

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1719

**Razvoj sustava globalne vizije za  
testiranje algoritama upravljanja  
bespilotnim letjelicama**

Bojan Spahija

Zagreb, srpanj 2018.

Zagreb, 14. ožujka 2018.

## DIPLOMSKI ZADATAK br. 1719

Pristupnik: **Bojan Spahija (0036476788)**  
Studij: Elektrotehnika i informacijska tehnologija  
Profil: Automatika


Zadatak: **Razvoj sustava globalne vizije za testiranje algoritama upravljanja  
bespilotnim letjelicama**

**Opis zadatka:**

U radu je potrebno osmisliti i razviti jednostavan i ekonomski prihvatljiv sustav globalne vizije za testiranje algoritama upravljanja bespilotnim letjelicama. Sustav treba omogućiti pouzadano određivanje pozicije i orijentacije letjelice na temelju informacija s više infracrvenih kamera postavljenih na prikladne pozicije u prostoru. Razvijeni sustav je potrebno eksperimentalno testirati i validirati.

Zadatak uručen pristupniku: 16. ožujka 2018.  
Rok za predaju rada: 29. lipnja 2018.

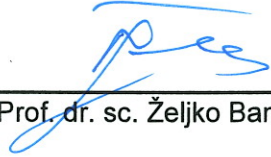
Mentor:

  
Izv. prof. dr. sc. Jadranko Matuško

Djelovođa:

  
Doc. dr. sc. Matko Orsag

Predsjednik odbora za  
diplomski rad profila:

  
Prof. dr. sc. Željko Ban

*Svim ljudima dobre volje*

# SADRŽAJ

<b>Popis slika</b>	<b>vi</b>
<b>1. Uvod</b>	<b>1</b>
<b>2. Postav</b>	<b>2</b>
2.1. Izbor kamera, IC izvora i markera . . . . .	2
2.1.1. Kamera . . . . .	3
2.1.2. Infracrveni LED izvor . . . . .	4
2.1.3. Marker . . . . .	5
2.2. Pozicija kamera i markera . . . . .	7
2.2.1. Nosač za Wiimote . . . . .	8
2.2.2. Radni prostor . . . . .	9
<b>3. Komunikacija</b>	<b>11</b>
3.1. ROS - Robot Operating System . . . . .	11
3.2. ROS - Implementacija . . . . .	12
<b>4. Izračun pozicije i orijentacije</b>	<b>14</b>
4.1. Geometrijski postupak . . . . .	15
4.1.1. Formiranje slike u pinhole kameri . . . . .	15
4.1.2. Određivanje točke pravicima . . . . .	16
4.2. Postupak pomoću neuronske mreže . . . . .	20
4.2.1. Osnove neuronskih mreža . . . . .	20
4.2.2. Implementacija neuronske mreže . . . . .	22
<b>5. Rezultati</b>	<b>25</b>
5.1. Geometrijska metoda . . . . .	25
5.2. Metoda s neuronskom mrežom . . . . .	27

<b>6. Zaključak</b>	<b>28</b>
<b>Literatura</b>	<b>29</b>

# POPIS SLIKA

2.1. Prikaz infracrvenog zračenja na spektru elektromagnetskih zračenja . . . . .	3
2.2. Testirane infracrvene kamere . . . . .	4
2.3. Infracrveni SMD LED modul . . . . .	5
2.4. Okrugli OptiTrack markeri postavljeni na dronu . . . . .	6
2.5. Odbijanje svjetlosti od retroreflektirajuće prizme . . . . .	6
2.6. Demonstracija osvjjetljenja reflektirajućeg prsluka . . . . .	7
2.7. Dron sa dva LED markera . . . . .	7
2.8. Pano sa postavljenim kamerama . . . . .	8
2.9. Modeli elemenata modularnog nosača . . . . .	9
2.10. Modularni Wiimote nosač . . . . .	9
2.11. Označen radni prostor . . . . .	10
3.1. ROS Kinetic Kame logo . . . . .	11
3.2. Pojednostavljeni prikaz komunikacijske strukture ROS-a . . . . .	12
4.1. Projekcija markera na Wiimoteove . . . . .	14
4.2. Projekcija točke na slikovnu ravninu u <i>pinhole</i> kameri . . . . .	15
4.3. Prikaz pravaca $l_1$ i $l_2$ te najbližih točaka $x_1$ i $x_2$ . . . . .	18
4.4. Model biološkog neurona . . . . .	21
4.5. Model perceptrona . . . . .	21
4.6. Neuronska mreža korištena u MATLAB-u . . . . .	23
5.1. Pravci, izračunate točke i stvarna točka za centralnu točku . . . . .	25
5.2. Pravci, izračunate točke i stvarna točka za točku u blizini centralne . . . . .	26
5.3. Pravci, izračunate točke i stvarna točka za rubnu točku . . . . .	26

# 1. Uvod

U mnogim granama industrije veliki značaj ima praćenje kretanja. Praćenje pokreta koristi se kod animacija za filmsku industriju i industriju video igara. Koristi se u vojnoj industriji za simulatore i u znanstvene svrhe za proučavanje kretanja ljudi i životinja. Zbog napretka robotike u posljednjih nekoliko godina, sve je veća potreba za preciznim praćenjem dronova i drugih robota s ciljem poboljšanja i razvoja algoritama autonomnog upravljanja.

Trenutna rješenja praćenja pokreta poput OptiTrack-a omogućuju iznimno precizno praćenje, ali vrlo su skupa. Cijene se kreću od nekoliko tisuća dolara do nekoliko desetaka tisuća dolara što nije prikladno za male firme, pojedince ili projekte za koje nije potrebna tolika preciznost.

## 2. Postav

Za 3D lokalizaciju bespilotne letjelice potrebne su minimalno dvije kamere. Pomoću dvije kamere i markera koje te kamere mogu detektirati moguće je dobiti lokaciju tog markera odnosno x-y-z koordinate markera u globalnom koordinatnom sustavu. Kod upravljanja letjelicom važno je poznavati i orijentaciju letjelice (yaw).

Upravljačke vrijednosti koje šalje letjelici imaju direktan utjecaj na gibanje letjelice u njezinom lokalnom koordinatnom sustavu. Da bi se poznao utjecaj upravljačkih vrijednosti na gibanje u globalnom koordinatnom sustavu, potrebno je znati za koliko je zarotiran lokalni koordinatni sustav letjelice u odnosu na globalni koordinatni sustav (orijentacija). Kako bismo dobili informaciju o orijentaciji, potreban je specifični raspored markera koji nam omogućuje prepoznavanje orijentacije bez poznavanja orijentacije u prethodnim trenucima.

### 2.1. Izbor kamera, IC izvora i markera

Sustav globalne vizije mora zadovoljiti određene zahtjeve i uvjete:

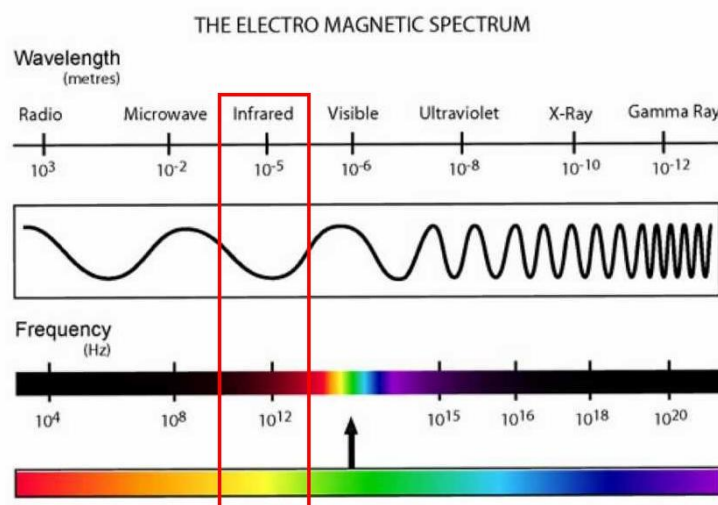
- Detekcija će se odvijati u zatvorenom prostoru
- Maksimalna udaljenost markera od kamere je oko 4m
- Letjelica je malih dimenzija što limitira dimenziju markera

Zbog relativno velike udaljenosti na kojoj detekcija markera mora raditi i malih dimenzija markera, detekcija pomoću RGB kamera i markera u bojama nije pouzdana. U sustavu globalne vizije koristit će se detekcija pomoću infracrvene svjetlosti.

Infracrvena svjetlost je pogodna za detekciju na većim udaljenostima i uvjetima promjenjivog osvijetljenja. Većina materijala iz okoline reflektira vrlo malo infracrvene svjetlosti što osigurava pouzdanu detekciju infracrvenih izvora i materijala koji dobro reflektiraju infracrvenu svjetlost. Za detekciju pomoću infracrvene svjetlosti potrebni su okrugli markeri koji dobro reflektiraju infracrvenu svjetlost u svim smjerovima, kamere koje će detektirati svjetlost reflektiranu od markera i izvor infracrvene svjetlosti



dovoljno velikog intenziteta i kuta zračenja da pokrije cijelo vidno polje kamere i željenu maksimalnu udaljenost markera.



**Slika 2.1:** Prikaz infracrvenog zračenja na spektru elektromagnetskih zračenja

### 2.1.1. Kamera

Testirane kamere su bile Pixy kamera sa IR-LOCK infracrvenom lećom (slika 2.2b) i kamera unutar Wiimotea (slika 2.2a) koja ima infracrvenu leću već ugrađenu. Obje kamere vraćaju piksele slike na kojima je detektirano infracrveno zračenje. Valna duljina infracrvenog zračenja koje kamere najbolje detektiraju iznosi oko 940nm. Wiimote šalje piksele od 4 najintenzivije infracrvene točke dok ih Pixy kamera može detektirati mnogo više, što u ovom slučaju ne igra ulogu zbog samo 3 potrebne točke.

Veza između Wiimotea i računala postiže se Bluetooth povezivanjem. Takva veza omogućuje veću udaljenost kamere i veću fleksibilnost kod postavljanja kamere. Direktna veza između računala i Pixy kamere može se postići mini USB kablom, a ako je nužna bežična veza potrebno je spojiti Pixy kameru na mikrokontroler koji će onda komunicirati bežično s računalom.

Također je važan vidni kut kamere jer on određuje veličinu prostora koji će se moći nadzirati. Kod Wiimote kamere horizontalni vidni kut iznosi oko  $42^\circ$ , dok vertikalni vidni kut iznosi oko  $32^\circ$ . Pixy kamera pokriva veće područje s horizontalnim vidnim kutem od  $75^\circ$  i vertikalnim vidnim kutem od  $47^\circ$ . U segmentu vidnog polja Pixy kamera ima prednost.

Rezolucija kamere će utjecati na preciznost izračuna 3D globalnih koordinata točke. Niže rezolucije rezultiraju većim pogreškama određivanja pozicije objekata u daljini.

Wiimote kamera ima vrlo malu rezoluciju od 128x96. Procesor unutar Wiimotea povećava tu rezoluciju na 1024x768 pomoću postupka analize intenziteta rubnih piksela i izračuna pozicije infracrvene točke unutar piksela (8x subpixel analysis). Ovaj postupak povećanja rezolucije kamere uvelike poboljšava preciznost detekcije usprkos korištenju jeftinih komponenata (infracrvena kamera niske rezolucije i mikroprocesor). S druge strane, Pixy kamera ima ugrađenu mnogo kvalitetniju Omnivision OV9715 kameru veće rezolucije (1280x800). Pri takvoj rezoluciji, Pixy kamera radi na 25 sličica u sekundi. Kako bi se postigla veća brzina rada kamere (50 sličica u sekundi), smanjena je rezolucija na 640x400. Zbog ograničenja RAM-a koji Pixy koristi za obradu slike, rezolucija je dodatno smanjena na 320x200 piksela. Veća rezolucija Wiimotea ga čini boljim izborom zbog manje greške detekcije, a i manje cijene. Wiimote je moguće nabaviti po cijeni nižoj od \$20, dok Pixy kamera košta \$69.99.

Wiimote je bolji izbor zbog rezolucije, lakšeg povezivanja i postavljanja te niže cijene. Jedina mana mu je relativno usko vidno polje, ali taj problem se može riješiti dobrim pozicioniranjem kamere na dovoljnu udaljenost od radnog područja drona.



(a) Wiimote kontroler



(b) Pixy kamera

**Slika 2.2:** Testirane infracrvene kamere

### 2.1.2. Infracrveni LED izvor

Potrebno je montirati izvor infracrvene svjetlosti uz kameru i usmjeriti ga u smjeru gledanja kamere kako bi se dobila čim bolja detekcija markera. Izvor mora biti dovoljnog intenziteta da infracrvena svjetlost bude detektirana kamerom nakon što prođe udaljenost između kamere i markera, reflektira se od markera i vrati opet do kamere. U najudaljenijoj točki radnog prostora drona ta će udaljenost biti oko 4m. Obične jeftine

infracrvene diode male snage takvu udaljenost neće moći uspješno pokriti za detekciju. Dodatni problem s takvim infracrvenim diodama je relativno mali kut zračenja koji iznosi najčešće između  $15^\circ$  i  $30^\circ$ . Da bi se pokrio cijeli radni prostor drona i vidno polje kamere potrebno bi bilo staviti više takvih dioda oko kamere.

Kako bi cijeli radni prostor bio dobro osvijetljen, koristiti će se infracrveni LED čipovi visoke snage od 3W (SMD LED Module) koji generira svjetlost valne duljine od 940nm (Slika 2.3). Osim veće udaljenosti koju pokriva, prednost ovakve vrste LED izvora je u širokom kutu zračenja koji iznosi  $120^\circ$ . Tijekom testiranja ispostavilo se da 3W infracrveni LED izvori u kombinaciji s reflektirajućim markerom omogućuju detekciju markera na premaloj udaljenosti od oko 1m. Da bi se ostvarila detekcija pomoću reflektirajućeg markera potrebno bi bilo koristiti izvor jači od 10W. Zbog širokog kuta zračenja i visoke snage, LED izvori mogu se dobro detektirati na udaljenosti do 3m pa su pogodni za markere ako se montiraju na drona.

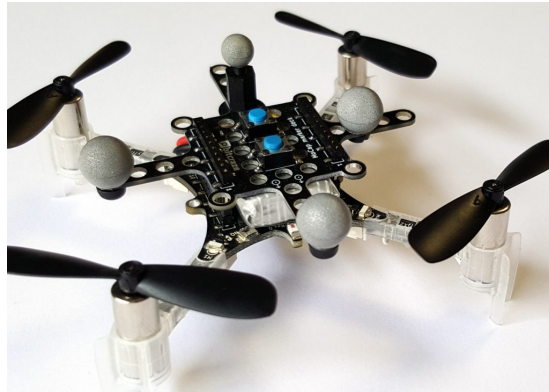


**Slika 2.3:** Infracrveni SMD LED modul

### 2.1.3. Marker

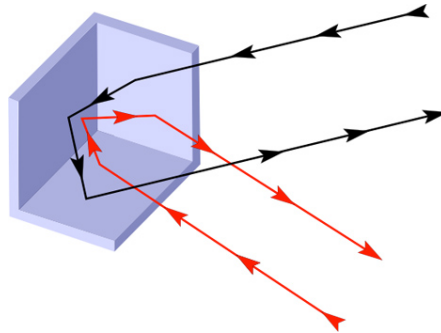
Markeri koji će se nalaziti na bespilotnoj letjelici moraju biti što lakši zbog iznimno male nosivosti letjelice. Drugi zahtjev je da reflektiraju što više svjetlosti natrag prema kameri neovisno o položaju letjelice i markera. Materijali koji dobro reflektiraju infracrvenu svjetlost su neki metali (Al, Ag i Au) te oksidi metala poput  $\text{SiO}_2$  i  $\text{Al}_2\text{O}_3$ . Ukoliko bi se koristili premazi ili boje koje sadrže navedene spojeve i elemente, oni bi se trebali nanjeti na lagani materijal oblika kugle koji će reflektirati svjetlost natrag kameri. U ovom slučaju važno je da marker ima oblik kugle kako bi se dio reflektirane svjetlosti, neovisno o poziciji markera i kamere, uvijek vraćao istim putem natrag

prema kameri. Stiroporne kuglice zadovoljavaju zahtjeve za marker koji će biti premazan.



**Slika 2.4:** Okrugli OptiTrack markeri postavljeni na dronu

Druga opcija je korištenje retroreflektirajućih materijala. Retroreflektirajući materijali reflektiraju svjetlost natrag izvoru svjetlosti uz minimalno rasipanje. Zraka svjetlosti koja dolazi prema retroreflektoru je paralelna i suprotnog smjera od reflektirane zrake (Slika 2.5). Retroreflektori su najčešće male staklene ili plastične prizme. Koriste se u prometu u boji za linije na cesti i prometne znakove, kod reflektirajućih prsluka, očiju za bicikl, itd. U slučaju odabira retroreflektora nije nužno koristiti marker oblika kugle zbog svojstva da vraća svjetlost prema izvoru neovisno o kutu upada.



**Slika 2.5:** Odbijanje svjetlosti od retroreflektirajuće prizme

Zbog infracrvenog izvora premalog intenziteta nije moguće koristiti reflektirajuće markere pa će se sam izvor koristiti kao marker. Tri izvora pričvršćena su na bespilotnu letjelicu i spojena su u paraleli na dronovu bateriju. Ovim se zadovoljavaju uvjeti detekcije, ali se skraćuje trajanje baterije zbog dodatnog napajanja LED izvora.



**Slika 2.6:** Demonstracija osvjetljenja reflektirajućeg prsluka

## 2.2. Pozicija kamera i markera

Za preciznu i robusnu detekciju važno je pozicionirati kamere i markere tako da u svakom mogućem stanju letjelice unutar radnog prostora markeri budu detektirani. Robusnost je moguće postići postavljanjem viška kamera kako bi u svakom slučaju barem tri kamere vidjele markere. Također je opcija ograničenje radnog prostora letjelice tako da smo sigurni da u svim pozicijama sve kamere vide markere.

Koriste se tri markera pozicionirana na vrhu letjelice. Markeri su postavljeni u formaciju trokuta sa jednim markerom ispred prednjeg kraja letjelice dok su druga dva markera iza stražnjeg kraja letjelice. Razmak između stražnjih markera je manji od razmaka između stražnjih i prednjeg kako bi se detektirala orijentacija letjelice.



**Slika 2.7:** Dron sa dva LED markera

Obje kamere moraju detektirati sva tri markera da bi se mogla odrediti pozicija i orijentacija letjelice u globalnom koordinatnom sustavu. Kamere su postavljene na

pano visine 1.88m i širine 1.25m. Pozicionirane su na rubovima panoa kako bi bile što udaljenije i pokrivale što veći radni prostor. Nagib kamera prema dolje iznosi  $48^\circ$  (rotacija oko x-osi kamere). Kamere su zarotirane oko njihove y-osi (lijevo-desno) za  $20^\circ$  jedna prema drugoj kako bi se dobio što veći radni prostor. Iz poznavanja maksimalne udaljenosti detekcije koja iznosi 3m te horizontalnog i vertikalnog vidnog kuta Wiimote kamera određuje se područje unutar kojeg će obje kamere detektirati sva tri LED markera. Rubovi zone detekcije obilježeni su ljepljivom vrpcom kako bi se dron uvijek mogao precizno postaviti za eksperimente.



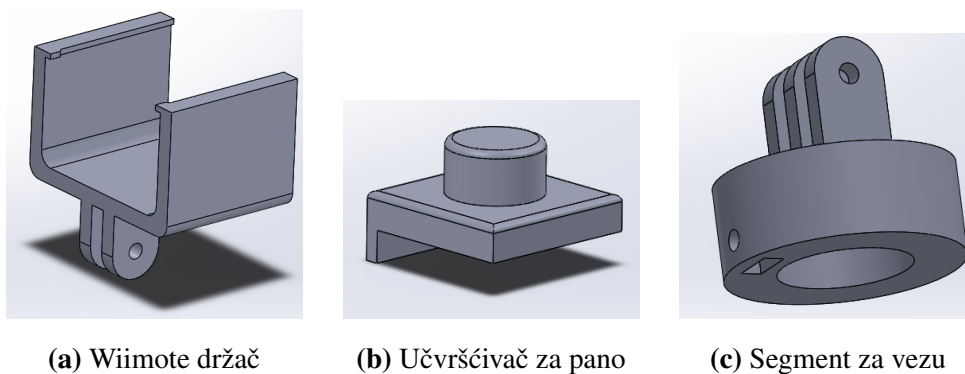
**Slika 2.8:** Pano sa postavljenim kamerama

### 2.2.1. Nosač za Wiimote

Potrebno je napraviti nosač za Wiimote kamere kako bi se dobila precizna orijentacija kamera i čvrsta veza sa objektom na kojem će kamera biti postavljena (pano). Nosač za kameru mora imati mogućnost zakreta kamere oko dvije osi, osi x i y (gore-dolje, lijevo-desno). Iz izmjerenih dimenzija Wiimotea i segmenta panoa na koji će se nosač pričvrstiti, napravljen je model modularnog nosač u Solidworksu i isprintan pomoću 3D printera. Nosač se sastoji od 3 dijela:

1. Wiimote držač
2. Učvršćivač za pano
3. Segment za vezu između fiksiranih dijelova

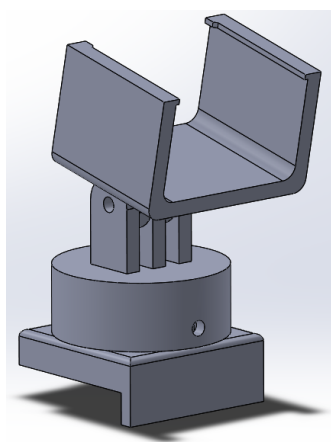
Veza između držača Wiimotea i segmenta za vezu ostvarena je preko ploha koje sjedaju jedne između drugih. Dok su plohe poravnate još postoji mjesta za micanje pa je



**Slika 2.9:** Modeli elemenata modularnog nosača

moguće postaviti željenu rotaciju Wiimotea oko x-osi (gore-dolje). Kada je postavljeni željeni kut rotacije, umetanjem M3 vijka kroz rupu u ploham i njegovim zatezanjem pomoću matice učvršćuje se veza i osigurava željena rotacija.

Veza između dijela učvršćenog na pano i segmenta za povezivanje ostvarena je preko cilindra na koji se postavlja segment za povezivanje. Segment za povezivanje se može slobodno rotirati oko vertikalne osi (lijevo-desno). Kad se postavi željeni zakret Wiimotea, veza se učvršćuje pomoću M3 matice i vijka.



(a) Spojeni Solidworks model



(b) Isprintani model

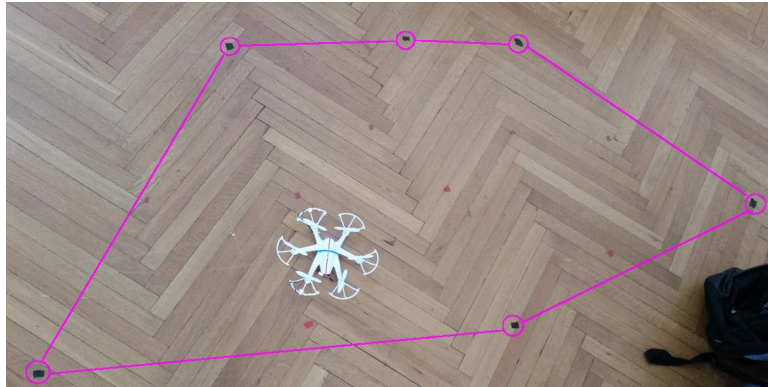
**Slika 2.10:** Modularni Wiimote nosač

### 2.2.2. Radni prostor

Iz pozicija i orijentacija kamera, poznatog vidnog kuta i maksimalne udaljenosti na kojoj Wiimoteovi detektiraju infracrveni LED marker računa se prostor unutar kojeg je kamera u mogućnosti detektirati marker. Taj postupak se radi za obje kamere. Presjek



prostora detekcije prve i druge kamere je radni prostor. Ako dron ostane unutar tog prostora njegova pozicija i orijentacija će biti dobivene iz sustava globalne vizije pa je važno postaviti ograničenja radnog prostora u algoritme upravljanja dronom kako on ne bi izašao tog prostora.



**Slika 2.11:** Označen radni prostor

Zbog ručnog postavljanja zakreta kamera koje unosi greške od nekoliko stupnjeva i loše detekcije markera na rubnim dijelovima vidnog polja kamera, određivanje rubova radnog prostora izvršeno je eksperimentalno.



## 3. Komunikacija

Oba Wiimotea se povezuju bežično preko Bluetooth veze sa osobnim računalom. Računalo mora na neki način dohvatiti bitne informacije iz kamera, iz njih izračunati poziciju i orijentaciju drona u prostoru te na kraju objaviti izračunate vrijednosti kako bi se one mogle dohvatiti od strane upravljačkog algoritma. Za preciznu detekciju i pouzdano upravljanje važno je da su mjerenja kamera sinkronizirana te da je poznata frekvencija izračuna i objave pozicije drona. Frekvencija kojom Wiimote šalje podatke iznosi 100Hz. Za upravljanje po poziciji takva brzina je nepotrebna pa će se uzimati podaci frekvencijom od 20Hz. Robotski operacijski sustav će se koristiti za uspostavu komunikacije jer zadovoljava sve uvjete i omogućuje precizno i brzo izvršavanje više Python ili C++ skripti za dohvat, obradu i objavu informacija.

### 3.1. ROS - Robot Operating System



Slika 3.1: ROS Kinetic Kame logo

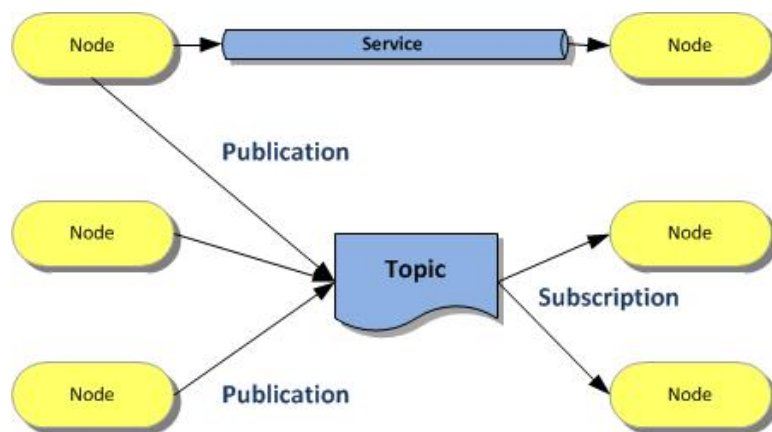
Velika složenost zadataka koji se postavljaju pred moderne robote zahtijeva korištenje programske infrastrukture koja omogućuje efikasno povezivanje nezavisno razvijenih podsustava u jedinstveni sustav koji može omogućiti autonomno djelovanje robota. Robotski operacijski sustav (ROS) pruža okolinu za razvoj modularne uprav-

ljačke programske podrške, komunikacijsku infrastrukturu koja povezuje programske komponente, te otvorenu biblioteku implementiranih algoritama. U posljednjih 5 godina, ROS je postao de facto standard za upravljanje robotima unutar akademske zajednice, te se njegova primjena širi i na industriju.

ROS nam u ovom projektu olakšava komunikaciju između nezavisnih Python skripti preko publish/subscribe sistema. Također nam omogućuje lako snimanje podataka koje objavljuje neka od skripti i ponovno objavljivanje tih podataka kasnije.

ROS čvor (node) je proces koji vrši izračune i obradu informacija. Više čvorova međusobno komunicira preko tema (topics). Kontrolni sustav će obično imati mnogo čvorova koji vrše manje specijalizirane zadatke. Prednost ovakvog sustava s mnogo čvorova je otpornost na greške jer se greške i rušenja u programu svode na rušenja pojedinog čvora, a ne cijelog sustava. Složenost koda je također smanjena zbog modularnosti te je testiranje sustava jednostavnije.

ROS teme su imenovani kanali kroz koje se odvija anonimna komunikacija između čvorova. Čvorovi se mogu pretplatiti na bilo koju od prisutnih tema pomoću Subscriber objekta ili mogu objavljivati podatke u zadanoj temi pomoću Publisher objekta. Tema može imati više pretplaćenih čvorova (subscribers) i objavljujućih čvorova (publishers). Vrsta teme ovisi o tipu poruke koja je objavljena na nju. Pretplaćeni čvorovi moraju dohvaćati tip podataka koji odgovara vrsti teme.



**Slika 3.2:** Pojednostavljeni prikaz komunikacijske strukture ROS-a

## 3.2. ROS - Implementacija

Sustav za globalnu viziju će se sastojati od tri čvora. Po čvor za primanje podataka sa svakog Wiimotea te čvor za obradu dobivenih mjerenja Wiimotea i objavu izračunate pozicije i orijentacije. Čvor za komunikaciju s Wiimoteom već postoji pa

je skinut s interneta. Koristi se "wiimote" paket iz skupa paketa "joystick\_drivers" ([https://github.com/ros-drivers/joystick\\_drivers](https://github.com/ros-drivers/joystick_drivers)). Ovaj paket se ne može koristiti za uspostavu komunikacije s više Wiimoteova. Ako se povežu oba Wiimotea pomoću njega, čvorovi će objavljivati mjerenja na istu temu i time brisati objave jedan drugom pa nije moguće dohvaćati mjerenja. Problem je riješen dodavanjem "id" parametra. Prilikom pokretanja čvora i povezivanja s Wiimoteom zadajemo identifikacijski broj koji mijenja naziv tema tako da dopiše parametar "id" na kraj postojećeg naziva teme. Novi čvor koji se koristi pokreće se Python skriptom `wiimote_node2.py` i nalazi se u paketu "wiimote".

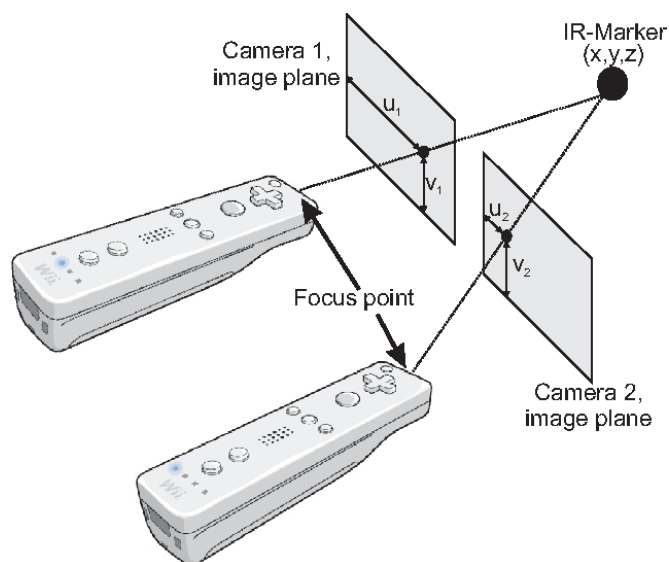
Ovime se postiže da umjesto jedne teme naziva `/wiimote/state` dobijemo dvije teme: `/wiimote/state1` i `/wiimote/state2`. Sada je lako dohvatiti mjerenja oba Wiimotea i možemo spojiti više Wiimoteova ako će biti potrebno.

Čvor za obradu mjerenja pretplaćeni je na `/wiimote/state1` i `/wiimote/state2` teme koje objavljuju modificirani čvorovi za dohvat podataka s Wiimoteova. Tip poruke koja se dohvaća je `wiimote/State.msg` tipa. Ovaj tip poruke sadrži mnoge nepotrebne podatke poput stanja tipki na Wiimoteu i njegovu akceleraciju pa je u funkciji dohvata potrebno izolirati samo važne podatke. Jedini potrebni podaci su pikseli na kojima su detektirani infracrveni izvori. Wiimote vraća taj podatak u obliku liste od 4 elementa, a svaki element je `wiimote/IrSourceInfo.msg` tipa podatka koji u sebi sadrži `x` i `y` detektiranog piksela.

Nakon što uspješno primi detektirane piksele od obje kamere, čvor za obradu izvršava izračun pozicije i orijentacije, koji će biti opisan kasnije u radu, te objavljuje dobivenu poziciju i orijentaciju na temu `/drone_position` u obliku tipa podatka `geometry_msgs/Quaternion.msg`. Quaternion tip poruke se koristi za objavu jer sadrži 4 varijable `float64` tipa, a objavljena poruka nema veze sa stvarnim kvaternionima. Varijable "x", "y" i "z" predstavljaju poziciju u globalnom koordinatnom sustavu dok varijabla "w" predstavlja orijentaciju drona, to jest zakret njegovog lokalnog koordinatnog sustava u odnosu na globalni oko z-osi (yaw). Čvor koji ima funkciju upravljačkog algoritma drona će čitati iz teme `/drone_position` što će omogućiti da se zatvori upravljačka petlja po poziciji i orijentaciji.

## 4. Izračun pozicije i orijentacije

Nakon što su kamere postavljene i usmjerene, markeri postavljeni na dronu i komunikacija uspostavljena, slijedi problem izračuna globalne pozicije točke iz poznatih piksela na kamerama. Svaki Wiimote objavljuje piksel na kojem je detektirao infracrveni marker. Taj piksel je zapisan u obliku  $(x,y)$  gdje  $x$  predstavlja stupac u kojem je piksel s prvim stupcem na lijevom kraju slike, a zadnjim stupcem (1024. stupac) na desnom kraju slike, dok  $y$  predstavlja redak s prvim retkom na dnu slike, a zadnjim na vrhu (768. red). Potrebno je iz para piksela  $(x_1, y_1)$  za prvi Wiimote i  $(x_2, y_2)$  za drugi Wiimote izračunati poziciju te točke u globalnim koordinatama  $(x_G, y_G, z_G)$ . Prvi postupak kojim se to pokušalo postići je pomoću geometrije i poznatih parametara kamere.



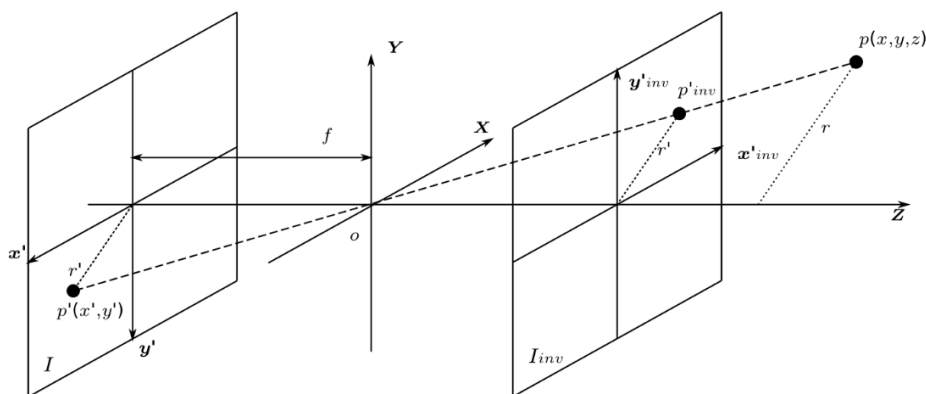
**Slika 4.1:** Projekcija markera na Wiimoteove

## 4.1. Geometrijski postupak

U ovom postupku koristiti će se "pinhole" model kamere za opis matematičkog odnosa koordinata točke u tri dimenzije i njezine projekcije na slikovnu ravninu. Ovaj model opisuje kameru točkom i ne uzima u obzir utjecaje leće poput iskrivljenja slike i замуćenje objekata koji nisu u fokusu. Točnost pinhole modela opada udaljavanjem od središta slike prema rubovima zbog povećanja iskrivljenja slike.

### 4.1.1. Formiranje slike u pinhole kameri

Sve zrake svjetlosti koje ulaze u točkasti model kamere i formiraju sliku prolaze kroz optički centar. Slikovna ravnina kamere je pomaknuta iza optičkog centra za udaljenost koja odgovara žarišnoj duljini kamere  $f$ . Dobivena slika na slikovnoj ravnini je izokrenuta pa se ona dodatno invertira kako bi dobili krajnju sliku. Inverzija slikovne ravnine odgovara projekciji točke na ravninu koja je postavljena ispred optičkog centra za žarišnu duljinu kamere. Središte 3-D koordinatnog sustava kamere ( $X$ ,  $Y$ ,  $Z$ ) postavljeno je u optički centar  $O$ . Koordinatni sustav slikovne ravnine ( $x'$ ,  $y'$ ) postavljen je u točki gdje  $Z$ -os koordinatnog sustava kamere siječe slikovnu ravninu.



**Slika 4.2:** Projekcija točke na slikovnu ravninu u *pinhole* kameri

Sve jednadžbe vezane uz projekciju slijede iz sličnosti trokuta formiranih stvarnom točkom  $p(x, y, z)$  u koordinatnom sustavu kamere, projekcijom te točke u koordinatnom sustavu slikovne ravnine i optičkim centrom.

$$\frac{x'}{x} = \frac{f}{z} \quad (4.1)$$

$$\frac{y'}{y} = \frac{f}{z} \quad (4.2)$$

### 4.1.2. Određivanje točke pravcima

Iz jednažbi 4.1 i 4.2 vidi se da je potrebno poznavati žarišnu duljinu kamere  $f$ , projiciranu točku  $(x', y')$  te barem jedan vektor  $(x, y$  ili  $z)$  točke  $p$  ako želimo odrediti koordinatu na kojoj se točka nalazi. Treba riješiti problem određivanja tri parametra iz dvije jednažbe. Premda je poznato da su točka, njezina projekcija i optički centar na istom pravcu, moguće je izračunati taj pravac. Žarišna udaljenost Wiimote kamere iznosi  $f = 1280$ . Poznat je broj piksela od sredine slike prema rubovima pa se računa maksimalni vertikalni i horizontalni kut pod kojim pravac može ulaziti u kameru.

$$\alpha_{max} = \arctg\left(\frac{x'_{max}}{f}\right) \quad (4.3)$$

$$\beta_{max} = \arctg\left(\frac{y'_{max}}{f}\right) \quad (4.4)$$

Iz rezolucije Wiimote kamere koja iznosi 1024x768 dolazimo do vrijednosti:

$$x'_{max} = \frac{1024}{2} = 512 \quad \text{i} \quad y'_{max} = \frac{768}{2} = 384 \quad (4.5)$$

Nakon uvrštavanja vrijednosti 4.5 u jednažbe 4.3 i 4.4 dolazimo do vrijednosti maksimalnog vertikalnog i horizontalnog kuta pravca koje iznose  $\alpha_{max} = 21.8^\circ$  i  $\beta_{max} = 16.7^\circ$ . Kako bi se saznalo za koliko svaki piksel po  $x'$  osi slike i po  $y'$  osi slike mijenja  $\alpha$  i  $\beta$  kut pravca, maksimalne vrijednosti kuteva dijele se s  $x'_{max}$  odnosno  $y'_{max}$ :

$$\Delta\alpha = \frac{\alpha_{max}}{x'_{max}} = 0.0426^\circ \quad (4.6)$$

$$\Delta\beta = \frac{\beta_{max}}{y'_{max}} = 0.0435^\circ \quad (4.7)$$

Jednažbe 4.6 i 4.7 vrijede u idealnom slučaju kada ne dolazi do iskrivljenja slike na rubovima. Poznato je za koliko stupnjeva svaki piksel po  $x'$  osi i  $y'$  osi mijenja usmjerenje pravca od pravca koji izlazi ravno iz kamere (po Z-osi koordinatnog sustava kamere). Slijede jednažbe za izračun kuteva zakreta iz poznatog piksela  $(x', y')$ :

$$\alpha = (512 - x')\Delta\alpha \quad (4.8)$$

$$\beta = (-384 + y')\Delta\beta \quad (4.9)$$

Iz poznatih kuteva slijedi izračun jediničnog vektora u koordinatnom sustavu kamere  $O$  koji pokazuje u smjeru pravca na kojem leži točka  $p$ :

$$x = \sin(\alpha)\cos(\beta) \quad (4.10)$$

$$y = \sin(\beta) \quad (4.11)$$

$$z = \cos(\alpha)\cos(\beta) \quad (4.12)$$

Sljedeći korak je transformacija vektora koji pokazuje na točku iz lokalnog koordinatnog sustava kamere ( $X, Y, Z$ ) u globalni koordinatni sustav ( $X_G, Y_G, Z_G$ ). Prvo će se izvršiti rotacija tog vektora pa onda translacija. Iz poznate pozicije kamere i orijentacije njenog lokalnog koordinatnog sustava računamo translacijsku i rotacijsku matricu. Koristi se Z-Y-X rotacijska matrica, što znači da se lokalni koordinatni sustav prvo rotira za kut  $\psi$  oko svoje Z-osi, onda rotira za kut  $\theta$  oko svoje novo pozicionirane Y-osi, te na kraju rotira za kut  $\phi$  oko X-osi. Za svaku od ovih rotacija, koristi se pripadna rotacijska matrica te se množe u ispravnom poretku  $R_Z(\psi)R_Y(\theta)R_X(\phi)$ .

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = R_Z(\psi)R_Y(\theta)R_X(\phi) \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} X_{trans} \\ Y_{trans} \\ Z_{trans} \end{bmatrix} \quad (4.13)$$

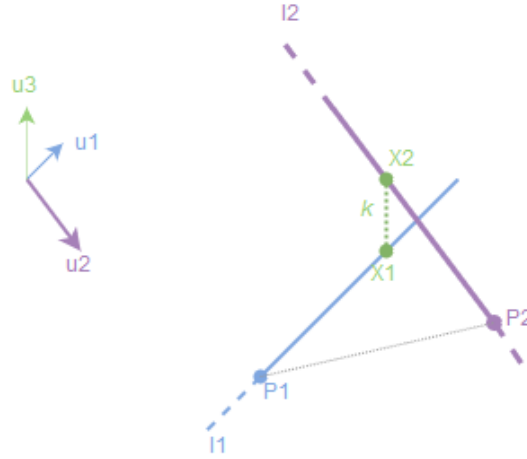
$$R_Z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

$$R_Y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (4.15)$$

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (4.16)$$

Nakon što je završena transformacija vektora u globalni koordinatni sustav, poznat nam je pravac na kojem leži točka. Da bi se mogla odrediti točna pozicija te točke na pravcu potrebna je druga kamera. Kada se isti postupak određivanja pravca na kojem leži točka obavi i za drugu kameru, traži se presjek pravca iz prve i iz druge kamere

u idealnom slučaju. Zbog ručnog postavljanja kuteva zakreta kamera, diskretizacije piksela i nesavršenosti kamera, slučaj da se pravci sijeku je skoro nemoguć. Zato se traže točke na oba pravca gdje su pravci najbliži jedan drugom.



**Slika 4.3:** Prikaz pravaca  $l_1$  i  $l_2$  te najbližih točaka  $x_1$  i  $x_2$

Za izračun najbližih točki na pravcima, kreće se od parametarske jednadžbe pravca koja glasi:

$$l = p_0 + d \cdot t \quad (4.17)$$

U jednadžbi 4.17  $p_0$  je neka poznata točka na pravcu,  $d$  je vektor smjera pravca, a  $t$  je skalar koji određuje pomak točke od  $p_0$  po pravcu u smjeru  $d$ .

Ako su poznate jednadžbe dvaju pravaca:

$$l_1 = p_1 + d_1 \cdot t_1 \quad \text{i} \quad l_2 = p_2 + d_2 \cdot t_2 \quad (4.18)$$

Pravac koji povezuje najbliže točke pravca  $l_1$  i  $l_2$  mora biti okomit na oba pravca. Vektorskim produktom vektora smjera pravaca  $d_1$  i  $d_2$  dobiva se vektor tog okomitog pravca,  $n$ .

$$n = d_1 \times d_2 \quad (4.19)$$

Najbliža točka na pravcu  $l_1$  pravcu  $l_2$  je na presjecištu pravca  $l_1$  i ravnine koja se dobi širenjem pravca  $l_2$  u smjeru  $n$ . Jednadžba te ravnine može se izraziti kao skup točaka  $p$  za koje vrijedi:

$$(p - p_0) \cdot n_R = 0 \quad (4.20)$$



U jednadžbi 4.20,  $p_0$  označava točku koja se nalazi na ravnini, a  $n_R$  označava vektor normale te ravnine. Ravnina koja nam je potrebna za pronaći najbližu točku na pravcu  $l_1$ , sadrži točku  $p_2$ , a jednadžba njene normale  $n_2$  dobit će se vektorskim produktom vektora smjera pravca  $d_2$  i  $n$ .

$$n_2 = d_2 \times n = d_2 \times (d_1 \times d_2) \quad (4.21)$$

Jednadžba ravnine koja sadrži pravac  $l_2$  i najbližu točku pravca  $l_1$  tada glasi:

$$(p - p_2) \cdot n_2 = 0 \quad (4.22)$$

Kako bi se dobila točka u kojoj pravac  $l_1$  sječe ovu ravninu, u jednadžbu 4.22 uvrštava se jednadžba za pravac  $l_1$ .

$$(p_1 + d_1 \cdot t_1 - p_2) \cdot n_2 = 0 \quad (4.23)$$

Potrebno je dobiti  $t_1$  za koji vrijedi jednadžba 4.23. Rješavanjem jednadžbe za  $t_1$  dobije se:

$$t_1 = \frac{(p_2 - p_1) \cdot n_2}{d_1 \cdot n_2} \quad (4.24)$$

Ponavljanjem postupka za pravac  $l_2$  dobit će se  $t_2$ . Uvrštavanjem izračunatih  $t_1$  i  $t_2$  u jednadžbe pravaca  $l_1$  i  $l_2$  (jednadžba 4.18) dobit će se najbliže točke na pravcima,  $c_1$  i  $c_2$ .

$$c_1 = p_1 + \frac{(p_2 - p_1) \cdot n_2}{d_1 \cdot n_2} d_1 \quad (4.25)$$

$$c_2 = p_2 + \frac{(p_1 - p_2) \cdot n_1}{d_2 \cdot n_1} d_2 \quad (4.26)$$

Opisani postupak pronalaska najbližih točaka dvaju pravaca koristi se u ROS čvoru zaduženom za obradu informacija pristiglih od kamera i objavu pozicije i orijentacije drona. Ovaj čvor pokreće se Python skriptom *wii\_detection.py*. Funkcija *line\_distance* zadužena je za pronalazak točaka i njihove udaljenosti.

Svi koraci opisani u ovom odlomku izvršavaju se unutar *wii\_detection* čvora. Prvi korak je učitavanje mjerenja oba Wiimotea. Iz dobivenih piksela vrši se izračun pravca na kojem je marker u lokalnom koordinatnom sustavu kamere. Izračunati pravci se prebacuju iz lokalnih koordinata u globalne. Od svake kamere se dobiva od nula do

tri pravca. Potrebno je upariti ispravne pravce prve i druge kamere kako bi se dobile dobre točke. Svaki pravac prve kamere se uspoređuje sa pravcima druge kamere i traži se najmanja udaljenost. Uzima se pravac druge kamere kojem je pravac prve kamere najbliži za izračun pozicije markera prema kojem ti pravci gledaju. Ukoliko nije niti jedan marker detektiran, objavljuje se pozicija drona  $x = -1.0$ ,  $y = -1.0$ ,  $z = -1.0$ . Ukoliko je detektiran samo jedan marker, objavljuje se pozicija tog markera. Kod dva ili tri markera dodatno se računa pozicija centra drona te njegova orijentacija.

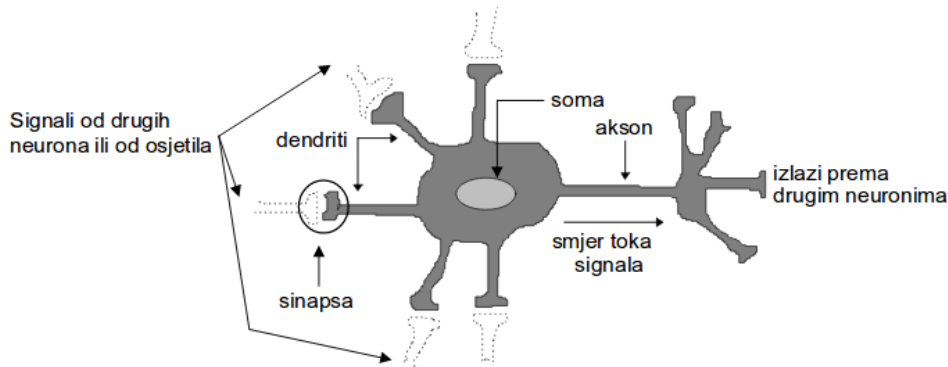
## 4.2. Postupak pomoću neuronske mreže

Drugi postupak koji se koristio za određivanje pozicije markera iz para piksela dobivenih od kamera bio je postupak pomoću neuronske mreže. Kod geometrijskog postupka nisu se uzeli u obzir intrinzični parametri kamera te su kamere morale biti vrlo precizno postavljene kako ne bi došlo do promašivanja pravaca i krive detekcije. Cilj ovog postupka je naučiti jednostavnu unaprijednu (*engl.* feedforward) neuronsku mrežu odgovarajućih dimenzija kako preslikati parove piksela  $(x_1, y_1)$  za prvi Wiimote i  $(x_2, y_2)$  za drugi Wiimote u globalne koordinate točke  $(x_G, y_G, z_G)$ . Dobro naučena neuronska mreža bi trebala moći precizno modelirati to preslikavanje, čak ako postoje neki nepoznati utjecaji i nelinearnosti. Mana ovog postupka je da se ne smije mijenjati nagib i pozicija kamera jer se učenje vrši s točno određenim položajem kamera.

### 4.2.1. Osnove neuronskih mreža

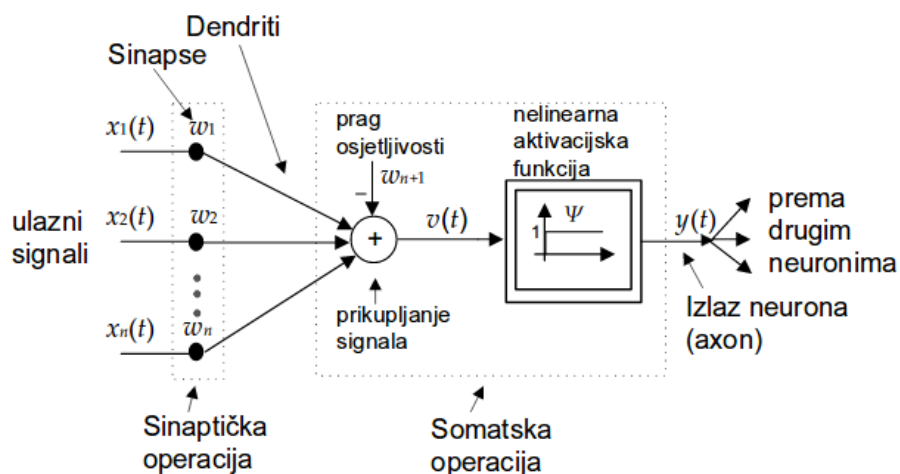
Ideja umjetnih neuronskih mreža proizašla je iz načina funkcioniranja bioloških neuronskih mreža. Biološki neuron, kao osnovna gradivna jedinica biološke neuronske mreže, prima i obrađuje informacije od drugih neurona i/ili od osjetilnih organa. Može se pojednostavljeno prikazati kao stanica sastavljena od tijela (soma), mnoštva dendrita i aksona (Slika 4.4). Mali razmak između završetka aksona prethodnoga neurona i dendrita sljedećeg neurona naziva se sinapsa. Akson jednoga neurona formira sinaptičke veze s mnoštvom drugih neurona. Impulsi, koji se generiraju u tijelu neurona, putuju kroz akson do sinapsi. Ovisno o učinkovitosti svakog pojedinačnog sinaptičkoga prijenosa, signali različitog intenziteta dolaze do dendrita. Učinkovitost sinaptičkoga prijenosa kroz neku sinapsu ovisi o njezinom elektrokemijskom stanju, koje je rezultat prethodnih sinaptičkih prijenosa kroz nju. Signali se od sinapsi dendritima prosljeđuju do tijela neurona, gdje se prikupljaju i obrađuju. Ovi signali mogu za tijelo neurona biti pobuđujući ili smirujući. Matematički gledano, pobuđujući i smirujući signali imaju

suprotan predznak. Ako je njihova kumulativna vrijednost tijekom kratkog vremenskog intervala veća od praga osjetljivosti neurona, tijelo neurona generira impulse (tzv. aktivacijske potencijale) koji se šalju duž aksona prema drugim neuronima, a ako je manja, neuron ostaje nepobuđen i ne generira impulse.



**Slika 4.4:** Model biološkog neurona

*Perceptron* je jednostavan model umjetnog neurona koji vrši slične operacije kao i biološki neuron (Slika 4.5). Sinaptička je operacija predstavljena množenjem svakog ulaznog signala  $x_i$  s težinskim koeficijentom  $w_i$ . Otežani se ulazni signali zbrajaju i njihov se zbroj uspoređuje s pragom osjetljivosti neurona  $w_{n+1}$ . Ako je zbroj otežanih signala veći od praga osjetljivosti neurona, nelinearna aktivacijska funkcija  $\psi$  generira izlazni signal neurona  $y$  iznosa jedan, a ako je manji, izlaz neurona je nula.



**Slika 4.5:** Model perceptrona

U radu će se koristiti statički model neurona. Njegov izlaz ovisi isključivo o trenutnim vrijednostima ulaznih signala i težinskim koeficijentima. Matematički se može opisati sljedećim izrazima:

$$v(t) = \sum_{i=1}^n w_i(t) \cdot x_i(t) - w_{n+1} \quad (4.27)$$

$$y(t) = \psi(v) \quad (4.28)$$

gdje je:

$\mathbf{X}_u(t) = [x_1(t), \dots, x_n(t)]^T$  - vektor ulaznih signala neurona;

$\mathbf{W}_s(t) = [w_1(t), \dots, w_n(t)]^T$  - vektor sinaptičkih težinskih koeficijenata;

$w_{n+1}$  - prag osjetljivosti neurona;

$v(t)$  - izlaz operacije konfluencije;

$\psi(v)$  - nelinearna aktivacijska funkcija;

$y(t)$  - izlaz neurona

Većina neuronskih mreža zahtijeva učenje, to jest primjenu algoritama koji podešavaju iznose sinaptičkih težinskih koeficijenata. Ciljevi učenja mreže ovise o njezinoj primjeni, pa tako i izbor odgovarajućeg algoritma učenja. Iako je razvijen veliki broj raznih algoritama učenja neuronskih mreža, moguće ih je po načinu učenja mreže podijeliti na algoritme učenja temeljene na pogrešci, algoritme učenja temeljene na izlazu mreže i algoritme učenja s ojačanjem. U radu će biti korišteni algoritmi učenja temeljeni na pogrešci.

*Algoritmi učenja temeljeni na pogrešci* često se nazivaju i algoritmi s “učiteljem” jer zahtijevaju vanjski referentni signal (učitelj) s kojim uspoređuju dobiveni odziv neuronske mreže generirajući signal pogreške. Na temelju signala pogreške algoritam učenja mijenja sinaptičke težinske koeficijente neuronske mreže s ciljem poboljšanja njezina vladanja, to jest smanjenja pogreške. Ovi se algoritmi mogu primijeniti samo ako je unaprijed poznato željeno vladanje neuronske mreže, to jest podaci na osnovi kojih se mreža uči moraju sadržavati parove vrijednosti ulazno-izlaznih signala.

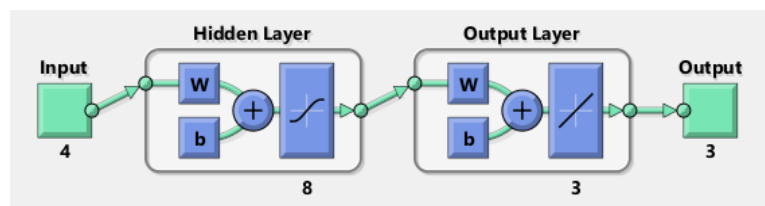
#### 4.2.2. Implementacija neuronske mreže

Neuronska mreža koja će se koristiti za izračun pozicije ima četiri ulaza i tri izlaza. Ulazi su koordinate piksela prve kamere ( $x_1, y_1$ ) i koordinate piksela druge kamere ( $x_2, y_2$ ). Izlaz iz neuronske mreže su globalne koordinate markera ( $X_G, Y_G, Z_G$ ). Iz jednadžbi izvedenih u geometrijskoj metodi izračuna pozicije točke može se uočiti da su jedini nelinearni članovi jednadžbi sinusi i kosinusi. Unaprijedna (*engl.* feed-forward) jednoslojna mreža će biti dovoljno precizna za opis funkcije preslikavanja.

Broj neurona u skrivenom sloju bi trebao biti veći od pet jer se s pet neurona skrivenog sloja postiže relativno dobro praćenje sinusne funkcije.

Kako bi se proveo algoritam učenja mreže temeljen na pogrešci, moraju se snimiti kombinacije ulaza i izlaza sustava. Za snimanje točaka, prvo se okvirno na podu označio rub radnog prostora. Onda su naznačene neke točke po podu unutar radnog prostora. Ukupno je 13 označenih točaka. Izmjerene su x i y koordinate označenih točaka u odnosu na globalni koordinatni sustav. Za hvatanje piksela kamera, napisana je skripta koja na pritisak tipke na tipkovnici učitava aktivne piksele s Wiimoteova i sprema ih u .csv datoteku. Slikanjem infracrvenog markera postavljenog na označene točke, saznaje se koja kombinacija piksela odgovara kojoj koordinati markera u prostoru. Marker je bio postavljen na visinama od 0cm, 15cm i 30cm nad označenim točkama, što daje ukupno 39 mjerenja i ulazno-izlaznih kombinacija. Za trening mreže koristi se MATLAB Neural Network Toolbox.

*MATLAB Neural Network Toolbox* je alat unutar programa MATLAB koji sadrži algoritme, već istrenirane modele i aplikacije za stvaranje, treniranje, vizualizaciju i simulaciju jednostavnih i dubokih neuronskih mreža. Ovaj alat, zbog svoje jednostavnosti, ubrzava proces treniranja i testiranja neuronske mreže koja će dobro preslikavati piksele kamera u točku u prostoru. Potrebno je samo stvoriti matrice u MATLAB-u koje odgovaraju ulazima i izlazima iz sustava, pokrenuti aplikaciju upisom naredbe *nnstart* i odabrati kakva nam neuronska mreža treba. Nakon treninga može se vidjeti preciznost mreže pa se može opet trenirati mreža sa promijenjenim parametrima kako bi se što više poboljšala preciznost. Unaprijedna mreža s jednim skrivenim slojem veličine 8 neurona dala je vrlo dobre rezultate te će se ona koristiti za određivanje pozicije markera (Slika 4.6).



**Slika 4.6:** Neuronska mreža korištena u MATLAB-u

Istrenirana neuronska mreža dobivena u MATLAB-u implementirana je u programskom jeziku Python kako bi se mogla koristiti u ROS čvoru. Čvor koji koristi neuronsku mrežu pokreće se skriptom *wii\_neural.py*. Problem kod ovog algoritma je kako upariti piksele prve i druge kamere ako je detektirano više markera. Wiimote vraća listu detektiranih piksela tako da je na prvom mjestu liste infracrveni marker najvećeg

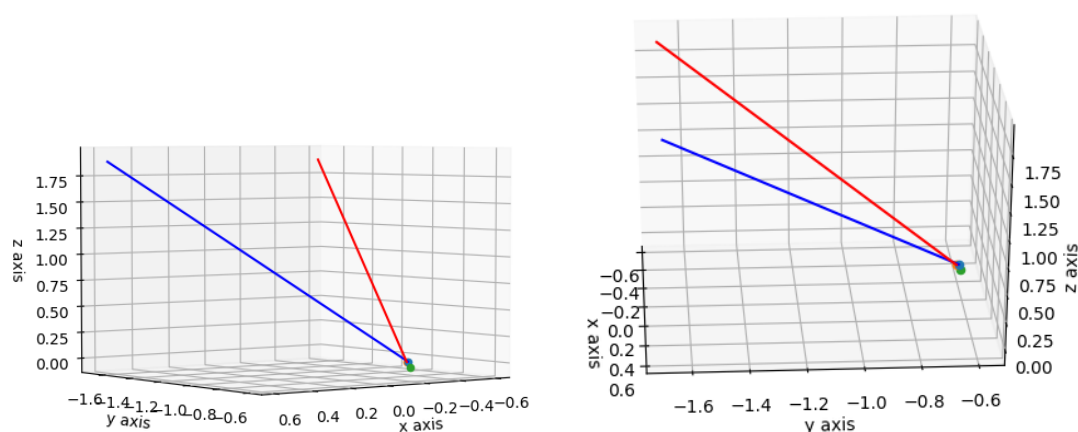
intenziteta. Ovaj problem je riješen postavljanjem dva infracrvena markera na dron, jedan naprijed, drugi iza, i održavanjem *yaw* kuta drona da dron uvijek bude okrenut prema panou s kamerama. Ako dron ostane u tom položaju, znamo da prvi piksel liste odgovara prednjem markeru, a drugi piksel stražnjem. Čvor objavljuje poziciju i orijentaciju drona na temu *drone\_position* kao što radi čvor za geometrijski postupak.

## 5. Rezultati

U nastavku će biti prikazani rezultati geometrijske metode i metode pomoću neuronskih mreža. Preciznost metoda biti će uspoređena i razmatrati će se moguća poboljšanja.

### 5.1. Geometrijska metoda

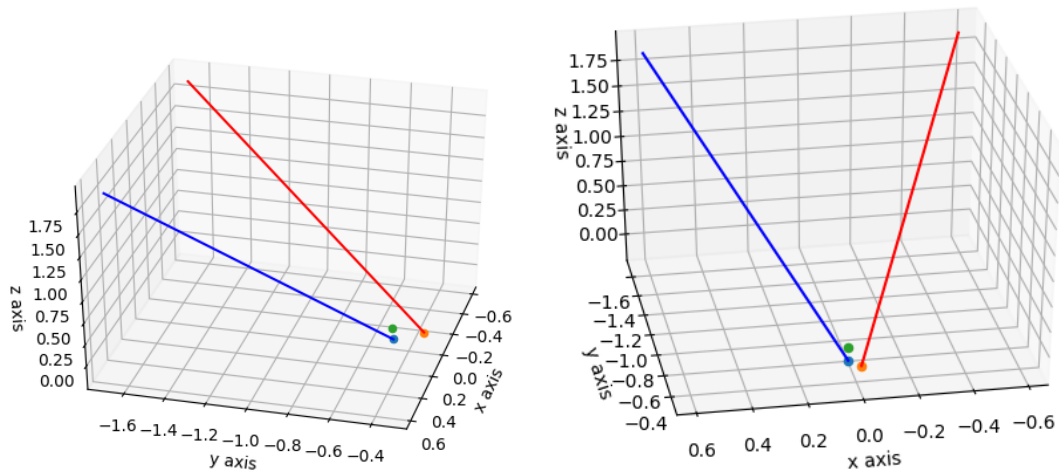
Čvor koji je vraćao poziciju na temelju geometrijske metode imao je vrlo lošu preciznost i nije bio pouzdan. Zbog dijelova nosača Wiimotea, čiji su se kutevi morali ručno postavljati i zatezati, te zbog načina spajanja tog nosača na pano koji ne osigurava horizontalnost spoja, postavljeni kutevi u rotacijskim matricama nisu bili točni. Dodatnim mjerenjem kuteva i testiranjem dobilo se relativno precizno lociranje točaka (unutar 5cm), ali na vrlo malom području.



**Slika 5.1:** Pravci, izračunate točke i stvarna točka za centralnu točku

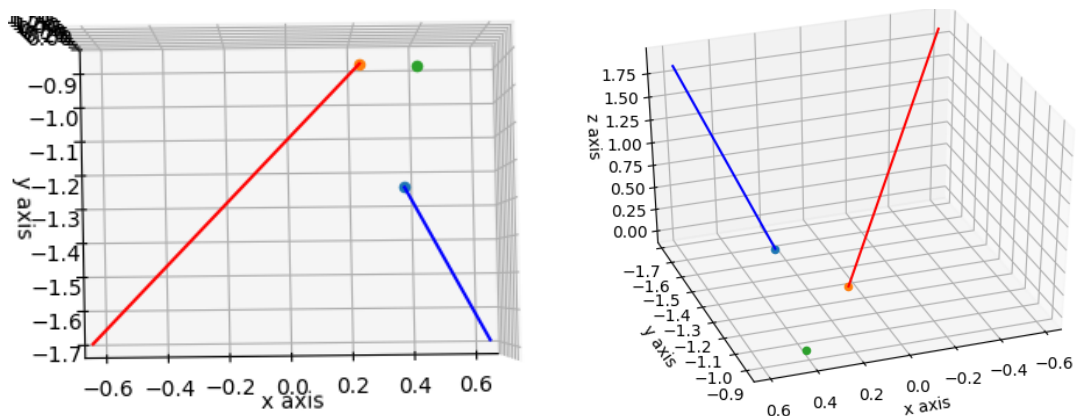
Slike 5.1 pokazuju detekciju za točku koja je bila za obje kamere blizu centra slikovne ravnine. Plavi pravac je pravac prve kamere, a crveni pravac pripada drugoj

kameri. Pravci prolaze vrlo blizu jedan drugom i udaljenost njihovih najbližih točaka je unutar 2cm od stvarne pozicije markera, naznačene zelenom točkom.



**Slika 5.2:** Pravci, izračunate točke i stvarna točka za točku u blizini centralne

Slike 5.2 pokazuju detekciju za točku koja se nalazi u radijusu od 20cm od prethodne točke (slike 5.1) koja je dobro detektirana. Može se uočiti da pada preciznost odmakom projekcije markera od centra slikovne ravnine kamera. Usrednjavanjem najbližih točaka pravaca dobila bi se greška od oko 5cm. Slična greška je i za ostale točke na takvoj udaljenosti od pozicije na kojoj je detekcija precizna.



**Slika 5.3:** Pravci, izračunate točke i stvarna točka za rubnu točku

Slike 5.3 prikazuju detekciju za točku blizu ruba radnog prostora. Kako se marker odmiče od središta prema rubu prostora detekcije, dolazi do porasta greške. Primjećuje se da se pravci presporo zakreću od središta prema rubu. Do toga može doći zbog



premalog  $\Delta\alpha$  i/ili  $\Delta\beta$ . Također je vjerojatno da se zbog iskrivljenja slike na rubovima povećavaju  $\Delta\alpha$  i  $\Delta\beta$ . Problem kod ove metode je u nepoznavanju intrinzičnih parametara kamere. Riješenje bi bila kalibracija kamere koja bi nam dala uvida u intrinzične parametre kamere. Kalibracija bi se mogla provesti pomoću dvije ili više infracrvenih LED dioda postavljenih na objekt na poznatim udaljenostima.

Metoda s neuronskom mrežom bila je brža opcija od osmišljanja i pisanja algoritma kalibracije za geometrijsku metodu.

## **5.2. Metoda s neuronskom mrežom**

## **6. Zaključak**

Zaključak.

# LITERATURA

## **Razvoj sustava globalne vizije za testiranje algoritama upravljanja bespilotnim letjelicama**

### **Sažetak**

U mnogim granama industrije veliki značaj ima praćenje kretanja. Prisutna rješenja poput OptiTrack-a osiguravaju precizno praćenje, ali su iznimno skupa. Cilj rada je razvoj jednostavnog i ekonomski prihvatljivog sustava globalne vizije za praćenje pozicije i orijentacije bespilotnih letjelica. Sustav se sastoji od dvije infracrvene kamere i tri infracrvena markera postavljena na dronu. Za određivanje pozicije markera u globalnom koordinatnom sustavu potrebne su minimalno dvije kamere. Minimalno tri markera potrebna su kako bi se mogla izračunati orijentacija drona uz njegovu poziciju. Kao kamera se koristi Wiimote koji sadrži jednostavnu infracrvenu kameru, a markeri su infracrveni LED izvori velike snage (3W). Komunikacija između Wiimote-ova i osobnog računala ostvarena je pomoću Bluetootha i ROS čvora. Iz poznatih pozicija kamera i dobivenih točaka od svakog Wiimotea, osobno računalo računa globalnu poziciju svake točke te iz tih pozicija računa orijentaciju i poziciju drona. Orijentacija i pozicija se objavljuju preko ROS-a kako bi se mogli dohvatiti od strane upravljačkog algoritma.

**Ključne riječi:** praćenje pokreta, sustav vizije, dronovi, infracrvene kamere, Wiimote, markeri, pozicija, orijentacija, ROS

## **Development of a Global Vision System for Testing of Control Systems for Unmanned Aerial Vehicles**

### **Abstract**

Motion tracking systems are of great importance in many industries. Current solutions like OptiTrack ensure precise tracking, but are very expensive. The goal of the project is the development of a simple and economically acceptable global vision system for drone position and orientation tracking. The system is made of two static infrared cameras and three infrared markers placed on the drone. To determine the position of the markers in the global coordinate system a minimum of two cameras is needed. At least three markers are needed in order to calculate the orientation of the drone. The camera used in the project is the IR camera inside a Wiimote and the markers used are high power IR LED sources (3W). The communication between the Wiimotes and the PC is realized using Bluetooth and a ROS node for each Wiimote. From the known Wiimote positions and the points detected by the Wiimotes, the PC calculates the global position of each point and then uses those positions to calculate the orientation and position of the drone. Orientation and position are published to a ROS topic, so that the control algorithm can acquire them.

**Keywords:** motion tracking, vision system, drones, infrared cameras, Wiimote, markers, position, orientation, ROS