

Ventspils Valsts 1. ģimnāzija

Valters Vieško
12. VID klases skolnieks

**Baxter robota inversās kinemātikas uzlabošana,
izmantojot Kinect2 sensoru**

Zinātniskās pētniecības darbs dabaszinātņu nozaru grupas
datorzinātnes un informācijas zinātnes nozarē

Zinātniskais vadītājs:
PhD.sc.comp., pētnieks
Ē. URTĀNS

Ventspils - 2024

ANOTĀCIJA

Baxter robotam RTU IKSA laboratorijā ir bojāti vai neprecīzi motora iekodētāji. Šajā darbā tika izveidots risinājums Baxter robota inversās kinemātikas vadišanai, lietojot ārēju Kinect2 sensoru, lai uzlabotu precizitāti. Darba autora jaunieviestais risinājums aprēķina robota locītavu leņķus un mērķa leņķus, lai nogādātu tā rokas galu mērķa galapunktā telpā. Robota locītavas iteratīvi tiek pārvietotas, izmantojot soļa vērtību, līdz mērķa pozīcija ir sasniegta. Lai analizētu risinājuma precizitāti, darba autors salīdzināja jaunieviesto risinājumu ar jau esošu risinājumu. Balstoties uz darbā aprakstītajiem eksperimentiem, jaunieviestais risinājums ir 336% precīzāks par jau esošo risinājumu. Darba gaitā tika izstrādāts papildus alternatīvs risinājums, kas bija 360% precīzāks par jau esošo risinājumu.

Lai šo darbu saprastu lasītājam ir jābūt vismaz vispārējām zināšanām par programmēšanu un augstāko matemātiku.

Jaunieviestais risinājums apskatāms: <https://youtu.be/NULSNqqtyNg>

Darbs rakstīts latviešu valodā, satur 16 lapas, 9 attēlus un tajā izmantoti 18 literatūras avoti.

Atslēgas vārdi: Inversā Kinemātika, Baxter, Autonomie roboti, Robotika, Kinect2

ABSTRACT

The Baxter robot in the RTU IKSA laboratory has damaged or imprecise motor encoders. In this paper, a solution was developed for controlling the Baxter robot's inverse kinematics using an external Kinect2 sensor to improve precision. The paper's author's solution calculates the joint angles of the robot and target angles to position its arm at the target endpoint in space. The robot's joints are iteratively moved using a previously determined step value until the target position is reached. To analyze the precision of the solution, the author compared the newly introduced solution with an already existing one. Based on the experiments described in the paper, the newly introduced solution is 336% more precise than the existing solution. During the development of this paper, the author created an additional alternative solution, which was 360% more precise than the existing solution.

To understand this work, the reader should have at least general knowledge of programming and higher mathematics.

The paper's author's solution's demonstration: <https://youtu.be/NULSNqqtyNg>

The paper is written in Latvian, comprising 16 pages, incorporating 9 images, and referencing 18 sources in the literature.

Keywords: Inverse Kinematics, Baxter, Autonomous robots, Robotics, Kinect2

Saturs

1. Apzīmējumu saraksts	3
2. Ievads	4
3. Literatūras apskats	5
3.1. Inversā kinemātika	5
3.1.1. Inversās kinemātikas risināšana, pielietojot parciālus atvasinājumus	5
3.2. Baxter robots	6
3.3. Kinect2 sensors	7
3.4. ROS	8
4. Eksperimentālā daļa	8
4.1. Kinect2 risinājums	8
4.1.1. IK un robota pozīcijas aprēķinātājs	8
4.1.2. Baxter robota kontrolieris	9
4.2. IMU risinājums	10
4.3. IMU-multi risinājums	10
4.4. Risinājuma salīdzināšanas metodoloģija	10
5. Rezultāti	11
6. Secinājumi	12
7. Tālākie pētījumi	13
8. Izmantotā literatūra	14
9. Pielikumi	16

1. Apzīmējumu saraksts

3D - Trīs dimensijas

DOF - Degrees Of Freedom (brīvības grādi)

EE - End Effector (galapunkts)

IK - Inversā kinemātika

IFR - International Federation of Robotics

IKSA - Informācijas, komunikācijas un signālu apstrādes

IR - Infrared (Infrasarkana)

RGB - Red, Green, Blue (sarkans, zaļš, zils)

ROS - Robot Operating System (robotu operētājsistēma)

RTU - Rīgas tehniskā universitāte

SDK - Software Development Kit (aplikāciju izstrādes komplekts)

ToF - Time of Flight (lidojuma laika)

2. Ievads

Pēdējos gados ir strauji pieaugusi automatizācija rūpnīcās, lietojot robotus, kas veic svarīgas darbības, kurām vajadzīga augsta precizitāte. Balstoties uz IFR (*International Federation of Robotics*), globāli ir tikuši instalēti 553052 industriāli roboti, turklāt šis skaits aug par 5% ik gadu [7]. Kā arī, balstoties uz mājaslapu *www.semanticscholar.org*, no 2013 līdz 2023 gada ik gadu publicēto darbu skaits ar atslēgas vārdu "Autonomous robots" (latv. Autonomie roboti) ir pieaudzis no 50066 līdz 68605.

Viens no šiem nolūkiem pielietojamiem robotiem ir Baxter, kuru izstrādāja 2012. gadā Rethnik Robotics. Baxter robotam ir divas rokas ar septiņām brīvības pakāpēm (7-DoF) un galva ar animētu displeju. Tomēr RTU IKSA laboratorijas Baxter robota motoru iekodētāji ir bojāti, neprecīzi un iegūst neprecīzus datus, rezultātā nav iespējams patstāvīgi, precīzi ar robota roku nonākt iepriekš definētā punktā telpā. Tā kā Baxter robotu beidza ražot 2018. gadā, kad *Rethnic robotics* tika slēgts [2], ir nepieciešams izstrādāt risinājumu šij neprecizitātei pašiem robota lietotājiem.

Alternatīvs robota iekšējo motoru izmantošanai ir pielietot ārejus sensorus, piemēram, ToF sensorus. Šos sensorus izmanto, lai savāktu datus par objektu pozīcijām telpā, tāpēc šie sensori ir pielietojami šim mērķim. Autors šī darba nolūkos ir izvēlējis izmantot Kinect2 sensoru un pārbaudīt, cik praktiski ir pielietot, šo sensoru Baxter robota kontrolēšanai.

Lai izmērītu risinājuma precizitāti, eksperimentālajā daļā tiks salīdzināts darba autora jaunieviestais risinājums, jau eksistējošais risinājums un darba gaitā ieviests alternatīvs risinājums Baxter robota kontrolēšanai. Konkrētā risinājuma precizitāti noteiks mērot robota rokas gala attālumu no kāda iepriekš izvēlēta punkta telpā. Lai aprēķinātu vēlamos leņķus, lai nogādātu robota rokas galu vēlamajā punktā, tiks pielietota inversā kinemātika.

Pētījuma hipotēze: Ir iespējams praktiski pielietot Kinect2 sensorus Baxter robota inversās kinemātikas kontrolēšanai, lai uzlabotu precizitāti.

Darba mērķis: Izstrādāt jaunievestu risinājumu un eksperimentālo vidi, kur vadīt Baxter robota inverso kinemātiku, lietojot Kinect2 sensoru un salīdzināt veiktspēju ar jau esošo metodi.

Darba uzdevumi:

1. Veikt literatūras izpēti par Baxter robotu, ROS un Kinect2. Apgūt inversās kinemātikas risināšanas metodes.
2. Izstrādāt programmu, kas pielietojama Baxter robota kontrolēšanai.
3. Izstrādāt programmu, kas nosaka Baxter robota roku pozīciju, lietojot Kinect2 sensorus.
4. Izstrādāt risinājumu, kas pielieto abas iepriekš minētās programmas, lai kontrolētu Baxter robotu.
5. Eksperimentāli salīdzināt jaunieviesto risinājumu ar jau esošo risinājumu un noteikt labāko.
6. Apgūt LaTeX programmēšanas valodu, lai uzrakstītu darbu.

3. Literatūras apskats

Literatūras apskatā tiks aprakstīta inversās kinemātikas risināšana, Baxter robots, Kinect2 sensors un ROS.

3.1. Inversā kinemātika

Tiešā kinemātika (angl. *Forward Kinematics*) tiek pielietota, lai noteiktu artikulētas struktūras (piem. cilvēka rokas) galapunktu, zinot struktūras segmentu izmērus un leņķus. [10] Tomēr inversā kinemātika (angl. *Inverse Kinematics*) strādā pretēji - katra segmenta leņķis tiek noteikts, zinot struktūras galapunktu un segmentu garumus. [10]

Tiešās kinemātikas risināšanu var izteikt ar vienādojumu [10]

$$x = f(q) \quad (1)$$

kur q - leņķis starp blakus esošiem segmentiem un x - visas struktūras galapunkts (angl. *End effector*) jeb EE. Tomēr inversās kinemātikas risināšanai ir nepieciešama inversā transformācija, kuru var izteikt kā vienādojumu

$$q = f^{-1}(x) \quad (2)$$

Pielietojot (1.), vienu rezultātu var iegūt ar vairākām q vērtībām. Tas nozīmē, ka (2.) pastāv vairāki atrisinājumi un IK risināšanas algoritmiem ir jāizvēlas konkrēts (2.) risinājums. [10]

3.1.1. Inversās kinemātikas risināšana, pielietojot parciālus atvasinājumus

Viens inversās kinemātikas aprēķināšanas risinājums ir pakāpeniski optimizēt tiešās kinemātikas vienādojumu (1.), mainot leņķus par konkrētām vērtībām, kuras izskaitļo, pielietojot parciālus atvasinājumus.

Šīs metodes pirmais solis ir izstrādāt tiešās kinemātikas vienādojumu (1.) [14]. To var izteikt vienam struktūras posmam ar (3.) funkciju

$$f(\theta, x, t) = R(\theta) \cdot x + t \quad (3)$$

kur θ - segmenta leņķis, t - segmenta sākumpunkts, x - segmenta vektors un $R(\theta)$ - rotācijas matrica [15]

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (4)$$

Lai demonstrētu šo metodi, var izstrādāt funkciju tiešās kinemātikas aprēķināšanai ar diviem posmiem, sasummējot abu esošo posmu vektorus. Piemērs šādai funkcijai, kur pirmā segmentā sākumpunkts ir $(0, 0)$, ir

$$f(\theta_1, \theta_2, x_1, x_2) = R(\theta_1) \cdot x_1 + R(\theta_1) \cdot (R(\theta_2) \cdot x_2) \quad (5)$$

Nejauši izvēloties θ_1 un θ_2 vērtības, iegūtais galapunkts būs kādā nejaušā punktā un vajadzēs mainīt θ_1 un θ_2 vērtības. Šīs metodes otrais solis ir aprēķināt parciālos atvasinājumus leņķiem θ_1 un θ_2 [14]. Lai noteiktu aprēķinātā nejaušā galapunkta ticamību, var izmantot lokālo ticamības funkciju (6.), kur y - vēlamais galapunkts un y' - tagad esošais galapunkts, kas iegūts no (5.) [16]

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=0}^N (y_i - y'_i)^2 \quad (6)$$

Tālāk, mēs varam izteikt parciālos atvasinājumus leņķiem θ_1 un θ_2 . Atvasinājumu ticamības funkcijai (6.) varam izteikt ar vienādojumu (7.) [17]

$$\frac{\partial(y - y')^2}{\partial y} = 2 \times (y' - y) \quad (7)$$

Lai iegūtu parciālo atvasinājumu priekš θ_1 un θ_2 , ir jāpielieto atvasinājums rotācijas matricai (4.), ko var izteikt kā (8.) [15]

$$\frac{R(\theta)}{\partial \theta} = \begin{bmatrix} -\sin(\theta), & -\cos(\theta) \\ \cos(\theta), & -\sin(\theta) \end{bmatrix} \quad (8)$$

Tālāk, var izteikt atvasinājumus θ_1 un θ_2 ar vienādojumiem (9.) un (10.) [17]

$$\frac{\mathcal{L}_{MSE}}{\partial \theta_1} = \frac{\partial(y - y')^2}{\partial y} \times \left(\frac{R(\theta_1)}{\partial \theta_1} \cdot x_1 \right) \quad (9)$$

$$\frac{\mathcal{L}_{MSE}}{\partial \theta_2} = \frac{\partial(y - y')^2}{\partial y} \times \left(R(\theta_1) \cdot \frac{R(\theta_2)}{\partial \theta_2} \cdot x_2 \right) \quad (10)$$

Šīs metodes trešais solis ir aprēķināt jaunas θ vērtības, tā lai samazinātu ticamības funkcijas rezultātu. Jaunos leņķus iegūst ar vienādojumu (11.) [1]

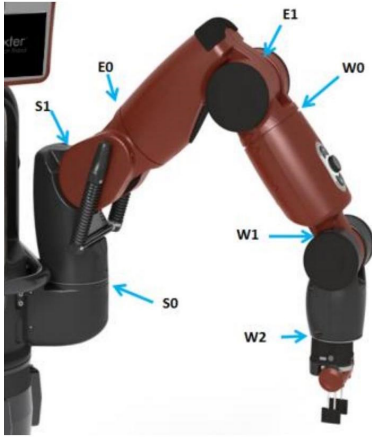
$$\theta' = \theta - \frac{\mathcal{L}_{MSE}}{\partial \theta} \times \alpha \quad (11)$$

Tālāk atkārti šīs metodes soļus līdz ticamības funkcijas rezultāts ir apmierinoši mazs [14]. Šo metodi var izmantot arī struktūrām ar vairāk segmentiem, pievienojot funkcijai (5.) vērtības θ_3 un x_3 , θ_4 un x_4 utt.

3.2. Baxter robots

Baxter robots ir industriālais robots, kuru uzņēmums *Rethink Robotics* prezentēja plašākai publikai 2012. gadā. [2] Tas formā atgādina cilvēku, tam ir divas cilvēka rokām līdzīgas rokas 7-DoF (*Degrees of Freedom*) un galva ar ekrānu.

Robota dators ir aprīkots ar trešās paaudzes Intel Core i7-3770 8MB 3.4 GHz procesoru, HD4000 video karti, 4GB DDR3 operatīvo atmiņu un 128 GB cieto disku. [18] Robota kameras maksimālo izšķirtspēja ir 1280 x 800 pikseli (640 x 400 pikseli efektīvā izšķirtspēja),



- S0 - Pleca rotācija
- S1 - Pleca celšana
- E0 - Elkoņa rotācija
- E1 - Elkoņa saliekšana
- W0 - Plaukstas rotācija
- W1 - Plaukstas saliekšana
- W2 - Plaukstu rotācija

1. att. Baxter robota motori vienā rokā [6]

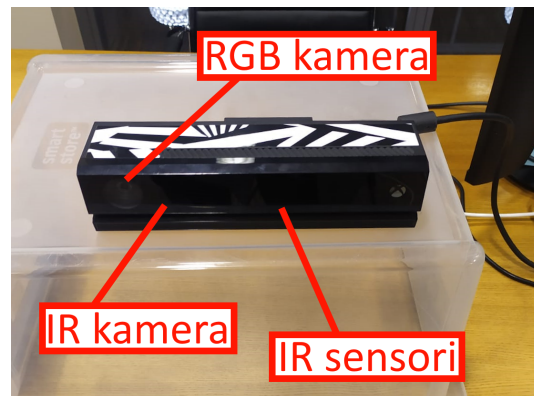
ar 30 kadru skaitu sekundē un 1,2 mm fokusa attālumu [18]. Animētajam sejas ekrānam ir izšķirtspēja 1024 x 600 pikseli. [18]

Šim darbam vissvarīgāk ir izprast robota roku uzbūvi. 1. attēlā tiek demonstrēti septiņi Baxter robota motori, kuri atrodami vienā rokā. Kā arī šie motori tiek apzīmēti ar saīsinājumiem, kurus šajā darbā izmantos, lai apzīmētu konkrētas locītavas.

Lai vienkāršotu sazināšanos starp datoru un Baxter robotu, ir pielietojams Baxter SDK, kas darbojas, izmantojot ROS, un satur *baxter_interface* bibliotēku [12]. Tā satur vairākus rīkus, lai manipulētu robota locītavas, kontrolētu tā galvu, piekļūtu robota kamerai un veiktu vairākas citas rīcības. [12] Šajā darbā darba autors izmantos šo bibliotēku, lai manipulētu robota roku - noteiktu tā locītavu pozīcijas un nosūtītu pavēles locītavu kustināšanai.

3.3. Kinect2 sensors

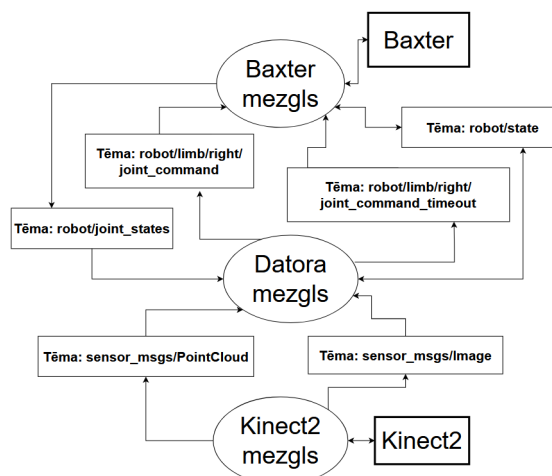
Kinect2 jeb *Microsoft Kinect V2* ir ierīce, kuru ražoja uzņēmums Microsoft, sākotnēji kā video-spēļu ierīci, bet tā tiek plaši izmantota pētniecībā tās precizitātes un zemās cenas dēļ [4]. Ierīce sastāv no RGB kameras, IR kameras un IR izstarotājiem (skat. 2. attēlā), kā arī no vairākiem mikrofoniem, kuri šajā darbā netiks raksturoti un izmantoti. Ierīcē spēj noteikt kadra 3D ģeometriju ar 30 kadru sekundē ātrumu. [3] Lai sensors noteiktu attālumu līdz objektam telpā, tas izmanto ToF metodi. IR izstarotājs tiek izslēgts un ieslēgts ar konkrētu frekvenci. Sensora IR kamera aprēķina objekta attālumu, izmantojot gaismas ātrumu un laika posmu, kas izmantots, lai IR stars atgrieztos. [5]



2. att. Kinect V2 sensors

3.4. ROS

ROS (*Robot Operating System*)[9] ir atvērta koda platforma, kas tika izstrādāta, lai atvieglotu robotu programmatūras izveidi un piedāvātu plašu rīku klāstu atšķirīgu robotu kontrolei. [9] Šī darba nolūkos, ROS sistēmas sastāv no trīs galveniem konceptiem. [9] Pirmais no šiem konceptiem ir "mezgls" (angl. *Node*), kas ir atdalīts process. [9] Šie "mezgli" sazinās ar konceptu vārdā "ziņas" (angl. *messages*), kas ir vienkāršas informācijas kopas. [9] Informācijas punkti iegūst "ziņas", abonējot "tēmas" (angl. *topics*), kas ir vienkārši teksta apzīmējumi konkrētas informācijas tēmai. [9] Piemērs ROS sistēmas izkārtojumam redzams 3. attēlā. 3. att. Eksperimenta konfigurācija ROS vidē



Šī platforma ietver arī *rospy* bibliotēku, kas ir ROS oficiālā bibliotēka *Python* valodai. [13] Baxter robota kontekstā ROS un *rospy* bibliotēka ir ļoti noderīgs instruments robota kontrolēšanai. Tāpēc darba autors to izmanto izvirzītā problēmas risinājuma izstrādē, lai nodrošinātu saziņu starp datoru un Baxter robotu.

4. Eksperimentālā daļa

Eksperimentālajā daļā tiek aprakstīti trīs šajā darbā salīdzinātie risinājumi un to salīdzināšanas metodoloģija. Darba autora jaunieviešto risinājumu šajā darbā sauks "Kinect2 risinājums" un jau esošais risinājums, kas šajā darbā tiks saukts "IMU risinājums". Kā arī darba autors jaunievieštajā risinājuma izstrādes laikā atklāja vēl vienu alternatīvu risinājumu, kuru sauks "IMU-multi risinājums".

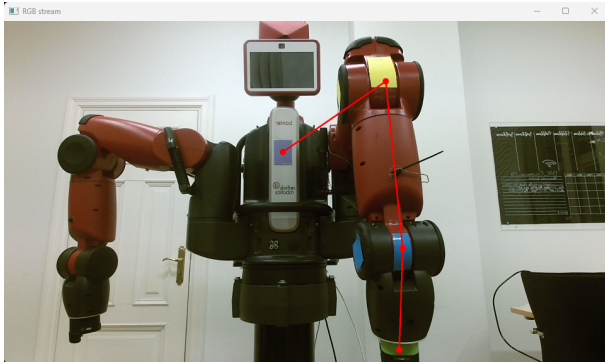
4.1. Kinect2 risinājums

Darba autors šajā darbā izvirza risinājumu Baxter robota inversās kinemātikas risināšanai, pat ja robota motoru iekodētāji ir bojāti, neprecīzi. Šis risinājums sastāv no divām daļām un tiek izstrādāts *Python3* programmēšanas valodā. (Risinājuma kodu skatīt 1. pielikumā)

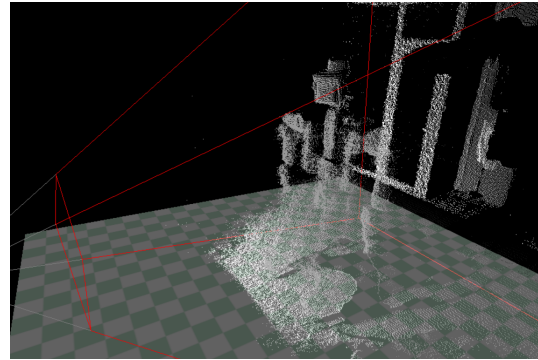
4.1.1. IK un robota pozīcijas aprēķinātājs

Pirmā risinājuma daļa - kods, kas aprēķina Baxter robota locītavu koordinātas telpā, šo locītavu leņķus, vēlamā galapunkta koordinātas telpā, un aprēķina vēlamos robota locītavu leņķus, lai robota rokas gals atrastos vēlamajā galapunktā. No sākuma, pielietojot populāro *opencv-python* bibliotēku, kods atrod locītavu x , y koordinātas RGB kameras kadrā. Darba autors aplāja robota locītavas ar krāsainu papīru, lai kods varētu noskaidrot šīs koordinātas, izolējot konkrētus krāsu "pleķus" un aprēķinot to vidējās koordinātas (skat. 4. attēlā). Tālāk

kods pārtulko šīs koordinātas no RGB kadra uz IR kadra koordinātām; zinot pikseļa koordinātas IR kadrā, kods aprēķina punkta koordinātas 3D telpā pielietojot *pykinect2* bibliotēku (skat. 5. attēlā).

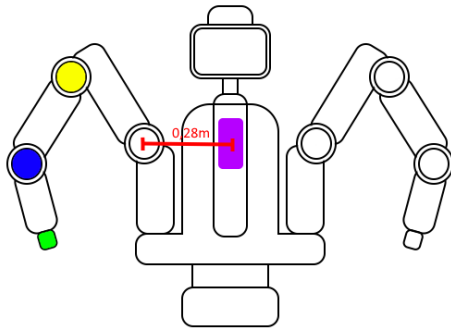


4. att. Punktu koordinātas kadrā

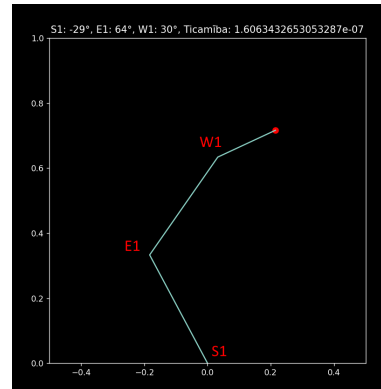


5. att. Punktu koordinātas 3D telpā

Viens ierobežojums ir tas, ka robota plecs ir reti redzams, tāpēc darba autora risinājums noskaidro robota krūškurvja koordinātas un tās pārrēķina, lai iegūtu nekustīgā pleca S_0 punkta koordinātas telpā (skat. 6. attēlu). Katru reizi, kad punktu koordinātas telpā tiek pārrēķinātas, kods aprēķina tagad esošos locītavu leņķus, pielietojot elementārus trigonometriskus vienādojumus, un aprēķina robota IK, lai nogādātu tā roku vēlamajā galapunktā. Robota IK tiek aprēķināta divos soļos - no sākuma aprēķina vēlamo S_0 locītavas leņķi un tad, pieņemot, ka locītavas E_0 un W_0 atrodas nulles pozīcijās, aprēķina locītavu S_1 , E_1 un W_2 koordinātas, pielietojot metodi ar parciāliem atvasinājumiem (skat. 7. attēlu).



6. att. S_0 koordinātu noteikšana



7. att. IK risināšana

Tālāk, lai nodrošinātu piekļuvi datiem, dati tiek publicēti lokālajā tīklā, lietojot serveri ar *Flask* bibliotēku.

4.1.2. Baxter robota kontrolieris

Trešā risinājuma daļa ir pirmkods, kas tiek izmantots, lai nosūtītu pavēles Baxter robotam, izmantojot *rospy* un *baxter_interface* bibliotēkas. Programmā no sākuma definē šļa izmēru katrai locītavai (0,5 radiāni). Tālāk tā iegūst vēlamos locītavu leņķus no servera,

kas raksturots otrajā risinājuma daļā. Tālāk tā iteratīvi salīdzina robota locītavu tagad esošos aprēķinātos leņķus ar vēlamajiem leņķiem un pakustina locītavu vajadzīgajā virzienā par vienu soli. Ja attālums starp tagad esošu aprēķinātu leņķi un vēlamo leņķi konkrētai locītavai ir mazāks par soli, soļa izmērs tiek pārmainīts uz pusi mazāks. Programma iteratīvi kustina robota locītavas uz vēlamajiem leņķiem (robota rokas gals tuvojas vēlamajam galapunktam) līdz tās tos sasniedz vai arī programma sasniegusi specificēto maksimālo iterāciju skaitu.

Risinājuma demonstrāciju var skatīt 2. pielikumā.

4.2. IMU risinājums

Alternatīvs risinājums Baxter IK kontrolēšanai ir pielietot Kinect2 robota nepieciešamo leņķu aprēķināšanai un paļauties uz jau eksistējošo risinājumu robota locītavu nogādāšanai vajadzīgajos leņķos. Šis risinājums ir lielākoties identisks 4.1. aprakstītajam risinājumam, bet robota kontrole nepaļaujas uz Kinect2. Šis risinājums pieprasa noskaidrot vajadzīgos leņķus no servera, kas aprakstīts 4.1. sadaļā, tad sūta robotam pavēli kustināt roku uz iegūtajiem leņķiem, lietojot *baxter_interface* bibliotēku ar ROS.

4.3. IMU-multi risinājums

Darba izstrādes laikā darba autors novēroja, ka robota neprecīzās pavēļu izpildes var uzlabot, robotam nosūtot vienu un to pašu pavēli vairākas reizes. Tāpēc šis risinājums nosūta robotam pavēli doties uz vēlamajām koordinātām atkārtoti. Lai noskaidrotu, kad beigt sūtīt robotam pavēles, pēc katras nosūtītās pavēles tiek pārbaudīts robota rokas galapunkts. Pavēles beidz sūtīt, kad robota roka ir sasniegusi vēlamo galapunktu, Kinect2 sensors nenovēro izmaiņas rokas pozīcijā pēc pavēles izpildes vai pavēle tiek nosūtīta jau trīsdesmito reizi.

4.4. Risinājuma salīdzināšanas metodoloģija

Darba eksperimentālajā daļā tiks salīdzināti Kinect2 risinājums, IMU risinājums un IMU-multi risinājums. Robots un Kinect2 sensors tiek novietoti 2,5 metru attālumā. Katrs risinājums tiek pārbaudīts atsevišķi pie deviņiem vēlamajiem galapunktiem (skat. 1. tabulu). 1. tabulā tiek rādīti deviņi galapunkti, to attālums no robota (\mathbf{l}), galapunkta augstums (\mathbf{h}) un leņķis starp galapunktu un robota centru ($\boldsymbol{\alpha}$). Risinājuma precizitātes pārbaudei pie katra punkta tiek izvirzīti šādi soļi:

1. Novietot vidē galapunktu, kuram Baxter robotam ir precīzi jāpieskaras ar rokas galu. Vēlamais galapunkts ir taisnas metāla stieples gals, kas pievienots pie kameras statīva (trīskāja).
2. Dot robotam pavēli pieskarties iepriekš minētajam punktam, izmantojot konkrēto risinājumu.
3. Izmērīt robota rokas galu pozīciju (milimetros) ar mērlentes palīdzību ($\pm 10\text{mm}$) relatīvi konkrētā punkta (\mathbf{EE}) atrašanās vietai - izmērīt \mathbf{x} , \mathbf{y} un \mathbf{z} novirzi, attiecīgi atzīmētu kā $\Delta\mathbf{x}$, $\Delta\mathbf{y}$, $\Delta\mathbf{z}$.

5. Rezultāti

Balstoties uz rezultātiem, Kinect2 risinājuma vidējā precizitāte (\bar{l}) ir 121mm un precizitātes mediāna (\tilde{l}) ir 86mm (skat. 2. tabulu). Tomēr IMU risinājuma vidējā precizitāte (\bar{l}) ir 407mm un precizitātes mediāna (\tilde{l}) ir 353mm (skat. 3. tabulu). Turklāt, IMU-multi risinājuma vidējā precizitāte (\bar{l}) ir 113mm un precizitātes mediāna (\tilde{l}) ir 92mm (skat. 4. tabulu).

Eksperimenta laikā darba autoram nācās eksperimentu veikt ar mākslīgo apgaismojumu, lai Kinect2 risinājums spētu veiksmīgi atpazīt visas krāsas kadrā. Turklāt, darba autoram nācās nodrošināt, ka kadra fonā nebija redzama neviena no izvēlētajām locītavu krāsām.

<i>EE</i>	<i>l</i> (cm)	<i>h</i> (cm)	α (grādi)
1.	90	100	60
2.	75	100	45
3.	90	100	90
4.	80	100	30
5.	60	80	70
6.	65	80	90
7.	75	80	45
8.	75	120	45
9.	45	120	45

1. tabula. Galapunktu (***EE***) koordinātas

<i>EE</i> nr.	Δx (mm)	Δy (mm)	Δz (mm)	<i>l</i> (mm)
1.	10	50	70	87
2.	0	50	65	82
3.	110	150	180	259
4.	80	60	40	108
5.	100	10	110	149
6.	100	40	160	193
7.	10	50	70	87
8.	20	10	50	55
9.	20	10	60	65

2. tabula. Kinect2 risinājuma rezultāti

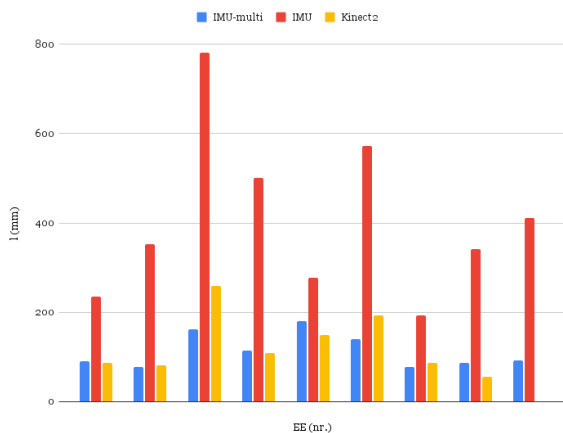
<i>EE</i> nr.	Δx (mm)	Δy (mm)	Δz (mm)	<i>l</i> (mm)
1.	80	70	210	235
2.	300	50	180	353
3.	420	250	610	782
4.	400	20	300	500
5.	120	100	230	278
6.	170	130	530	572
7.	140	30	130	193
8.	130	310	60	341
9.	410	40	20	412

3. tabula. IMU risinājuma rezultāti

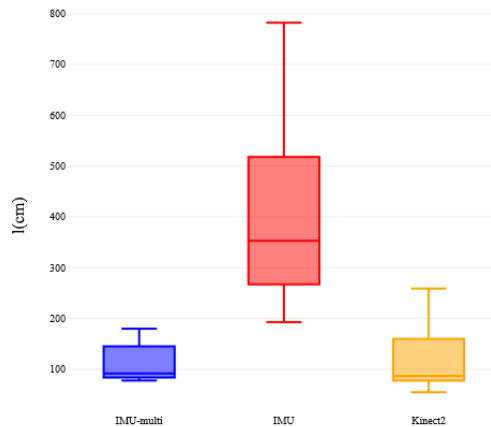
<i>EE</i> nr.	Δx (mm)	Δy (mm)	Δz (mm)	<i>l</i> (mm)
1.	30	50	70	91
2.	0	50	60	78
3.	80	50	130	161
4.	70	90	0	114
5.	30	170	50	180
6.	0	140	10	140
7.	30	60	40	78
8.	40	30	70	86
9.	45	25	80	92

4. tabula. IMU-multi rezultāti

Vieglākai informācijas uztverei, rezultāti apkopoti divās diagrammās. Diagrammās redzams Kinect2 risinājums (dzeltenā krāsā), IMU risinājums (sarkanā krāsā) un IMU-multi (zilā krāsā). Vienā no divām diagrammām ir redzama risinājuma precizitāte pie katra punkta (skat. 8. attēlu). Otrā no divām diagrammām redzams rezultātu datu apkopojums kastu diagrammā (skat. 9. attēlu).



8. att. Precizitātes vērtības (l)



9. att. Kinect, IMU un IMU-multi risinājumu rezultāti

Lai noskaidrotu, vai risinājumu rezultātu starpības ir statistiski nozīmīgas, šajā sadaļā rezultāti tiek salīdzināti ar neatkarīgu kopu T testa palīdzību, kas aprakstīts [11] avotā. Šī pētījuma nolūkos, mēs pieņemsim, ka rezultāti ir statistiski nozīmīgi, ja testa p-vērtības ir mazākas par 0,05.

Analizējot Kinect2 un IMU risinājuma rezultātus, darba autors secina, ka rezultātu starpība ir statistiski nozīmīga ($t(8) \approx 0,0006$, p-vērtība $\approx 0,0005$). Tātad, Kinect2 risinājums bija par 336% precīzāks par IMU risinājumu, un tā vidējā nobīde (\bar{l}) ir par 286mm mazāka.

Turklāt analizējot IMU-multi un IMU risinājuma rezultātus, var secināt, ka rezultātu starpība arī ir statistiski nozīmīga ($t(8) \approx 0,0004$, p-vērtība $\approx 0,0003$). Tātad, IMU-multi risinājums bija par 360% precīzāks par IMU risinājumu, un tā vidējā nobīde (\bar{l}) ir par 294mm mazāka.

Tomēr, analizējot Kinect2 un IMU-multi risinājumu, var secināt, ka rezultātu starpība nav statistiski nozīmīga ($t(8) \approx 1,331$, p-vērtība $\approx 0,7836$). Tāpēc nav iespējams apgalvot, ka eksperimenta rezultāti ir statistiski nozīmīgi un viens risinājums ir noteikti precīzāks par otru.

6. Secinājumi

Darba autors secina, ka jaunieviestie Kinect2 un IMU-multi risinājumi ir praktiski pielietojami Baxter robota inversās kinemātikas uzlabošanai.

1. Pētījuma hipotēze ir apstiprināta, jo Kinect2 un IMU-multi risinājumi ir precīzāki par jau esošo IMU risinājumu un ir praktiski pielietojami Baxter robota inversās kinemātikas vadīšanai.
2. Salīdzinot risinājumu vidējo precizitāti, secina, ka Kinect2 risinājums ir 336% precīzāks par IMU risinājumu un IMU-multi risinājums ir vidēji 360% precīzāks. Tomēr eksperimentālajā daļā netika pierādīts, ka ir statistiski nozīmīga starpība starp Kinect2 un IMU-multi risinājuma precizitāti.

3. Kinect2 risinājums paļaujas uz krāsu atpazīšanu kadrā. Šī iemesla dēļ, šis risinājums ir pielietojams tikai vidē ar piemērotu apgaismojumu. Kā arī, šis risinājums paļaujas, ka fonā nav kāda no izvēlētajām locītavu krāsām.
4. Tika sasniegti visi izvirzītie darba uzdevumi.

7. Tālākie pētījumi

Kinect2 risinājums balstās uz krāsu atpazīšanu, lai noteiktu robota locītavas koordinātas RGB kadrā. Tomēr šis risinājums nav pielietojams, piemēram, situācijā, kurā vajadzīgs kontrolēt abas robota rokas viegli atpazīstamu krāsu trūkuma dēļ. Kā arī šis risinājums paļaujas, ka kadra fonā nekur citur nav redzamas izvēlētas locītavu krāsas, un ka kadra apgaismojums ir piemērots iestatītajām krāsu vērtībām. Uzlabots risinājums varētu izmantot objektu detekciju ar mākslīgo intelektu [8], lai atpazītu locītavas.

Turklāt autora risinājums aprēķina inversās kinemātikas (IK) divās dimensijās un paļaujas uz S_0 , lai virzītu roku uz vēlamo galapunktu, izmantojot tikai S_0 , S_1 , E_1 un W_1 locītavas. Uzlabojums varētu ietvert alternatīvu izteiksmi (5.), kas aprēķina septiņas θ vērtības ar struktūras galapunktu 3D telpā.

8. Izmantotā literatūra

- [1] Mahmoud Ayach. “A Guide to Gradient Descent and Stochastic Gradient Descent”. *Tech Blog* (2023). Skatīts 13.01.2024.
- [2] Baxter. <https://robotsguide.com/robots/baxter?interactive=3>. Skatīts 13.01.2024. 2012.
- [3] C. Dal Mutto u. c. “Time-of-Flight Cameras and Microsoft Kinect™”. *Springer* (2013). Skatīts 13.01.2024. URL: <https://lstm.dei.unipd.it/nuovo/Papers/ToF-Kinect-book.pdf>.
- [4] E. Lachat u. c. “FIRST EXPERIENCES WITH KINECT V2 SENSOR FOR CLOSE RANGE 3D MODELLING”. *ISPRS Archives* (2015). Skatīts 13.01.2024. URL: <https://isprs-archives.copernicus.org/articles/XL-5-W4/93/2015/isprsarchives-XL-5-W4-93-2015.pdf>.
- [5] Ala Al-Fuqaha u. c. “Techniques of Indoor Positioning Systems (IPS): A Survey”. *IEEE Access* (2019). Skatīts 13.01.2024. URL: https://www.researchgate.net/figure/The-internals-of-Kinect-V2-showing-the-principle-of-TOF_fig2_333151664.
- [6] T.L. Harman un C. Fairchild. *INTRODUCTION TO BAXTER*. https://sceweb.sce.uhcl.edu/harman/CENG5931Baxter2015/Guides/BAXTER_Introduction_2_08_2016.pdf. Skatīts 13.01.2024. 2015.
- [7] International Federation of Robotics. *World Robotics 2023 Report: Asia ahead of Europe and the Americas*. 2024. URL: <https://ifr.org/ifr-press-releases/news/world-robotics-2023-report-asia-ahead-of-europe-and-the-americas>.
- [8] Ayush Jain u. c. “AI-enabled object detection in UAVs: challenges, design choices, and research directions”. *IEEE Network* (2021). Skatīts 13.01.2024.
- [9] Morgan Quigley u. c. “ROS: an open-source Robot Operating System”. Skatīts 13.01.2024. IEEE. 2009. URL: <http://robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf>.
- [10] M. Soch un R. Lórencz. “Solving Inverse Kinematics – A New Approach to the Extended Jacobian Technique”. Skatīts 13.01.2024. 2005. URL: <https://api.semanticscholar.org/CorpusID:59584981>.
- [11] R. R. Wadhwa un R. Marappa-Ganeshan. “T Test”. *StatPearls [Internet]* (2023). Skatīts 13.01.2024. URL: <https://www.ncbi.nlm.nih.gov/books/NBK553048/>.
- [12] RethinkRobotics. *Baxter Research Robot Python Interfaces for the Baxter SDK*. https://github.com/RethinkRobotics/baxter_interface. Skatīts 13.01.2024.
- [13] ROS Development Team. *rospy*. <http://wiki.ros.org/rospy>. Skatīts 13.01.2024.
- [14] Steve Rotenberg. “Inverse Kinematics 1”. Lekcijas prezentācija, Skatīts 13.01.2024. 2017. URL: https://cseweb.ucsd.edu/classes/wi17/cse169-a/slides/CSE169_08.pdf.
- [15] M. W. Spong, S. Hutchinson un M. Vidyasagar. “Time Derivative of Rotation Matrices: A Tutorial”. *Robot Modeling and Control* (2006). Skatīts 13.01.2024.

- [16] Juan Terven u. c. *Loss Functions and Metrics in Deep Learning. A Review*. Skatīts 13.01.2024. 2023.
- [17] Ē. Urtāns. “2023-Q1-AI 3. Inversā matemātika - parciāl-atvasinājumi”. *Youtube* (2023). Skatīts 13.01.2024. URL: <https://www.youtube.com/watch?v=tUCKoy8Vv8M>.
- [18] R. L. Williams. *Baxter Humanoid Robot Kinematics*. Skatīts 13.01.2024. 2012. URL: https://www.academia.edu/35398090/Baxter_Kinematics.

9. Pielikumi

1. pielikums - Darbā izstrādātā risinājuma kods: <https://github.com/theglobehead/baxter-kinect2>
2. pielikums - Kinect2 risinājuma demonstrācija <https://www.youtube.com/watch?v=NULSNqqtyNg>