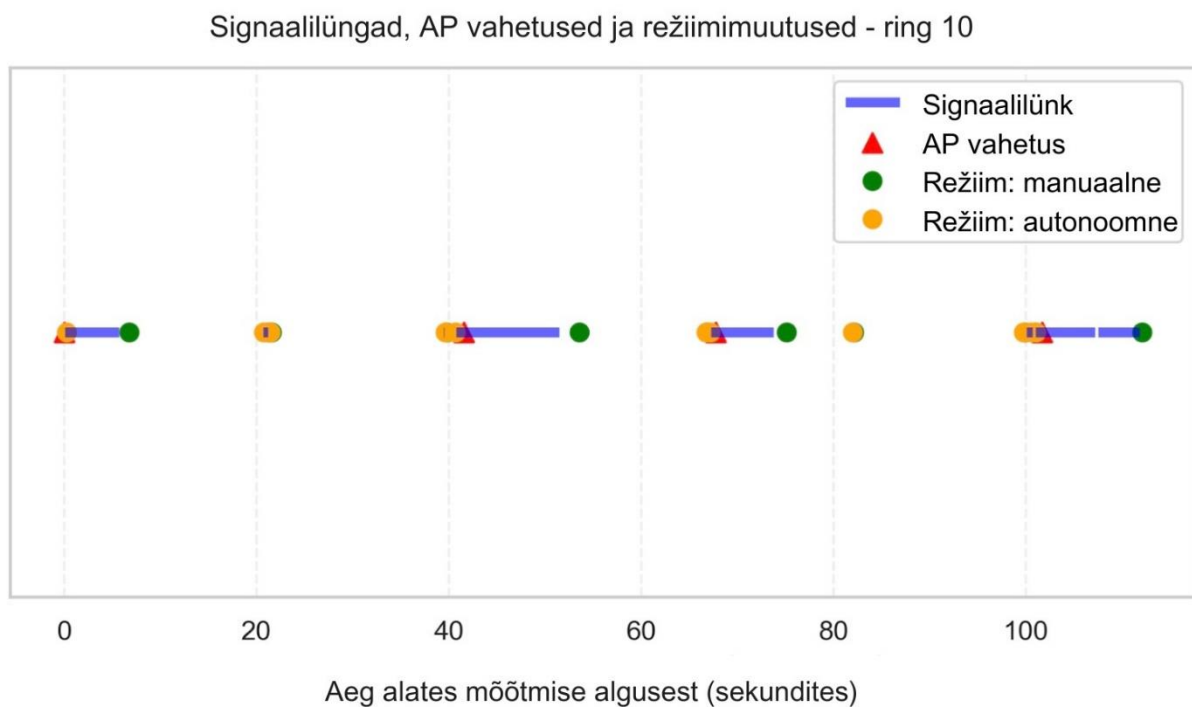


Joonis 8. Videovoog signaalilüngad ja pääsupunktide (AP) vahetused 2. ringil.

3.3.3 Režiimivahetuste töökindlus ja õiguspärasus

Režiimivahetused käsitsi ja autonoomse juhtimisrežiimi vahel toimusid süsteemi otsustusloogika järgi, lähtudes ühenduse olemasolust ja videopildist. Režiimivahetused registreeriti sündmustena `mode_auto_switch`. Logiandmete analüüsi eesmärk oli hinnata, kuidas režiimivahetused vastasid tegelikule olukorrale ning kas esines valesid või põhjendamatu ümberlülitusi.

Joonisel 9 on visualiseeritud 10. ringi sündmued ajateljel. Punased kolmnurgad tähistavad pääsupunkti (AP) vahetusi, sinised jooned signaalilünki ning värvilised punktid režiimimuutusi – roheline tähistab manuaalset režiimi ning oranž autonoomset režiimi. Analüüsist selgus, et enamik režiimimuutusi järgnesid signaalilünkadele või pääsupunkti (AP) vahetustele, mis viitab sellele, et süsteem reageeris ühenduse katkemisele ootuspäraselt ja kooskõlas eeldatud loogikaga.



Joonis 9. Signaalilüngad, pääsupunktide (AP) vahetused ja režiimimuutused 10. ringil.

Siiski esines ka lühikesi episoode, kus süsteem lülitus mõne sekundi jooksul korduvalt manuaalse ja autonoomse režiimi vahel. Need olukorrad võivad viidata liiga tundlikule otsustuskünnise seadistusele või ühenduse ajutisele katkemisele, millele järgnes ühenduse taastumine. Selliseid episoode võib pidada valedeks režiimivahetusteks, kuna need ei lähtunud

tegelikust vajadusest muuta juhtimisrežiimi. Katse läbiviimisel ei olnud sellised vaheldumised operaatorile tajutavad, kuid ilmnid selgelt hilisemas logianalüüsis.

Üldiselt võib järeldada, et režiimivahetuse loogika toimis korrekselt ja reageeris sündmustele loogiliselt. Küll aga võib süsteemi töökindlust suurendada ajapõhiste viivituste rakendamisega, et vältida kiireid ja mittevajalikke režiimimuutusi ebastabiilse võrgu tingimustes.

4. Tulevikusuunad ja rakendusvaldkonnad

Käesoleva töö tulemused on näidanud süsteemi head toimivust kontrollitud tingimustes, kuid kogutud kogemus ja tekkinud ideed avavad mitmekesised võimalused edasiseks arenduseks ning laitemateks rakendusteks. Järgnevates alampeatükkides esitatakse esmalt tehnilised ettepanekud süsteemi jõudluse, stabiilsuse ja funktsionaalsuste täiendamiseks, seejärel kirjeldatakse täiendavaid kasutusvõimalusi nii akadeemilises õppes kui ka tööstuslikes ja demo-rakendustes.

4.1 Edasised arengusuunad

Käesoleva töö tulemuste ja testimise käigus kogutud kogemuste põhjal ilmnesid mitmed valdkonnad, kus süsteemi tööd saab oluliselt täiustada ja laiendada. Alljärgnevalt on välja toodud peamised arendusideed, mis keskenduvad nii videoedastuse stabiilsuse parandamisele kui ka autonoomse juhtimisloogika töökindluse ja skaleeritavuse tõstmisele:

- **Dünaamiline bitikiiruse kohandamine vastavalt võrguoludele.** Testimised näitasid, et Wi-Fi pääsupunkti vahetamise ajal langeb videopildi kvaliteet ja suureneb latentsus, mistõttu on vajalik bitikiiruse automaatne reguleerimine müra ja andmekao põhjal.
- **Andurite lisamine ja nende kombineerimine.** Ühe kaamera põhine mudel toimib usaldusväärsetel teatud valgusoludes ja jätab koridoride pimedad nurgad katmata. 360° kaamerasüsteem või lidar pakuks valgusest sõltumatut, täielikku ruumipilti, kuid nõuaks nii süsteemi arhitektuuri kohandamist kui ka operaatori kasutajaliidese ümberkujundamist.
- **Mitme autonoomse sõiduki samaaegne juhtimine.** Platvormi arhitektuuri laiendada nii, et üks operaator saab paralleelselt hallata mitut autonoomset sõidukit. See võimaldaks uurida paralleeljuhtimise eripära, operaatori kognitiivset koormust ja otsustusprotsesse erinevates tingimustes, näiteks erineva isejuhtimise määra juures. Näiteks, kas operaator suudab jooksvalt ette näha, et mudel vajab varsti abi ning kuidas sujuvalt lahendada olukordi kui mitu sõidukit vajavad samaaegselt abi.
- **Operaatori käitumise analüüs erinevates võrgutingimustes.** Võrgu stabiilsuse kõikumine mõjutab operaatori otsustusprotsessi ja juhtimisstiili manuaalses režiimis. Uurida, kuidas võrgu kvaliteedi kõikumine mõjutab kaugjuhi reaktsiooniaega, täpsust ja kognitiivset koormust ning milliseid üldisi strateegiaid võiks rakendada juhtimiskogemuse ja turvalisuse parandamiseks.

- **Takistuste ja mudeli ennustuste visualiseerimine.** Operaatori olukorratedadlikkuse tõstmiseks kuvatakse reaajas mudeli ennustatud roolinurk ja kiirus koos andurite tuvastatud takistustega. See võimaldab operaatoril mudeli soovitusi kiiresti hinnata ning vajadusel trajektoori sujuvalt kohandada.

4.2 Võimalikud rakendused

Lõputöö praktilises osas valminud platvorm pakub rakendusvõimalusi teadusuuringutes ja haridustöös kui ka näidisplatvormina erinevatel messidel ning esitlustel. Platvorm pakub usaldusväärset ja paindlikku võimalust kaugjuhtimise valdkonna edasiseks uurimiseks.

Platvorm võimaldab teadusuuringutes süstemaatiliselt hinnata teleoperatsiooni juhtimisviiside ja võrguolukordade mõju. Näiteks annab see võimaluse kvantifitseerida käsijuhtimise ja autonoomrežiimi vahetuste kiirust ning täpsust Wi-Fi pääsupunktide ümberlülitumise ajal, pakkudes väärtuslikku teaduslikku ettekujutust signaalikao ja latentsuse mõjust süsteemi reageerimisvõimele.

Hariduslikus kontekstis pakub platvorm väärtuslikku projektipõhist õppevahendit isejuhtivate sõidukite ja teleoperatsiooni aluste õppimiseks. Intuiitvne veebiliides koos reaajas videovoo ja telemeetriaga võimaldab õpilastel lahendust disainida, testida ja analüüsida ning vajadusel kohandada.

Platvorm on sobiv ka messide, konverentside ja koostööürituste demonstreerimiseks. Osalejad saavad ise sõidukit juhtida ja autonoomset loogikat jälgida, mis aitab süvendada arusaama kaugtöö ja tehisintellekti integreerimisest ning soodustada huvi ja koostöövõimalusi.

Kokkuvõte

Bakalaureusetöö eesmärk oli arendada väiksemõõtmelise mänguauto kaugjuhtimissüsteem, mis võimaldaks Tartu Ülikooli isejuhtivate sõidukite laboris turvaliselt ja efektiivselt testida erinevaid juhtimis- ning autonoomialahendusi. Käsitletud probleem seisnes usaldusväärse ja minimaalse viitega kaugjuhtimissüsteemi loomises, mis oleks piisavalt täpne ning kiire, et toetada autonoomsete süsteemide arendust ja testimist kontrollitud keskkonnas.

Töö raames valmis kaugjuhtimislahendus väiksemõõtmelisele sõidukile, kasutades Donkey Car avatud lähtekoodiga platvormi ja Raspberry Pi 4. Platvormi tarkvaraarhitektuur ehitati ümber nii, et see võimaldaks reaajas juhtimissignaali vastuvõttu ja töötlemist üle Wi-Fi. Juhtimiskäsud (näiteks roolinurk ja mootori kiirus) saadeti brauseripõhisest kasutajaliidesest WebSocket-protokolli kaudu, mis tagas madala latentsuse. Samal ajal edastati sõidukilt kaamerapilt ja süsteemi telemeetria tagasi liidesesse. Kui ühendus katkes või videopilt muutus ebastabiilseks, aktiveerus automaatselt autonoomne juhtimisrežiim. Lisaks loodi loogika, mis tuvastas ArUco-markereid ning käivitas ohutuspeatuse autonoomses režiimis vastavalt markerite olemasolule. Loodud süsteem võimaldas läbi viia katseid, mille eesmärk oli testida katkestusest taastumist, režiimide vahetamise töökindlust ning süsteemi toimivust mitme Wi-Fi pääsupunkti tingimustes.

Testimise tulemused näitasid, et loodud süsteem suudab hoida stabiilset kaadrisagedust keskmiselt 15,6 kaadrit sekundis, ulatudes maksimaalselt 20 fps-ni. Ressursikasutuse analüüs kinnitas, et Raspberry Pi 4 platvormil jäi piisavalt CPU ja RAM töövaru, võimaldades sujuvat andmeedastust ka autonoomses režiimis. Kümnel testiringil mõõdeti keskmiselt 4,4 Wi-Fi pääsupunkti vahetust ühe ringi kohta, millest enamik põhjustas lühiajalise videopildi katkestuse ja autonoomsele režiimile lülitumise. Keskmise videopildi taastumise aeg pärast pääsupunkti vahetust oli 7,37 sekundit. Üldiselt toimis režiimide vahetus loogikapõhiselt ja ootuspäraselt, kuid üksikutel juhtudel esines lühiajalisi korduvaid ümberlülitusi, mis viitavad tundlikule otsustuskünnisele. Sellest hoolimata võib järeldada, et välja töötatud süsteem on sobilik teleoperatsiooni ja autonoomse juhtimisloogika töökindluse hindamiseks mitme Wi-Fi pääsupunktiga keskkonnas.

Viited

- [1] U. Petrović, „Small-scale cars for autonomous driving research“, 2023.
- [2] İ. Karaböcek, B. Kavak, ja E. Özdemir, „Remote Control of ADAS Features: A Teleoperation Approach to Mitigate Autonomous Driving Challenges“, *Engineering Proceedings*, kd 82, nr 1, Art. nr 1, 2024, doi: 10.3390/ecsa-11-20449.
- [3] „Donkey Car“. Vaadatud: 31. märts 2025. <https://docs.donkeycar.com/>
- [4] H.-M. Huang, „Terminology for specifying the autonomy levels for unmanned systems: version 1.0“.
- [5] A. Schimpe ja F. Diermeyer, „Steer with Me: A Predictive, Potential Field-Based Control Approach for Semi-Autonomous, Teleoperated Road Vehicles“, *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, sept 2020, lk 1–6. doi: 10.1109/ITSC45102.2020.9294702.
- [6] „J3016_202104: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International“. Vaadatud: 14. mai 2025. https://www.sae.org/standards/content/j3016_202104/
- [7] „SAE Levels of Driving Automation™ Refined for Clarity and International Audience“. Vaadatud: 3. märts 2025. <https://www.sae.org/site/blog/sae-j3016-update>
- [8] C. Team, „The Road to Autonomous Vehicles: How Close Are We to Full Self-Driving?“, carstyle. Vaadatud: 14. mai 2025. <https://carstyle.au/the-road-to-autonomous-vehicles-how-close-are-we-to-full-self-driving/>
- [9] A. Podhurst, „The complete guide to autonomous vehicle teleoperation“, DriveU. Vaadatud: 10. märts 2025. <https://driveu.auto/blog/the-complete-guide-to-av-teleoperation/>
- [10] S. Lu, R. Zhong, ja W. Shi, „Teleoperation Technologies for Enhancing Connected and Autonomous Vehicles“, *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, Denver, CO, USA: IEEE, okt 2022, lk 435–443. doi: 10.1109/MASS56207.2022.00068.
- [11] D. Elinav, „Teleoperation: cellular coverage and why one modem is not enough“, DriveU. Vaadatud: 10. märts 2025. <https://driveu.auto/blog/why-one-modem-is-not-enough-for-teleoperation-of-robots-and-autonomous-vehicles/>

- [12] N. Goodall, „Non-technological challenges for the remote operation of automated vehicles“, *Transportation Research Part A: Policy and Practice*, kd 142, lk 14–26, dets 2020, doi: 10.1016/j.tra.2020.09.024.
- [13] N. Grabbe, S. Brettschneider, ja K. Bengler, „The work organization in teleoperated driving – a human reliability and socio-technical systems perspective“, *at - Automatisierungstechnik*, kd 72, nr 12, lk 1169–1194, dets 2024, doi: 10.1515/auto-2024-0089.
- [14] K. Huang *et al.*, „Researches on A Tele-operated Robot Based on Wi-Fi“, *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, märts 2019, lk 648–651. doi: 10.1109/ITNEC.2019.8729472.
- [15] S. B. Kamtam, Q. Lu, F. Bouali, O. C. L. Haas, ja S. Birrell, „Network Latency in Teleoperation of Connected and Autonomous Vehicles: A Review of Trends, Challenges, and Mitigation Strategies“, *Sensors*, kd 24, nr 12, Art. nr 12, jaan 2024, doi: 10.3390/s24123957.
- [16] „The Common Challenges and Limitations of WiFi Networks | actcorp“. Vaadatud: 12. märts 2025. <https://www.actcorp.in/blog/common-challenges-and-limitations-wifi-networks>
- [17] „Campus Wi-Fi Coverage Mapping and Analysis“, ResearchGate. Vaadatud: 10. märts 2025. https://www.researchgate.net/publication/340452569_Campus_Wi-Fi_Coverage_Mapping_and_Analysis
- [18] „IEEE 802.11 Handover — INET 4.5.4 documentation“. Vaadatud: 14. mai 2025. <https://inet.omnetpp.org/docs/showcases/wireless/handover/doc/index.html>
- [19] S. Verma, T. K. Rodrigues, Y. Kawamoto, M. M. Fouda, ja N. Kato, „A Survey on Multi-AP Coordination Approaches over Emerging WLANs: Future Directions and Open Challenges“, *IEEE Commun. Surv. Tutorials*, kd 26, nr 2, lk 858–889, 2024, doi: 10.1109/COMST.2023.3344167.
- [20] M. B. E. Zarrouki, V. Klös, M. Grabowski, ja S. Glesner, „Fault-Tolerance by Graceful Degradation for Car Platoons“, *OASICS, Volume 68, ASD 2019*, kd 68, lk 1:1-1:15, 2019, doi: 10.4230/OASICS.ASD.2019.1.
- [21] O. Akbarzadeh ja H. Attar, „Investigating The Effect of Latency in A 5G Infrastructure for Applications Requiring Ultra-Low Latency“, *2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, Zarqa, Jordan: IEEE, nov 2022, lk 1–5. doi: 10.1109/EICEEAI56378.2022.10050470.

- [22] A. Slalmi, H. Chaibi, A. Chehri, R. Saadane, G. Jeon, ja N. Hakem, „On the Ultra-Reliable and Low-Latency Communications for Tactile Internet in 5G Era“, *Procedia Computer Science*, kd 176, lk 3853–3862, 2020, doi: 10.1016/j.procs.2020.09.003.
- [23] „HTTP vs WebSocket: Real-Time Web Communication Guide“. Vaadatud: 14. märts 2025. <https://www.digitalsamba.com/blog/websocket-vs-http>
- [24] L. Srinivasan, J. Scharnagl, ja K. Schilling, „Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation“, *IFAC Proceedings Volumes*, kd 46, nr 29, lk 83–88, jaan 2013, doi: 10.3182/20131111-3-KR-2043.00032.
- [25] „MQTT vs WebSocket - Which protocol to use when in 2024“, Aply Realtime. Vaadatud: 14. märts 2025. <https://ably.com/topic/mqtt-vs-websocket>
- [26] E. Kamrani, „Real-Time Internet-Based Teleoperation“, *Intelligent Control and Automation*, kd 3, nr 4, Art. nr 4, nov 2012, doi: 10.4236/ica.2012.34041.
- [27] „H.264 vs. MJPEG in Embedded Vision – All You Need to Know“, TechNexion. Vaadatud: 15. märts 2025. <https://www.technexion.com/resources/h-264-vs-mjpeg-in-embedded-vision-all-you-need-to-know/>
- [28] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, ja T. Wiegand, „Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)“, *IEEE Transactions on Circuits and Systems for Video Technology*, kd 22, nr 12, lk 1669–1684, dets 2012, doi: 10.1109/TCSVT.2012.2221192.
- [29] C. Gregersen, „Guide to WebRTC vs. RTSP Video Streaming Protocols“, Nabto. Vaadatud: 15. märts 2025. <https://www.nabto.com/webrtc-vs-rtsp/>
- [30] Y. Li, N. Raison, S. Ourselin, T. Mahmoodi, P. Dasgupta, ja A. Granados, „AI solutions for overcoming delays in telesurgery and telementoring to enhance surgical practice and education“, *J Robot Surg*, kd 18, nr 1, lk 403, 2024, doi: 10.1007/s11701-024-02153-9.
- [31] J. Corredor, J. Sofrony, ja A. Peer, „Decision-Making Model for Adaptive Impedance Control of Teleoperation Systems“, *IEEE Transactions on Haptics*, kd 10, nr 1, lk 5–16, jaan 2017, doi: 10.1109/TOH.2016.2581807.
- [32] N. Feizi, M. Tavakoli, R. V. Patel, ja S. F. Atashzar, „Robotics and AI for Teleoperation, Tele-Assessment, and Tele-Training for Surgery in the Era of COVID-19:

Existing Challenges, and Future Vision“, *Front Robot AI*, kd 8, lk 610677, apr 2021, doi: 10.3389/frobt.2021.610677.

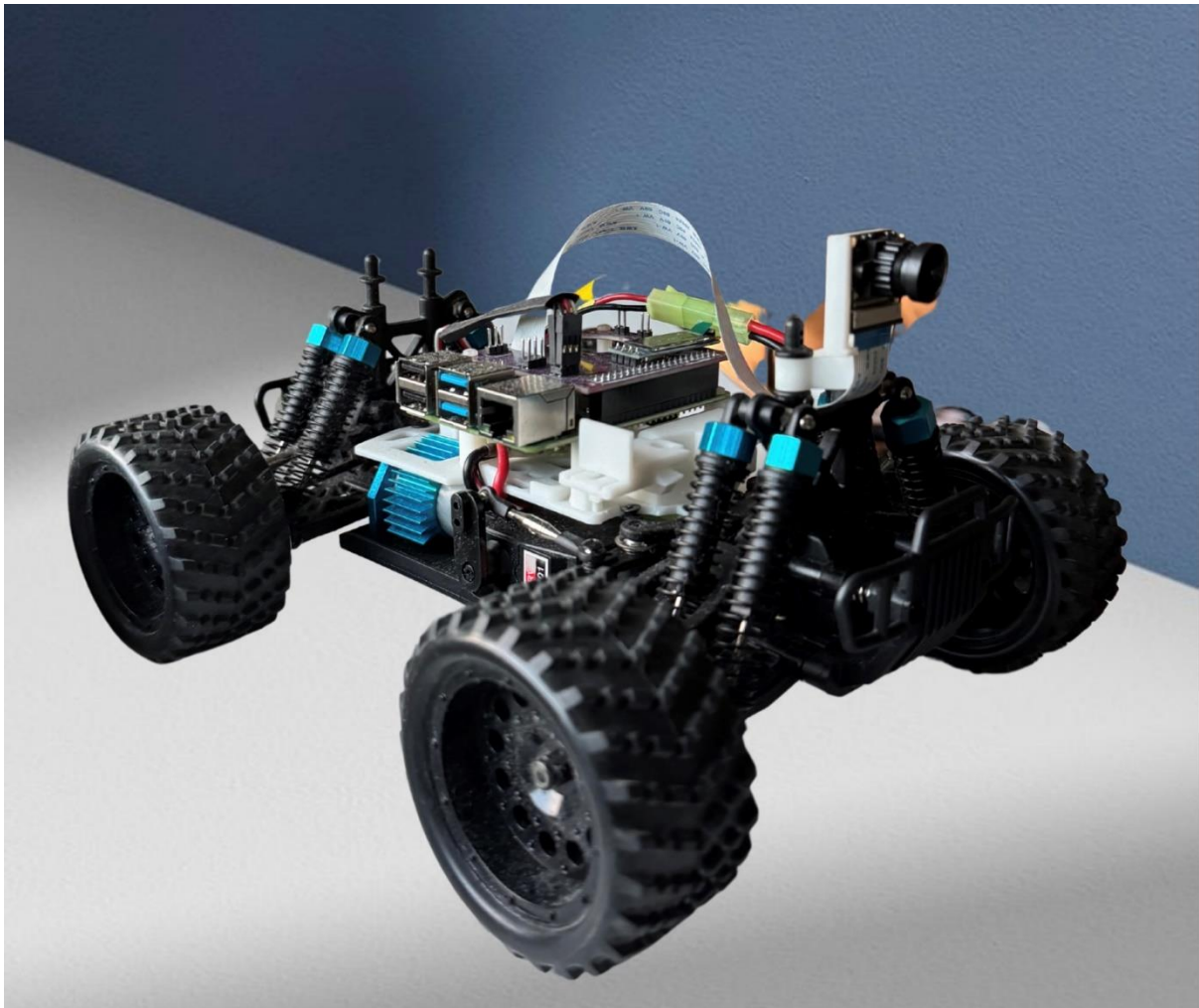
[33] J. Bohren, K. Guerin, T. Xia, G. D. Hager, P. Kazanzides, ja L. L. Whitcomb, „Toward practical semi-autonomous teleoperation: Do what i intend, not what i do“, *Advanced Robotics and its Social Impacts*, okt 2011, lk 20–23. doi: 10.1109/ARSO.2011.6301974.

[34] M. Rubagotti, T. Taunyazov, B. Omarali, ja A. Shintemirov, „Semi-Autonomous Robot Teleoperation With Obstacle Avoidance via Model Predictive Control“, *IEEE Robotics and Automation Letters*, kd 4, nr 3, lk 2746–2753, juuli 2019, doi: 10.1109/LRA.2019.2917707.

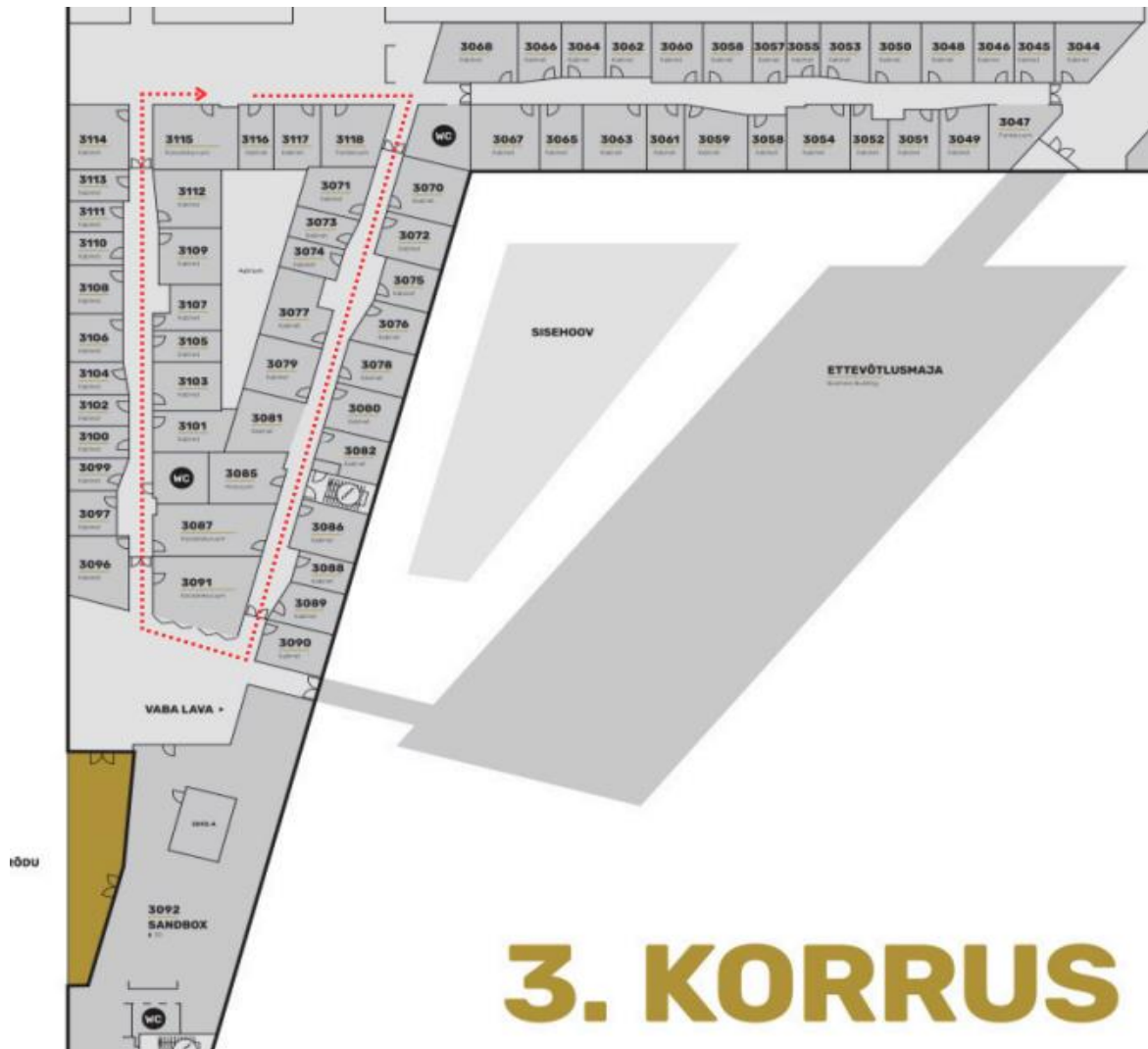
[35] R. Harris, „Virtual DOM is pure overhead“. Vaadatud: 22. aprill 2025. <https://svelte.dev/blog/virtual-dom-is-pure-overhead>

Lisad

I. Väiksemõõtmeline kaugjuhitav sõiduk



II. Testraja asukoht Delta õppehoone kolmandal korrusel



III. Kasutajaliidese paigaldamise juhend

Järgnevalt on välja toodud kasutajaliidese paigaldamise juhend:

1. Veenduge, arvutis on olemas Node.js (katsetatud versiooniga 21) ja Git
2. Laadige koodihoiglast alla lähtekood käsuga:
 - a. `git clone https://github.com/MarkkusKoddala/donkey-teleop-ui.git`
3. Liikuge käsureal alla laetud projekti kausta:
 - a. `cd donkey-teleop-ui`
4. Paigalde projekti sõltuvused
 - a. `npm install`
5. Looge fail `.env` ning lisage sinna järgmine muutuja:
 - a. `PUBLIC_CAR_IP=xxx.xxx.xx.xxx`
6. Käivitage arendusserver järgmiseks äsuga:
 - a. `npm run dev`

IV. Sõidukipoolse rakenduse paigaldamise juhend

Järgnevalt on välja toodud sõidukipoolse rakenduse paigaldamise juhend. Juhend on koostatud ja testitud järgmistel tingimustel: Raspberry Pi 4 Model B Rev 1.5, Raspberry Pi OS (Debian 11 "Bullseye"), Python 3.9.2 ja DonkeyCar v5.0.0.

1. Laadige koodihoidlast alla lähtekood käsuga:
 - a. `git clone https://github.com/MarkkusKoddala/donkey-teleop-controller.git`
2. Looge sõiduki projekt ja liikuge projekti kausta:
 - a. `donkeycar createcar --path ./donkey-teleop-controller/`
 - b. `cd donkey-teleop-controller`
3. Paigaldage projekti sõltuvused:
 - a. `pip install -r requirements.txt`
4. Asendage konfiguratsioonifail järgmise käsuga:
 - a. `cp my_preconfig.py myconfig.py`
5. Muutke `manage.py` faili, lisades funktsiooni `add_user_controller` algusesse järgmine kood:

```
if cfg.USE_CUSTOM_CONTROLLER:
    from core.teleop_control_part import TeleopControlPart
    controller = TeleopControlPart(cfg)
    V.add(
        controller,
        inputs=['cam/image_array'],
        outputs=['user/angle', 'user/throttle', 'user/mode',
'recording', 'cam/image_array'],
        threaded=True
    )
    return controller
```

6. Käivitage auto käsuga (ühtegi mudelit ei rakendada):
 - a. `python manage.py drive`
7. (Valikuline) Kasutage kaustas `resources` olevat eelnevalt treenitud mudelit:
 - a. `python manage.py drive --model=resources/delta_model.h5`

V. Lahenduse demontsratsioon katseringil

Järgnevalt on viidatud videole, mis demonstreerib väikesemõõtmelise mänguauto liikumist katseringil kaugjuhtimissüsteemi abil. Video näitab, kuidas kasutaja annab käske veebipõhise liidese kaudu ning sõiduk reageerib reaalsajas vastavalt. Video asub lingil: https://drive.google.com/file/d/1kmvpCJD3vnSd_HNt6E-dSlr3ioGfKNCu/view?usp=sharing

VI. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Markkus Koddala,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Tartu Ülikooli isejuhtivate sõidukite labori väikesemõõtmelise auto kaugjuhtimise arendus“, mille juhendaja on Ardi Tampuu, 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 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.

Markkus Koddala

15.05.2025