

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**BEZPILOTA MOTORIZĒTAS IEKĀRTAS
ORIENTĒŠANĀS TĒLPĀ UN OPTIMĀLĀ
MARŠRUTA IZVĒLE**

BAKALaura DARBS

Autors: **Gustavs Šenne**
Studenta apliecības nr.: gs12027
Darba vadītājs: Dr.dat. Reinholds Zviedris

RĪGA 2016

ANOTĀCIJA

Šajā darbā tiek apskatīta un risināta austoša problēma ar autonomu “robotu” orientēšanos telpā, bez iepriekšējām priekšzināšanām par to, optimālu maršruta izvēli, kā arī risinājums, kā to izveidot neatkarīgi no ārējiem resursiem, piemēram, globālās pozicionēšanas sistēmām, mobilo datu pieslēguma datiem, bezvadu tīklu vai radio frekvenču identifikāciju. Risinājums veidots kā vairāku sensoru kombinācija, iekļaujot sevī 10 dimensiju jeb DOF, ultraskaņas un vibrāciju sensorus. Rezultātā ir iegūta sistēmas koncepcija, kura pati spēj autonomi apstaigāt lauku, paralēli veidojot apstaigātās platības karti.

Atslēgvārdi: INS, SLAM, DOF, pilnīga lauka apceļošana, MEMS, zemu izmaksu sensori.

ABSTRACT

Unmanned motorized machine orientation in space and optimal route selection

This research examines the rising problem of the orientation of autonomous robots in a space without any prior knowledge of it, an optimal choice of path, how to solve this problem without any external help, for example from satellites, mobile data, wireless sensors or radio frequency identification. The solution is made of a combination of sensors, including 10 dimension or DOF, ultrasound and vibration sensors. The result is a concept of a system which can autonomously traverse a space while building a map of said space.

Keywords: INS, SLAM, DOF, complete field traversal, MEMS, low-cost sensors.

SATURS

APZĪMĒJUMI.....	4
IEVADS.....	5
1. PAŠREIZĒJĀ SITUĀCIJA / LĪDZĪGI DARBI.....	7
2. PIELIETOTĀS IEKĀRTAS / SENSORI.....	11
2.1. Ultraskaņa.....	11
2.2. Žiroskops un akselerometrs.....	12
2.3. Magnetometrs.....	13
2.4. Barometrs.....	14
3. SENSORU KĻŪDU APSTRĀDE.....	15
3.1. Kalmana filtrs.....	17
3.2. Komplimentārais filtrs.....	19
3.3. Digitālā kustību apstrāde.....	19
3.4. Darba ietvaros izpētītās žiroskopa kļūdu novēršanas metodes.....	20
3.5. Darba ietvaros izpētītās akselerometra kļūdu novēršanas metodes.....	22
4. TELPAS APCEĻOŠANA.....	26
4.1. Kartēšana (Mapbuilding).....	26
4.2. Ceļa atrašana (Pathfinding).....	27
REZULTĀTI.....	29
ALTERNATĪVI RISINĀJUMI / NĀKOTNES PLĀNI.....	30
SECINĀJUMI.....	31
IZMANTOTĀ LITERATŪRA UN AVOTI.....	32

APZĪMĒJUMI

DOF - Degrees of Freedom - brīvības pakāpes ir veids, kā apraksta cik iekārtai vai sensoram ir dimensijas, kurās orientēties, piemēram, 6-DOF parasti apzīmē 3 asu žiroskopa un 3 asu magnetometra kombināciju. 6-DOF neapzīmē noteikti akcelerometra un žiroskopa kombināciju - tā var būt jebkādu sensoru kombinācija, kuru kopējais asu skaits ir 6.

DR - Dead Reckoning - pašreizējās pozīcijas noteikšana balstoties uz sākotnēji uzstādītu sākuma punktu un pārvietošanos laikā un telpā no šī punkta.

RISS - Reduced Inertial Sensor System - reducētā inerciālo sensoru sistēma ir kāda 3 asu navigācijas sistēma, samazināta versija, kur tiek izmantota daļa no pieejamajām brīvības pakāpēm.

IMU - Inertial Measurement Units - inerciālās mērvienības ir mērvienības, ko iegūst no inerciālajiem sensoriem, no žiroskopa - leņķiskais paātrinājums, no akcelerometra - paātrinājums.

INS - Inertial Navigation System - inerciālās navigācijas sistēma ir uz inerciāliem sensoriem jeb inerciālajām mērvienībām balstīta sistēma

GPS - Globālā Pozicionēšanas Sistēma ir sistēma, kurā ar satelītu palīdzību var iegūt savu atrašanās vietu apzīmētu ar koordinātēm uz zemes "fona".

WIFI - Wireless Fidelity - bezvadu tīklājs, ir tīkla savienojums, kuru nodrošina starp divām iekārtām specifiskā frekvencē raidīti radioviļņi.

Drons jeb UAV - Unmanned Aerial Vehicle - cilvēka-nevadīts gaisa transportlīdzeklis.

USB - Universal Serial Bus - universāls savienojuma veids, kuru izmanto, lai nodrošinātu datu savienojumus ar dažādām iekārtām.

HPF - High pass filter - augstas caurlaidības filtrs ir filtrs, pēc kura ieviešanas no tam padotām vērtībām tas atgriež vērtības, kas ir augstākas par konkrēti norādītu sliekšni.

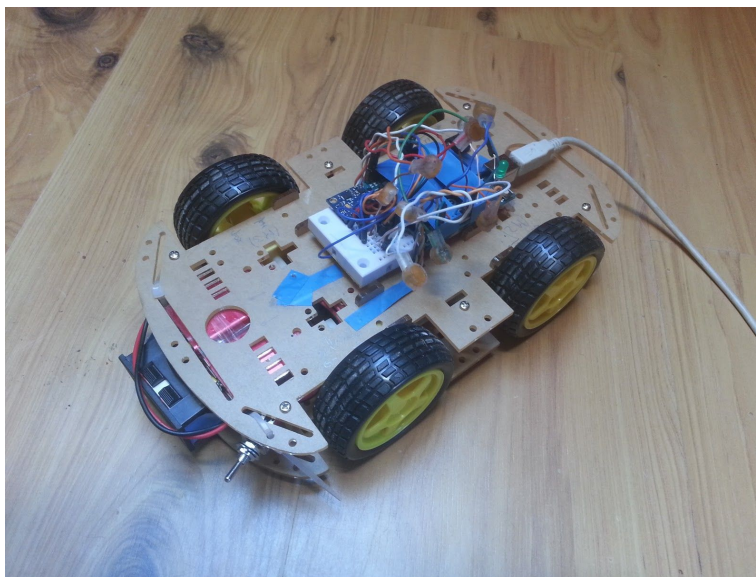
LPF - Low pass filter - zemas caurlaidības filtrs ir filtrs ar tādu pašu principu kā augstās caurlaidības filtrs, tikai atgriež vērtības, kas ir zemākas par norādīto sliekšni.

IEVADS

Jau no laiku sākuma cilvēkiem ir bijusi nepieciešamība kaut ko darīt. Vai tas būtu strādāt darbā, lai ar to nopelnītu iztiku, vai tas būtu mājas vides uzturēšana kārtībā, vairumā gadījumu ar to saprotot pamata vajadzību nodrošināšanu, piemēram, uzkopšanu vai ēst gatavošanu. Nokļūšana līdz darbam pat prasa uzmanību un enerģiju, vai tā būtu iešana vai braukšana ar savu auto. Tā visa pamatā ir darbs. Lai darbu veiktu cilvēkam ir jāpieliek pūles un uzmanība, kas, savukārt, patērē enerģiju, kuru varētu patērēt krietni noderīgāk.

Galvenā doma būtu netērēt enerģiju lietām, kuras nepatīk darīt. Lai 'kāds' varētu paveikt darbu, kas pašam nepieciešams, tad līdz šim tam ir bijuši nepieciešami ārēji līdzekļi kā, piemēram, nauda, lai samaksātu citam par šī darba paveikšanu.

Tas neattiecas tikai uz subjektīvu darbu darīšanu. Bieži vien pētnieciskos nolūkos ir nepieciešams doties bīstamās vietās, piemēram dziļās alās, kur ir skābekļa trūkums vai temperatūras pārsniedz cilvēka spējas. Mūsdienās arī lēnām pieaugošāka problēma ir katastrofu seku novēršana, piemēram, Fukušimas atomelektrostacijas avārija, kur dēļ paaugstinātās radiācijas noplūdes cilvēkiem bez īpaša aprīkojuma pieeja vairs nebija iespējama.



1.att. Radītā iekārta

Līdz ar tehnoloģiju attīstību kļūst aktuāls jautājums par cilvēku iesaisti daudzās sfērās. T.i. padarīt darbus, ko cilvēki dara ikdienā, autonomus, neatkarīgus no cilvēka. Saistībā ar šo plašo tēmu, pasaulē jau notiek daudz un dažādī pētījumi un tiek strādāts pie tā lai ar vien

vairāk darbus varētu automatizēt. Plašāk par šiem risinājumiem ir rakstīts nākošajā tēmā.

Šī darba mērķis ir izpētīt autonomu iekārtu orientēšanās iespēju robežas, izmantojot zemu izmaksu sensorus, kā arī kādas ir iespējas iekārtām orientēties telpā, neizmantojot tādas ārējus resursus, kā globālās pozicionēšanas sistēmas. Paredzēts aplūkot, kādas ir atsevišķu sensoru un iekārtu mērījumu iespējas un precizitāte, kā arī izveidot universālu risinājuma modeli, kuru varētu izmantot kā pamatu izstrādājot līdzīgas autonomas iekārtas.

1. PAŠREIZĒJĀ SITUĀCIJA / LĪDZĪGI DARBI

Izpēte tika sākta ar sabiedrībā nesen darītiem pieejamiem risinājumiem, kas guvuši augstu atpazīstamību un ir aktuālas tēmas ne tikai konkrēto sfēru pētniekiem, bet arī sabiedrībai, kas no tā varētu gūt vislielāko labumu.

- Google self-driving car project [1] - projekts, ko šobrīd aktīvi veic plaši pazīstamā Google kompānija. Aprīkojot automašīnu ar lāzeriem, radariem, kamerām un citiem sensoriem, tā spēj noteikt kur tā atrodas un kas atrodas tai apkārt. Ar programmatūras palīdzību tā cenšas paredzēt esošo situāciju, ņemot vērā savu stāvokli un apkārtējos objektus, piemēram cilvēkus, riteņbraucējus un citas automašīnas, lai neizraisītu negadījumus. Ar šīm automašīnām autopilotā jau ir nobraukti 1,5 miljoni jūdžu.

- Autonomo automašīnu projektu ir krasi uzsākusi attīstīt arī autokompānija 'Tesla'. Līdz ar tās šī gada automašīnas modeli 'Model S', kompānija ir nākusi klajā ar daudz iespējām, piemēram, 'Autosteer', 'Autopark' [2] u.c. Kopumā kompānija lēš, ka ar tās automašīnām, autopilotā ir nobraukti jau pāri 100 miljoniem jūdžu.

- Zāles pļaušana mūsdienās arī kļūst par problēmu, ko var atrisināt bez cilvēka iesaistes. Kompānija Husqvarna [3], tāpat kā dažas citas kompānijas, kas nodarbojas ar dārzniecību un tai paredzēto iekārtu izveidi, ir izveidojušas autonomos zāles pļāvējus, kas ļauj pļaut zāli, pat neatrodoties tuvumā mājai.

- Tepat pie kaimiņiem ziemeļos, kompānija Starship Technologies ir izveidojusi autonomu kurjeru, kurš spēj piegādāt iepakojumus līdz pat durvīm [4]. Tam ir pilnīga un konstanta izsekošana, t.i. tiek uzskaitītas visas atrašanās vietas, kur tas atrodas un ir bijis.

- iRobot korporācijai ir vairāki robotiski risinājumi ikdienišķiem darbiem, piemēram grīdu sūknēšana, mazgāšana, baseinu tīrīšana [5] u.c.

- Šogad arī pasaule tika iepazīstināta ar OceanOne - Stenfordas universitātē izstrādātu un izveidotu bimanuālu zemūdens humanoīdu robotu [6]. Aprīkots ar atgriezenisko atsauksmi (haptic feedback), kas atgriež lietotājam kādu "sajūju" par robota stāvokli, OceanOne šī gada aprīlī, apmeklēja vidusjūrā grimuša kuģa vraku, kuru neviens cilvēks līdz šim nebija apmeklējis un no tā izcēla greipfrūta lieluma vāzi.

- Tāda korporācija kā Amazon.com, piedāvā piegādāt saviem klientiem preces ļoti īsā laikā ar bezpilota lidaparātiem jeb droniem. Šie lidaparāti lido augstumā, kas nepārsniedz 400

pēdas, ar kravnesību 5 mārciņas, tie solās piegādāt iepakojumus 30 minūtēs vai mazāk [7].

Iepriekš minētajām automašīnām un kurjeriem svarīgi ir zināt savu globālo atrašanās vietu, to nosakot, galvenokārt, ar globālajām pozicionēšanas sistēmām, līdz ar to tas nav pielietojams tādām iekārtām, kas ceļo pa vidi kurā nav pieejams GPS, piemēram zem ūdens vai alās.

Tas pats attiecas arī uz autonomajiem zāles pļāvējiem, bet tiem nav tik kritiski zināt savu globālo atrašanās vietu. To neņemot vērā, tiem svarīgi ir zināt savu relatīvo atrašanās vietu no sākuma punkta, un cik daudz tie ir nopļāvuši. Atrašanās vietas problēmu, līdz šim apskatītie pļāvēji risina ar stieples, jeb vada ierakšanu ap pļaujamo teritoriju, kas ir optimāls risinājums, ja grib automatizēt zināma lauka apstaigāšanu ilgtermiņā. Šis risinājums, savukārt, noteikti neatbilst autonomizācijai nezināmā vidē.

iRobotiem, kas ir ļoti līdzīgi savā pārvietošanās principā, nav tik kritiski nepieciešams zināt savu atrašanās vietu. Risinājumi, kas ar savu zemo izmaksu ir ieguvuši salīdzinoši daudz lielāku popularitāti, nav paši efektīvākie. Pēc kādas vietnes salīdzinājuma starp vairākiem tīrītājiem [8], un veiklas šo robotu cenas izpētes interneta veikalā Amazon.com var secināt, ka tie roboti kuri tīra ātrāk, maksā arī dārgāk. Konkrētajā gadījumā robots, kurš tīrīja tikai 3 reizes ātrāk un neapstaigāja lauku vairāk kā dažas reizes, maksāja 2 reizes dārgāk.

Visbeidzot, OceanOne risinājums, kas pats par sevi ir ļoti inovatīvs risinājums, kas agrāk bija tikai zinātniskā fantastika, nu ir realizēts. Ar visu tā inovativitāti, to ir iespējams attīstīt tālāk. Šim robotam pirmkārt nav pilnīga autonomija, nedz arī tas var pilnīgi brīvi pārvietoties, jo tam ir nepieciešams savienojums ar bāzes staciju un vadītāju.

Var secināt, ka ir daudz līdzīgi risinājumi, bet neviens nebalstās tikai uz sensoriem kuri iegūst informāciju no tūlītējās apkārtējās vides (t.i. Neizmanto ārējus resursus, kā GPS, mobilie dati vai Wi-Fi). Šajā nodaļā tika apskatīti un aprakstīti daži pētījumi un risinājumi līdzīgām problēmām.

- 1) Magicol [9] - Galvenokārt balstīts uz zemes ģeomagnētiskā lauka, uz kura pamata strādā arī parastie kompasī. Konkrētā ēkā, apdzīvotā vidē apstaigājot tika veikti magnetometra lasījumi. Iegūtie rezultāti tika analizēti, kā arī salīdzināti ar DR jeb IMU sistēmu, izveidojot sistēmu, kura tika analizēta citā pētījumā. Balstoties uz šiem iegūtajiem datiem vienā no konkrētajām eksperimentu vidēm - birojā - 80% gadījumu varēja noteikt atrašanās vietu tikai ar 4m kļūdu, salīdzinot ar 9,5m kļūdu kas bija DR sistēmā uz kuru šis pētījums atsaucās.

- 2) Inertial Navigation Systems for Mobile Robots [10] - Šis pētījums ir vislīdzīgākais. Te tika veikti mērījumi ar un bez paplašinātā kalmana filtra. Uzskatāmākais, kas šeit tiek aptverts ir sensoru kļūdas - žiroskopam - bez kļūdu kompensēšanas rodas 5-15° nobīde minūtē, ar kompensēšanu šo nobīdi var samazināt par 5 grādiem, ja tiek dots piemērots kļūdu modelis, savukārt - akselerometram (jeb attāluma noteikšanai) - 1-8cm sekundē nobīde. Šī kritiski lielā kļūda akselerometram rodas dēļ tā, ka, lai iegūtu attālumu no paātrinājuma, ir nepieciešams divreiz integrēt šos mērījumus. Minētais pētījums tika veikts uz konkrētas sistēmas, kur iekārtas ātrums bija līdz 21 km/h. Šī ir būtiska atšķirība, kas tiks ņemta vērā risinot šajā darbā izvirzītās problēmas.
- 3) Mobile Robot Localization Using Sonar [11] - Tika apskatīts arī risinājums, kā ar ultraskaņas sensoru var noteikt orientāciju telpā. Šāds princips netiek lietots, jo uz iekārtas būtu jānovieto vairāki ultraskaņas sensori un jāpārklāj viss rādiuss apkārt iekārtai, līdz ar to tas būtu pārāk laikietilpīgi, lai būtu lietderīgi.
- 4) Lightweight Map Matching for Indoor Localisation using Conditional Random Fields [12] - Izmantojot dotu telpu plānu apvienojumā ar wifi signāliem, no kuriem nevienu manā gadījumā nav plānots pielietot. Tiek apskatīta arī karšu zīmēšana kurai ir lielāka lietderība manā konkrētajā gadījumā. Kā tā nosaukumā arī minēts tika pētīta karšu atbilstība ar kondicionālajiem nejaušajiem laukiem. Diezgan plašs pētījums, jāapskata atkārtoti.
- 5) Digitālā audio apstrāde un efektu sintēze pielietojot bezvadu sensorus [13] - Šajā pētījumā vienā no tēmām ir aprakstīta līdzīga problēma un tās risinājums. Konkrētāk tiek pētīta inerciālās navigācijas sistēma lai orientētos telpā, apskatītas kļūdas jeb nobīdes kas rodas sensoros, aprakstīts kalmana filtrs ar tā darbības principiem, kā arī realizēts sensoru datu saplūdumu algoritms.
- 6) An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter [14] - Pētījums veikts uz "helikoptera" (rotoriem bāzēta lidaparāta), kur apskatīja tā autonomu vadību ar GPS, MEMS-INS, kā arī pāreju starp šīm abām sistēmām. Uz INS tika pielietots sešu stāvokļu kalmana filtrs.
- 7) Mobile Robot Navigation Using Active Vision [15] - Neņemot vērā šī darba nosaukumu tajā bija aprakstīts SLAM, karšu veidošana, kā arī kalmana filtrēšana. Autonomā navigācija arī tika aprunāta, konkrētāk - automātiskā pozīcijas bāzētā navigācija, arī apvienota ar karšu būvēšanu.

- 8) Robust Mobile Robot Localisation from Sparse and Noisy Proximity Readings using Hough Transform and Probability Grids [16] - Darbs ar mērķi izveidot mazu-izmaksu navigācijas sistēmu universālam lietojumam. Šajā metodē gan tika izmantota jau dota telpas karte, kur pretī manā darbā šī karte tiks veidota. Tiks sīkāk apskatīts, kā notiek kartes zīmēšana un, iespējams, kas līdzīgs tiks ieviests būvējot manis pētītās problēmas karti.
- 9) Simultaneous Map Building and Localization for an Autonomous Mobile Robot [17] - Pētījumā realizēta SLAM ar attāluma sensoru palīdzību. Attāluma sensori novietoti uz servo motora, tādējādi dodot iespēju noklāt lielāku telpu. Tas prasa papildu laiku lai noskenētu, līdz ar to, tas nav optimāli. Apskatīta arī reālā laika karšu veidošana.
- 10) Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter [18] - Aprakstīts kalmana un papildinošā filtra dizains, lai apvienotu dažāda veida sensorus vienā inerciālās sistēmas dinamiskā reakcijā. Šī ir svarīga puse, ko nepieciešams apskatīt, lai neveidotos lielas nobīdes iegūtajos datos un palielinātu vispārējo sistēmas lokalizācijas precizitāti.

2. PIELIETOTĀS IEKĀRTAS / SENSORI

Lai veiktu autonomu iekārtas atrašanās vietas noteikšanu, no tās sākuma pozīcijas, un lai tā spētu veiksmīgi atgriezties šajā pozīcijā, ir nepieciešams to aprīkot ar sensoriem kuri spēs reģistrēt sekojošus aspektus: Žiroskops - leņķi; Akselerometrs - paātrinājumu; Magnetometrs - stāvokli attiecībā pret ziemeļiem; Ultraskaņa - tuvumā esošos objektus; Barometrs - augstumu.

Šāda sensoru kombinācija tika izvēlēta, jo tā dod iekārtai 11-dof, jeb 11 brīvības pakāpes. Lielākā daļa šo sensoru jau tiek veidoti kombinācijās cits ar citu, līdz ar to tiek arī samazināta to izmaksa. Modulis gy-87 ir uzskatāmākais piemērs, kas arī tiek lietots šajā darbā. Tas sevī iekļauj akselerometru, žiroskopu, magnetometru un barometru, kas kopumā dod 10 brīvības pakāpes uz 2x1,7cm lielas plates.

2.1. Ultraskaņa

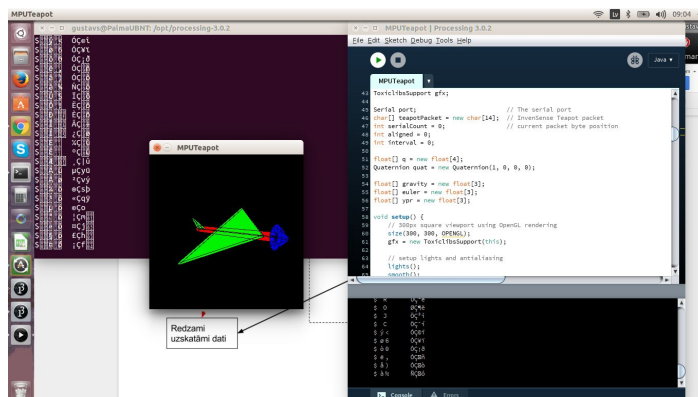
Attāluma noteikšanai tiek izmantots ultraskaņas sensors HC-SR04. Raidot ultraskaņas viļni un pēc tam klausoties šī viļņa "atbalsi", tiek noteikts laiks kurā šis vilnis atbalsojas un izmantojot skaņas ātruma konstanti $\sim 340\text{m/s}$, ar šo laiku aprēķina attālumu līdz objektam/-iem no kura šī skaņa atbalsojās.

Konkrēti šī sensora tuvākais attālums kurā tas var noteikt attālumu ir 2cm un lielākais 4m. Ņemot vērā, ka autonomā iekārta kura tiks aprīkota ar šo sensoru nepārsniegs ātrumu lielāku par 1m/s, tad ar šādu sensora lielāko attālumu pietiks gadījumā, ja būs nepieciešams nekavējoties apstāties.

Ar šī sensora palīdzību arī tiks noteikti tuvumā esoši objekti. Tas būs nepieciešams, ja pie īsākā ceļa izvēles plānošanas pieļaus iespēju ceļot pāri lauka daļai kura iepriekš nav bijusi apstaigāta, kā arī gadījumos, kad izvēlēties ceļot pa apstaigāto daļu, bet uz tās būs novietots šķērslis.

2.2. Žiroskops un akcelerometrs

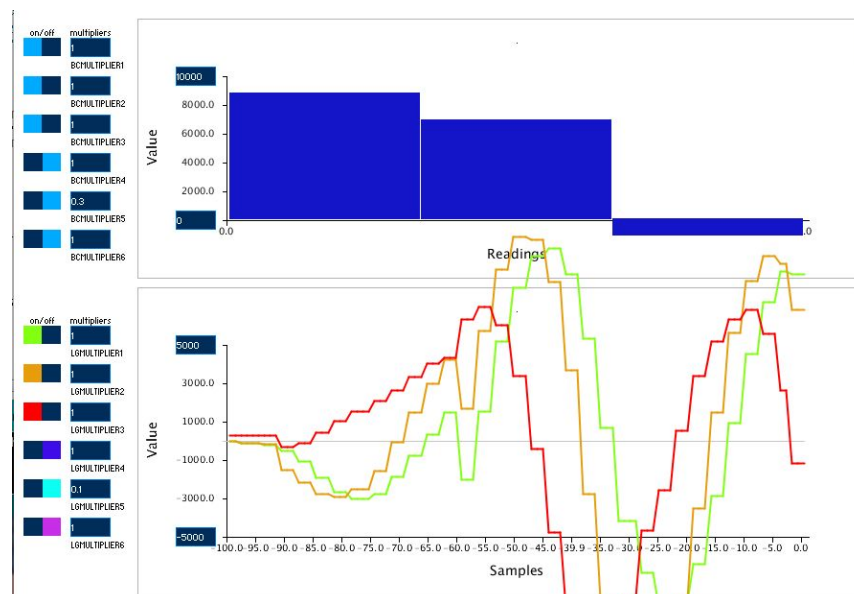
Leņķa stāvokļa un paātrinājuma mērīšanai tiks izmantots sensoru modulis GY-521 un GY-87, abi ar MPU6050 čipiem. Šie konkrētie čipi tika veidoti priekš viedtālrunu un citu iekārtu, neliela enerģijas patēriņa, nelielu izmaksu un augsta snieguma prasībām. Šim sensoram ir vairāki darbības veidi: no tā var nolasīt jēlus datus (raw data) kas, atkarībā pie uzstādīta augstākā jūtīguma var būt robežās +/- 16384; var nolasīt arī no DMP bufera, kur dati tiek apstrādāti, ņemot vērā leņķisko paātrinājumu no žiroskopa; ir iespēja arī mainīt, vai attēlo paātrinājumu relatīvi no sākuma pozīcijas, vai tikai no esošā stāvokļa. Konkrētā sensora specifikācijā ir norādīts, ka ir iespēja mainīt tā jūtīgumu, akcelerometram tas var būt viens no 4 veidiem: +/-2g, +/-4g, +/-8g un +/-16g, kur 2g ir visjūtīgākais un 16g vismazgāk jūtīgs. ‘g’ šajā gadījumā ir gravitācijas paātrinājums kas atbilst $1g = 9.80665m/s^2$. Šobrīd veiktajos pētījumos ir minēts, ka žiroskopam rodas nobīde, ja to lieto ilgāk par ~25 minūtēm, savukārt akcelerometram dažās sekundēs rodas nobīde. Tiks izpētīts vai ar atbilstošu kalmana filtru ir iespējams samazināt vai atbrīvoties no nobīdes.



2.1. att. Žiroskopa datu vizualizācija

Attēlā redzams darba autora mēģinājums vizualizēt žiroskopa sūtītos mērījumus ar 3d objektu - “lidmašīnu”.

Praktiski viss notiek tā - Žiroskops mēra kādā leņķī iekārta ir pagriezusies attiecībā no sākuma stāvokļa. Akcelerometrs mēra paātrinājumu un no tā iegūts noieta attālumu. Ideālā pasaulē ar to vien pietiktu, bet reāli šiem sensoriem veidojas nobīdes, jo nav nekas cits uz ko balstīties vai salīdzināt mērījumus. Ar žiroskopu nav tik lielas neērtības, jo tam korekciju veiks ar magnetometra palīdzību. Noteiktā laika intervālā nosakot leņķi attiecībā pret ziemeļiem, varēs veikt labojumus, ja žiroskopa mērījumi neatbilst attiecīgajam leņķim.



2.2. att. Akselerometra datu vizualizācija

Attēlā 2.2. Redzama žiroskopa datu vizualizācija, kad sensors tika strauji pakratīts. Sensors MPU-6050 tika pievienots pie Arduino Uno, kas savukārt tika pievienots datoram ar seriālo USB portu. Mikrokontrolerī Arduino tika ierakstīta programmatūra, kas nolasa datus no sensora un sūta šos datus ārā caur seriālo portu. Uz datora tika palaista processing vides programmatūra kura spēj nolasīt datus no šī seriālā porta un uzreiz attēlot tos grafikā [20]. Lai šo veiksmīgi atrisinātu, nācās veikt nelielas izmaiņas sensora kontroles reģistros, kuri atbild par to cik ātri tiek nolasīti dati no sensora. Pēc noklusējuma tas ir 100Hz, kas ir 100 mērījumi sekundē, ko šī konkrētā processing vides programma nespēja apstrādāt, līdz ar to lasīšanas ātrums tika samazināts līdz 25Hz.

2.3. Magnetometrs

Magnetometrs HMC5883L ir mikroshēmas modulis radīts zema lauka elektromagnētisma uztveršanai ar digitālu saskarni priekš tādām lietotnēm, kurām nepieciešams zemu resursu izmaksu kompass un magnetometrija. Šī mikroshēma sevī iekļauj augstas izšķirtspējas HMC118X sērijas magnēt-rezistatīvos sensorus, lietotnei specifisku integrēto mikroshēmu kas satur pastiprināšanu, kā arī 12 bitu pārveidotāju no analoga signāla uz digiālo kas dod 1 līdz 2 grādu kompassa virziena precizitāti. Attiecībai pret magnētisko ziemeļu noteikšanai. Magnetometrs ir nepieciešams žiroskopa leņķa korekcijai, ja tāda rodas.

2.4. Barometrs

BMP180 sensora ļoti zemais enerģijas patēriņš ir pielāgots lietošanai mobilajos telefonos, plaukstdatoros, globālās pozicionēšanas navigācijas (orientēšanās) iekārtās, kā arī āra aprīkojumā.

```
Temperature: 23.20deg C
Pressure: 102496 Pa
Standard Atmosphere: 1.0116
Altitude: -97.03 M

Temperature: 23.20deg C
Pressure: 102502 Pa
Standard Atmosphere: 1.0116
Altitude: -97.52 M

Temperature: 23.20deg C
Pressure: 102499 Pa
Standard Atmosphere: 1.0116
Altitude: -97.27 M

-----
pēdējie 3 mērījumi kad biju ārā 1. stāvā.

Temperature: 21.30deg C
Pressure: 102522 Pa
Standard Atmosphere: 1.0118
Altitude: -99.17 M

Temperature: 21.20deg C
Pressure: 102520 Pa
Standard Atmosphere: 1.0118
Altitude: -99.00 M

Temperature: 21.20deg C
Pressure: 102514 Pa
Standard Atmosphere: 1.0117
Altitude: -98.51 M
```

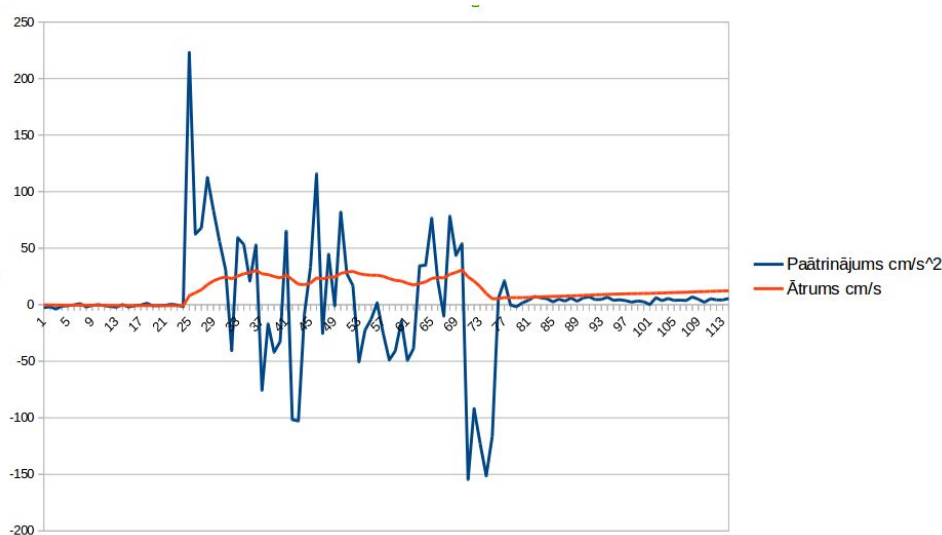
2.2. att. Dati iegūti no magnetometra

Attēlā redzami dati pēc sensora atrašanās vietas izmaiņš.

Ar zemu augstuma “troksni” - tikai 0.25m ātrā pārveidošanas laikā, šis sensors piedāvā pārāku sniegumu (pār savu iepriekšējo modeli BMP085 ?). Tam ir I²C saskarne kas dod iespēju ērtai sistēmas savietojamībai ar mikrokontrolieri. Tas ir bāzēts uz piezo-rezistīvās tehnoloģijas priekš spēcīgas elektro magnētiskās savietojamības, augstas precizitātes un linearitātes, kā arī ilgtermiņa stabilitātes.

3. SENSORU KĻŪDU APSTRĀDE

Veicot iepriekšējā nodaļā apskatīto sensoru tuvāku izpēti, kā arī praktisku datu nolasīšanu [20 - 22] ir manāmas kļūdas mērījumos. Šie mērījumi tika veikti uz šasijas ar 4 riteņu piedziņu un divām D298L mikroshēmām, kas katra spēj vienlaicīgi kontrolēt divus motorus. Konkrēti šim pārbaudījumam visiem motoriem signāls uzsākt kustību tika padots vienlaicīgi, jo tie tika tika pievienoti pie viena slēdža.

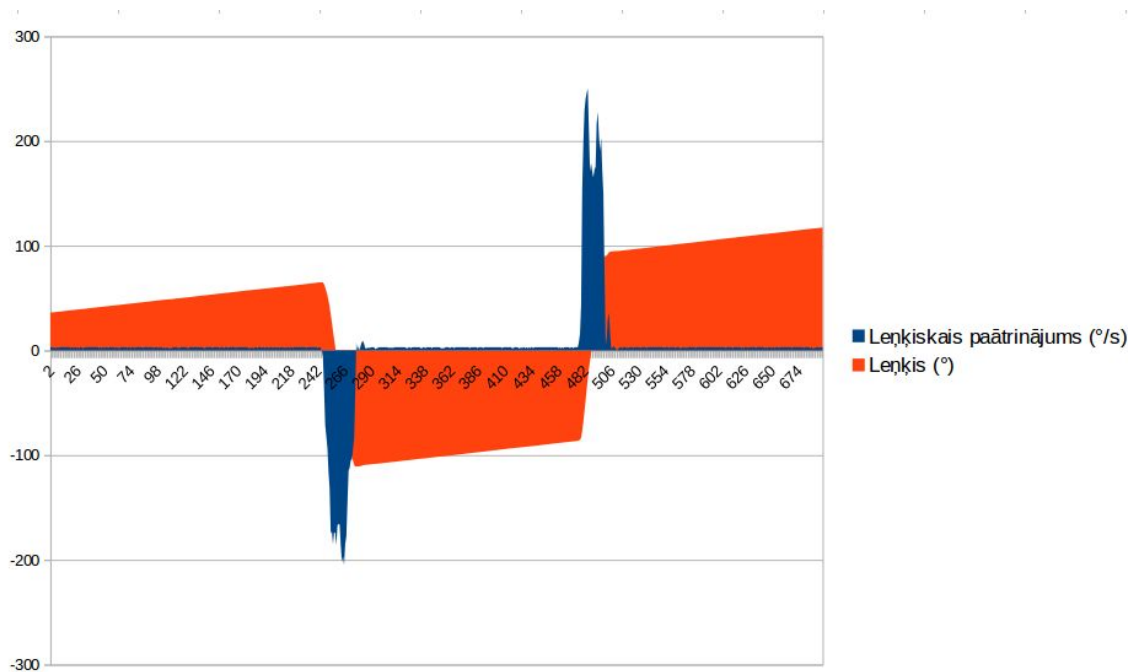


3.1. att. Paātrinājuma dati iegūti no sensora MPU-6050

Attēlā 3.1. ar zilo līniju ir redzami dati, ko darba autors ieguva nolasot mērījumus no sensora attēlojot tos pēc jēlu datu transformācijām uz cilvēkam lasāmiem datiem - paātrinājuma centimetros sekundes kvadrātā (cm/s^2) un sarkano līniju ir redzams ātrums, kas tika iegūts aprēķinot paātrinājumu konkrētā laika vienībā. Uz y-ass attēloti mērījumi, uz x-ass - mērījumu skaits. Mērījumi tika veikti 25Hz frekvencē, kas nodrošināja 25 mērījumus sekundē, kā rezultātā laiks starp katru mērījumu ir vidēji 0,04 sekundes. Pirmajā Intervālā no 1. līdz 25. mērījumam sensors netika kustināts, otrajā intervālā no 26. līdz 70. tika padots signāls darboties un trešajā intervālā pie 71. mērījuma šis signāls tika izslēgts. Kā redzams pirmajā intervālā paātrinājums un ātrums vidēji nav mainījies, lai arī uz paātrinājuma datiem redzams troksnis. Otrajā intervālā sensors paātrinājās līdz noteiktam ātrumam un uz šiem 46 mērījumiem saglabāja savu vidējo ātrumu un trešajā intervālā pazūdot motora darbībai strauji samazināja savu ātrumu, jo ieguva negatīvu paātrinājumu bremzēšanas rezultātā. Trešajā intervālā ir arī novērojams, ka paātrinājums nav samazinājies līdz sākotnējai vidējai 0

vērtībai, kā rezultātā ātrums sāk lineāri pieaugt. Šī ir aktuāla problēma kuras risinājumi tiks apskatīti šajā darbā.

Otra lielākā problēma ir žiroskopa dati. Lai veiktu žiroskopa leņķiskā paātrinājuma sensora datu uzskatāmu attēlošanu, tas tika novietots uz zemes, netika kustināts, tad pagriezts, atstāts uz mirkli un tad atkal pagriezts.



3.2. att. Leņķiskā paātrinājuma dati

Attēlā 3.2. ar zilu laukumu ir attēlots leņķiskais paātrinājums un ar sarkanu ir attēlots leņķis. Sensors netika kustināts pirmo trešdaļu no attēlā redzamā intervāla, tad pagriezts pa 180 pa pulksteņrādītāja virzienā, atstāts uz vidējo intervālu, kas bija 10 sekundes un tad atgriezts atpakaļ. Kā redzams sarkanais laukums diezgan stabili pieaug. Tam par iemeslu ir pavisam nelielā sensora sākotnējā kļūda. Ar ļoti blāvu laikumu attēlā ir redzama līnija tuvu nullei, kas ir leņķiskā paātrinājuma mērījums ar vērtību $3^{\circ}/\text{sek}$. Pie 242. mērījuma šī vērtība nokrīt līdz $\sim -200^{\circ}/\text{sek}$ tajā brīdī kad sensors tika pagriezts, un nokrītās atpakaļ, kad tas tika atstāts nekustīgi, tas pats ir novērojams arī pie 482. mērījuma tikai pretējā virzienā.

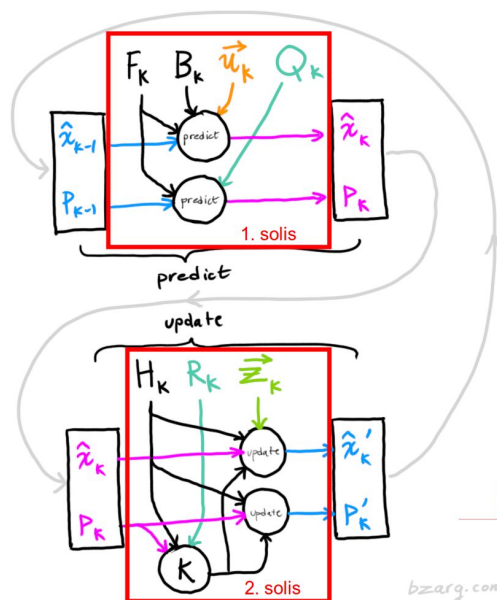
Žiroskopa gadījumā sensora problēma nav saistīta ar mērījumu nobīdi pēc sensora kustināšanas, bet gan ar sākotnējo datu nekorektumu. Lai šīs problēmas risinātu, tika izpētīti otrajā nodaļā aprakstītie darbi [10, 13, 18], kā arī cits risinājums - Invensense Digital Motion Processing, konkrētajam sensoram MPU-6050 iebūvēta iespēja apstrādāt datus uz paša sensora, nenoslogojot mikrokontrolieri kurš to darbina.

3.1. Kalmana filtrs

“1960. gadā R.E. Kalmans nopublicēja savu slaveno rakstu par diskrēta-laika lineāro filtrēšanas problēmas rekursīvo risinājumu. Rakstā bija piedāvāts matemātiskais rīks priekš dinamiskas sistēmas stāvokļa vektora novērtēšanas, izmantojot nepilnu un trokšņainu mērījumu virkni. Kalmana filtrs tiek pielietots inženierzinātņu un ekonometrisko problēmu risināšanai – no radaru un video sistēmu līdz makroekonomisko modeļu parametru novērtējumiem. Kopš tiem laikiem Kalmana filtrs bija ekstensīvi pētīts un plaši pielietots, it īpaši saistībā ar autonomo un asistēto navigāciju.” [23, 24] tā kalmana filtru apraksta kāds LU maģistrantūras students [13].

Uzskatāmākais piemērs ar kuru varētu aprakstīt filtrus ir šāds: Telpā atrodas iekārta kurai nepieciešams zināt savu ģeometrisko stāvokli, konkrēti - orientāciju. Lai šo uzdevumu izpildītu, iekārtai ir pieejams žiroskops. Veicot rotācijas darbību un nolasot mērījumus no šī sensora iekārta iegūs konkrēto leņķi kādā tā ir pagriezusies. Problēma ir sensorā, jo tas nav absolūti precīzs. Vienā mērījumā kļūda nav dramtiski liela, bet attiecīgi ilgtermiņā tā palielināsies un jo ilgāk iekārtai atradīsies darbībā, jo lielāka kļūda akumulēsies. Jāņem vērā arī “fona trokšņi”, kuri nav paša sensora radīti, bet mēdz rasties no ārējiem apstākļiem. Tādēļ uz šādas sistēmas tiek pielietots filtrs. Zemāk tiek skaidrots kā strādā kalmana filtrēšana, ar piemēru no interneta.

Kalman Filter Information Flow



3.3. att. Kalmana Filtra informācijas plūdums [25]

Attēlā 3.3. Redzama Kalmana filtra plūduma diagramma, kā ar to tiek apstrādāti dati.

Apzīmējumu skaidrojumi:

\hat{x}_k - nākošais, jeb paredzētais stāvoklis, tas kas mums interesē;

P_k - nākošā kovariances matrica (nobīde kas ietekmē rezultātu);

\hat{x}_{k-1} - iepriekšējais stāvoklis;

P_{k-1} - iepriekšējā kovariances matrica (nobīde jeb kļūda);

F_k - prognozes matrica;

B_k - kontroles matrica;

\bar{u}_k - zināmas ārējas ietekmes;

Q_k - nezināmas ārējas ietekmes;

H_k - sensoru modelējums;

R_k - trokšņu nenoteiktības kovariācija;

\bar{z}_k - sensoru lasījumi;

K - Kalmana guvums;

Attēlā 3.3. ir redzami divi soļi:

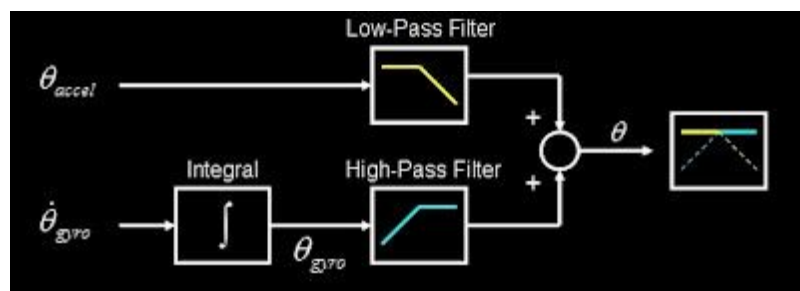
a) Pirmajā solī ir redzama pozīcijas aprēķināšana izmantojot iepriekš paredzamas varbūtības: Nākošo stāvokli iegūst, saskaitot (iepriekšējā mērījuma reizinājumu ar prognozes matricu) un (kontroles matricas reizinājumu ar zināmām ārējām ietekmēm). Nākošo kovariances matricu iegūst saskaitot (prognozes matricas reizinājumu ar iepriekšējo stāvokli) un nezināmajām ārējām ietekmēm.

b) Otrajā solī ir redzams, kā iepriekšējā solī iegūto pozīciju, apstrādā ar reāliem mērījumiem: Kalmana guvumu aprēķina izmantojot iepriekšējo kovariācijas matricu, sensoru modelējumu un sensoru trokšņu nenoteiktības kovariāciju. Lai pielabotu kovariācijas matricu tiek aprēķināta starpība starp pirmajā solī iegūtās prognozes par kov.m., un (kalmana guvuma, sensoru modelējuma un kov.m. reizinājumu). Savukārt lai pielabotu pozīcijas reālo atrašanās vietu, ņemot vērā mērījumus tiek saskaitīts iepriekšējā paredzētā pozīcija, ar kalmana guvuma reizinājumu (ar starpību, starp sensoru lasījumiem un (pirmajā solī paredzētā punkta un sensoru modelējuma reizinājumu)).

Paplašinātais Kalmana filtrs - tas tiek lietots nelineārām sistēmām. Principā praktiski no parastā kalmana filtra tas atšķiras ar to, ka veidojot aprēķinu nākošajai paredzamajai vērtībai, elementi no kuriem nākošais tiek aprēķināts, tiek aprēķināti caur kādu nelinearitāti (piem. Log, sin, u.c.) [26].

3.2. Komplimentārais filtrs

Tipisks papildinošā filtra pielietojums ir kombinēt vertikālā paātrinājumu un barometrisko vertikālo ātrumu lai iegūtu vertikālā ātruma aprēķinu [27]. Papildinošie filtri ir kas tāds, kas pats par sevi pasaka ko tas dara - tas papildina. Ņemot par piemēru akcelerometra un žiroskopa sensrou mērījumus ir zināms, ka saņemot jēlus datus no akcelerometra ilgtermiņā, tie var kļūt neprecīzi un tā pat arī saņemot datus no žiroskopa īstermiņā. Papildinošais filtrs papildina žiroskopa mērījumus ar akcelerometra mērījumiem lai rezultātā iegūtu korektus datus reālajā laikā.



3.4. att. Papildinošais jeb komplimentārais filtrs [28]

Attēlā 3.4. parādīts, kā no datiem attēloti ar dzeltenu un zilu līniju, kuriem nekonkrētā laikā rodas kļūda, tiek iegūts apvienojums lai atbrīvotos no šīs kļūdas.

3.3. Digitālā kustību apstrāde

Invensense ir kompānija kas veido un ražo šī brīža populārāko un zemāko-izmaksu žiroskopa un akcelerometra sensorus. Līdz ar savu plašo izpratni par MEMS iekārtām, viņi ieviesuši arī risinājumu resurs-efektīvākai datu apstrādei. DMP (Digital Motion Processing), jeb digitālā kustību apstrāde atrodas uz paša MPU-6050 un nodrošina sensoru mērījumu apstrādes algoritmu izpildi. DMP iegūst datus no akcelerometriem, žiroskopiem un citiem sensoriem ko iespējams pievienot, kā piemēram magnetometriem un apstrādā šos datus. Rezultējošie dati var tikt nolasīti no paša DMP reģistriem vai tikt ievietots FIFO uzgaidē [29].



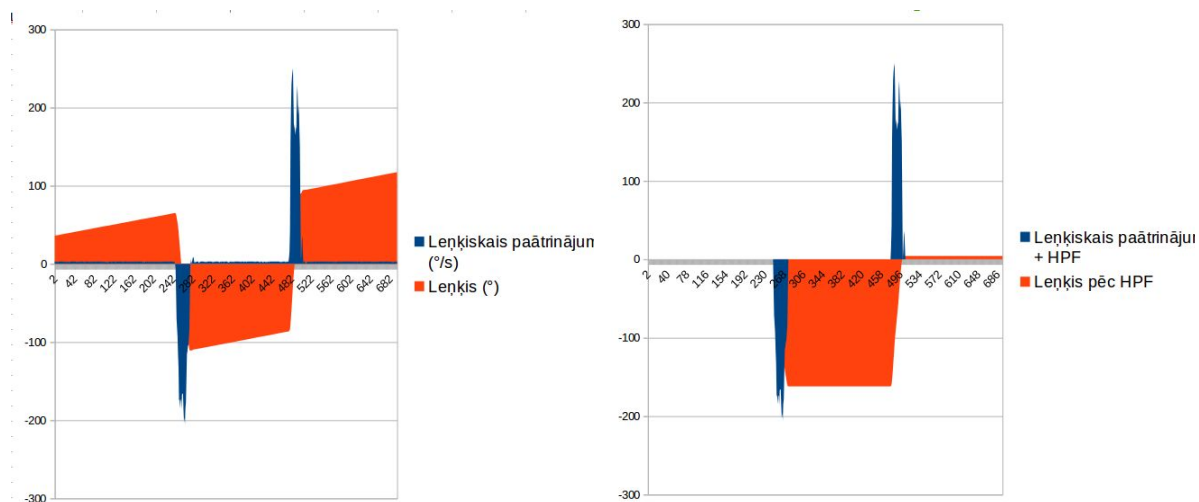
3.5. att. MPU-6000 Ģimenes Bloku Diagramma [30]

Attēlā 3.5. attēlota saziņa starp sensoru un iekārtu. Pirmais bloks no kreisās attēlo pašu sensoru, otrais bloks attēlo MPU-6000 sērijas moduļa saskarni ar šo sensoru, kā arī tālāku informācijas nodošanu iekārtām kas to spēj lasīt, un trešais bloks attēlo šīs iekārtas jeb lietotņu processorus, kas šos datus ir spējīgi saņemt un apstrādāt no šī moduļa. Šīs funkcijas mērķis ir pārnest laika prasības un apstrādes jaudu no resursprocesora [29, 30].

3.4. Darba ietvaros izpētītās žiroskopa kļūdu novēršanas metodes

Lai novērstu žiroskopa kļūdas tika apskatīti divi risinājumi:

- 1) Augstās caurlaidības filtrs (High pass filter)



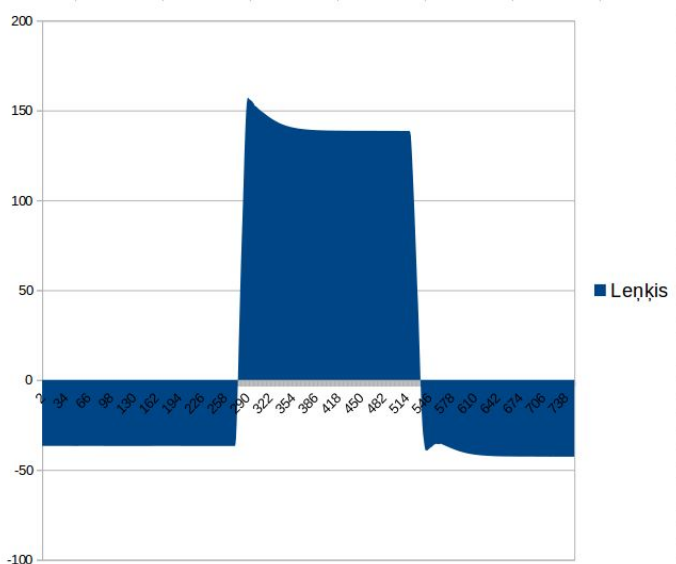
3.6. att. Sensoru datu mērījumu salīdzinājums

Attēlā 3.6. redzami dati kas iegūti ar MPU-6050 žiroskopu. Abos gadījumos tika nolasīti dati un tika veikta transformācija kas bija pārveidošana no “jēliem” datiem uz saprotamiem, t.i. leņķisko paātrinājumu (attēlots ar zilu laukumu), un pagrieziena leņķis, kas tika iegūts šo paātrinājumu reizinot ar laika vienību (attēlots ar sarkanu laukumu). Attēlā labajā pusē ir redzami šie dati pēc augstās caurlaidības filtra pievienošanas. Šajā gadījumā

tika pievienots noteikums izmantot tikai jēlos datus, kas ir ar vērtībām mazāki par -10 vai lielāki par 10, tādējādi zaudējot troksni kas ir redzams kreisajā pusē. Bez šī nelielā trokšņa, kas redzams kreisajā pusē pie pašas x-ass, vairs nesummējas leņķis, kas tika iegūts no itkā esošā konstantā paātrinājuma.

Šāda pieeja ir piemērota gadījumos, kad trokšņi ir pilnībā paredzami, taču tas ir reti, jo katram sensoram var būt sava rūpnīcas kļūda, kā arī šīs vērtības var sākt nobīdīties ar laiku, kā tika novērtos veicot datu izpēti ar nākošo metodi.

2) Digitālā kustību apstrāde (DMP)

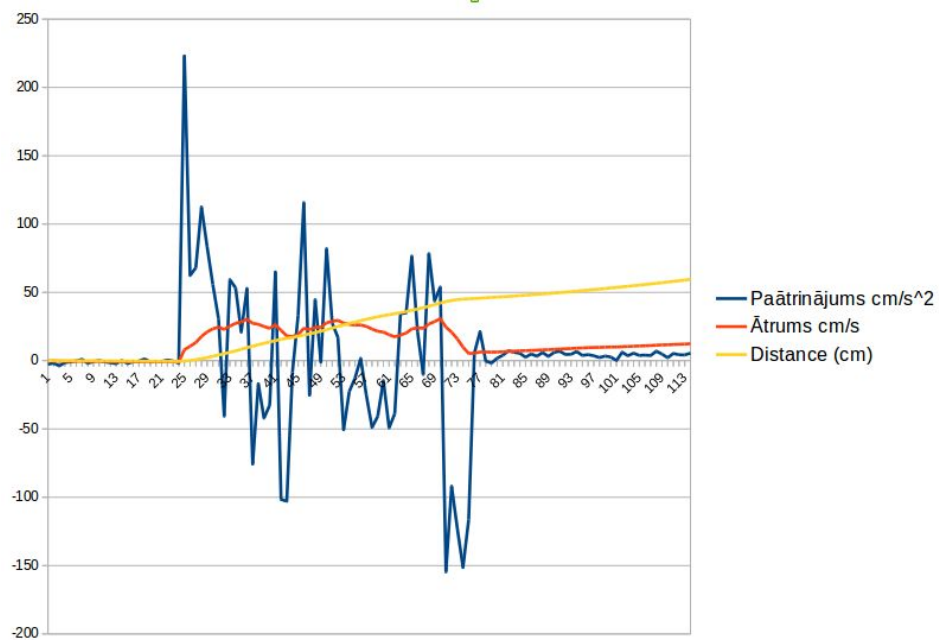


3.7. att. Leņķa noteikšana ar DMP

Šajā attēlā (3.7.) ir redzami sensora mērījumi, kas tika iegūti no moduļa MPU-6050 DMP. Šiem datiem, atšķirībā no “jēlajiem” datiem, kas tika iegūti no šī paša sensora, ir redzams inerces izliekums. Attēla kreisajā malā redzami dati atspoguļo sensora nekustību un tā orientāciju -40° pret sensora noteiktu 0° līdz datu otrajai trešdaļai, kad sensors tika strauji pagriezts $\sim 180^\circ$. Ievērojams ir nelielais kāpums līdz 160° , pēc kura dati nokrītās līdz attiecīgajiem 140° , kā tas arī tika pagriezts. Un pēc tam šī līdzīga tendence parādās arī kad sensors tika pagriezts atpakaļ pretējā virzienā 180° . Šajā gadījumā, gan šī inerces pat nepārsniedza galējo vidējo vērtību, kas bija $\sim 45^\circ$, bet sasniedza 40° , nedaudz vēl nokrītās, un tad tikai atguva pareizo vērtību 45° . Šī anomālija ir skaidrojama ar DMP datu savienošānu starp sensoriem un iespējamu kāda kalmana filtra pielietošanu tā algoritmos.

3.5. Darba ietvaros izpētītās akselerometra kļūdu novēršanas metodes

Tika nolasīti sākotnējie “jēlie” dati, bez jebkādas papildināšanas vai konkrētas tīšas datu manipulācijas no tiem aprēķināts paātrinājums cm/s^2 , no tā aprēķināts ātrums vienā laika vienībā, un no tā veiktā distance. Eksperiments tika veikts uz iepriekš minētās šasijas, ar 4 riteņu piedziņu, uzsākot viena virziena kustību ar visiem riteņiem vienlaicīgi. Motori tika darbināti 2000 milisekundes, bet kustība norisinājās 2200 milisekundes, jo iekārta bija uzņēmusi inerci.



3.8. att. Paātrinājuma, ātruma un attāluma dati

Mērījumi tika veikti 25Hz frekvencē, līdz ar to katru sekundi tika veikti 25 mērījumi. Iedaļas uz x-ass norāda mērījuma nummuru. Attēlā ir redzams, ka pirms kustības uzsākšanas vidējās vērtības visām taisnēm ir ap 0, savukārt pēc kustības veikšanas šie mērījumi ātruma un paātrinājuma gadījumos neatgriezās līdz 0, kā tiem būtu jādara. Kā rezultātā ātrums pakāpeniski palielinās un veiktā distance attiecīgi arī. Neņemot vērā šos pakāpeniskos pieaugumus, attāluma mērījums tika veikts pārsteidzoši precīzi - iekārta bija pārvietojusies ~48cm, ko ap 78.mērījumu arī var nolasīt no dzeltenās taisnes.

Lai novērstu akselerometra kļūdas tika mēģināti sekojoši risinājumi:

1) Kalmana filtrs

Izvērtējot situāciju, tika izvēlēts risināt paātrinājuma nepilnīgo datu lasījumu problēmu. Ieviešot sekojošu kalmana filtra algoritmu ir iespējams noteiktā mērā ietekmēt kādi dati tiks

attēloti - cik daudz sensoru trokšņus ņemt vērā, cik daudz procesa trokšņus ņemt vērā, kā arī kāds ir to ‘svars’ attiecībā pret pārējiem datiem. [31] Algoritms sekojošs:

$$p = p+q;$$

$$k = p/(p+r);$$

$$x = x+k(\text{mērījums} - x);$$

$$p = (1-k)*p;$$

x - šajā gadījumā ir gala vērtība ko iegūst pēc mērījumu filtrēšanas.

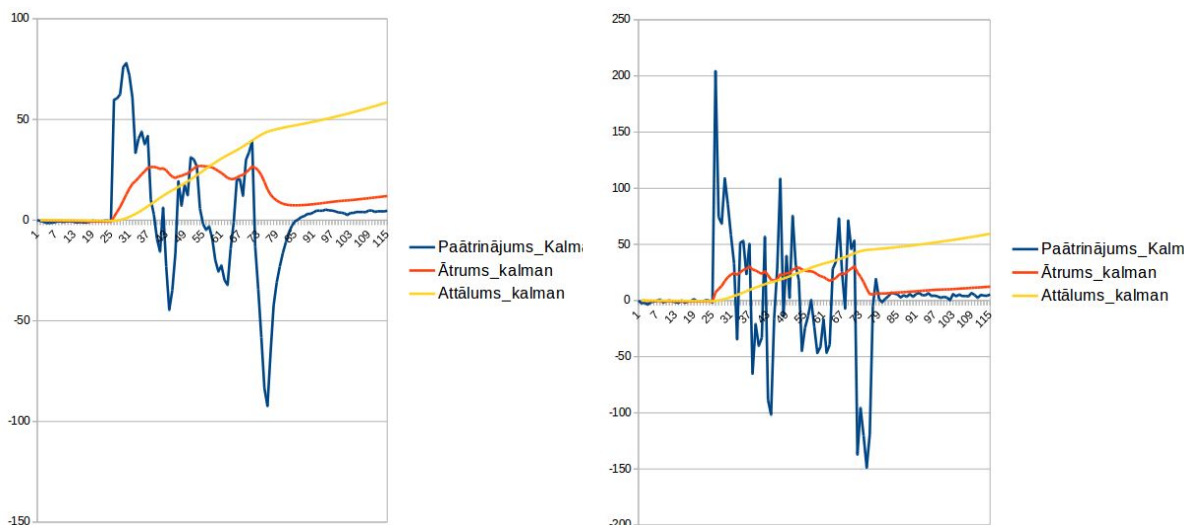
q - procesa troksnis.

r - sensoru troksnis.

p - aprēķinātā kļūda.

k - kalmana guvums.

Pēc šī algoritma ieviešanas, var uzskatāmi salīdzināt attēlus 3.8. un 3.9.



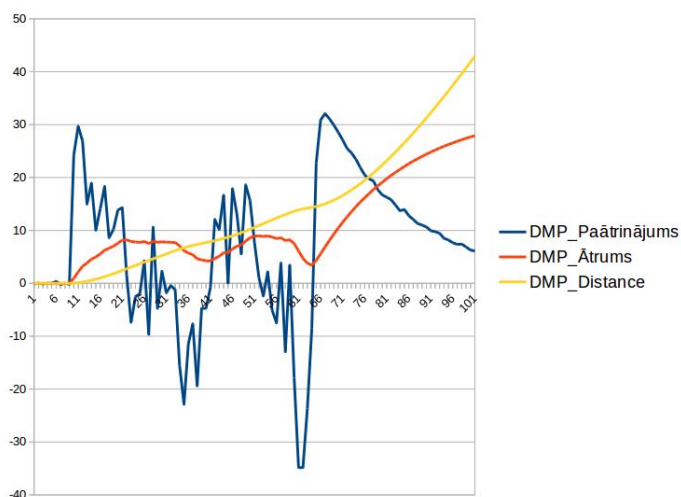
3.9. att. Datu apstrāde ar kalmana filtra algoritmu

Attēla kreisajā malā ir redzams filtrs ar procesa troksni 10 un sensoru troksni 1. Ievērojami ir izlīdzinātas vērtības un pirmā visaugstākā vērtība tiek ņemta vērā krietni mazāk. Savukārt attēla labajā malā ir šis algoritms ar procesa troksni 1 un sensoru troksni 10. Šajā grafikā vērtības ir izgludinātas, bet ne tik kritiski cik otrā grafikā.

2) Digitālā kustību apstrāde (DMP)

Šajā mēģinājumā tika izmantota čipā iebūvētā digitālā kustību apstrādes funkcija. Pēc attēla spriežot, sākotnējie dati ir labi, līdz uzsākas kustība. Salīdzinoši ar parastajiem datiem, neviena no vērtībām nav bijusi tik tuvu, kā mērījumiem bez šī DMP. Kas visu padara vēl sliktāku šim datu apstrādes veidam, ir pēdējais mērījums, kurš sasniedz pāspīlēti augstu

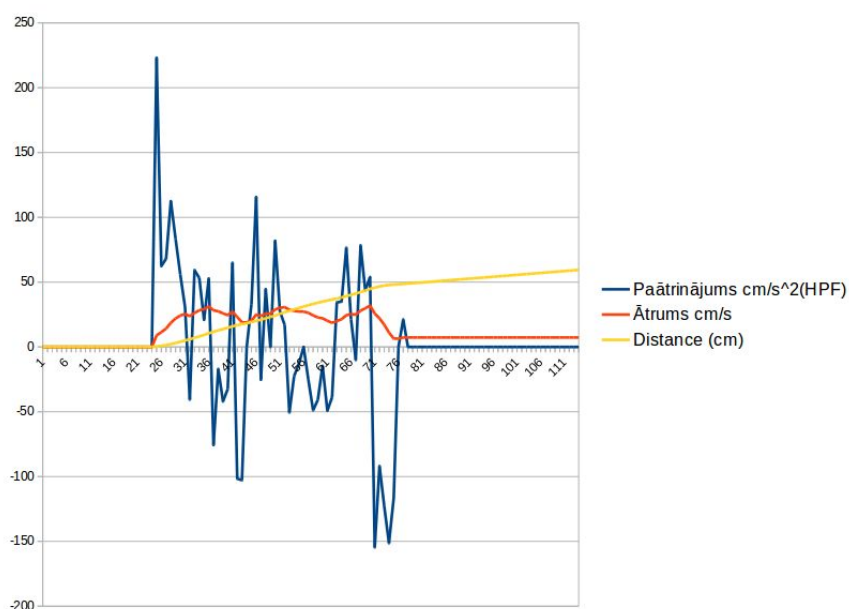
vērtību. To, kā tieši šis DMP algoritms strādā tā ražotāji neatklāj, bet manāma ir līdzība ar kalmana filtru ieviešanu, kas izlīdzina jebkādas iespējamas nepareizas vai trokšņainas vērtības.



3.10. att. Datu apstrāde ar “Digital Motion Processing”

Paskatoties no tālāka rakursa šo pārspīlēto pieaugumu varētu arī interpretēt kā mēģinājumu kompensēt kopējo vērtību - ja paskatās uz “oriģinālajiem” lasījumiem, kur veidojas ātruma nobīde, tad var redzēt, ka ātrums nemaz nesasniedz 0 vērtību, kā tam vajadzētu, jo paātrinājumam uzsākot braukt un paātrinājumam (palēninājumam) beidzot braukt, būtu visa kopējā brauciena laikā jābūt diezgan vienādam. Iespējams tas nesakrīt sensoru kļūdu un trokšņu dēļ. Līdz ar to, 3.10. Attēlā redzamo paātrinājuma lēcieni varētu interpretēt kā pēdējā mērījuma korekciju, kas šajā gadījumā gan ir pilnīgi nevietā.

3) Augstās caurlaidības filtrs (High pass filter)



3.11. att. Datu apstrāde ar “High-pass” filtru

Izmantojot augstās caurlaidības filtru, tika noņemtas jebkādas vērtības, kas bija mazākas par 10cm/s^2 no iepriekš veiktajiem mērījumiem. Kā redzams attēlā 3.9. Šis filtrs veiksmīgi izlīdzina jebkuras vērtības, kas ir jebkādā mērā trokšņainas nelielā līmenī, tādējādi atrisinot pieaugošā ātruma problēmu. Šajā situācijā gan parādās cita problēma, kā redzams attēlā - ātrums nerasniedz nulli. Līdz ar to var novērot, ka distance, mazāk, bet tik un tā palielinās.

4. TELPAS APCEĻOŠANA

Lai iekārta spētu ne tikai orientēties telpā, bet arī izvēlēties optimālu maršrutu, vai tas būtu lai pārvietotos no punkta A uz punktu B, vai arī lai pilnībā apstaigātu šo lauku, ir jāveic tā izpēte. Šī maršruta izveidei ir nepieciešams apskatīt divus tā aspektus: apkārtējās vides apzināšanu - kartēšanu, un efektīvu maršruta izvēli - pathfinding“.

Kādā darbā par robotu navigāciju izmantojot aktīvo redzi, tiek spriests par pieeju kartēšanai telpā. Viens no variantiem būtu kartes centru attiecināt, kā sensora jeb iekārtas centru un pārvietošanās gadījumos nevis pa karti pārvietotos robots, bet visi apkārtējie objekti pārvietotos attiecīgi. Šāds variants varētu šķīst pievilcīgs gadījumos kad iekārta atrodas nezināmā telpā, bet šādā gadījumā, būtu grūti noteikt cik tālu robots ir pārvietojies starp diviem punktiem, vai pat likt tam atgriezties sākotnējā pozīcijā. Dēļ šī iemesla un šajā darbā aplūkotās problēmas tas nekvalificējās kā risinājums [32].

4.1. Kartēšana (Mapbuilding)

IMU - Risinājums, kurš neiekļauj nekādu papildu sensoru pievienošanu ir balstīts uz iepriekšējā nodaļā minētajiem IMU lokalizācijas variantiem. Uzsākot darbību, esošais stāvoklis tiktu atzīmēts, kā 0 punkts no kura tiktu zīmēta karte. Šis nav drošākais risinājums spriežot pēc sekojoša piemēra. Lai iekārta varētu veiksmīgi veidot karti tai būtu jānosaka kur ir kāds šķērslis, telpu vidē - sienas. To var risināt iebraucot šajā sienā, nosakot, ka vairs nav ātruma, bet ja tiek pieļauta turpmāka kustība turp, pēc signāla motoriem, tad šī punkta konkrēto pusi var atzīmēt kā sienu. Šis risinājums nestrādātu, ja kāda iemesla dēļ nebūtu šķēršļu, bet motoram darbojoties, nemainītos ātrums - saķeres zuduma gadījumā. Līdz ar to optimāls risinājums tas nav. Alternatīvi šo varētu izpildīt ar vibrāciju sensoru, kurš spētu noteikt vai iekārta kaut kur ir ietriekusies. Pielietojot šādu risinājumu varētu spriest arī par ceļa kvalitāti gadījumos, kad kartēšana notiek ne telpās, bet ārā.

Alternatīva iepriekšminētajai pieejai būtu [33] rakstā pētīta nejaušā lauka apstaigāšana, kas konkrētajos salīdzinājumos izrādījās efektīvāka par līdzīgām pieejām.

Sonārs - Rakstā par kartēšanu un lokalizāciju autonomiem robotiem tiek piedāvāts risinājums izmantot sonārus. Šis ir samērā nobriedis risinājums kas ir ļoti efektīvs lai izpētītu karti tālāk par pašu iekārtu neliekot tai skriet sienās. Tehnoloģija arī ir sasniegus diezgan augstu līmeni spējot ieviest šādus sonārus izmēros kas nav lielāki par 3cm³. Negatīvais aspekts šim risinājumam būtu tā darbības laiks. Šie sensori pārsvarā mēra cik tālu ir objekti

kas tiem ir tieši priekšā. Lai no vienas iekārtas mērītu vairāk, kā tikai taisnu līniju sev priekšā, šis sensors būtu jānovieto uz soļu motora un pagriežoties pa vienu soli jāveic mērījums. Minētajā rakstā tika spriests par 4 šādu sensoru izmantošanu uz vienas iekārtas. Šādā gadījumā uzaust problēma ar laiku, jo lai veiktu šādu telpas analīzi būtu jāgaida krietns laiks vienā punktā [34].

Kartēšanai ir vēl dažas alternatīvas kuras nav tik resursefektīvas, kā risinājumi kuri tika pieminēti šajā nodaļā, līdz ar to tās tiks aprakstītas Nodaļā par alternatīviem risinājumiem.

4.2. Ceļa atrašana (Pathfinding)

Ceļa plānošana mūsdienās ir ļoti daudz pētīta tēma. Saukti arī par koku pārmeklēšanas algoritmiem, dēļ līdzības grafu apstaigāšanai. Tie tiek izmantoti arī citās analogiskās sfērās, kur šīs pašas kartes pielīdzina grafiem. Lai veiktu veiksmīgu ceļa plānošanu ir jāpakļaujās zināmiem mērķa-specifiskiem ierobežojumiem. Ceļa, vai tas būtu īsākais ceļš, vai garākais ceļš, atrašanā ir svarīgi apzināt karti. Ja karte jeb telpa kurā jāorientējas ir zināma un šajā telpā ir jānokļūst no viena punkta uz citu, tad var pielietot piemērotu specifisku algoritmu, piemēram Dijkstras algoritmu, lai palaižot sistēmu, vēl pirms tiek uzsākta darbība, jau tiktu aprēķināts īsākais ceļš starp šiem punktiem. Gadījumos, kad mērķis ir zināms, bet karte ir daļēji zināma vai pavisam nezināma efektīvāk būtu izmantot D* algoritmu [35].

Nemsim autonoma zālesplāvēja gadījumu par piemēru. Pieņemsim, ka zālesplāvējs ir aprīkots ar motoriem, līdz ar to tas var pārvietoties, bateriju, līdz ar to tas var brīvi pārvietoties un sensoriem, kas ne tikai ļauj tam orientēties, bet arī noteikt cik daudz enerģija tam ir, lai varētu plānot cik tālu tas var pārvietoties. Zinot šos parametrus var secināt, ka jārisina būs divas problēmas :

- 1) Īsākā ceļa problēma.
- 2) Lauka pilnīga apceļošana, jeb ceļojošā pārdevēja problēma.

Uzsākot darbību, šīs problēmas nebūtu jārisina, ja pieņem, ka telpa nav zināma. Tādā gadījumā ir jānododas uz tuvāko neizpētīto pusi. Iepriekš minētā īsākā ceļa problēma parādās, kad iekārtai ir jāatgriežas uz sākuma pozīciju, kā arī, kad no sākuma pozīcijas uz pozīciju no kuras tikko tā bija atgriezusies. Savukārt pilnīgai lauka apceļošanas problēmai būtu jāstājas pretī gadījumā kad viss lauks būtu vienreiz pilnībā apstaigāts. Arī tad nevar pilnībā paļauties uz pilno telpas karti, jo vide var mainīties, zālesplāvēja gadījumā, piemēram, kurmim izrokot

zemes kaudzi. Veicot izpēti tika noskaidrots, ka šobrīd efektīvākie algoritmi ir A* un D*, kuri arī tiks apskatīti.

A* algoritms. Izstrādāts 1968. Gadā to pirmo reizi aprakstīja Stenfordas pētījumu institūta pētnieki. Tas apvieno sevī tādas hjeristiskas pieejas kā “Greedy Best-First-Search” un formālas pieejas, kā Dijkstra algoritmu. “Greedy BFS” algoritms izmanto informāciju par to kurā no četriem apkārtējiem virzieniem objekts atrodas un neņemot vērā šķēršļus dodas uz mērķi. Dijkstra algoritms savukārt apstaigā sev vistuvākos ceļus vispirms, līdz atrod ceļu līdz mērķim [36].

D* algoritms. Izpētīts 1993. gadā, Karnegijas Melonas universitātē. Šajā rakstā [37] tika piedāvāts optimāls risinājums ceļu plānošanai nezināmās un dinamiskās vidēs. Ļoti līdzīgā, kā A* algoritms, šis risinājums piedāvā iespēju plānot ceļu un šķēršļu gadījumā to pārplānot eksponenciāli ātrākā laikā attiecībā pret lauka izmēru, kā attēlots [37] rakstā, eskperimentu sadaļā [37, 38].

Savukārt, lai veiktu telpas pilnīgu apceļošanu, tika apskatīts šāds [39] risinājums, kur tika veikts vairāku populāru pilnu lauku apceļošanas algoritmu salīdzinājums. Rezultātā var secināt, ka piedāvātais algoritms ir visefektīvākais, lai pilnībā apceļotu lauku. Iespējams, pētījumā norādītais pilnā pārklājuma D* algoritms ar “gandrīz 100% pārklājumu mainīgā vidē” ir nekorekts, jo rezultāti netika pārbaudīti praksē. Neņemot šo faktu vērā, pilnā pārklājuma D* algoritms un piedāvātais TRACC (TRaversal Algorithm for Complete Coverage - Ceļošanas Algoritms Pilnam Pārklājumam) būtu piemēroti risinājumi šajā darbā aprakstītajai inerciālās navigācijas sistēmai. Tomēr, lai izvēlētos labāko algoritmu, būtu jāievieš katrs no tiem un jāizvērtē dažādi tā aspekti, konkrēti - nokļūjums, ātrdarbība un spēja izvairīties no šķēršļiem.

REZULTĀTI

Pētījuma sākumā tika izvirzīti mērķi izpētīt inerciālās navigācijas sistēmas, autonomizēt šādu sistēmu, izpētīt telpas apceļošanu un izveidot risinājumu telpas pilnīgai apstaigāšanai.

Lai rastu skaidrību par tēmu un uzsvērtu tās aktualitāti, tika apskatīti mūsdienīgi projekti un risinājumu atklājumi. Pēc tā tika izpētīti risinājumi, kas ir saistīti vai līdzīgi ar šajā darbā apskatīto problēmu ar autonomu navigāciju, mazu resursu patēriņu sistēmām, un spēju izvēlēties dažādus lauku apstaigāšanas veidus.

Tika apskatīti pieejami un zemu resursu patēriņu sensori, tie izpētīti un aprakstīti. Līdz ar šo tika apzinātas galvenās problēmas, kas var rasties, un daļa šo problēmu arī apskatīta sensoru kļūdu labošanas nodaļā. Šajā nodaļā tika veikti vairāki eksperimenti ar iekārtām, lai apzinātu to ieviešanas ērtību un iespējas.

Eksperimentālā pieredze pierādīja, ka mikroelektro mehāniskās iekārtas nav absolūti precīzas un mēdz veidot nobīdes laika posmos. Eksperimentu laikā tika izpētīti dažādi risinājumi un daži no tiem realizēti dzīvē. Tomēr netika sasniegts vēlamais rezultāts - pēc paātrinājuma dubultas integrēšanas iegūts precīzs veiktā attāluma mērījums, jo rezultātam labvēlīgākā eksperimenta otrajā pusē, akselerometrs uzrādīja nepareizas vērtības - paātrinājumu, kaut arī iekārta atradās nekustīgi. Šī iemesla dēļ nākošā problēma ar ceļu plānošanu un lauka pilnīgas apstaigāšanas algoritmiem tika apskatīta tikai teorētiskā līmenī un nekas praktisks ar to netika paveikts.

ALTERNATĪVI RISINĀJUMI / NĀKOTNES PLĀNI

Kā alternatīvus risinājums var minēt sistēmas, kuras nav ar tik nelielu resursu patēriņu, ir kādā mērā lēnākas, vai arī tām nepieciešami kādi ārēji resursi, piemēram, satelīti.

Rakstos [40, 41] ir minēta inovatīva apkārtējās vides apzināšana izmantojot attālumu nosakošu lāzeru palīdzību. Šādi risinājumi ir prognozējami efektīvāki gadījumos, kad līdzīgas iekārtas, kā šajā darbā izstrādātā, spēj pārvietoties arī trešajā dimensijā, t.i. lidot, vai arī iekārtai tiek dots uzdevums izpētīt savu apkārtni. Risinājumā [42] tika aplūkota spēļu konsoles XBOX papildinājums Kinect, un kā to izmantot, lai veidotu telpas 3D karti un noteikt savu atrašanās vietu tajā, papildinoši tika izmantoti IMU. Šie risinājumi netika apskatīti tuvāk, jo šīs iekārtas nav tik plaši pieejamas vai zemu izmaksu, kā arī nav kritiski nepieciešami, lai sasniegtu 2D orientēšanās mērķi.

Vienā no apskatītajiem rakstiem [9] risinājumam ir visai neierasta pieeja - lokalizācija izmantojot zemes ģeomagnētiskā lauka parakstus. Apvienojumā ar bezvadu tīkla signāliem, šis risinājums aprakstīja pozitīvus rezultātus. Tika apskatīts arī risinājums, kas nav pilnībā paredzēts autonomām inerciālās navigācijas sistēmām. Citā rakstā [43] aprakstīts, kā ar viedtālruni un bezvadu tīklu palīdzību, lietotājs, precīzāk lietotne, var noteikt atrašanās vietu telpas apmilitūdā. Šiem risinājumiem ir nepieciešams bezvadu tīklājs un/vai iepriekšēja kartes apstaigāšana.

Protams, jāpiemin arī globālās pozicionēšanas risinājumi, ar kuriem iekārta varētu noteikt savu globālo atrašanās vietu, kā arī no dispozīcijas laikā iegūt savu ātrumu. Šādi risinājumi balstās uz ārējiem resursiem - satelītiem, kas arī ir iemesls kādēļ tas netika izmantots.

SECINĀJUMI

Šajā darbā tika izvirzīti sekojoši mērķi - izpētīt inerciālo mērvienību sistēmu ieviešanu lokalizācijas mērķiem, analizēt inerciālo navigācijas sistēmu precizitāti, izpētīt vienlaicīgus kartēšanas un lokalizācijas algoritmus un realizēt šos risinājumus apvienotā sistēmā.

Veicot esošu risinājumu izpēti autora meklējumu robežās, tika secināts, ka tikai uz inerciālām sistēmām bāzēta navigācija līdz šim tiešā veidā nav realizēta. Ir realizētas sistēmas kur inerciālā navigācija tika izmantota, kā papildinošs līdzeklis, lai samazinātu kļūdainus mērījumus.

Darbā tika veikta izmantoto sensoru analīze un izpētīta to datu apstrādes sarežģītība. Pēc šo eksperimentu rezultātiem, tika izpētīti risinājumi, kā veikt sensoros pastāvošās kļūdas likvidāciju, izmantojot gan sensorā iebūvētas funkcijas, gan arī veicot iegūto datu pēcapstrādi. Diemžēl ne ar vienu no apskatītajiem risinājumiem neizdevās realizēt ilgstošu (ilgāk par 2 sekundēm) nekļūdīgu (atrašanās vietas noteikšana ar mazāk kā 1cm kļūdu) atrašanās vietas noteikšanu.

Iepriekšējo eksperimentu rezultātā tika izpētīti lauka apceļošanas algoritmi īsākā ceļa aprēķināšanai, kā arī pilna lauka apstaigāšanai. Vismaz vienā no risinājumiem tika piedāvāts pseidokoda paraugs, ar kuru varētu veikt lauka analīzi tad, kad tiktu atrisināta autonomas lokalizācijas problēma.

Tālākais attīstības virziens noteikti būtu padziļināta inerciālās navigācijas sistēmu, konkrētāk akcelerometru, izpēte un dažādu algoritmu ieviešana kļūdu novēršanai. Autora plāns ir turpināt uz šī darba bāzes izveidotās inerciālās navigācijas sistēmas tālāku izstrādi.

IZMANTOTĀ LITERATŪRA UN AVOTI

- [1] Google Self-Driving Car Project [tiešaiste] – [atsauce 25.05.2016.] Pieejams:
<https://www.google.com/selfdrivingcar/>
- [2] Model S Software Version 7.0 - [tiešaiste] – [atsauce 25.05.2016.] Pieejams:
<https://www.teslamotors.com/presskit/autopilot>
- [3] Husqvarna Robotic Lawn Mowers - [tiešaiste] – [atsauce 25.05.2016.]Pieejams:
<http://www.husqvarna.com/us/products/robotic-lawn-mowers/>
- [4] Starship - [tiešaiste] – [atsauce 25.05.2016.] Pieejams: <https://www.starship.xyz/>
- [5] iRobot - [tiešaiste] – [atsauce 25.05.2016.] Pieejams: <http://www.irobot.com/>
- [6] OceanOne - [tiešaiste] – [atsauce 25.05.2016.] Pieejams:
<http://cs.stanford.edu/groups/manips/ocean-one.html>
- [7] Amazon Prime Air - [tiešaiste] – [atsauce 25.05.2016.] Pieejams:
<http://www.amazon.com/b?node=8037720011>
- [8] Long Exposure Pictures of Robots Cleaning - [tiešaiste] – [atsauce 5.05.2016.] Pieejams:
<http://spectrum.ieee.org/automaton/robotics/home-robots/long-exposure-pictures-of-robots-cleaning>
- [9] **Y. Shu, C. Bo, G Shen, C. Zhao, L. Li, F. Zhao**, Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing, 2015.
- [10] **Billur Barshan and Hugh F. Durrant-Whyte**, Inertial Navigation Systems for Mobile Robots, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 11, NO. 3, JUNE 1995.
- [11] **Michael Drumheller**, Mobile Robot Localization Using Sonar, Massachusetts institute of technology, 1985.
- [12] **Zhuoling Xiao, Hongkai Wen, Andrew Markham, Niki Trigoni**, Lightweight Map Matching for Indoor Localisation using Conditional Random Fields, Department of Computer Science, University of Oxford, 2014.
- [13] **Georgijs Kanonirs**, Digitālā audio apstrāde un efektu sintēze pielietojot bezvadu sensorus, LATVIJAS UNIVERSITĀTE DATORIKAS FAKULTĀTE, 2010.
- [14] **Jan Wendel, Oliver Meister, Christian Schlaile, Gert F. Trommer**, An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter, Institut fur Theoretische

- Elektrotechnik und Systemoptimierung, Universität Karlsruhe, 2006.
- [15] **Andrew John Davison, Keble College**, Mobile Robot Navigation Using Active Vision - Department of Engineering Science University of Oxford, 1998.
- [16] **Axel Großmann and Riccardo Poli**, Robust Mobile Robot Localisation from Sparse and Noisy Proximity Readings using Hough Transform and Probability Grids - School of Computer Science, The University of Birmingham, 2001.
- [17] **John J. Leonard, Hugh F. Durrant-Whyte**, Simultaneous Map Building and Localization for an Autonomous Mobile Robot - 1991.
- [18] **Eric Foxlin**, Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter, Research Laboratory of Electronics Massachusetts Institute of Technology, 1996.
- [19] **K. A Drishya, Divya K .S, Sakthiprasad. K .M, Anjaly Krishnan**, Localization and Mapping of Flying Robot in GPS-Denied Environments, IJSTE - International Journal of Science Technology & Engineering | Volume 1 | Issue 10 | April 2015
- [20] MPU-6050 Sensor Tutorial - [tiešaiste] – [atsauce 25.05.2016]. Pieejams: <http://diyhacking.com/arduino-mpu-6050-imu-sensor-tutorial/>
- [21] Invensense MPU-6050 - [tiešaiste] – [atsauce 25.05.2016]. Pieejams: <http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- [22] Arduino Guide using MPU-6050 and nRF24L01 - [tiešaiste] – [atsauce 25.05.2016]. Pieejams: http://dhrc.snu.ac.kr/wp-content/uploads/2015/11/MPU-6050_RF24L01.pdf
- [23] **R.E. Kalman**, “A New Approach to Linear Filtering and Prediction Problems”, Transactions of the ASME–Journal of Basic Engineering, 82 (Series D): pp. 35-45, 1960.
- [24] **G. Welch, G. Bishop**, “An Introduction to the Kalman filter”, SIGGRAPH, 2001.
- [25] How a Kalman filter works - [tiešaiste] – [atsauce 25.05.2016]. Pieejams: <http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>
- [26] The Extended Kalman Filter: An Interactive Tutorial for Non-Experts - [tiešaiste] – [atsauce 25.05.2016]. Pieejams: http://home.wlu.edu/~levys/kalman_tutorial/
- [27] **WALTER T. HIGGINS, JR.**, A comparison of complementary and Kalman filtering - Arizona State University, 1975.
- [28] Fun with the Complementary Filter - [tiešaiste] – [atsauce 25.05.2016]. Pieejams: <http://scolton.blogspot.com/2012/09/fun-with-complementary-filter-multiwii.html>
- [29] MPU-6000 and MPU-6050 Product Specification Revision 3.4
- [30] InvenSense Introduces Next Evolution of 6-Axis SoC With Enhanced Digital Motion

- Processor™ - [tiešaiste] – [atsauce 25.05.2016]. Pieejams:
<http://ir.invensense.com/phoenix.zhtml?c=237953&p=irol-newsArticle&ID=2021322>
- [31] Filtering Sensor Data with a Kalman Filter - [tiešaiste] – [atsauce 25.05.2016]. Pieejams:
<http://interactive-matter.eu/blog/2009/12/18/filtering-sensor-data-with-a-kalman-filter/>
- [32] **Andrew John Davison, Keble College**, Mobile Robot Navigation Using Active Vision - Department of Engineering Science University of Oxford, 1999.
- [33] Experimental investigation on coverage performance of a chaotic autonomous mobile robot
- [34] IROS '91, Nov. 3-5. 1991. "Simultaneous Map Building and Localization for an Autonomous Mobile Robot".
- [35] **Anthony Stentz**, Optimal and Efficient Path Planning for Partially-Known Environments - The Robotics Institute; Cmegie Mellon University, 1994.
- [36] Introduction to A* - [tiešaiste] – [atsauce 25.05.2016]. Pieejams:
<http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html#algorithms>
- [37] **Anthony Stentz**, Optimal and Efficient Path Planning for Unknown and Dynamic Environments, The Robotics Institute Carnegie Mellon University, 1993.
- [38] **David Rathbun, Ph.D., Sean Kragelund, Anawat Pongpunwattana, Brian Capozzi, Ph.D.**, An evolution based path planning algorithm for autonomous motion of a uav through uncertain environments, University of Washington, Seattle, WA.
- [39] **Kavitha Thiayagarajan and Coimbatore Ganeshsankar Balaji** , Traversal algorithm for complete coverage, Journal of Computer Science 2012, 8 (12), 2032-2041.
- [40] **Hartmut Surmann*, Andreas Nüchter, Joachim Hertzberg**, An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments, Fraunhofer Institute for Autonomous Intelligent Systems (AIS), Schloss Birlinghoven, 2003.
- [41] **David M. Cole and Paul M. Newman**, Using Laser Range Data for 3D SLAM in Outdoor Environments*, Oxford University Robotics Research Group Department of Engineering Science University of Oxford.
- [42] Bas des Bouvrie, Improving RGBD Indoor Mapping with IMU data, Delft University of Technology Master's Thesis in Embedded Systems, 2011.
- [43] The framework for internal navigation and discovery - [tiešaiste] – [atsauce 25.05.2016]. Pieejams: <https://github.com/schollz/find>

Bakalaura darbs „Bezpilota motorizētas iekārtas orientēšanās telpā un optimālā maršruta izvēle”
izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie
informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Gustavs Šenne

Rekomendēju/nerekomendēju darbu aizstāvēšanai (nevajadzīgo izsvītrot)

Vadītājs: Dr.dat. Reinholds Zviedris

Recenzents:

Darbs iesniegts Datorikas fakultātē 30.05.2016.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

prot. Nr.

Komisijas sekretār__: