

Servo sustav za lansiranje balističkih projektila

Golemović, Franjo

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:104352>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-11-25**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Franjo Golemović

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Mladen Crneković, dipl. ing.

Student:

Franjo Golemović

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, Prof. Dr. Sc. Mladenu Crnekoviću na izdvojenom vremenu, savjetima i pomoći pri izradi ovog rada.

Franjo Golemović



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomске ispite
Povjerenstvo za završne i diplomске ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 24 - 06 / 01	
Ur.broj: 15 - 24 -	

ZAVRŠNI ZADATAK

Student: **Franjo Golemović** JMBAG: **0035235139**

Naslov rada na hrvatskom jeziku: **Servo sustav za lansiranje balističkih projektila**

Naslov rada na engleskom jeziku: **Servo system for launching ballistic missiles**

Opis zadatka:

Učinkovito lansiranje raketnih balističkih projektila pretpostavlja postojanje automatskog uređaja koji će u što kraćem vremenu s bilo kojeg položaja lansera, usmjeriti raketu prema cilju. Za to je potrebno imati servo sustav automatskog pozicioniranja lansera po traženom azimutu i elevaciji.

Azimut lansera izračunava se prema trenutačnom položaju lansera i položaju cilja, a elevacija lansera prema unaprijed definiranim tablicama za svaki tip projektila zasebno, a ovisno o udaljenosti i vremenskim uvjetima.

U radu je potrebno:

- projektirati servomehanizam azimuta i elevacije lansera te izračunati njihove karakteristike (točnost, brzina ...)
- odabrati potrebne senzore položaja i orijentacije lansera
- definirati način zadavanja cilja
- izvesti algoritam izračuna potrebnog azimuta

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2023.

Datum predaje rada:

1. rok: 22. i 23. 2. 2024.
2. rok (izvanredni): 11. 7. 2024.
3. rok: 19. i 20. 9. 2024.


Predviđeni datumi obrasci:

1. rok: 26. 2. - 1. 3. 2024.
2. rok (izvanredni): 15. 7. 2024.
3. rok: 23. 9. - 27. 9. 2024.

Zadatak zadao:


Prof. dr. sc. Mladen Crneković

Predsjednik Povjerenstva:


Prof. dr. sc. Damir Godić

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA.....	III
POPIS TEHNIČKE DOKUMENTACIJE	IV
POPIS OZNAKA	V
1. UVOD.....	1
1.1. Balistički projektili.....	2
1.2. Lanseri balističkih projektila.....	4
2. MEHANIZAM I KONSTRUKCIJA.....	6
2.1. Rotaciju po azimutu	8
2.2. Rotacija po elevaciji.....	12
2.3. Raketni paket.....	14
2.4. Napajanje.....	15
2.5. Baza lansera.....	15
2.6. Servisiranje i hlađenje lansera.....	16
3. MOTORI.....	17
3.1. BLDC Motor	17
3.2. Upravljanje motora.....	18
3.3. Odabir motora za rotaciju po azimutu.....	20
3.4. Odabir motora za rotaciju po elevaciji	24
3.5. Napajanje i kalibracija motora	26
3.5.1 Napajanje	26
3.5.2. Kalibracija motora	27
4. SENZORI	30
4.1. Mikrokontroler	30
4.2. Senzor položaja	31
4.3 Senzor orijentacije.....	33
4.4. Algoritam postavljanja cilja i potrebnog azimuta	33
4.5. Rotacija po elevaciji	38
4.6. Integracija senzora.....	39
4.7. Alternativni izbor procesora.....	41
5. UPRAVLJANJE MOTORA.....	43
5.1. Komunikacija sa motorom	44
5.2. Komande i kod motora.....	46
6. ZAKLJUČAK.....	48
LITERATURA.....	49

POPIS SLIKA

Slika 1. Balistička raketa V-2 [1]	3
Slika 2. Stadiji balističkih raketa [2]	4
Slika 3. Sustav za lansiranje balističkih raketa [3].....	5
Slika 4. Idejno rješenja servo sustava za lansiranje	6
Slika 5. Podjela konstrukcije	7
Slika 6. Okretni ležaj sa unutarnjim ozubljenjem [4].....	8
Slika 7. Poligon sila u ležaju [6]	9
Slika 8. Mehanizam rotacije po Azimutu.....	11
Slika 9. Elementi za prijenos momenta	12
Slika 10. Mehanizam rotacije po elevaciji	14
Slika 11. Ravnotežni položaj lansera	14
Slika 12. Organizacija napajanja motora.....	15
Slika 13. Baza lansera	15
Slika 14. Poklopci i perforacije za hlađenje	16
Slika 15. BLDC motor [7].....	17
Slika 16. ESC kontroler [8].....	18
Slika 17. Princip rada BLDC motora [9].....	19
Slika 18. Određivanje položaja rotora pomoću Hallovih senzora [9]	19
Slika 19. Određivanje položaja rotora pomoću elektromotorne sile [9]	20
Slika 20. Dimenzije motora PD6-EB [10]	21
Slika 21. Motor PD6-EB sa intergriranim mjenjačem GPLE80 [10]	21
Slika 22. Poligon sila na pogonskom i gonjenom zupčaniku [12]	23
Slika 23. Konstrukcija za osiguranje prvog motora	24
Slika 24. Konstrukcija za osiguranje drugog motora	26
Slika 25. Agregat Honda EU22i [14]	27
Slika 26. Sučelje Plug and Drive Studio 3 [15].....	28
Slika 27. Arduino Uno Pločica [16]	30
Slika 28. GPS modul NEO-M8 [17]	32
Slika 29. CMPS14 kompas [18]	33
Slika 30. Platforma za elektroničke komponente	40
Slika 31. Senzori na lanseru	41
Slika 32. MAX485 modul [19]	43

POPIS TABLICA

Tablica 1. Karakteristike gonjenog zupčanika[5]	8
Tablica 2. Karakteristike motora za rotaciju po azimutu [10].....	22
Tablica 3. Karakteristike pogonskog zupčanika [11].....	22
Tablica 4. Karakteristike motora za rotaciju po elevaciji [10].....	25
Tablica 5. Kut elevacije prema tipu rakete.....	38
Tablica 6. Pakiranje komande u niz bajtova	44

POPIS TEHNIČKE DOKUMENTACIJE

1. Kod za Arduino
2. Sklopni crtež lansera

POPIS OZNAKA

Oznaka	Jedinica	Opis
m_{ko}	kg	Masa podkonstrukcije
m_{ra}	kg	Masa pojedine rakete
m_{la}	kg	Masa lansera
n_{r1}	o/min.	Broj okretaja motora za rotaciju po azimutu
m_2	-	Modul gonjenog zupčanika
z_2	-	Broj zubi gonjenog zupčanika
d_2	mm	Diobena kružnica gonjenog zupčanika
F_N	N	Normalna sila koja djeluje na ležaj
g	m/s^2	Ubrzanje zemljine sile teže
F_{TR}	N	Sila trenja u ležaju
μ_1	-	Koeficijent trenja u ležaju
T_{TR}	Nm	Moment trenja u ležaju
d_{TR}	mm	Promjer po kojemu djeluje sila trenja
I_D	kgm^2	Moment inercije diska
r_{ro}	mm	Polumjer rotacije
ω_{r1}	rad/s	Kutna brzina rotacije po azimutu
α_{r1}	rad/s^2	Kutna akceleracija rotacije po azimutu
t_{r1}	s	Vrijeme rotacije po azimutu
T_{I1}	Nm	Okretni moment inercije po azimutu
T_{UK1}	Nm	Ukupni okretni moment po azimutu
i_1	-	Prijenosni omjer rotacije po azimutu
F_O	N	Obodna komponenta normalne sile
T_1	Nm	Okretni moment pogonskog zupčanika
d_1	mm	Promjer diobene kružnice pogonskog zupčanika
T_2	Nm	Okretni moment gonjenog zupčanika
d_2	mm	Promjer diobene kružnice gonjenog zupčanika
η_z	-	Stupanj djelovanja zupčanog prijenosa
η_l	-	Stupanj djelovanja lančanog prijenosa
m_{pa}	kg	Masa raketnog paketa
n_{r2}	o/min.	Broj okretaja motora za rotaciju po elevaciji

I_K	kgm^2	Moment inercije kvadra
a	mm	Prva dimenzija kvadra okomita na os rotacije
b	Mm	Druga dimenzija kvadra okomita na os rotacije
T_{I2}	Nm	Okretni moment inercije po elevaciji
i_2	-	Prijenosni omjer rotacije po elevaciji
z_3	-	Broj zubi pogonskog lančanika
z_4	-	Broj zubi gonjenog lančanika
T_3	Nm	Okretni moment pogonskog lančanika
T_4	Nm	Okretni moment gonjenog lančanika

SAŽETAK

Servo sustavi za lansiranje balističkih projektila omogućuju brzu i preciznu orijentaciju lansera, što je ključno za uspješno izvođenje vojnih operacija, lansiranje projektila i znanstvena istraživanja. Ovi sustavi poboljšavaju točnost, brzinu, sigurnost i fleksibilnost operacija, smanjujući potrebu za ljudskom intervencijom i povećavajući ukupnu učinkovitost.

Ovaj rad pruža sveobuhvatan pregled dizajna i implementacije ovih sustava, ističući njihovu vitalnu ulogu u modernim tehnološkim primjenama.

Ključne riječi: Raketne, lanser, rotacija, mehanizam, elevacija, azimut

SUMMARY

Servo systems for launching ballistic missiles enable quick and precise orientation of the launcher, which is crucial for the successful execution of military operations, satellite launches, and scientific research. These systems enhance the accuracy, speed, safety, and flexibility of operations, reducing the need for human intervention and increasing overall efficiency.

This paper provides a comprehensive overview of the design and implementation of these systems, highlighting their vital role in modern technological applications.

Keywords: Rockets, launcher, rotation, mechanism, elevation, azimuth

1. UVOD

Projektilna oružja igrala su ključnu ulogu u evoluciji ratovanja i obrane, prateći svoje korijene do drevnih vremena kada su rani ljudi koristili jednostavne alate poput prački i lukova za lansiranje projektila. Ova primitivna oružja postavila su temelje za razvoj sofisticiranijih projektilnih oružja, koristeći tehnološke i materijalne napretke kroz stoljeća.

Povijesna putanja projektilnih oružja zabilježila je značajne prekretnice s pojavom baruta u 9. stoljeću, revolucionirajući ratovanje kroz stvaranje ranih vatrenih oružja i topova. Korištenje baruta omogućilo je ispaljivanje projektila pri mnogo većim brzinama i s većom snagom, označavajući početak modernog topništva i malokalibarskog oružja.

Kako je tehnologija napredovala kroz industrijsku i modernu eru, projektilna oružja dramatično su se razvijala. Puške, strojnice i velikokalibarsko topništvo postali su standard u vojnim arsenalima, s poboljšanjima u preciznosti, dometu i vatrenoj moći. Dvadeseto stoljeće svjedočilo je razvoju vođene raketne tehnologije, uvodeći novu eru preciznosti i strateških sposobnosti.

U suvremenom vremenu, projektilna oružja obuhvaćaju širok spektar, od malokalibarskog oružja koje koristi pješništvo do naprednih balističkih projektila. Balistički projektili predstavljaju vrhunac tehnologije projektilnog oružja. Ovi sofisticirani sustavi sposobni su isporučiti bojne glave na velike udaljenosti s preciznošću, često koristeći složene sustave navođenja i pogonjene raketnim gorivom. Balistički projektili mogu nositi konvencionalne ili nuklearne bojne glave, čineći ih ključnim elementom moderne vojne strategije i odvraćanja.

Transformacija projektilnih oružja od rudimentarnih alata do naprednih sustava balističkih projektila naglašava neumornu potragu za tehnološkim inovacijama u ratovanju. Danas balistički projektili služe ne samo kao kritični dijelovi nacionalne obrane već i kao simboli tehnološke moći i geopolitičke moći. Njihov razvoj nastavlja oblikovati globalnu vojnu dinamiku, utječući na obrambene politike i međunarodne odnose na duboke načine.

Jedan od ključnih aspekata učinkovitosti balističkih projektila je sustav lansiranja. Ovi sustavi omogućuju ne samo brzo i precizno ispaljivanje projektila, već i prilagodljivost u različitim

vojnim situacijama. Sustavi lansiranja, kao što su mobilne platforme ili silosi, igraju ključnu ulogu u povećanju preživljavanja i iznenađenja, čime se osigurava strateška prednost. Efikasnost lansiranja može odrediti ishod misija, a modernizacija i razvoj novih lansirnih sustava postali su prioriteti u vojnoj strategiji mnogih zemalja. Ovi sustavi stoga ne samo da doprinose snazi oružanih snaga, već i oblikuju način na koji se balistički projektili integriraju u šire vojne operacije.

Zbog izrazite bitnosti lansirnih sustava u ovom radu će se pokazati kako bi se jedan od tih sustava napravio.

1.1. Balistički projektili

Balistički projektili predstavljaju jednu od najznačajnijih inovacija u povijesti ratovanja, simbolizirajući tehnološki napredak i evoluciju vojnih strategija. Povijest balističkih projektila počinje u ranom 20. stoljeću, ali njihovi korijeni sežu mnogo dalje, do prvih pokušaja korištenja raketne tehnologije.

Prvi zabilježeni primjeri korištenja raketa u ratne svrhe datiraju iz srednjovjekovne Kine, gdje su se jednostavne rakete pogonjene barutom koristile kao oružje. No, tek je 20. stoljeće, s napretkom u razumijevanju aerodinamike, kemije goriva i preciznog navođenja, omogućilo razvoj pravih balističkih projektila.

Razvoj balističkih projektila značajno je napredovao tijekom Drugog svjetskog rata, kada je nacistička Njemačka razvila V-2 raketu, prvi balistički projektil dugog dometa. Ovaj projektil bio je tehnološko čudo svog vremena, koristeći tekuće gorivo i inercijalni navigacijski sustav. Nakon rata, američki i sovjetski znanstvenici usvojili su i unaprijedili ovu tehnologiju, što je dovelo do hladnoratovske utrke u naoružanju i razvoja interkontinentalnih balističkih projektila (ICBM).



Slika 1. Balistička raketa V-2 [1]

Balističke rakete funkcioniraju prema načelu balističke putanje, koja je zakrivljeni put projektila pod utjecajem gravitacije i početnog pogona.

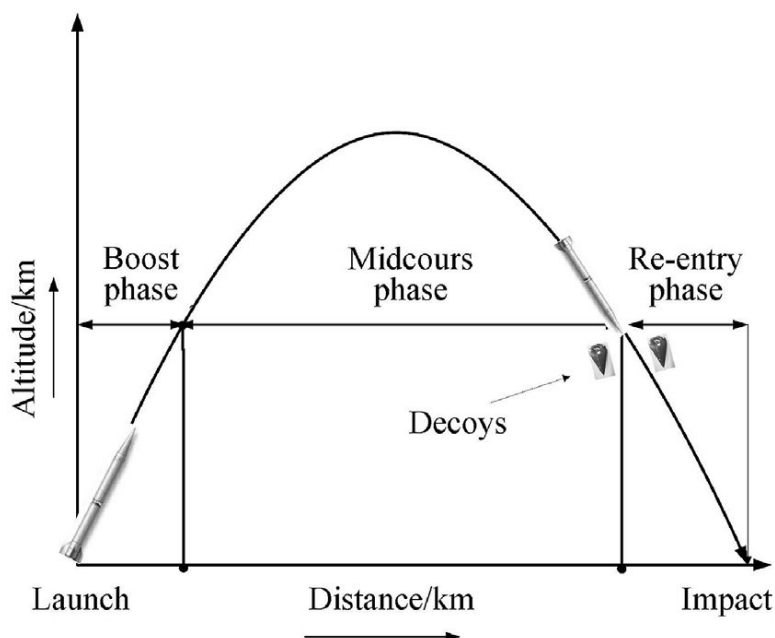
Balističke rakete obično prolaze kroz tri glavna stadija tijekom svog leta: paljenje, srednji stadij i povratni stadij.

Prvi stadij, paljenje(Boost phase), započinje u trenutku kada se raketa lansira i završava kada motor prestane s radom. Ovaj stadij obično traje od nekoliko sekundi do otprilike 2-3 minute, ovisno o tipu rakete. Tijekom paljenja raketa koristi svoj raketni motor (ili više njih) za generiranje potiska, uspinjući se prema nebu, često vertikalno ili pod kutom. Ovaj je stadij karakteriziran velikim brzinama i temperaturama, što ga čini najlakšim za detekciju od strane protuzračnih sustava. Također, u ovom stadiju troši se velika količina goriva, a raketa može dostići visoke brzine.

Nakon paljenja, prelazimo na srednji stadij(Midcourse phase), koji započinje kada raketa izlazi iz atmosfere i završava kada se vraća u atmosferu. Ovaj stadij može trajati od nekoliko minuta do desetaka minuta, ovisno o ruti i visini leta. U srednjem stadiju raketa izlazi iz atmosfere i prelazi u svemir, često dostižući visine od nekoliko stotina kilometara. U ovoj fazi raketa može koristiti inercijalne navigacijske sustave i često je opremljena različitim sustavima za navođenje. S obzirom na veliku udaljenost i visinu, raketa može biti manje vidljiva za detekciju,

ali se može koristiti i satelitska navigacija. Ako raketa nosi više bojnih glava, u ovoj se fazi često odvajaju pojedinačne glave.

Treći stadij je povratni stadij(Re-entry phase), koji započinje kada raketa ili njezine bojne glave ponovno ulaze u atmosferu i završava kada se bojne glave isporučuju na cilj. Ovaj stadij traje od nekoliko sekundi do nekoliko minuta. Tijekom povratka u atmosferu, bojne glave prolaze kroz ekstremne uvjete, uključujući visoke temperature i brzine. Zbog trenja i kompresije zraka, temperature mogu dostići nekoliko tisuća stupnjeva, što zahtijeva posebne materijale za zaštitu. U ovoj se fazi bojne glave kreću velikim brzinama, što otežava njihovo presretanje. Na kraju ovog stadija, bojne glave mogu se aktivirati kako bi isporučile svoj sadržaj, bilo putem eksplozije ili druge vrste djelovanja.



Slika 2. Stadiji balističkih raketa [2]

1.2. Lanseri balističkih projektila

Lanseri balističkih raketa su sustavi dizajnirani za lansiranje balističkih projektila. Ovi sustavi su ključni za isporuku projektila s visokom preciznošću i učinkovitošću. Postoji nekoliko vrsta lansera za balističke rakete, a svaki ima svoje prednosti i mane ovisno o specifičnim zahtjevima i uvjetima korištenja

Svaka vrsta lansera za balističke rakete ima svoje specifične prednosti i koristi se u različitim strategijskim kontekstima. Silosni lanseri pružaju visoku razinu zaštite i spremnosti, mobilni kopneni lanseri omogućuju visoku mobilnost i fleksibilnost, željeznički lanseri koriste postojeće infrastrukture za skrivenost, podmornice nude izvanrednu prikrivenost i sposobnost drugog udara, dok zračni lanseri omogućuju brzu reakciju i fleksibilnost. Izbor lansera ovisi o vojnoj strategiji, geografskim uvjetima i specifičnim potrebama obrane.



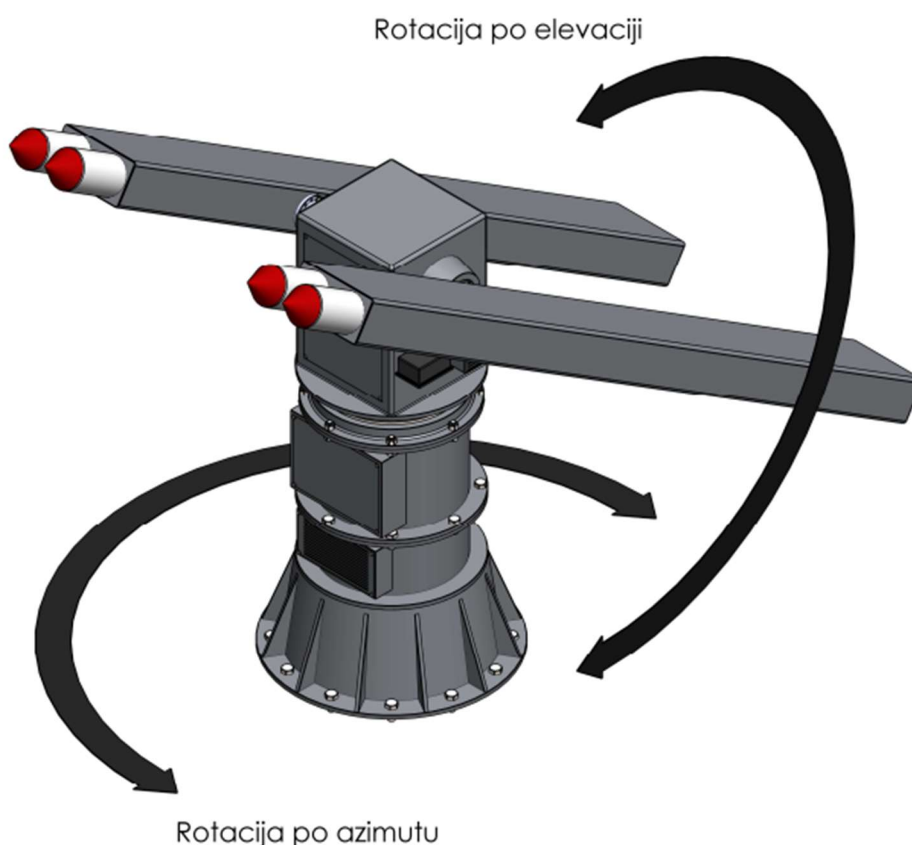
Slika 3. Sustav za lansiranje balističkih raketa [3]

Postoje razne izvedbe lansera balističkih projektila sa rotacijom po azimutu i elevaciji, no postoji vrlo malo javno dostupnih informacija o njima. Kombinirajući dostupne informacije o postojećim lanserima i sličnim tehnologijama napravljen je lanser u ovome radu.

2. MEHANIZAMI I KONSTRUKCIJA

Servo sustav koji se projektira mora imati mogućnost rotacije po azimutu i elevaciji kako bi se osigurala točnost i preciznost ciljanja. Od postojećih izvedba lansera jedina koja omogućuju pozicioniranje prema azimutu i elevaciji je izvedba s mobilnim lanserom na kopnu, iz tog razloga naš lanser ćemo projektirati tako da se montira na prijevozno vozilo.

Pomoću računalnog programa za 3D modeliranje Solidworks napravljeno je idejno rješenje servo sustava za lansiranje. Mehanizam je projektiran tako da ima dva stupnja slobode gibanja tj. dvije rotacije. Pogon sustava će dobivati upute o rotiranju od senzora orijentacije i položaja koji će kasnije biti odabrani u radu.

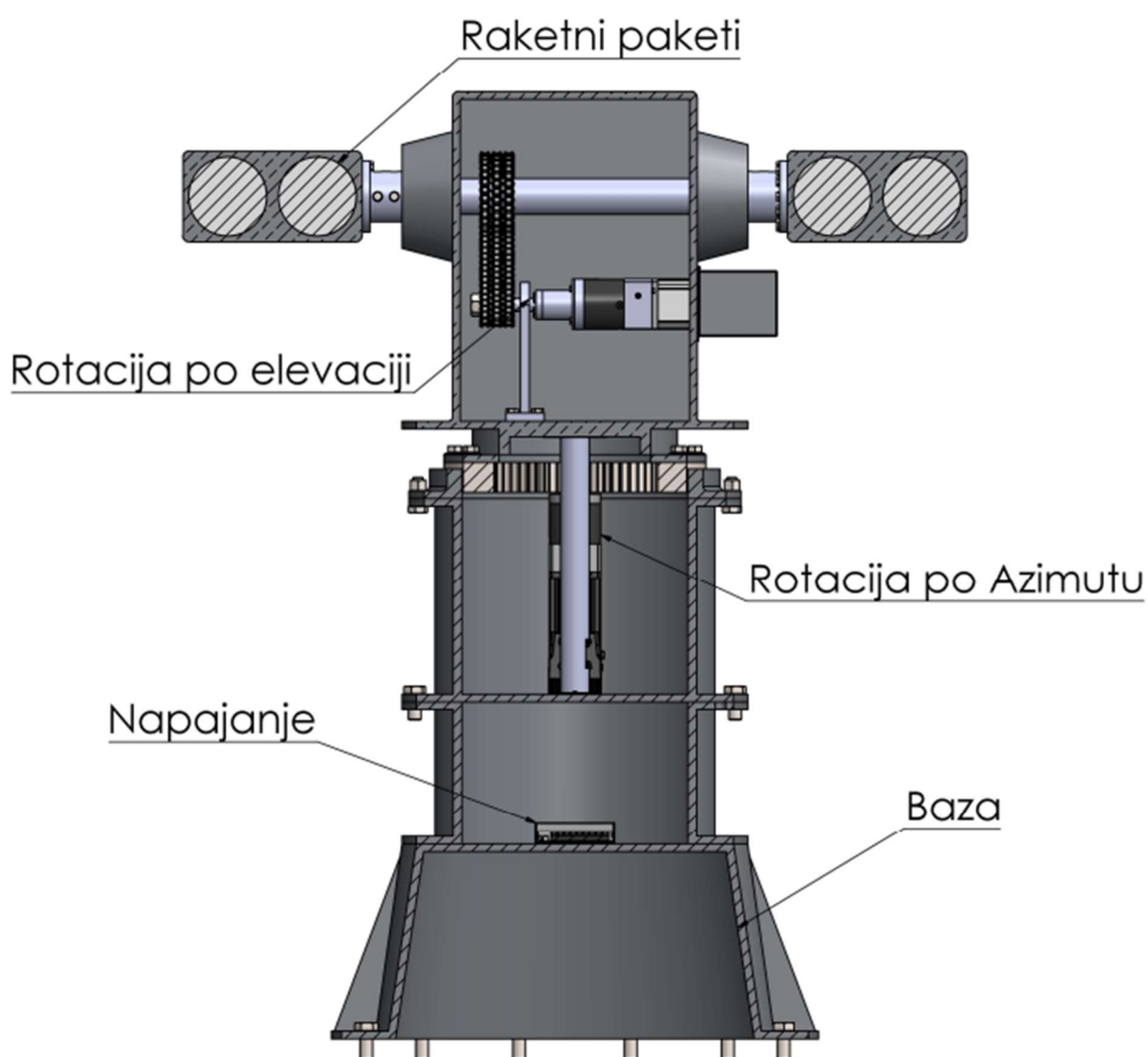


Slika 4. Idejno rješenja servo sustava za lansiranje

Rakete prema kojima ćemo dimenzionirati lanser su M-21OF-M. Maksimalni doseg koji želimo na našem lanseru je 20 kilometara. Podkonstrukcija se izvodi velikim dijelom od aluminijske legure Al-Li 2195 koja osigurava lanseru nisku masu uz održavanje visoke čvrstoće i otpornosti na visoke temperature koje se pojavljuju tijekom lansiranja raketa. Ukupna masa

podkonstrukcije za lansiranje raketa je približno $m_{ko} = 144 \text{ kg}$, masa je očitana iz programa Solidworks zadavanjem gustoće materijala dijelovima koje motor mora pokretati. Masa pojedine rakete iznosi $m_{ra} = 66 \text{ kg}$ te s obzirom da ih na lanseru imamo 4, ukupna masa raketa i konstrukcije iznosi $m_{la} = 408 \text{ kg}$.

Konstrukcija lansera je podijeljen na 5 glavnih dijelova: Rotacija po Azimutu, Rotacija po elevaciji, Raketni paket, Napajanje i Baza.



Slika 5. Podjela konstrukcije

2.1. Rotaciju po azimutu

Brzina okretaja motora za rotaciju po azimutu je prema preporuci za lansere $n_{r1} = 20 \frac{o}{min}$.

Kako bi se naš sustav mogao rotirati potreban je mehanizam koji će prenijeti okretni moment s motora na lanser. Pri konstruiranju ovakvih lansera i Okretnih topova na tenkovima ili sličnih konstrukcija, za rotaciju po azimutu se upotrebljavaju okretni ležajevi. Okretni ležajevi se koriste jer omogućuju precizno i stabilno pozicioniranje teških konstrukcija s minimalnim trenjem. Njihova robusnost i pouzdanost osiguravaju sigurno usmjeravanje raketa prema cilju, čak i u ekstremnim uvjetima.



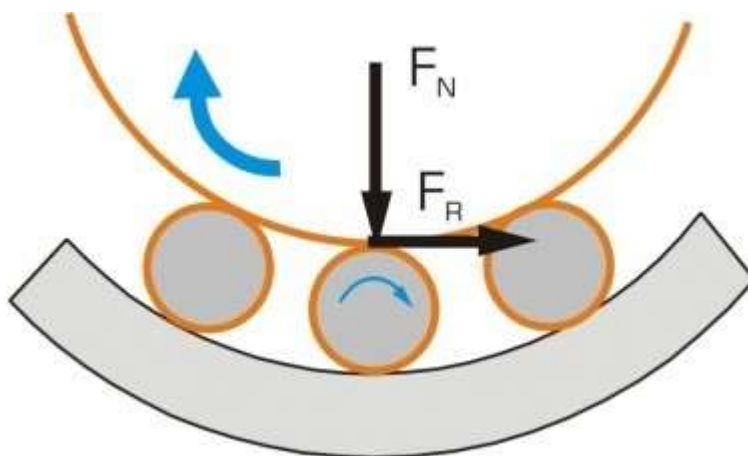
Slika 6. Okretni ležaj sa unutarnjim ozubljenjem [4]

Dvije stvari koje sprječavaju okretanje ležaja su sila trenja koja se stvara u samom ležaju te inercija. Kako bi bili sigurni da se ležaj može nesmetano okretati potrebno je izračunati okretni moment koji se prenosi na njega putem motora. Okretni ležaj koji je odabran za prijenos momenta je RKS.22 041 četveroredni kuglični rotirajući ležaj s unutarnjim zupčanikom lagane serije iz SKF-ovog kataloga.

Tablica 1. Karakteristike gonjenog zupčanika[5]

Modul- m_2	5
Broj zubi- z_2	67
Diobena kružnica- d_2	335 mm

Kako bi ležaj nesmetano rotirao okretni moment koji se stvara na gonjenom zupčaniku mora biti veći od zbroja momenta trenja i momenta inercije. Moment trenja se stvara na kuglicama u samome ležaju. Ležajevi koriste maziva kako bi se utjecaj trenja maksimalno smanjio te samim time je koeficijent trenja podosta nizak. Sila trenja djeluje da mjestu kontakta kuglice sa samim ležajem. Ležaj koji koristimo ima kuglice s kontaktom u četiri točke što čini analizu trenja kompleksnom, kako bi se izbjegla kompleksnost zamišljamo kako ima kontakt u jednoj točki te povećavamo koeficijent trenja kako bi kompenzirali što računamo s jednom točkom.



Slika 7. Poligon sila u ležaju [6]

Normalna sila koja djeluje na ležaj se računa prema izrazu 1.

$$F_N = g \cdot (m_{la}) = 9,81 \cdot (408) = 4002,48 \text{ N} \quad (1.)$$

gdje je:

F_N – Normalna sila na ležaju, N

g – Ubrzanje zemljine sile teže, m/s^2

m_{ra} – masa lansera, kg

Opće poznati izraz za silu trenja iz normalne sile 2.

$$F_{TR} = F_N \cdot \mu_1 = 4002,48 \cdot 0.03 = 120.1 \text{ N} \quad (2.)$$

gdje je:

F_{TR} – Sila trenja u ležaju, N

μ_1 – Koeficijent trenja u ležaju

Te konačno iz izraza 3 možemo dobiti moment trenja u ležaju.

$$T_{TR} = F_{TR} \cdot \frac{d_{tr}}{2} = 120.1 \cdot \frac{0.411}{2} = 24.68 \text{ Nm} \quad (3.)$$

gdje je:

T_{TR} – Moment trenja u ležaju, Nm

d_{tr} – Promjer po kojem djeluje sila trenja, mm

Rotacijska inercija je svojstvo svakog objekta koji se može rotirati. To je skalarna vrijednost koja nam govori koliko je teško promijeniti brzinu rotacije objekta oko određene osi rotacije. Rotacijska inercija igra sličnu ulogu u rotacijskoj mehanici kao masa u linearnoj mehanici. Naime, rotacijska inercija objekta ovisi o njegovoj masi, ali i o raspodjeli te mase u odnosu na os rotacije. Kada se masa pomakne dalje od osi rotacije, postaje sve teže promijeniti brzinu rotacije sustava. Intuitivno, to je zato što masa sada nosi više momenta dok se kreće po krugu (zbog veće brzine) i jer se vektor momenta brže mijenja. Oba ova učinka ovise o udaljenosti od osi rotacije.

Tijelo koje rotira po azimutu ima podosta složenu geometriju za analizu, te ovisi u kojoj se poziciji mijenja iznos inercije. Iz toga razloga računat ćemo kao da imamo solidan disk. Izraz za računanje momenta inercije diska je dan u izrazu 4.

$$I_D = \frac{1}{2} \cdot (m_{la}) \cdot r_{ro}^2 = \frac{1}{2} \cdot 408 \cdot 0.5^2 = 51 \text{ kgm}^2 \quad (4.)$$

gdje je:

I_D – Moment inercije diska, kgm^2

r_{ro} – polumjer rotacije, mm

Okretni moment inercije se računa kao umnožak kutne akceleracije i momenta inercije. Prvo moramo izračunati kutnu brzinu po izrazu 5. kako bi mogli dobiti akceleraciju.

$$\omega_{r1} = \frac{\pi}{30} \cdot n_{r1} = \frac{\pi}{30} \cdot 20 = 2.094 \frac{\text{rad}}{\text{s}} \quad (5.)$$

gdje je:

ω_{r1} – kutna brzina rotacije po azimutu, $\frac{\text{rad}}{\text{s}}$

Akceleraciju dobivamo tako da podijelimo brzinu po vremenu, s time da uzimamo da je vrijeme potrebno deset sekundi te uvrstimo u izraz 6.

$$\alpha_{r1} = \frac{\omega_{r1}}{t_{r1}} = \frac{2,094}{10} = 0,2094 \frac{rad}{s^2} \quad (6.)$$

gdje je:

α_{r1} – kutna akceleracija rotacije po azimutu, $\frac{rad}{s}$

t_{r1} – vrijeme rotacije po azimutu, s

I konačno prema izrazu 7. dobivamo okretni moment inercije.

$$T_{I1} = \alpha_{r1} \cdot I_D = 0,2094 \cdot 51 = 10,6794 Nm \quad (7.)$$

gdje je:

T_{I1} – Okretni moment inercije po azimutu, $\frac{rad}{s}$

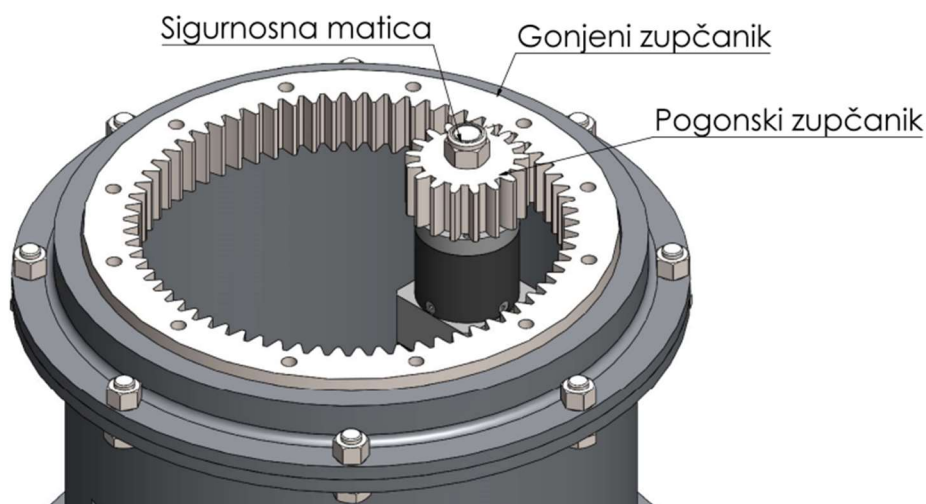
Ukupni moment koji se mora oboriti je zbroj inercije i trenja te iznosi:

$$T_{UK1} = (T_{I1} + T_{TR}) \cdot S_1 = (10.6794 + 34.1) \cdot 1.5 = 67,17 Nm \quad (8.)$$

gdje je:

T_{UK1} – Ukupni okretni moment po azimutu, $\frac{rad}{s}$

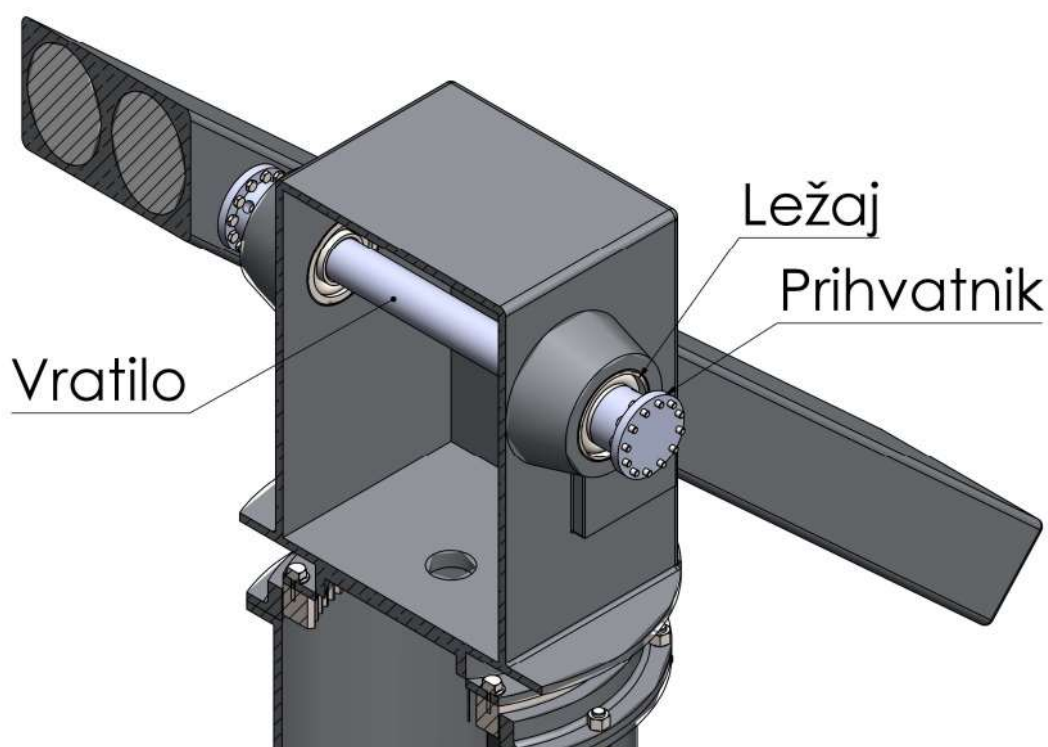
Čeoni zupčanik je osiguran na motoru s pomoću M12 matice i motor je pričvršćen na ostatak konstrukcije kako bi prenosio moment bez problema. Okretni ležaj je pričvršćen na ostatak konstrukcije s osam M18 vijaka s pripadajućim maticama. Na gonjenom zupčaniku se nalaze M14 navojne rupe na koje se pričvršćuje ostatak konstrukcije s dvanaest vijaka.



Slika 8. Mehanizam rotacije po Azimutu

2.2. Rotacija po elevaciji

Kako bi se raketni paket mogao pozicionirati pod idealnim kutom za lansiranje za predviđenu udaljenost lansiranja potreban je mehanizam koji to omogućuje. Na raketne pakete moment će se prenositi preko vratila koje je povezano s prihvatnikom na raketni paket te ima uležištenje s pomoću dva ležaja s obje strane. Prihvatnik je povezan s pomoću dvanaest M12 vijaka na raketni paket te s pomoću osam M8 vijaka na vratilo.



Slika 9. Elementi za prijenos momenta

Pri prijenosu moment dolazi kao i kod rotacije po azimutu dolazi do stvaranja trenja i inercije, no u slučaju elevacije ležajevi koji se koriste su malog promjera te okretni moment koji se stvara je reda veličine tisućinke te se ne uzima u obzir pri odabiru motora. Glavne stavka prema kojoj odabiremo motor je inercija pri rotaciji raketnog paketa. Raketni paket je oblika kvadra te ima masu $m_{pa} = 182 \text{ kg}$ koja se rotira pri brzini od $n_{r2} = 10 \frac{o}{min}$.

Moment inercije se izračunava s pomoću izraza 13. pošto se radi o geometrijskom tijelu oblika kvadra.

$$I_K = \frac{1}{12} \cdot m_{pa} \cdot (a^2 + b^2) = \frac{1}{12} \cdot 182 \cdot (0,14^2 + 2,87^2) = 125.225 \text{ kgm}^2 \quad (13.)$$

gdje je:

I_K – Moment inercije kvadra, kgm^2

a i b – dimenzije kvadra okomite na os rotacije, mm

Kutna brzina i akceleracija se izračunavaju kao i kod rotacije po azimutu te se dobiva kutna brzina $\omega_{r2} = 1,05 \frac{rad}{s}$ i kutna akceleracija pri vremenu od pet sekundi $\alpha_{r2} = 0,21 \frac{rad}{s^2}$. Prema izrazu 14. dobivamo okretni moment te množimo s dva kako bi uzeli u obzir dva raketna paketa koja se nalaze na lanseru.

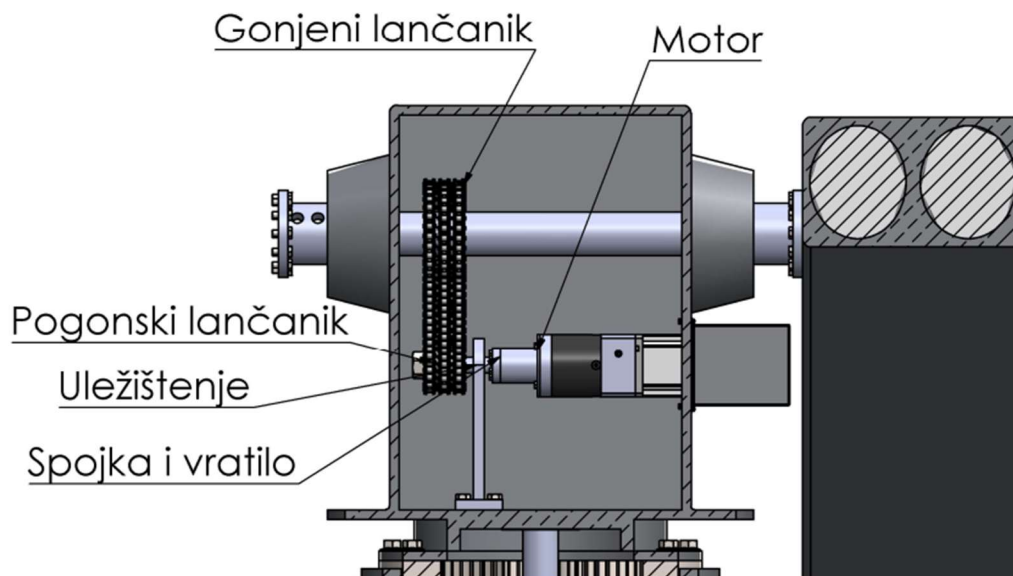
$$T_{I2} = \alpha_{r2} \cdot I_K = 2 \cdot 0,21 \cdot 125,225 = 52,6 Nm \quad (14.)$$

gdje je:

T_{I2} – Okretni moment inercije po elevaciji, $\frac{rad}{s}$

Kako bi prenijeli moment s motora na vratilo potreban je mehanizam prijenosa okretnog momenta. Za prijenos je odabran lančani prijenos. Korištenje lančanog prijenosa za rotaciju po elevaciji lansera nudi brojne prednosti, uključujući visoku sposobnost prijenosa okretnog momenta, što je ključno za precizno podizanje teških tereta. Lančani prijenos osigurava pouzdan rad bez proklizavanja, omogućujući točno namještanje kuta elevacije, što je od velike važnosti za preciznost ciljanja. Osim toga, lančani prijenos je poznat po svojoj dugotrajnosti i niskom održavanju, što ga čini idealnim za vojnu upotrebu gdje su pouzdanost i trajnost ključni. Njegov kompaktan dizajn omogućuje učinkovitu primjenu u ograničenim prostorima, pružajući stabilno i precizno kretanje.

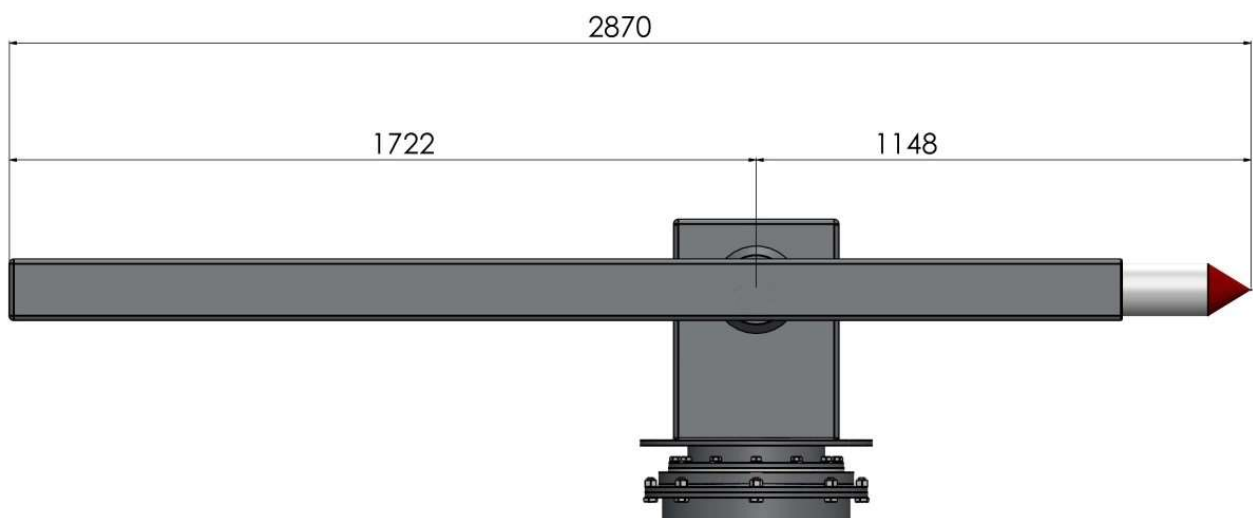
Kako bi osigurali sigurnu rotaciju lančanog prijenosa koristit ćemo sustav s tri lanca. Za pogonski lančanik koristimo valjkasti lančanik sa tri reda zubi koji ima, te isto takav koristimo za gonjeni lančanik. S motora na lančanik se moment prenosi s pomoću spojke i vratila. Vratilo ima ležaj koji pomaže da se sile koje djeluju na njemu rasporede po dužini vratila.



Slika 10. Mehanizam rotacije po elevaciji

2.3. Raketni paket

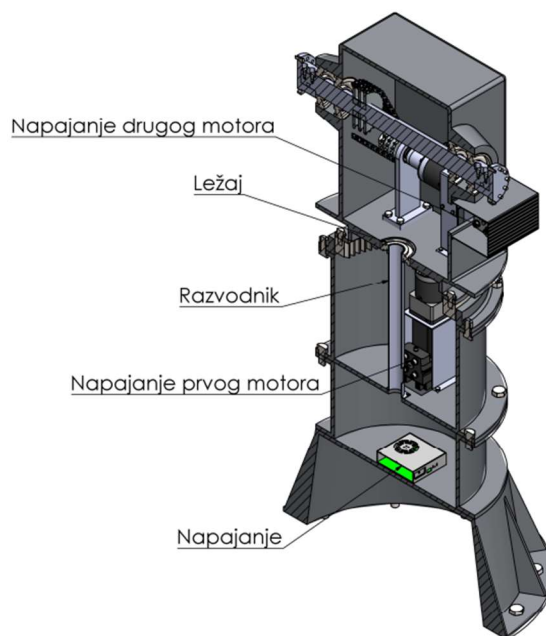
Kako ne bi dolazilo do stvaranja okretnog momenta zbog težine raketnih paketa, rakete se moraju postaviti u ravnotežni položaj. Balističke rakete sastoje se od bojevnje glave, motora na čvrsto ili tekuće gorivo, sustava za navođenje i kontrolu te strukturalnog okvira koji sve komponente drži zajedno. Bojevnja glava nosi eksplozivni ili drugi korisni teret, motor pruža potisak za lansiranje i let, a sustav za navođenje kontrolira putanju rakete. Težište rakete obično se nalazi na 40 % ukupne dužine rakete od prednjeg dijela. Ukupna dužina rakete je 2870 mm te se prihvatnik raketnog paketa postavlja na 40 % ukupne dužine od početka raketnog paketa što iznosi 1148 mm.



Slika 11. Ravnotežni položaj lansera

2.4. Napajanje

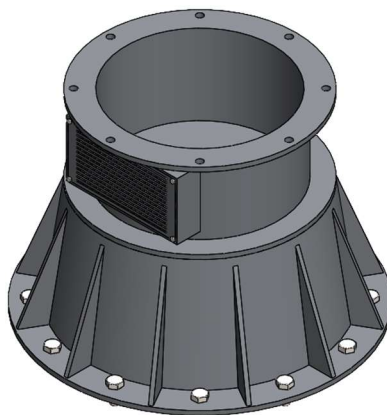
Napajanje motora se nalazi ispod prve rotacije lansera, kablovi koji služe za napajanje motora dolaze do napajanja kroz razvodnik koji služi kako se kablovi ne bi zapetljali. Razvodnik je na dnu povezan vijcima te umetnut u ležaj kako bi se lanser mogao rotirati oko cijevi te se tako ne povlače žice. Konkretno napajanje će biti odabrano kasnije u radu prema zahtjevima motora.



Slika 12. Organizacija napajanja motora

2.5. Baza lansera

Lanser u bazi ima dvanaest M20 rupa za povezivanje s vozilom. Lanser se treba prevoziti u neutralnom položaju s horizontalno postavljenim raketnim paketom kako bi gabariti vozila i lansera bili što manji i jednostavniji za transport. Baza na sebi ima rebra kao konstrukcijsko ojačanje protiv sile tlaka.

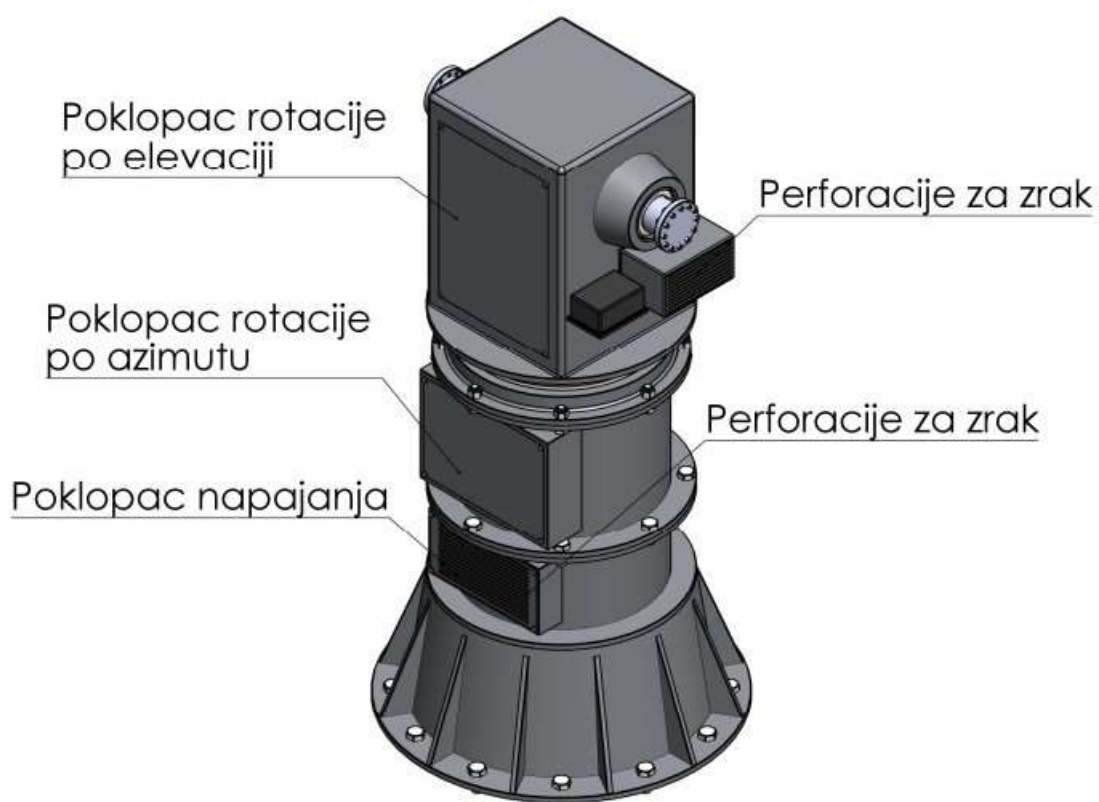


Slika 13. Baza lansera

2.6. Servisiranje i hlađenje lansera

U slučaju kvara ili potrebe za zamjenom određenih dijelova lansera potrebno ga je konstruirati tako da se s lakoćom može doći do svakog dijela lansera. Iz tog razloga svaki glavni dio lansera u kojemu se nalaze komponente ima svoj poklopac preko kojeg se servisira.

Sve komponente se pri radu zagrijavaju a pogotovo motori i napajanje, iz tog razloga je potrebno napraviti perforacije na mjestu gdje se nalaze hladnjaci motora i samo napajanje kako bi mogao ući svježi zrak također se može ugraditi poseban sistem hlađenja po potrebi.



Slika 14. Poklopci i perforacije za hlađenje

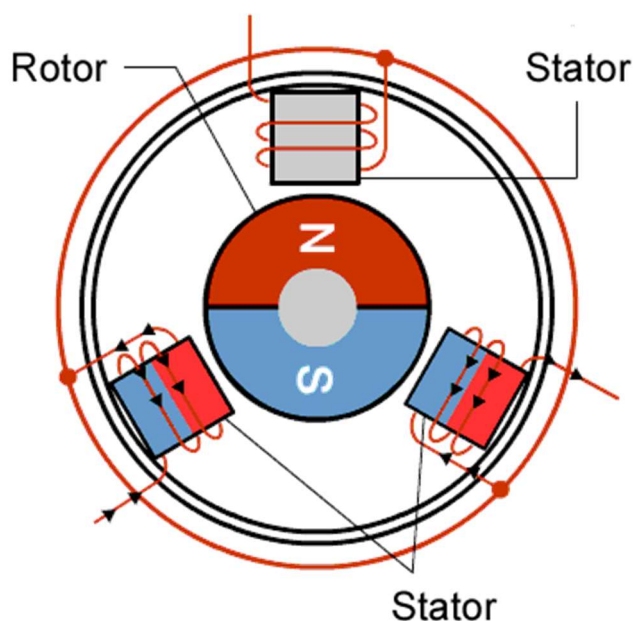
3. MOTORI

3.1. BLDC Motor

Kako bi se lanser mogao rotirati po azimutu i elevaciji potrebni su mu motori koji će to omogućiti. Pri odabiru motora za pogon postoji više različitih vrste koje možemo odabrati poput: pneumatskih, hidrauličkih ili električnih. Za izradu ovog rada su izabrani električni DC motori. DC motori imaju nekoliko ključnih prednosti u usporedbi s pneumatskim i hidrauličkim sustavima kada se koriste za upravljanje lanserima balističkih raketa. Prvo, omogućuju izuzetno preciznu kontrolu položaja i brzine, što je kritično za točno usmjeravanje raketa prema cilju. Pneumatski i hidraulički sustavi ne mogu postići istu razinu preciznosti zbog kompresibilnosti zraka i gubitaka u tekućinama.

DC motori mogu zahtijevati složeniju električnu infrastrukturu i osjetljiviji su na pregrijavanje. Nasuprot tome, pneumatski i hidraulički sustavi mogu biti robusniji u ekstremnim uvjetima, no njihova niža preciznost i učinkovitost često ih čine manje prikladnima za precizno upravljanje lanserima balističkih raketa.

U izradi ovog projekta koristit će se BLDC motori tj. Brushless DC motori. BLDC motori su napredna vrsta elektromotora koji koristi elektronsku komutaciju, umjesto četkica i komutatora. Ova tehnologija omogućuje tiši rad, manju potrebu za održavanjem i veću energetska učinkovitost.

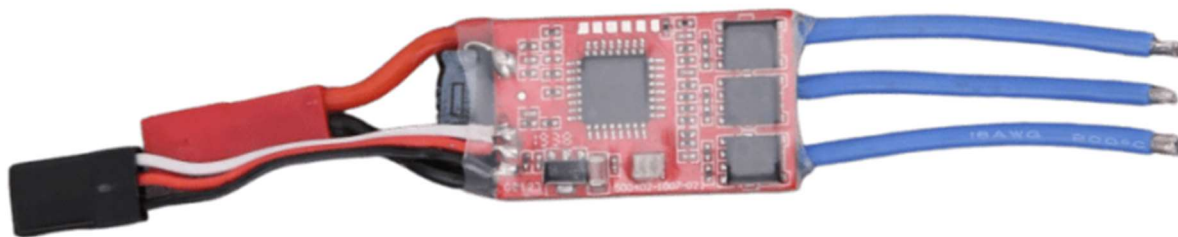


Slika 15. BLDC motor [7]

BLDC motor radi tako da elektronski kontroler, koristeći podatke iz senzora položaja, usmjerava struju kroz zavojnice statora, stvarajući rotirajuće magnetsko polje koje povlači rotor s permanentnim magnetima, omogućujući njegovu kontinuiranu rotaciju bez potrebe za fizičkim četkicama.

3.2. Upravljanje motora

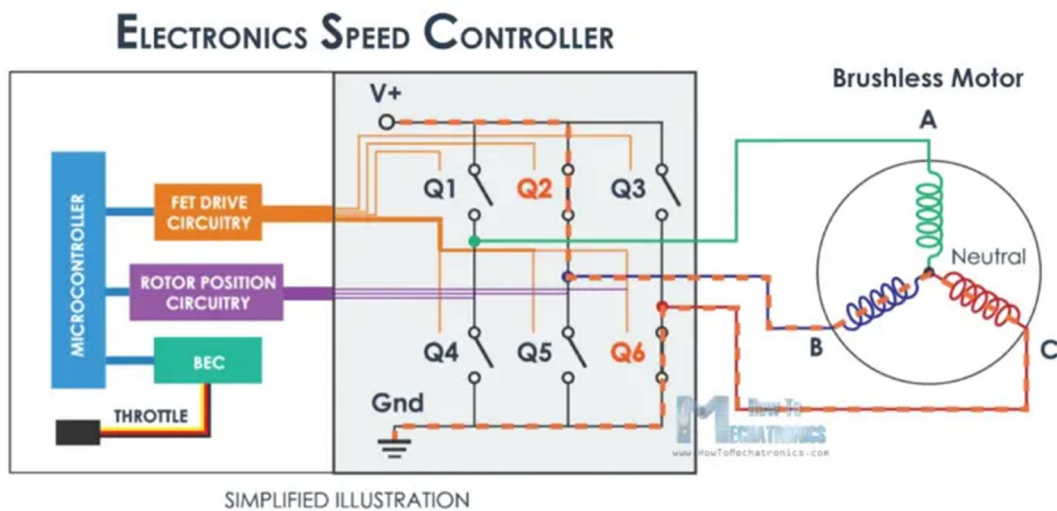
BLDC motori se upravljaju s pomoću ESC-a što znači “elektronička kontrola brzine”, regulira brzinu (gas) i funkciju električnog motora. Nudi trofazni izvor električne energije i može biti zasebna jedinica ili integrirana unutar prijemnika. Određeni proizvođači integriraju ove upravljače sa motorima.



Slika 16. ESC kontroler [8]

Brzina istosmjernog motora s četkicama može se mijenjati promjenom primijenjenog ulaznog napona. Međutim, u slučaju beskontaktnih istosmjernih motora, ulazni napon nije stalan istosmjerni napon. Upravljački napon je trofazni napon koji može biti u obliku kvadratnog ili sinusnog vala. Frekvencija, amplituda i faza/širina impulsa ovog ulaznog napona mogu promijeniti brzinu i moment beskontaktnog istosmjernog motora (BLDC). Stoga je potreban zaseban sklop za kontrolu brzine BLDC motora.

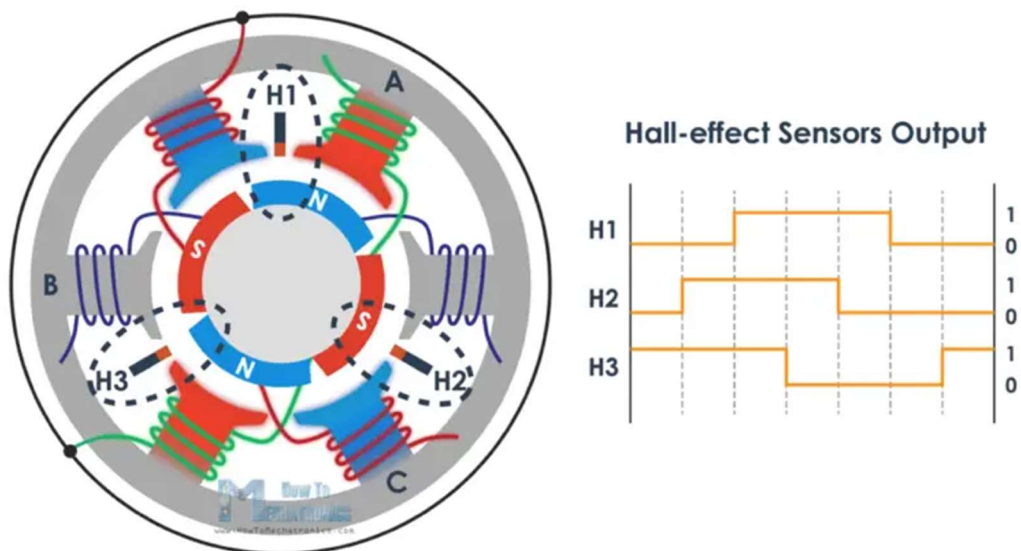
ESC ili elektronički regulator brzine kontrolira kretanje ili brzinu beskontaktnog motora aktiviranjem odgovarajućih MOSFET-ova kako bi se stvorilo rotirajuće magnetsko polje koje omogućuje rotaciju motora. Što je viša frekvencija ili što brže ESC prolazi kroz 6 intervala, to će brzina motora biti veća.



Slika 17. Princip rada BLDC motora [9]

Kako bi znali kada aktivirati koju fazu moramo znati položaj rotora, a za određivanje položaja rotora koriste se dvije uobičajene metode.

Prva uobičajena metoda uključuje korištenje Hallovih senzora ugrađenih u stator, koji su raspoređeni pod kutom od 120 ili 60 stupnjeva jedan od drugog.

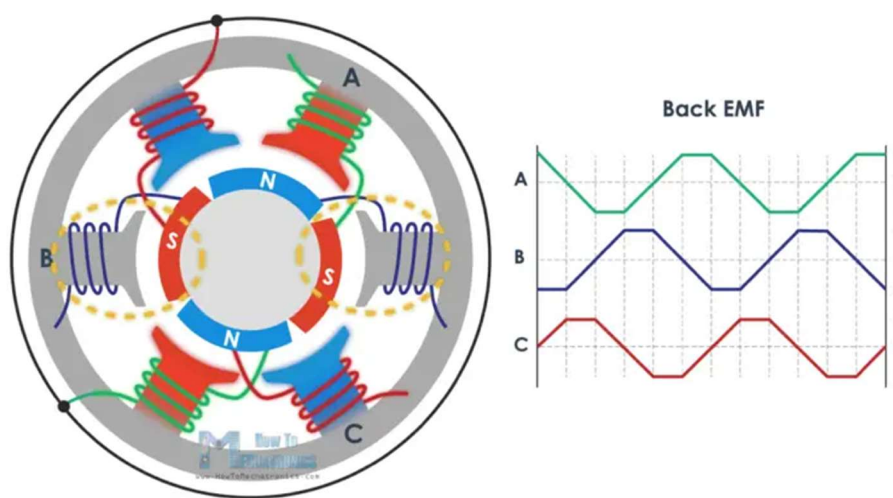


Slika 18. Određivanje položaja rotora pomoću Hallovih senzora [9]

Kako se permanentni magneti rotora okreću, Hallovi senzori detektiraju magnetsko polje i generiraju logičku vrijednost "high" za jedan magnetski pol ili logičku vrijednost "low" za

suprotni pol. Na temelju tih informacija ESC zna kada treba aktivirati sljedeći sekvencijalni prijelaz ili interval.

Druga uobičajena metoda za određivanje položaja rotora je s pomoću mjerenja povratne elektromotorne sile (back EMF). Povratna EMF nastaje kao rezultat suprotnog procesa stvaranja magnetskog polja, odnosno kada se pokretno ili promjenjivo magnetsko polje prolazi kroz zavojnicu, ono inducira struju u zavojnici.



Slika 19. Određivanje položaja rotora pomoću elektromotorne sile [9]

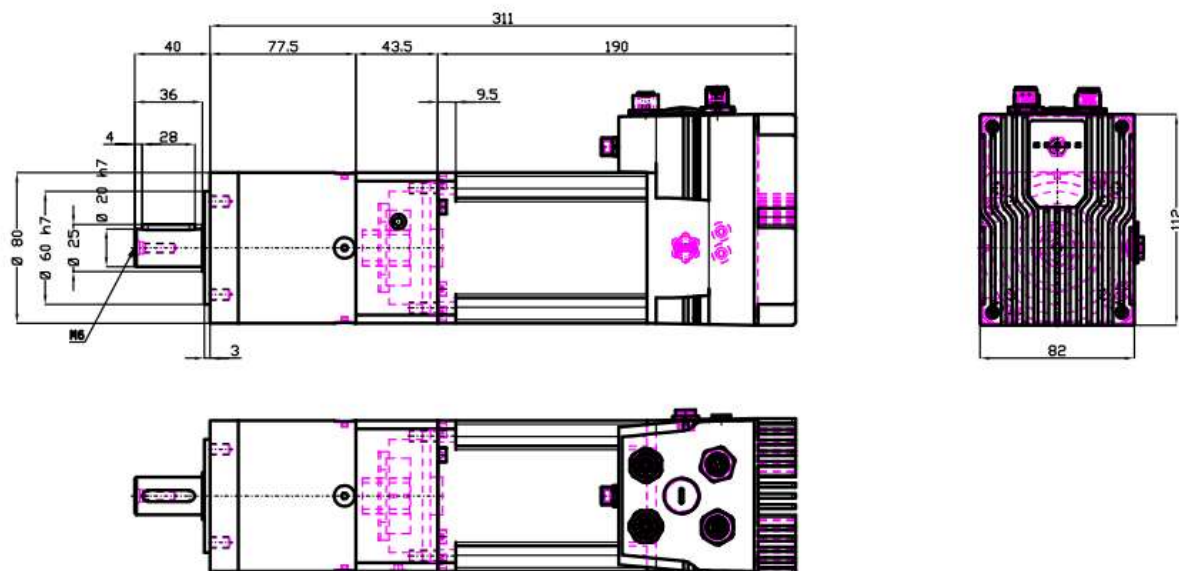
Kada se pokretno magnetsko polje rotora prolazi kroz slobodnu zavojnicu, odnosno onu koja nije aktivna, inducira struju u toj zavojnici i kao rezultat toga nastaje pad napona u zavojnici. ESC bilježi te padove napona kako se javljaju i na temelju njih predviđa ili izračunava kada bi sljedeći interval trebao nastupiti.

3.3. Odabir motora za rotaciju po azimutu

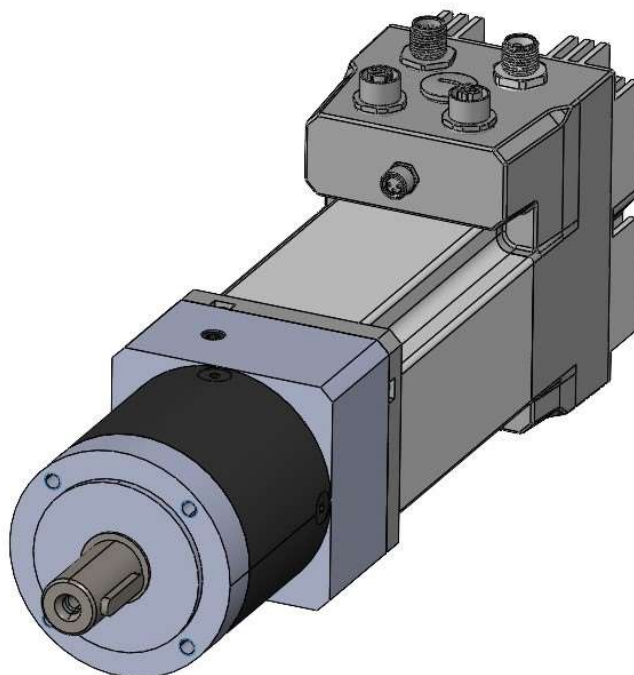
Pomoću izračunatih vrijednosti okretnog momenta rotacije se odabiru zadovoljavajući motori. Zadaća motora je da precizno postave projekte u poziciju iz koje će nakon lansiranja pogoditi cilj te zahtijevaju veliku preciznost.

Prvi motor koji će biti dimenzioniran je zaslužan za rotaciju po azimutu. Prvi motor mora biti sposoban rotirati kompletnu težinu svih raketa te konstrukcije na kojoj se nalaze rakete. U prethodnom poglavlju smo izračunali potrebni okretni moment za rotaciju po azimutu koji se treba oboriti.

Odabrani motor je PD6-EB BLDC motor od tvrtke Nanotec koji ima integrirani planetarni reduktor GPLE80, kontoler za upravljanje i inkrementalni kontroler.



Slika 20. Dimenzije motora PD6-EB [10]



Slika 21. Motor PD6-EB sa intergriranim mjenjačem GPLE80 [10]

Tablica 2. Karakteristike motora za rotaciju po azimutu [10]

Nazivna brzina vrtnje motora	3000 o/min.
Nazivna snaga	0,94 kW (1,26 KS)
Nazivni okretni moment	3 Nm
Maksimalni radni tlak	7 bar
Jačina zvuka	76 dB (A)
Radna temperatura	-10°C-40°C
Karakteristike nakon redukcije mjenjačem	
Nazivna brzina vrtnje nakon mjenjača	80 o/min.
Nazivni okretni moment nakon mjenjača	25 Nm
Težina	5,5 kg

Za prijenos momenta sa motora na unutarnji zupčanik se koristi čelni zupčanik. Pri kontaktu zupčanika u prijenosu momenta se stvara normalna sila koja se može rastaviti na obodnu i radijalnu komponentu. Kako bi se održavala ravnoteža obodne komponente moraju biti jednake.

Tablica 3. Karakteristike pogonskog zupčanika

Modul- m_1	5
Broj zubi- z_1	17
Diobena kružnica- d_1	85 mm

Prijenosni omjer iznosi :

$$i_1 = \frac{z_2}{z_1} = \frac{67}{17} = 3,94 \quad (9.)$$

gdje je:

i_1 – Prijenosni omjer rotacije po azimutu

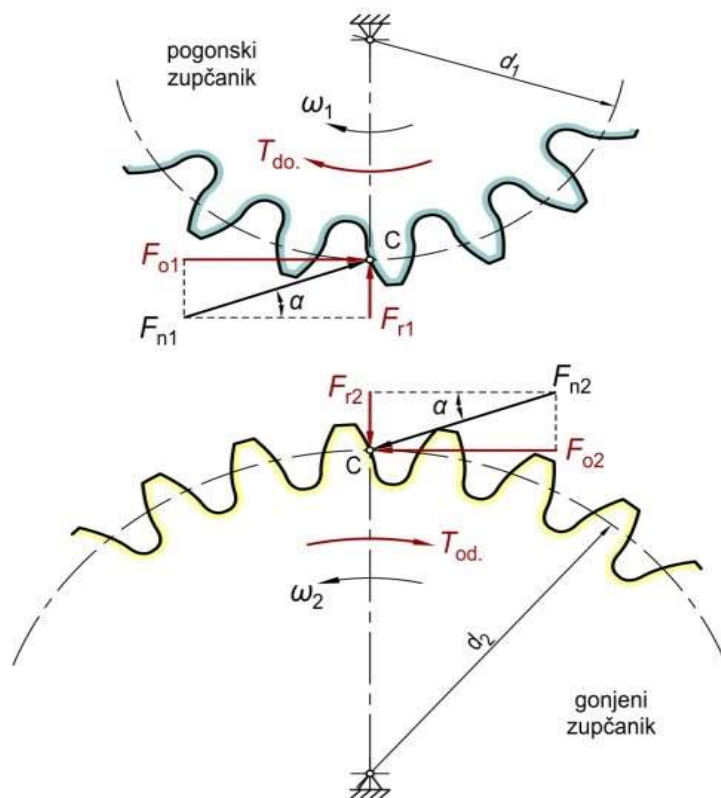
Zupčanici su odabrani tako da se preko prijenosnog omjera postigne željena brzina okretanja lansera pri radnom okretnom momentu. Pošto je radna brzina 80 okretaja po sekundi na pogonskom zupčaniku pomoću izraza 10. dobivamo brzinu na gonjenom zupčaniku koja zadovoljava potrebnu.

$$n_{r1} = \frac{n_1}{i_1} = \frac{80}{3,94} = 20,3 \frac{o}{min.} \quad (10.)$$

gdje je:

i_1 – Prijenosni omjer rotacije po azimutu

Pri kontaktu zupčanika u prijenosu momenta se stvara normalna sila koja se može rastaviti na obodnu i radijalu komponentu. Kako bi se održavala ravnoteža obodne komponente moraju biti jednake.



Slika 22. Poligon sila na pogonskom i gonjenom zupčaniku [11]

Pomoću okretnog momenta na pogonskom zupčaniku te njegovog diobenog promjera možemo izračunati obodnu komponentu normalne sile prema izrazu 11.

$$F_o = \frac{2 \cdot T_1}{d_1} = \frac{2 \cdot 30}{0.085} = 588,235 \text{ N} \quad (11.)$$

gdje je:

T_1 – Okretni moment pogonskog zupčanika, Nm

d_1 – Promjer diobene kružnice pogonskog zupčanika, mm

F_o – Obodna komponenta normalne sile

Obodna komponenta normalne sile su jednake na oba zupčanika iz tog razloga pomoću formule 12. možemo izračunati okretni moment koji se stvara na gonjenom zupčaniku.

$$T_2 = \frac{\eta_z \cdot F_O \cdot d_2}{2} = \frac{0,98 \cdot 588,235 \cdot 0.335}{2} = 96,559 \text{ Nm} > 67,17 \text{ Nm} \quad (12.)$$

gdje je:

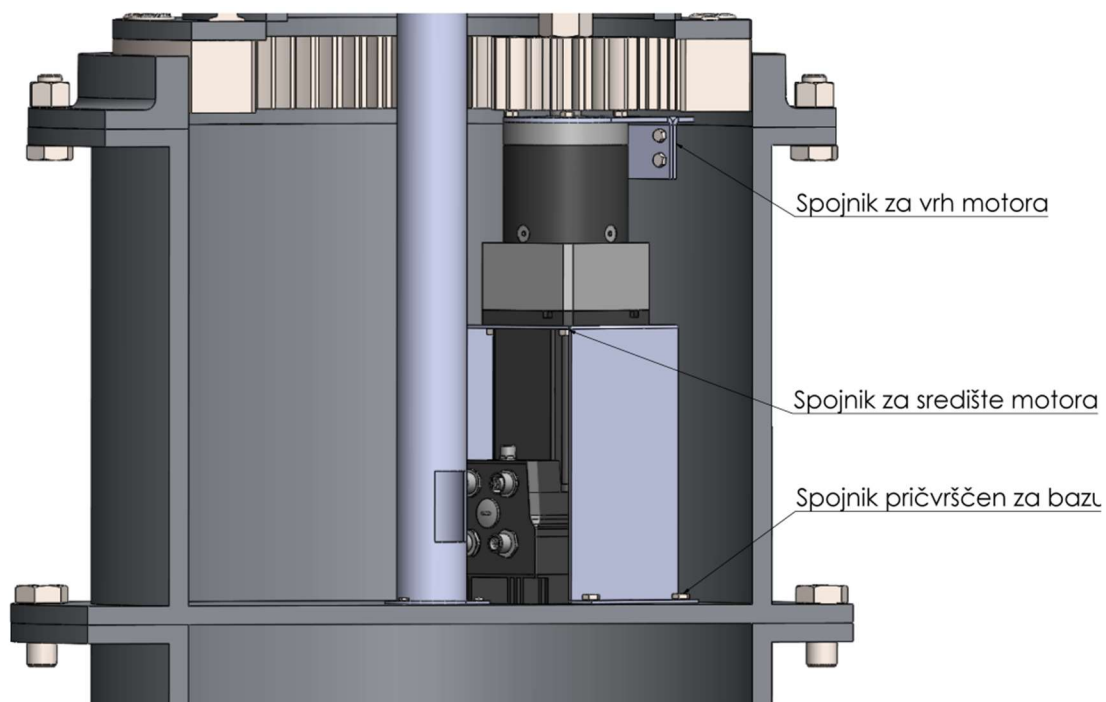
T_2 – Okretni moment gonjenog zupčanika, Nm

d_2 – Promjer diobene kružnice gonjenog zupčanika, mm

η_z – Stupanj djelovanja zupčanog prijenosa

Motor zadovoljava potrebni okretni moment

Motor je osiguran na ostatak konstrukcije pomoću dva spojnika, prvi se nalazi na vrhu gdje je spojen sa 4 M6 vijka na vrh motora koji u sebi ima navojne rupe. Spojnik je povezan na lim koji je zavaren na ostatak konstrukcije sa 4 M5 vijaka. Drugi spojnik se nalazi na sredini konstrukcije te je također kroz njega provučeno 4 M6 vijaka sa kojima je motor na sredini učvršćen, središnji spojnik je pričvršćen za bazu sa 4 M5 vijaka kako bi se u potpunosti osigurao prijenos momenta sa motora na mehanizam.



Slika 23. Konstrukcija za osiguranje prvog motora

3.4. Odabir motora za rotaciju po elevaciji

Postupak odabire motora za drugu rotaciju je skorom identičan postupku odabira prvog motora. Kako bi se raketni paket mogao pozicionirati pod idealnim kutom za lansiranje za predviđenu udaljenost lansiranja potreban je motor koji to omogućuje.

Motor odabran za rotaciju po elevaciji je isti kao i kod rotacije po azimutu sa jedinom razlikom da motor za elevaciju ima ugrađeni apsolutni višeokretni enkoder.

Tablica 4. Karakteristike motora za rotaciju po elevaciji [10]

Nazivna brzina vrtnje motora	3000 o/min.
Nazivna snaga	0,94 kW (1,26 KS)
Nazivni okretni moment	3 Nm
Maksimalni radni tlak	7 bar
Jačina zvuka	76 dB (A)
Radna temperatura	-10°C-40°C
Karakteristike nakon redukcije mjenjačem	
Nazivna brzina vrtnje nakon mjenjača	30 o/min.
Nazivni okretni moment nakon mjenjača	30 Nm
Težina	5,5 kg

Prijenosni omjer rotacije po elevaciji dobivamo pomoću izraza 15.

$$i_2 = \frac{z_4}{z_3} = \frac{45}{15} = 3 \quad (15.)$$

gdje je:

z_3 – Broj zubi na pogonskom lančaniku

z_4 – Broj zubi na gonjenom lančaniku

i_2 – Prijenosni omjer lančanog prijenosa

Lančanici su također odabrani tako da se preko prijenosnog omjera postigne željena brzina okretanja lansera pri radnom okretnom momentu. Pošto je radna brzina 30 okretaja po sekundi na pogonskom lančaniku pomoću izraza 16. dobivamo brzinu na gonjenom lančanika koja zadovoljava potrebnu.

$$n_{r2} = \frac{n_3}{i_2} = \frac{30}{3} = 10 \frac{o}{min.} \quad (16.)$$

gdje je:

i_2 – Prijenosni omjer rotacije po elevaciji

Okretni moment koji se stvara na gonjenom lančaniku dobivamo pomoću izraza 17.

$$T_4 = T_3 \cdot i_2 \cdot \eta_z = 30 \cdot 3 \cdot 0,85 = 76,5 \text{ Nm} > 52,6 \text{ Nm} \quad (17.)$$

gdje je:

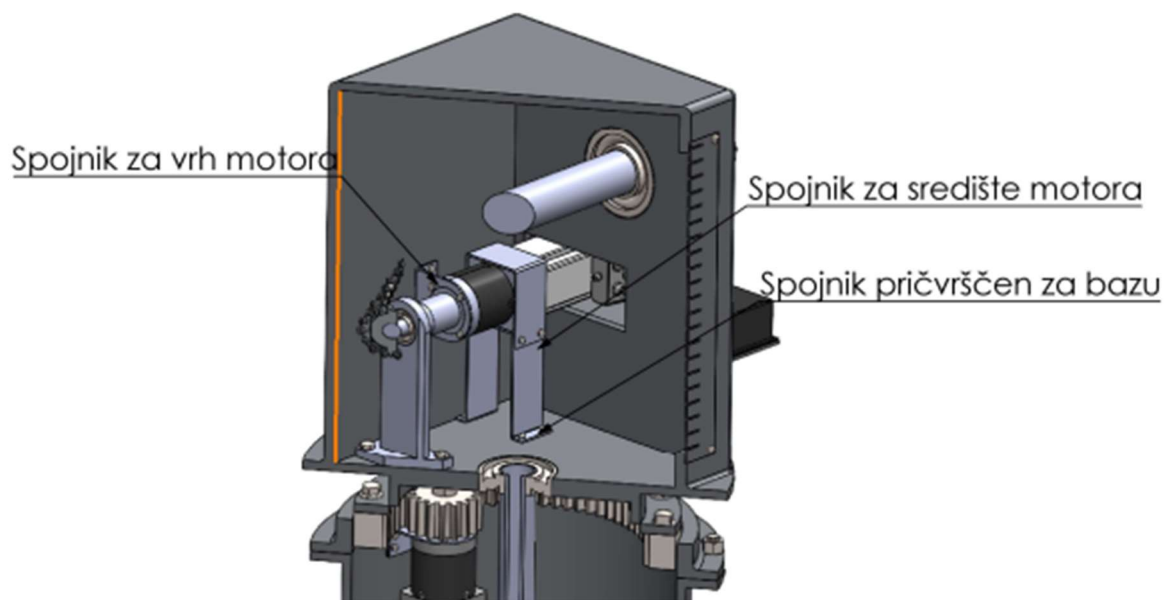
T_4 – Okretni moment gonjenog lančanika Nm

T_3 – Okretni moment pogonskog lančanika, Nm

η_l – Stupanj djelovanja lančanog prijenosa

Motor zadovoljava potrebni okretni moment

Motor je spojen



Slika 24. Konstrukcija za osiguranje drugog motora

3.5. Napajanje i kalibracija motora

3.5.1 Napajanje

Motori se napajaju naponom od 12 VDC do 57.6 VDC. Za motora se koristi agregat koji bi se trebao montirati na vozilo koje prevozi lanser. Korištenje agregata za napajanje lansera raketa predstavlja značajnu opciju, posebno kada je riječ o sustavima koji zahtijevaju visoke razine energije ili kada su operacije dugotrajne. Jedna od glavnih prednosti korištenja agregata je njegova sposobnost da pruži visoke razine snage. Agregati su dizajnirani za opskrbu velikih količina električne energije, što može biti presudno za sustave poput lansera raketa koji

zahtijevaju značajnu energiju za rad. Ova značajka čini agregate idealnim za dugotrajne operacije gdje baterijski sustavi ne bi bili dovoljno učinkoviti ili praktični.

Osim toga, agregati nude fleksibilnost u pogledu snage i napona. Mnogi moderni agregati omogućuju prilagodbu izlaznog napona, što omogućuje korisnicima da prilagode energiju prema specifičnim potrebama svojih sustava. Ova prilagodljivost može biti korisna u okruženjima gdje se energija mora prilagoditi različitim komponentama ili uvjetima rada.

Još jedna prednost korištenja agregata je smanjena ovisnost o baterijama. Baterijski sustavi mogu biti ograničeni kapacitetom i potrebom za čestim punjenjem. Agregat, koji koristi gorivo kao izvor energije, može pružiti neprekidnu opskrbu energijom, što omogućuje dugotrajne operacije bez potrebe za stalnim punjenjem.

Primjer agregat koji bi se mogao koristiti je Agregat Honda EU22i. Agregat Honda EU22i ima pouzdanu izlaznu snagu od 2200 W i prilagodljive potrošnje goriva, što ga čini idealnim za napajanje. Njegove kompaktnije dimenzije olakšavaju transport. Da bi se spojio na motore, koristi se AC-DC mjenjač za konverziju izlaza od 120 VAC na potreban DC napon. Izlaz agregata povezuje se s ulazom mjenjača, a zatim se izlaz mjenjača povezuje s motorima kako bi se osigurao odgovarajući DC napon. Agregat se uključuje, mjenjač se postavlja na željeni napon, te se provjerava stabilnost napajanja motora.



Slika 25. Agregat Honda EU22i [12]

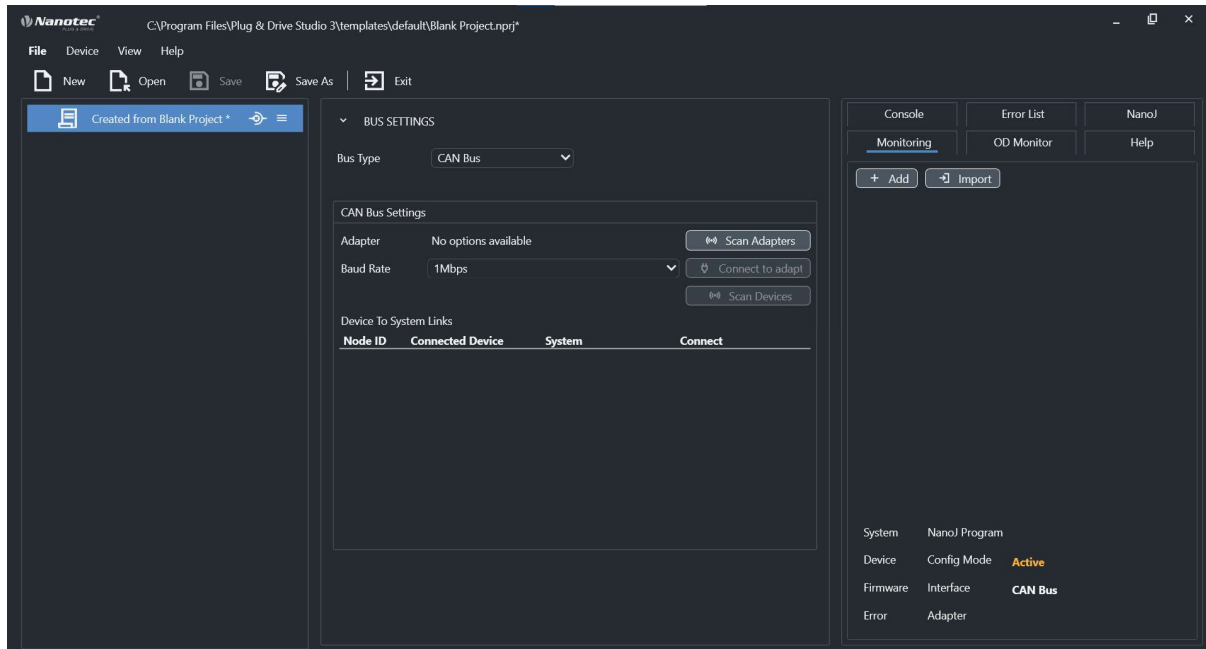
3.5.2. Kalibracija motora

Za kalibraciju motora se koristi program Plug and Drive Studio 3. U Plug and Drive Studio 3, korisnici mogu učinkovito konfigurirati i optimizirati BLDC motore tvrtke Nanotec. Nakon što

se poveže motor s računalom preko odgovarajućeg sučelja poput USB-a, RS485-a ili Ethernet-a, korisnik pokreće softver i kreira novi projekt ili otvara postojeći. Automatsko prepoznavanje motora omogućuje softveru da preuzme specifične podatke o motoru, što uključuje radne napone, struje, brzine i postavke enkodera.

Nakon inicijalne detekcije, korisnici unose ili učitavaju parametre motora, uključujući električne karakteristike kao što su otpor i induktivitet. Konfiguracija enkodera također je ključna, gdje se postavljaju vrsta enkodera, razlučivost i povratne informacije. Odabirom odgovarajućeg načina kontrole, poput kontrole momenta, brzine ili pozicije, korisnici mogu prilagoditi motor kako bi odgovarao specifičnim zahtjevima aplikacije.

Kalibracija motora je sljedeći korak, koji omogućava optimizaciju kontrolnih petlji. Podesivi PID parametri (proporcionalni, integralni i derivativni) omogućuju fino podešavanje performansi motora, što poboljšava njegovu stabilnost i odzivnost. Napredne postavke, kao što su profil brzine, akceleracija i deceleracija, također se mogu prilagoditi za postizanje željenih performansi.



Slika 26. Sučelje Plug and Drive Studio 3 [10]

Softver pruža alate za nadzor u stvarnom vremenu, što omogućuje korisnicima da promatraju podatke poput brzine motora, struje i položaja. Analizom ovih podataka moguće je identificirati

potencijalne probleme ili prilike za daljnje poboljšanje. Ako se pojave bilo kakvi problemi, alati za dijagnostiku pomažu u rješavanju tih problema.

Kada je optimizacija završena, konfiguracija se sprema ili direktno u motor ili u softver, omogućujući tako primjenu podešenih postavki u stvarnom aplikativnom okruženju. Ovaj pristup omogućuje korisnicima da osiguraju da motor radi prema očekivanjima, s poboljšanom performansom i pouzdanošću.

Kako bi kalibrirali konkretan motor, s obzirom da nemamo taj motor ovdje je dan općeniti postupak kalibracije.

4. SENZORI

Za pravilnu orijentaciju lansera prema cilju potrebni su senzori koji će omogućiti da znamo trenutnu lokaciju i orijentaciju lansera te mikrokontroler koji će služiti kao međuračunalo između senzora i motora. Senzori i mikrokontroleri vojnog standarda koji se koriste u realnim lanserima nisu javno dostupni te ih često proizvodi sama vojska. Kako bi što jednostavnije prikazali projektiranje ovakvog lansera koristit ćemo javno dostupan mikrokontroler i senzore.

4.1. Mikrokontroler

Za mikrokontroler pomoću kojeg ćemo očitavati informacije sa senzora ćemo koristiti Arduino Uno od kompanije Arduino. Arduino je talijanska tvrtka osnovana 2005. godine, poznata po razvoju pristupačnih i jednostavnih platformi za mikrokontrolere, namijenjenih za edukaciju i projekte u elektronici. Arduino Uno je jedna od najpopularnijih ploča iz njihove ponude, poznata po svojoj pouzdanosti i svestranosti. Opremljen s 14 digitalnih ulaza/izlaza i 6 analognih ulaza, Uno omogućava korisnicima povezivanje s raznim sensorima i uređajima, što ga čini idealnim za znanstvene primjene poput prikupljanja podataka, eksperimentiranja s automatikom i stvaranja prototipova.



Slika 27. Arduino Uno Pločica [13]

Napajanje Arduino Uno pločice može se ostvariti putem USB priključka ili korištenjem vanjskog izvora. Vanjski izvor napajanja može biti AC ili DC adapter, ili baterija. Ako koristite adapter, on se priključuje na 2.1mm priključak za napajanje, dok se baterija može spojiti na pinove Gnd i Vin.

Arduino pločica podržava rad s napajanjem u rasponu od 6 do 20V. Međutim, ako napon padne ispod 7V, napon na 5V pinu može biti manji od 5V, što može uzrokovati nestabilnost

rada. S druge strane, ako se koristi napon veći od 12V, postoji rizik od pregrijavanja naponskog regulatora, što može oštetiti pločicu. Stoga se preporučuje korištenje napajanja u rasponu od 7 do 12V. Ključni naponski pinovi na pločici uključuju: Vin, 5V, 3.3V, Gnd, i IOREF.

Svaki od 14 pinova na Arduino Uno-u može se upotrijebiti kao ulazni ili izlazni upotrebljavajući funkcije: `pinMode()`, `digitalWrite()`, `digitalRead()`. Svaki pin može pružati ili primati maksimum od 40mA. Grupe pinova imaju posebne funkcije:

- Serijski – 0 (Rx) i 1 (Tx)
- Vanjski interapti – 2 i 3
- PWM – 3, 5, 6, 9, 10 i 11
- SPI – 10, 11, 12, 13
- LED – 13 Uno ima i 6 analognih pinova, i to su redom od A0 do A5.

Arduino Uno se upravlja putem Arduino IDE programa, koji omogućava korisnicima da pišu, uređuju i prenose kod na ploču. Kroz Arduino IDE, korisnici mogu koristiti jednostavan programski jezik temeljen na C/C++-u kako bi definirali funkcionalnosti svog projekta. Kada je kod napisan, on se kompilira i učitava na Arduino Uno preko USB veze, nakon čega ploča automatski izvršava upute. Arduino IDE također nudi ugrađene primjere i biblioteke koje olakšavaju upravljanje različitim komponentama, poput senzora i motora.

4.2. Senzor položaja

Za senzor položaja se koristi GPS modul u-blox NEO-M8. U-blox NEO-M8 je napredni GPS modul koji nudi visoku preciznost i fleksibilnost za različite aplikacije. Pruža podršku za više globalnih navigacijskih sustava uključujući GPS, GLONASS, Galileo i BeiDou, čime omogućava veću pouzdanost i točnost praćenja, čak i u složenim okruženjima. Modul se odlikuje brzim početkom praćenja, velikom preciznošću (do 2.5 metra) i mogućnošću korištenja višestrukih frekvencija za poboljšanje signala, što ga čini idealnim za autonomna vozila, dronove i druge aplikacije koje zahtijevaju visoku točnost.



Slika 28. GPS modul NEO-M8 [14]

GPS modul NEO-M8 koristi serijsku komunikaciju za prijenos podataka prema Arduino, pri čemu se knjižnica TinyGPS++ koristi za obradu i izdvajanje relevantnih GPS informacija kao što su geografska širina, dužina i visina. Ovaj proces uključuje nekoliko ključnih koraka, počevši od hardverskog povezivanja pa sve do obrade podataka, što omogućuje Arduino da prima i interpretira podatke s modula.

NEO-M8 modul šalje GPS podatke u formatu NMEA (National Marine Electronics Association) rečenica. Ove rečenice su tekstualni nizovi koji sadrže različite informacije o poziciji, brzini, visini, vremenu i još mnogo toga. Primjer NMEA rečenice izgleda ovako:

`$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47`

Svaki dio rečenice predstavlja različitu informaciju. U ovom primjeru, "GPGGA" označava vrstu rečenice (Global Positioning System Fix Data), a kasniji dijelovi predstavljaju podatke o vremenu, geografskoj širini, dužini, kvaliteti GPS signala i visini.

Kada je NMEA rečenica uspješno obrađena, Arduino može pristupiti podacima putem funkcija poput `gps.location.lat()` i `gps.location.lng()`, koje vraćaju trenutnu geografsku širinu i dužinu.

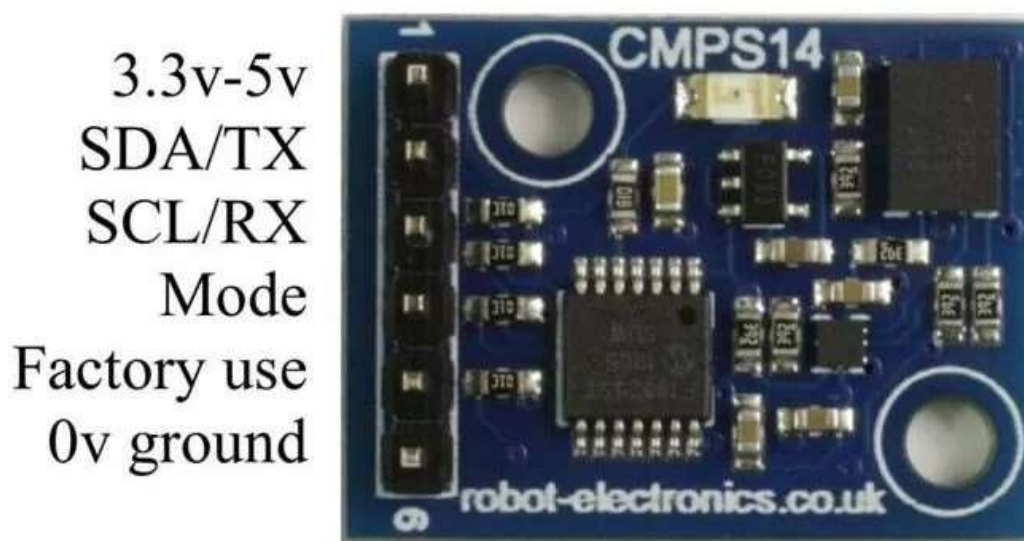
Modul se napaja u rasponu od 3.0V do 5.5V i omogućava energetski učinkovito djelovanje, što je važno za prijenosne uređaje. Za optimalan rad potrebno je koristiti vanjsku antenu koja

poboljšava prijem satelitskih signala. Kompaktne dimenzije modula olakšavaju njegovu integraciju u razne uređaje, čineći ga svestranim rješenjem za precizno određivanje položaja.

4.3 Senzor orijentacije

Za orijentaciju se koristi CMPS14. CMPS14 je kompas pete generacije s kompenzacijom nagiba. Koristi 3-osni magnetometar, 3-osni žiroskop i 3-osni akcelerometar. U srži modula nalazi se vrhunski BNO080 koji koristi algoritme za uklanjanje pogrešaka uzrokovanih nagibom PCB-a. Modul također omogućava zaustavljanje kalibracije i korištenje statičkog kalibracijskog profila umjesto dinamičke kalibracije.

Za napajanje, modul je fleksibilan i može se napajati s naponom između 3.3V i 5V, a troši nominalnih 18mA struje. Za komunikaciju se koristi I2C sučelje.



Slika 29. CMPS14 kompas [15]

4.4. Algoritam postavljanja cilja i potrebnog azimuta

Kod tradicionalnih vojnih operacija u kojima se postavlja nekakav cilj se koriste vojne karte prema pomoću kojih se očitavaju koordinate cilja. Očitavanje koordinata cilja s vojnih karti je pouzdan način za osiguranje točnosti, jer su te karte napravljene prema strogim vojnim standardima. Vojne karte pružaju detaljne informacije o terenu i taktičkim uvjetima, što pomaže u boljem planiranju i provođenju operacija. Korištenje standardiziranih koordinatnih sustava

također olakšava komunikaciju između različitih vojnih jedinica. Osim toga, karte su uvijek dostupne i pouzdane, čak i u situacijama kada moderna tehnologija zakaže.

Pomoću GPS senzora znamo trenutačne koordinate na kojima se lanser nalazi a pomoću kompasa znamo koja je njegova orijentacije tj. kako je orijentiran u odnosu na pravi sjever. Kako bi algoritam funkcionirao podrazumijeva se da znamo trenutne koordinate i orijentaciju pomoću senzore te da zadajemo cilj tako da unosimo njegove koordinate koje se očitavaju pomoću karte.

Algoritam se temelji na Haversine formuli. Haversine formula je matematička metoda koja se koristi za izračunavanje najkraće udaljenosti između dvije točke na sfernoj površini, poput Zemlje. Ova formula uzima u obzir zakrivljenost Zemlje, što je važno za precizne proračune na velikim udaljenostima.

Formula se temelji na trigonometriji i koristi geografske širine i dužine točaka izražene u radijanima. Izračun se vrši pomoću sljedećih formula:

$$A = \sin\left(\frac{\Delta\Phi}{2}\right)^2 + \cos\Phi_1 \cdot \cos\Phi_2 \cdot \sin\left(\frac{\Delta\lambda}{2}\right)^2 \quad (18.)$$

$$c = 2 \cdot \text{atan2}(\sqrt{A}, \sqrt{1-A}) \quad (19.)$$

$$d = R \cdot c \quad (20.)$$

gdje je:

Φ_1 – Geografska širina trenutačne lokacije

Φ_2 – Geografska širina cilja

$\Delta\Phi$ – Razlika u geografskim širinama

$\Delta\lambda$ – Razlika u geografskim duljinama

R – Polumjer zemlje

d – udaljenost između dviju točaka

Pomoću Haversine formula dobiva se formula za azimut koja glasi:

$$\text{Azimut} = \text{atan2}(\sin(\Delta\lambda) \cdot \cos(\Phi_2), \sin(\Phi_2) - \sin(\Phi_1) \cdot \cos(\Phi_2) \cdot \cos(\Delta\lambda)) \quad (21.)$$

Algoritam kreće tako da se prvo unose koordinate trenutne lokacije lansera i ciljne točke. Koordinate se sastoje od geografske širine i dužine. Da bi se izračunao azimut između ove dvije točke, geografske širine moraju se pretvoriti u radijane. Ovaj korak je ključan jer su matematičke funkcije koje se koriste za izračunavanje azimuta, kao što su sinus i kosinus, zasnovane na radijanima, a ne na stupnjevima.

Nakon pretvorbe širina u radijane, sljedeći korak je izračunavanje razlike u geografskoj dužini između ciljane i trenutne točke, također u radijanima. Ovo je bitno za precizno određivanje smjera jer razlika u dužini utječe na konačni azimut.

Uz pomoć Haversine formule, koja se koristi za izračunavanje azimuta na temelju ove razlike, računa se azimut između dvije točke. Azimut se izračunava kao kut između sjevera i linije koja povezuje trenutnu lokaciju s ciljem. Ovaj kut se prvo dobiva u radijanima, a zatim se pretvara u stupnjeve. Da bi se osiguralo da azimut bude u rasponu od 0° do 360° , primjenjuje se normalizacija.

Nakon što je azimut izračunat, potrebno je dohvatiti trenutni smjer lansera. Smjer lansera, koji se dobiva iz kompasa ili senzora, koristi se za usporedbu s izračunatim azimutom.

Daljnji korak uključuje izračunavanje najkraćeg kuta za rotaciju motora. Ovo uključuje izračunavanje kuta za rotaciju u smjeru kazaljke na satu i suprotno od kazaljke na satu. Odabir smjera rotacije temelji se na usporedbi ta dva kuta. Ako je kut u smjeru kazaljke na satu manji ili jednak kutu suprotno od kazaljke na satu, motor se okreće u smjeru kazaljke na satu za taj kut. U suprotnom, motor se okreće suprotno od kazaljke na satu. Nakon što su smjer i kut rotacije određeni taj kut se množi se prijenosnim omjerom zupčanika i šalje motoru zajedno sa smjerom rotacije kako bi se lanser pravilno pozicionirao pomoću mehanizma rotacije po azimutu.

Algoritam je napisan u Arduino Ide aplikaciji te će biti navedeni ključni dijelovi koda uz objašnjenje. Cijeli kod sa kratkim objašnjenjem pojedine linije koda će biti stavljen kao prilog radu, ovdje će se detaljno obraditi samo ključni dijelovi koda. Kod se može podijeliti u tri ključne cjeline: izračunavanje azimuta između dvije geografske točke, dohvaćanje trenutnog smjera iz kompasa te izračunavanje najkraćeg kuta za okretanje motora. Svaka cjelina ima specifičnu ulogu u rješavanju problema pozicioniranja lansera prema ciljnoj točki.

Funkcija `izracunajAzimut()` obavlja ključnu zadaću izračunavanja smjera između trenutne lokacije lansera i ciljane geografske točke. Azimut, koji predstavlja kut između sjevera i linije koja spaja trenutnu i ciljnu točku, potreban je za ispravno usmjerenje lansera. Ovaj kut se računa u rasponu od 0° do 360° , pri čemu je 0° sjever, 90° istok, 180° jug, a 270° zapad. Kako bi se izračunao azimut, geografske širine prvo se pretvaraju iz stupnjeva u radijane, jer matematičke funkcije poput sinus i kosinus u standardnoj biblioteci C++ rade s radijanima. Razlika u dužini između ciljane točke i trenutne lokacije također se pretvara u radijane, jer taj parametar značajno utječe na konačni izračun azimuta. Korištenjem Haversine formule, izračunavaju se dvije pomoćne vrijednosti, x i y , koje predstavljaju projekcije razlike u koordinatama na sfernoj površini Zemlje. Funkcija `atan2(y, x)` koristi te vrijednosti za izračunavanje kuta u radijanima, uzimajući u obzir znakove x i y kako bi se odredio ispravan kvadrant. Konačno, azimut se pretvara iz radijana u stupnjeve, a zatim normalizira u raspon od 0° do 360° , kako bi se osiguralo da rezultat bude uvijek unutar tog raspona, bez obzira na početne koordinate.

```
float izracunajAzimut(float sirina1, float duzina1, float sirina2, float
duzina2) {
    float radSirina1 = radians(sirina1); // Pretvaranje trenutne geografske
širine u radijane
    float radSirina2 = radians(sirina2); // Pretvaranje ciljane geografske
širine u radijane
    float deltaDuzina = radians(duzina2 - duzina1); // Razlika u dužini u
radijanima

    // Izračunavanje projekcija na sfernoj površini
    float y = sin(deltaDuzina) * cos(radSirina2);
    float x = cos(radSirina1) * sin(radSirina2) - sin(radSirina1) *
cos(radSirina2) * cos(deltaDuzina);
    float azimut = atan2(y, x); // Izračunavanje azimuta u radijanima

    azimut = degrees(azimut); // Pretvaranje azimuta u stupnjeve
    return fmod((azimut + 360), 360); // Normalizacija azimuta kako bi bio u
rasponu 0-360°
}
```

Funkcija `dohvatiSmjerIzCMPS14()` čita podatke iz CMPS14 kompasa koristeći I2C komunikaciju. Kompas vraća smjer u desetinkama stupnjeva, pa je potrebno kombinirati visoki

i niski bajt podataka da bi se dobio točan smjer. Konačni rezultat se vraća u stupnjevima, što omogućuje kasniju usporedbu s izračunatim azimutom.

```
// Funkcija za dobivanje trenutnog smjera iz CMPS14 kompasu
float dohvatiSmjerIzCMPS14() {
    // Započinjemo I2C komunikaciju s CMPS14 modulom
    Wire.beginTransmission(CMPS14_I2C_ADRESA);

    // Postavljamo zahtjev za čitanje podataka o smjeru iz registra 2 (2.
    bajt)
    Wire.write(2);

    // Završavamo I2C komunikaciju nakon postavljanja zahtjeva
    Wire.endTransmission();

    // Zahtijevamo 2 bajta podataka iz CMPS14 modula (smjer je 16-bitni broj)
    Wire.requestFrom(CMPS14_I2C_ADRESA, 2);

    // Čitamo visoki i niski bajt podataka iz modula
    byte visokiBajt = Wire.read(); // Prvo čitamo visoki bajt
    byte niskiBajt = Wire.read(); // Zatim čitamo niski bajt

    // Kombiniramo dva bajta u 16-bitni broj koji predstavlja smjer (u
    desetinkama stupnjeva)
    int smjer16 = ((visokiBajt << 8) + niskiBajt) / 10; // Konverzija u
    stupnjeve

    // Vraćamo smjer u stupnjevima
    return smjer16;
}
```

Funkcija `izracunajKutOkretanja()` izračunava najkraći kut za rotaciju motora prema ciljnom azimutu, uzimajući u obzir trenutni smjer lansera. Funkcija izračunava kut za rotaciju u smjeru kazaljke na satu (CW - clockwise) i u suprotnom smjeru (CCW - counterclockwise). Na temelju ovih izračuna, odabire se manji kut kako bi se optimizirala rotacija.

```
// Funkcija za izračunavanje najkraćeg kuta za okretanje motora
float izracunajKutOkretanja(float trenutniSmjer, float ciljaniAzimut) {
    // Izračunavanje kuta za okretanje u smjeru kazaljke na satu (CW -
    clockwise)
    float CW_kut = fmod((ciljaniAzimut - trenutniSmjer + 360), 360);
}
```

```

// Izračunavanje kuta za okretanje suprotno od kazaljke na satu (CCW -
counterclockwise)
float CCW_kut = fmod((trenutniSmjer - ciljaniAzimut + 360), 360);

// Provjera koji je kut manji i u kojem smjeru treba okretati motor
if (CW_kut <= CCW_kut) {
    Serial.println("Okretanje u smjeru kazaljke na satu");
    return CW_kut; // Vraćamo manji kut za CW rotaciju
} else {
    Serial.println("Okretanje suprotno od kazaljke na satu");
    return CCW_kut; // Vraćamo manji kut za CCW rotaciju
}
}

```

Ove tri funkcije predstavljaju temeljni algoritam za precizno pozicioniranje lansera prema ciljanoj točki. Njihova zajednička funkcionalnost omogućuje da sustav uspješno usmjeri lanser prema željenoj geografskoj točki.

4.5. Rotacija po elevaciji

Za rotaciju po elevaciji će se koristiti tablica koja nam određuje kut lansiranja za pojedine rakete. Kada se koriste realne rakete proizvođač uz informacije o raketama navodi kut pod kojim se lansiraju za idealne performanse. Pošto takve informacije nisu javno dostupne napraviti ćemo proizvoljnu tablicu u kojoj ćemo odrediti 4 kuta rotacije za rakete prema njihovom doseg. Prema ovoj tablici ćemo pomoću arduina slati informacije motoru za rotaciju po elevaciji kako pozicionirati lanser.

Tablica 5. Kut elevacije prema tipu rakete

Doseg rakete	Kut lansiranja
1-5 kilometara (Tip 1)	30°
5-10 kilometara (Tip 2)	35°
10-15 kilometara (Tip 3)	40°
15-20 kilometara (Tip 4)	45°

U kodu, ovo se postiže stvaranjem varijable koja označava trenutni tip rakete te nizom koji sadrži kutove lansiranja za svaki od tih tipova. Kod za ovu fazu je sljedeći:

```
// Definiranje tipa rakete i pripadajućih kutova lansiranja
int tipRakete = 1; // Odabrani tip rakete (1, 2, 3 ili 4)
float kutovi[4] = {30.0, 35.0, 40.0, 45.0}; // Kutovi lansiranja za tipove
1-4
```

Drugi korak u procesu je izračunavanje stvarnog kuta lansiranja na temelju odabranog tipa rakete. Ovo se postiže pristupanjem odgovarajućem elementu niza kutovi koristeći varijablu tipRakete kao indeks. Kod za ovu operaciju izgleda ovako:

```
// Određivanje kuta lansiranja na temelju tipa rakete
float kutLansiranja = kutovi[tipRakete - 1]; // Tip rakete je 1, 2, 3 ili 4
```

U ovom dijelu koda, tipRakete - 1 omogućava ispravno indeksiranje niza jer su indeksi niza 0-bazirani. Ova linija koda omogućava da se za svaki tip rakete dodijeli odgovarajući kut lansiranja. Prema odabranom tipu rakete se odgovarajući kut šalje dalje na motor za rotaciju. Upravljanje motora

4.6. Integracija senzora

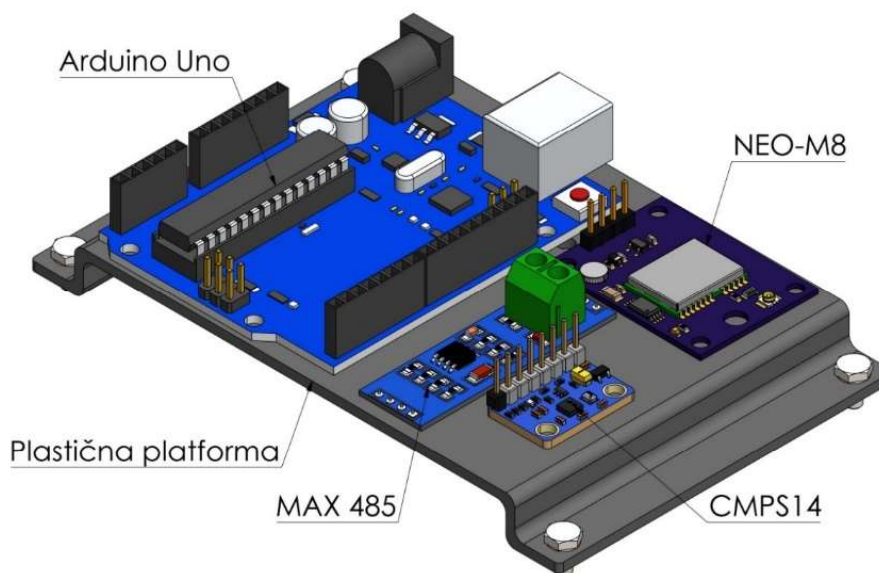
Kako bi s pomoću senzora mogli pravilo izračunati potrebni azimut, senzori se moraju nalaziti na samome lanseru. Kompas mora biti postavljen u smjeru raketa kako bi pravilno računao odstupanje od pravog senzora. GPS i kompas senzori za Arduino često su osjetljivi na prisutnost metala zbog nekoliko ključnih razloga. Prvo, kompas senzori, koji se temelje na magnetometrima, koriste magnetsko polje Zemlje kako bi odredili smjer. Međutim, metali, posebno feromagnetski materijali poput željeza i čelika, mogu značajno poremetiti ovo magnetsko polje. Kada se metalni predmeti nalaze u blizini kompasa, stvaraju lokalna magnetska polja koja iskrivljuju prirodno magnetsko polje Zemlje. Kao rezultat toga, senzor može dati netočne očitavanja, jer može biti pod utjecajem magnetskog polja metala umjesto stvarnog sjevernog pola Zemlje.

Drugo, GPS senzori koji se oslanjaju na signale iz satelita također su podložni smetnjama uzrokovanim metalima. GPS signali su relativno slabi kada dođu do Zemljine površine, a metalne prepreke, poput zgrada ili velikih metalnih objekata, mogu blokirati ili oslabiti ove signale. Nadalje, metali mogu uzrokovati refleksiju ili raspršenje GPS signala, što dovodi do

tzv. multipath pogrešaka. U takvim slučajevima, GPS prijemnik može primiti kako izravni signal tako i reflektirani, što uzrokuje zbunjenost i netočnosti u određivanju položaja.

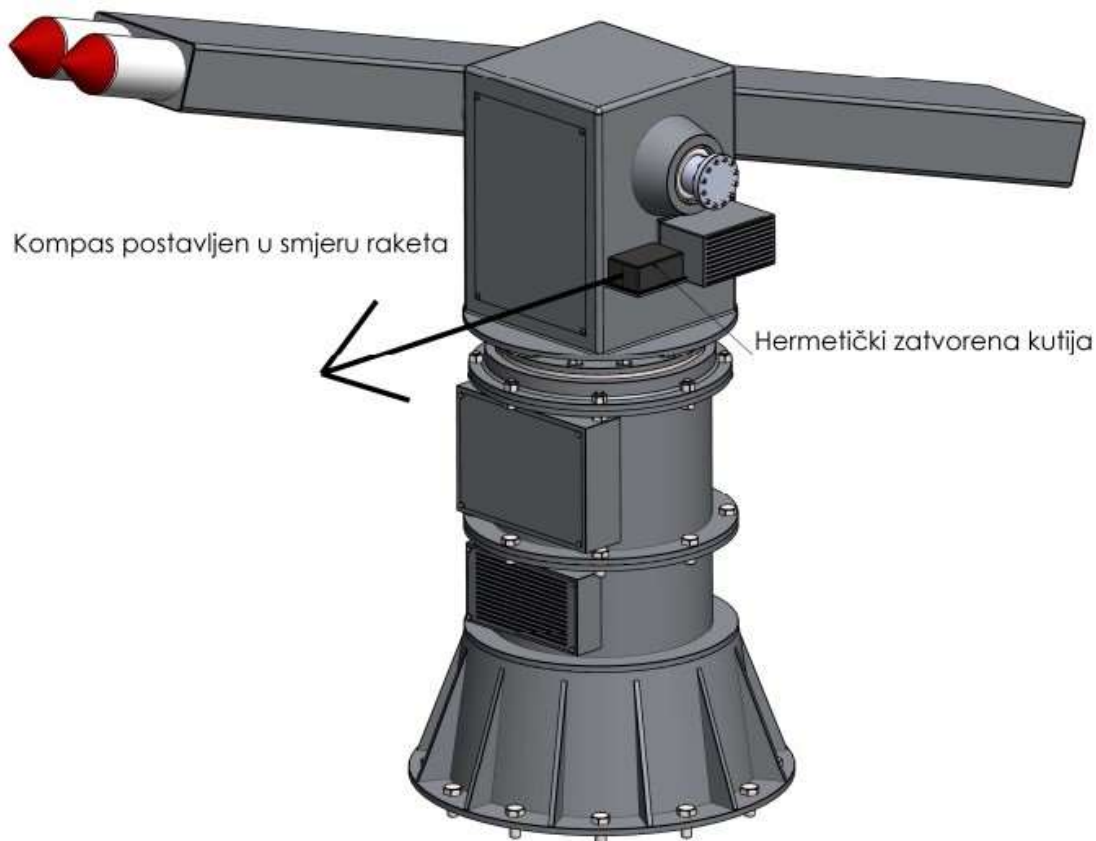
Zbog ovih potencijalnih smetnji, važno je GPS i kompas senzore zaštititi od vanjskih utjecaja. Jedan od najboljih načina za to je smjestiti ih u plastičnu, hermetički zatvorenu kutiju. Plastika je nevodljiva i nemetalna, što znači da ne uzrokuje magnetske smetnje i ne ometa prolaz elektromagnetskih signala. Hermetičko zatvaranje osigurava da unutar kutije ne može ući vlaga, prašina ili drugi vanjski faktori koji bi mogli oštetiti senzore ili ometati njihovu funkcionalnost. Na taj način, sensorima se osigurava stabilno i precizno radno okruženje, što je ključno za točne rezultate u projektima koji koriste Arduino.

Za senzore se izrađuje plastična platforma na koju se postavljaju senzori kako ne bi bili u kontaktu sa metalom.



Slika 30. Platforma za elektroničke komponente

Platforma se povezuje na lanser tako da kompas gleda u smjeru raketa te se hermetički zatvara sa plastičnom kutijom. Kutija se mora nalaziti na dijelu lansera koji se rotira po azimutu kako bi kompas pratio promjenu orijentacije samog lansera.



Slika 31. Senzori na lanseru

4.7. Alternativni izbor procesora

Korištenje SoftwareSerial na platformama poput Arduino Uno može dovesti do usporavanja rada, jer se serijska komunikacija emulira softverski, što opterećuje procesor. To postaje problematično u sustavima koji zahtijevaju obradu više serijskih uređaja istovremeno, poput GPS modula i RS-485 komunikacije. SoftwareSerial također onemogućava prekide tijekom prijenosa podataka, što može utjecati na preciznost upravljanja motorima.

Alternativno rješenje je korištenje Arduino Mega (ATmega2560), jer ima više hardverskih serijskih portova, što omogućuje bržu i pouzdaniju komunikaciju bez opterećivanja glavnog procesora. Tako je moguće koristiti više serijskih uređaja paralelno, bez potrebe za softverskom emulacijom.

Ispod je naveden kod koji koristi hardverske serijske portove Serial1 za GPS modul i Serial2 za RS-485 komunikaciju, čime eliminira potrebu za SoftwareSerial i povećava brzinu obrade. GPS podaci se kontinuirano čitaju i obrađuju unutar glavne petlje, pripremajući prostor za buduću implementaciju RS-485 komunikacije.

```
#include <TinyGPS++.h>

TinyGPSPPlus gps;

void setup() {
  Serial.begin(9600); // Serijski monitor
  Serial1.begin(9600); // GPS modul na Serial1
  Serial2.begin(9600); // RS-485 na Serial2
}

void loop() {
  // GPS obrada preko Serial1
  while (Serial1.available()) {
    gps.encode(Serial1.read());
  }

  // RS-485 obrada preko Serial2
}
```

Ovaj pristup poboljšava performanse, jer se koristi hardverska serijska komunikacija koja ne blokira rad procesora.

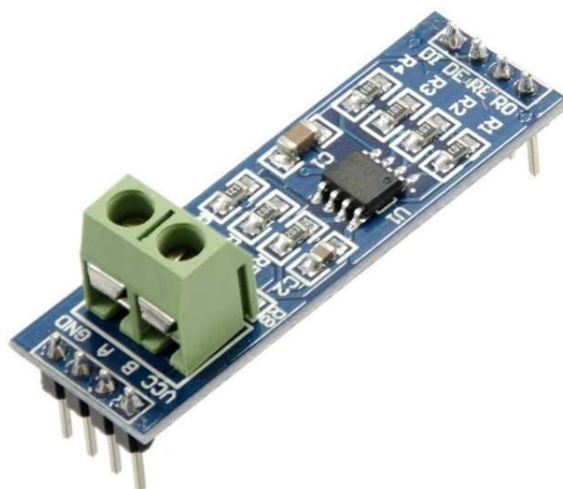
5. UPRAVLJANJE MOTORA

Motor PD6-EB nudi različite oblike komunikacije koji omogućuju fleksibilnost u upravljanju i integraciji u različite sustave. Među najpopularnijim oblicima komunikacije koje ovaj motor podržava su RS-232, RS-485 i CAN Bus. Svaki od ovih protokola ima svoje prednosti i specifične primjene, no za određene aplikacije, poput onih gdje se koristi Arduino kao kontroler, RS-485 predstavlja najbolji izbor.

RS-485 se ističe kao idealno rješenje za komunikaciju između Arduina i PD6-EB motora zbog nekoliko ključnih prednosti. Prvo, RS-485 omogućuje komunikaciju na većim udaljenostima (do 1200 metara) i podržava višepointne mreže, što znači da više uređaja može dijeliti istu komunikacijsku liniju. Ovo je posebno korisno u aplikacijama gdje se upravlja s više motora.

Druga velika prednost RS-485 komunikacije je njena otpornost na elektromagnetske smetnje. U uvjetima gdje postoji mnogo električnih uređaja ili u industrijskom okruženju, RS-485 pruža stabilnu i pouzdanu komunikaciju. To je ključno kada se koristi Arduino, koji nema naprednu zaštitu od smetnji kakvu nude neki drugi mikrokontroleri ili PLC-ovi.

RS-485 je lako integrirati s Arduinom koristeći pristupačne module poput MAX485, koji pretvaraju TTL signale Arduina u diferencijalne signale potrebne za RS-485 komunikaciju. Ovo omogućuje jednostavnu implementaciju i široku kompatibilnost s postojećim hardverom i softverom. Zbog ovih razloga, RS-485 predstavlja optimalan izbor za komunikaciju između Arduina i PD6-EB motora.



Slika 32. MAX485 modul [16]

5.1. Komunikacija sa motorom

Upravljanje motorima putem komunikacijskih protokola, kao što je RS-485, zahtijeva precizno formatiranje i pakiranje komandi kako bi se osiguralo da motor ispravno primi i izvrši željene upute

Prvi korak u slanju komandi motoru jest pravilno oblikovanje same komande. Svaka komanda počinje adresom uređaja kojem je namijenjena. Ova adresa omogućava razlikovanje više uređaja na istoj komunikacijskoj liniji, što je posebno važno u sustavima s više motora. Nakon adrese slijedi kod komande, koji definira specifičnu operaciju koju motor treba izvršiti. Na primjer, kod može označavati postavljanje brzine motora, promjenu smjera ili dohvaćanje statusa motora. Osim toga, neke komande mogu zahtijevati dodatne parametre, kao što su željena brzina ili smjer rotacije. Na kraju komande obično se dodaje kontrolna suma (checksum) ili ciklički redundantni kod (CRC), koji osigurava da su podaci ispravno preneseni i da nisu oštećeni tijekom prijenosa.

Nakon što je komanda formirana, ona se pakira u niz bajtova. Ovi bajtovi uključuju adresu, kod komande, parametre i checksum, organizirane u točno određenom redoslijedu. Ovaj niz bajtova spreman je za slanje putem RS-485 komunikacijskog kanala.

Tablica 6. Pakiranje komande u niz bajtova

Element	Opis	Veličina (bajtovi)	Primjer Vrijednosti
Adresa Uređaja	Identifikator motora na RS-485 mreži	1 bajt	0x01
Kod Komande	Identifikator operacije(npr. Postavi brzinu)	2 bajta	0x2100
Parametri	Vrijednost koje se šalju motoru(npr. brzina)	2 bajta	0x00C8 (200 u decimalnom)
Checksum	Kontrolna suma za provjeru ispravnosti podataka	1 bajt	0x3A

Ako spojimo sve gore navedene elemente, konačni niz bajtova koji se šalje motoru putem RS-485 izgleda ovako:

0x01 0x21 0x00 0x00 0xC8 0x3A

Ovaj niz bajtova predstavlja kompletan paket koji se šalje motoru kako bi on izvršio određenu komandu, u ovom slučaju, postavljanje brzine na vrijednost 200.

Komunikacija preko RS-485 protokola omogućava slanje ovako formatiranih komandi na velike udaljenosti, uz visoku otpornost na smetnje. To je posebno važno u industrijskim okruženjima gdje motori mogu biti smješteni na različitim lokacijama unutar velikih postrojenja. Uz pomoć RS-485 protokola, serijska komunikacija s motorom postaje pouzdana i učinkovita. Na primjer, koristeći Arduino, komande se mogu slati putem funkcija kao što su `Serial.write()`, koje omogućavaju slanje niza bajtova motoru.

Nakon što je komanda poslana, motor obično šalje odgovor natrag. Ovaj odgovor može sadržavati potvrdu da je komanda ispravno primljena i izvršena, ili podatke koje ste zatražili, poput trenutne brzine ili statusa motora. Odgovor se također prima u obliku niza bajtova, koji se dekodiraju kako bi se dobile korisne informacije. Provjera ispravnosti odgovora obavlja se pomoću checksuma ili CRC-a, što omogućava otkrivanje eventualnih grešaka u prijenosu.

Kako bi mogli slati podatke sa Arduina na kontroler potreban nam je MAX485 TTL u RS-485 modul pomoću kojeg ćemo pakirane kutove slati na motor. Kontroleri motora će pomoću izračunatih kutova i koda koji je isprogramiran u njima pravilno postaviti lanser na željeni azimut i rotaciju. Kodiranje samih motora nije dio ovog rada pošto bi to zahtjevalo edukaciju o samim motorima i same motore. Ovdje će biti naveden kod pomoću kojeg pakiramo i šaljemo kutove te smjer rotacije motora.

```
// Funkcija za slanje podataka na motore putem RS-485 s dodanim smjerom rotacije
void posaljiPodatkeMotorima(float azimut, float elevacija, bool smjerCW) {
    // Primjer formata pakiranja podataka: [adresni_bajt, naredba, podaci, checksum]
    byte adresaMotora = 0x01; // Adresa motora za azimut
    byte naredbaAzimut = 0x10; // Naredba za postavljanje kuta azimuta
    byte naredbaElevacija = 0x11; // Naredba za postavljanje kuta elevacije
```

```

// Pretvaranje azimuta i elevacije u bajtove (ovisno o protokolu motora)
byte azimutLowByte = lowByte((int)azimut);
byte azimutHighByte = highByte((int)azimut);
byte elevacijaLowByte = lowByte((int)elevacija);
byte elevacijaHighByte = highByte((int)elevacija);

// Dodavanje smjera rotacije: 0x00 za CW, 0x01 za CCW
byte smjerBajt = smjerCW ? 0x00 : 0x01;

// Slanje podataka za azimut, uključujući smjer rotacije
rs485Serial.write(adresaMotora);
rs485Serial.write(naredbaAzimut);
rs485Serial.write(azimutHighByte);
rs485Serial.write(azimutLowByte);
rs485Serial.write(smjerBajt); // Dodajemo smjer rotacije
rs485Serial.write(azimutHighByte ^ azimutLowByte ^ smjerBajt); //
Checksum

// Slanje podataka za elevaciju
rs485Serial.write(adresaMotora);
rs485Serial.write(naredbaElevacija);
rs485Serial.write(elevacijaHighByte);
rs485Serial.write(elevacijaLowByte);
rs485Serial.write(elevacijaHighByte ^ elevacijaLowByte); // Checksum
}

```

Prvo se definira adresa motora te naredbe za postavljanje azimuta i elevacije. Kutovi azimuta i elevacije pretvaraju se u 16-bitne vrijednosti, koje se zatim razdvajaju u dva bajta (visoki i niski). Ovi bajtovi se šalju motoru, zajedno s naredbom i kontrolnim zbrojem (checksum), koji osigurava ispravnost prenesenih podataka. Na taj način motor prima precizne podatke o smjeru i iznosu rotacije.

5.2. Komande i kod motora

Motor u sebi ima određene komande koje pomažu pri upravljanju motora te olakšavaju programiranje samog motora. Postojanje komadni znači da kod ne mora biti toliko kompleksan pošto postoje ugrađene komande za kompleksne funkcije motora. Jedna od bitnih komandi je postavljanje prijenosnog omjera samog motora i mjenjača, postavljanjem prijenosnog omjera povećava se i smanjuje brzina vrtnje motora i okretni moment. Također postoji komanda za postavljanje ciljane pozicije motora pomoću koje se lagano može upravljati okretom motora. Motor sadrži mnoge druge komande koje se mogu upotrebljavati pri programiranju.

Kako bi se lanser mogao pravilno postaviti za lansiranje potrebno je napisati kod koji pomoću izračunatih kutova i smjera rotacije postavlja pravilni azimut i elevaciju. Kod mora uzeti obzir prijenosne omjere pojedinih mehanizama rotacije. Kod za rotaciju po azimutu izračunava pomoću izračunatog kuta i određenog smjera rotacije ukupni potreban kut za koji se mora rotirati te izvršava tu naredbu. Motor za rotaciju po elevaciji ima ugrađeni apsolutni enkoder pomoću kojeg može očitavati trenutačnu poziciju tj. rotaciju motora, motor dobiva potreban kut na kojem se mora nalaziti te uz pomoć prijenosnog omjera izračunava apsolutni kut i rotira se. Na ovaj način se ostvaruje automatizirano postavljanje lanser u pravilnu poziciju za lansiranje balističkih raketa.

6. ZAKLJUČAK

U projektu razvoja servo sustava za lansiranje balističkih projektila uspješno su riješeni tehnički izazovi kroz nekoliko ključnih etapa, što je dovelo do konačnog proizvoda.

Prva etapa uključivala je razvoj mehanizama rotacije i konstrukcije sustava. Najprije su osmišljene mehaničke komponente za rotaciju po azimutu i elevaciji. Mehanizam za rotaciju po azimutu realiziran je pomoću sustava zupčanika koji omogućuje precizno pomicanje u horizontalnoj ravnini, dok je za elevaciju odabran lančani prijenos. Ovi mehanički sustavi projektirani su tako da osiguraju stabilnost i preciznost, uzimajući u obzir radne uvjete, dok je konstrukcija osigurala čvrstinu i izdržljivost sustava.

Nakon što je dovršena konstrukcija, u drugoj etapi se pristupa odabiru motora koji bi odgovarali zahtjevima za brzine, preciznosti, snagom i okretnog momenta. Korišteni su PD6-EB motori, zbog njihove sposobnosti da osiguraju dovoljan moment za preciznu kontrolu pozicije i brzine. Motor za azimut bio je povezan s mehanizmom zupčanika, dok je motor za elevaciju integriran s lančanim prijenosom, što je omogućilo finu kontrolu oba pokreta.

U trećoj etapi odabiru se senzora za mjerenje orijentacije i položaja lansera. Odabrani su CMPS14 senzor za orijentaciju i NEO-M8 GPS modul za precizno praćenje lokacije lansera. Ovi senzori su odabrani zbog njihove visoke točnosti i jednostavne integracije s Arduino platformom, a njihova povezanost s motorima preko RS485 komunikacije omogućila je brzu razmjenu podataka.

Konačna etapa projekta bila je razvoj algoritma za izračunavanje azimuta i kontrolu motora. Razvijeni algoritam se temelji na poznavanju koordinata cilja i trenutne pozicije lansera te se izračunava potreban azimut i elevacija. Algoritam je prilagođen različitim tipovima projektila, a motori su kontrolirani putem Arduino platforme, čime je postignuto precizno automatsko pozicioniranje lansera.

Rad uspješno demonstrira kako kombinacija mehaničkih sustava, senzora i algoritama može stvoriti pouzdan i učinkovit servo sustav za lansiranje balističkih projektila. Korištenje modernih tehnoloških rješenja, poput Arduino platforme i komunikacije putem RS485, omogućilo je visoku razinu automatizacije, preciznosti i učinkovitosti.

LITERATURA

- [1] <https://www.awm.gov.au/articles/blog/v2rocket>
- [2] https://www.researchgate.net/figure/Ballistic-missile-trajectory-with-three-phases_fig1_338192363
- [3] <https://www.istockphoto.com/photo/several-combat-missiles-isolated-on-a-white-background-missile-weapons-gm920255062-252906193>
- [4] <https://www.ldb-bearing.com/ldb-bearing-knowledge/slewing-bearing-knowledge/slewing-ring-gear-module.html>
- [5] <https://www.skf.com/us/products/slewing-bearings>
- [6] <https://www.gunt.de/en/products/bearing-friction/040.23200/tm232/glct-1:pa-148:pr-1391>
- [7] <https://www.renesas.com/us/en/support/engineer-school/brushless-dc-motor-01-overview>
- [8] <https://analogcircuitdesign.com/esc-brushless-dc-motors/>
- [9] <https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/>
- [10] <https://www.nanotec.com/eu/en/products/10834-pd6-eb80cd-m-65-1>
- [11] Krešimir Vučković; Podloge za predavanja iz kolegija “Elementi konstrukcija II”, Fakultet strojarstva i brodogradnje, Zagreb
- [12] <https://katalog.comterra.hr/product/honda-eu22i-18-kw-benzin-inverter-agregat-za-struju/>
- [13] https://en.wikipedia.org/wiki/Arduino_Uno
- [14] <https://www.u-blox.com/en/product/neo-m8l-series>
- [15] <https://www.active-robots.com/cms/14-tilt-compensated-compass.html>
- [16] <https://probots.co.in/rs485-to-serial-uart-ttl-max485-converter-module-board.html>

PRILOZI**I. Kod za Arduino**

```

#include <Wire.h> // Uključivanje knjižnice za I2C
komunikaciju
#include <TinyGPS++.h> // Uključivanje knjižnice za GPS module
#include <SoftwareSerial.h> // Uključivanje knjižnice za serijsku
komunikaciju

TinyGPSPlus gps; // Kreiranje GPS objekta iz TinyGPS++
knjižnice
SoftwareSerial ss(4, 3); // Definiranje serijskih pinova za GPS
modul (RX pin 4, TX pin 3)

// Definiranje I2C adrese za CMPS14 kompas
const int CMPS14_I2C_ADRESA = 0x60;

// Definiranje serijskih pinova za RS-485 komunikaciju
const int RS485_TX = 8; // Pin za TX (transmit) na RS-485
const int RS485_RX = 7; // Pin za RX (receive) na RS-485

// Definiranje serijskog porta za RS-485 komunikaciju
SoftwareSerial rs485Serial(RS485_RX, RS485_TX);

// Varijable za pohranu trenutnih GPS koordinata
float trenutnaSirina, trenutnaDuzina;

// Ciljne GPS koordinate (npr. Zagreb)
float ciljanaSirina = 45.8150;
float ciljanaDuzina = 15.9819;

// Definiranje tipa rakete i pripadajućih kutova lansiranja
int tipRakete = 1; // Odabrani tip rakete (1, 2, 3 ili 4)
float kutovi[4] = {30.0, 35.0, 40.0, 45.0}; // Kutovi lansiranja za tipove
1-4

void setup() {
    Serial.begin(9600); // Pokretanje serijske komunikacije za
ispis podataka u monitoru
    ss.begin(9600); // Inicijalizacija serijske komunikacije
s GPS modulom
    rs485Serial.begin(9600); // Inicijalizacija RS-485 komunikacije

    Wire.begin(); // Inicijalizacija I2C komunikacije za
CMPS14 kompas
}

void loop() {

```

```
// Provjera dostupnosti GPS podataka
while (ss.available() > 0) {
    gps.encode(ss.read());

    if (gps.location.isUpdated()) {
        trenutnaSirina = gps.location.lat();
        trenutnaDuzina = gps.location.lng();
    }
}

float trenutniSmjer = dohvatiSmjerIzCMPS14();
float ciljaniAzimut = izracunajAzimut(trenutnaSirina, trenutnaDuzina,
ciljanaSirina, ciljanaDuzina);
float kutZaOkretanje = izracunajKutOkretanja(trenutniSmjer,
ciljaniAzimut);

// Odabir kuta lansiranja na temelju tipa rakete
float kutLansiranja = kutovi[tipRakete - 1];

// Provjera smjera rotacije (true za CW, false za CCW)
bool smjerCW = (izracunajKutOkretanja(trenutniSmjer, ciljaniAzimut) ==
kutZaOkretanje);

// Slanje podataka na motor (azimut, elevacija i smjer rotacije)
posaljiPodatkeMotorima(kutZaOkretanje, kutLansiranja, smjerCW);

// Ispis podataka u serijski monitor
Serial.print("Trenutni smjer: ");
Serial.println(trenutniSmjer);
Serial.print("Ciljani azimut: ");
Serial.println(ciljaniAzimut);
Serial.print("Kut za okretanje: ");
Serial.println(kutZaOkretanje);
Serial.print("Kut lansiranja: ");
Serial.println(kutLansiranja);
Serial.print("Smjer rotacije: ");
Serial.println(smjerCW ? "CW" : "CCW");

delay(1000);
}

float dohvatiSmjerIzCMPS14() {
    Wire.beginTransmission(CMPS14_I2C_ADRESA);
    Wire.write(2);
    Wire.endTransmission();
    Wire.requestFrom(CMPS14_I2C_ADRESA, 2);

    byte visokiBajt = Wire.read();
```



```

byte niskiBajt = Wire.read();
int smjer16 = ((visokiBajt << 8) + niskiBajt) / 10;

return smjer16;
}

float izracunajAzimut(float sirina1, float duzina1, float sirina2, float
duzina2) {
    float radSirina1 = radians(sirina1);
    float radSirina2 = radians(sirina2);
    float deltaDuzina = radians(duzina2 - duzina1);

    float y = sin(deltaDuzina) * cos(radSirina2);
    float x = cos(radSirina1) * sin(radSirina2) - sin(radSirina1) *
cos(radSirina2) * cos(deltaDuzina);
    float azimut = atan2(y, x);

    azimut = degrees(azimut);
    return fmod((azimut + 360), 360);
}

float izracunajKutOkretanja(float trenutniSmjer, float ciljaniAzimut) {
    float CW_kut = fmod((ciljaniAzimut - trenutniSmjer + 360), 360);
    float CCW_kut = fmod((trenutniSmjer - ciljaniAzimut + 360), 360);

    if (CW_kut <= CCW_kut) {
        Serial.println("Okretanje u smjeru kazaljke na satu");
        return CW_kut;
    } else {
        Serial.println("Okretanje suprotno od kazaljke na satu");
        return CCW_kut;
    }
}

// Funkcija za slanje podataka na motore putem RS-485 s dodanim smjerom
rotacije
void posaljiPodatkeMotorima(float azimut, float elevacija, bool smjerCW) {
    // Primjer formata pakiranja podataka: [adresni_bajt, naredba, podaci,
checksum]
    byte adresaMotora = 0x01; // Adresa motora za azimut
    byte naredbaAzimut = 0x10; // Naredba za postavljanje kuta azimuta
    byte naredbaElevacija = 0x11; // Naredba za postavljanje kuta elevacije

    // Pretvaranje azimuta i elevacije u bajtove (ovisno o protokolu motora)
    byte azimutLowByte = lowByte((int)azimut);
    byte azimutHighByte = highByte((int)azimut);
    byte elevacijaLowByte = lowByte((int)elevacija);
    byte elevacijaHighByte = highByte((int)elevacija);
}

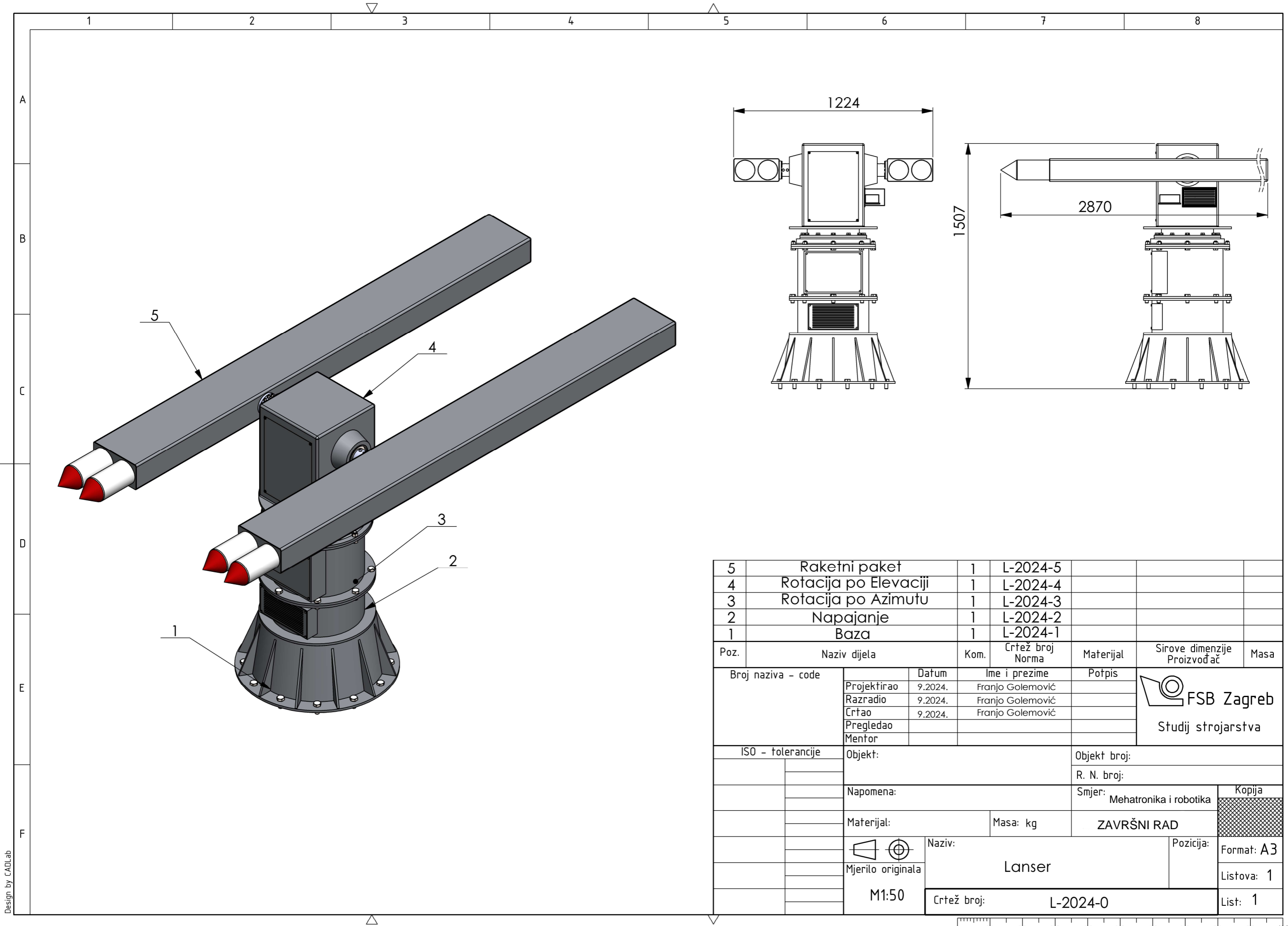
```


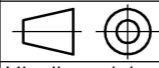
```
// Dodavanje smjera rotacije: 0x00 za CW, 0x01 za CCW
byte smjerBajt = smjerCW ? 0x00 : 0x01;

// Slanje podataka za azimut, uključujući smjer rotacije
rs485Serial.write(adresaMotora);
rs485Serial.write(naredbaAzimut);
rs485Serial.write(azimutHighByte);
rs485Serial.write(azimutLowByte);
rs485Serial.write(smjerBajt); // Dodajemo smjer rotacije
rs485Serial.write(azimutHighByte ^ azimutLowByte ^ smjerBajt); //
Checksum

// Slanje podataka za elevaciju
rs485Serial.write(adresaMotora);
rs485Serial.write(naredbaElevacija);
rs485Serial.write(elevacijaHighByte);
rs485Serial.write(elevacijaLowByte);
rs485Serial.write(elevacijaHighByte ^ elevacijaLowByte); // Checksum
}
```

II. Sklopni crtež lansera



5	Raketni paket	1	L-2024-5			
4	Rotacija po Elevaciji	1	L-2024-4			
3	Rotacija po Azimutu	1	L-2024-3			
2	Napajanje	1	L-2024-2			
1	Baza	1	L-2024-1			
Poz.	Naziv dijela	Kom.	Crtež broj Norma	Materijal	Sirove dimenzije Proizvođač	Masa
Broj naziva - code		Datum	Ime i prezime		Potpis	
Projektirao		9.2024.	Franjo Golemović		 FSB Zagreb Studij strojarstva	
Razradio		9.2024.	Franjo Golemović			
Crtao		9.2024.	Franjo Golemović			
Pregledao						
Mentor						
ISO - tolerancije		Objekt:			Objekt broj:	
					R. N. broj:	
		Napomena:			Smjer: Mehatronika i robotika	
		Materijal:		Masa: kg	ZAVRŠNI RAD	
		 Mjerilo originala		Naziv:		Pozicija:
		M1:50		Lanser		Format: A3
		Crtež broj:		L-2024-0		Listova: 1
						List: 1

