

Analiza performansi algoritma za izbjegavanja sudara na moru temeljenog na COLREG-u

Stojanović, Petar

Master's thesis / Diplomski rad

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:490080>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 768

**ANALIZA PERFORMANSI ALGORITMA ZA IZBJEGAVANJA
SUDARA NA MORU TEMELJENOG NA COLREG-U**

Petar Stojanović

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 768

**ANALIZA PERFORMANSI ALGORITMA ZA IZBJEGAVANJA
SUDARA NA MORU TEMELJENOG NA COLREG-U**

Petar Stojanović

Zagreb, veljača 2025.

DIPLOMSKI ZADATAK br. 768

Pristupnik: **Petar Stojanović (0036511477)**

Studij: Računarstvo

Profil: Računalno inženjerstvo

Mentor: prof. dr. sc. Nikola Mišković

Zadatak: **Analiza performansi algoritma za izbjegavanja sudara na moru temeljenog na COLREG-u**

Opis zadatka:

Izbjegavanje prepreka na moru predstavlja bitan preduvjet za sigurnost autonomnog broda. Potrebno je proučiti COLREG pravila za izbjegavanje sudara na moru te algoritme koji ih primjenjuju. Postojeću implementaciju algoritma izbjegavanja prepreka sudara treba proširiti sa simulacijskim okruženjem u kojem bi se sustavno testirale performanse algoritma u slučajno generiranim scenarijima.

Rok za predaju rada: 14. veljače 2025.

Zahvaljujem dragom Bogu, roditeljima, bratu i djevojci na bezuvjetnoj podršci, ljubavi i razumijevanju. Posebno hvala Matku Batošu na spremnosti da pomogne čak i u kasnim satima vikendom te cijelom LABUST-u na golemom strpljenju i toleriranju mojih kašnjenja i neurednosti. Tijekom protekle dvije godine, uz njihovu pomoć i primjer, naučio sam vrijednost mukotrpnog rada i težnje za najboljim rezultatima. Uz njih sam naučio što je istraživački rad i vrijednosti odnosa prema poslu, koje ću pamtiti i primjenjivati cijeli život. Moj uspjeh i kreiranje osobe koja jesam, ne bi bio moguć bez Vas. Hvala.

Sadržaj

Uvod	1
1. Metoda prepreka brzine za izbjegavanja sudara na temelju COLREG konvencije	3
1.1. Opis metode.....	3
1.1.1. Matematička formulacija.....	12
1.1.2. Geometrijska interpretacija.....	13
1.1.3. Implementacija	13
1.1.4. Arhitektura sustava	15
1.2. Simulacijski eksperiment.....	17
1.2.1. Sadržaj eksperimenta.....	17
1.2.2. Prikupljanje i obrada podataka	18
1.2.3. Analiza podataka	19
1.3. Prikaz testiranih scenarija.....	20
2. Metoda izbjegavanja sudara na temelju dubokog podržanog učenja	33
2.1. Opis metode.....	34
2.1.1. Navigacija i upravljanje pomorskim plovilima	34
2.1.2. Podržano učenje (Reinforcement Learning – RL).....	36
2.2. Implementacija	42
2.2.1. Alati i biblioteke	43
2.2.2. Okruženje.....	43
2.2.3. Agent	43
2.2.4. Nagradna funkcija.....	44
2.3. Simulacijski eksperiment.....	46
2.3.1. Trening i evaluacija	46
3. Rezultati i diskusija	48
3.1. Rezultati VO metode	49

3.2. Diskusija VO metode	70
3.3. Rezultati DRL metode	79
3.4. Diskusija DRL metode	80
Zaključak	85
Literatura	87
Sažetak.....	89
Summary.....	90
Skraćenice.....	91
Tablica slika.....	92

Uvod

U posljednjim desetljećima značajan je porast interesa za razvoj autonomnih sustava u pomorskoj navigaciji, potaknut sve većom gustoćom prometa te posljedično potrebom za povećanom sigurnošću plovidbe. S obzirom na to da je sigurnost u pomorskom prometu primarno regulirana međunarodnim COLREG pravilima (Convention on the International Regulations for Preventing Collisions at Sea, 1972), implementacija ovih pravila u autonomne sustave predstavlja ključni izazov i prioritet za istraživače i inženjere. Tradicionalni pristupi, koji se oslanjaju na fiksne matematičke modele i algoritamske procedure, dokazali su svoju učinkovitost u standardnim uvjetima.

Jedno od rješenja koje implementira COLREG pravila u autonomnoj navigaciji je metoda prepreke brzine (Velocity Obstacle, VO). Koristi napredne matematičke alate, uključujući primjenu Minkowski sume, kako bi se odredio siguran prostor za kretanje plovila, te se temelji na modularnoj arhitekturi koja omogućava fleksibilnu obradu podataka prikupljenih putem senzora.

Suvremeni trendovi u području autonomne navigacije sve više ukazuju na potrebu za adaptivnim i inteligentnim sustavima koji mogu učiti i prilagođavati se promjenjivim uvjetima u stvarnom vremenu. U tu svrhu razvijena je metoda prepreke brzine, čija se implementacija temelji na metodama dubokog podržanog učenja. Metoda prepreke brzine omogućava kontinuirani trening agenta kroz interakciju sa simulacijskim okruženjem, čime se postiže visoka prilagodljivost i efikasnost u procesu izbjegavanja sudara.

Hibridni pristupi koji integriraju tradicionalne matematičke modele s metodama strojnog učenja mogu rezultirati značajnim poboljšanjima u performansama izbjegavanja sudara. Konkretno, cilj je postići poboljšanja u točnosti predviđanja kolizijskih situacija te u brzini prilagodbe promjenjivim uvjetima, čime se smanjuje broj kritičnih situacija za značajan postotak. Nadalje, tradicionalni pristupi pate od problema uslijed nepreciznosti senzorskih podataka, što se reflektira u povećanom broju lažno pozitivnih indikacija potencijalnih sudara.

Stoga, glavni cilj ovog rada je pružiti sveobuhvatnu i kritičku analizu dvaju pristupa za implementaciju sustava izbjegavanja sudara autonomnih plovila.

Opseg istraživanja

S jedne strane, razmatra se tradicionalni pristup implementiran u prepreke brzine metodi (Velocity Obstacle, VO), koja se oslanja na fiksne matematičke modele usklađene s COLREG pravilima. S druge strane, istražuje se suvremeni pristup realiziran kroz duboko podržano učenje, koji primjenjuje metode kontinuirane adaptacije i optimizacije putanje plovila. Ovi pristupi uspoređuju se na temelju rezultata simulacija koje obuhvaćaju tipične scenarije pomorske navigacije, poput head-on susreta, crossing situacija i pretjecanje. Ovaj rad s toga ne samo da pruža detaljnu analizu navedenih metoda već i identificira izazove te nudi smjernice za razvoj hibridnih sustava koji mogu poboljšati sigurnost i učinkovitost autonomnih plovila u stvarnim uvjetima. Integracijom tradicionalnih algoritama s naprednim metodama dubokog podržanog učenja, potencijalno je moguće razviti sustave koji će značajno unaprijediti pomorsku sigurnost i postaviti nove standarde u autonomnoj navigaciji.

Okvir rada

Rad je strukturiran tako da kroz poglavlja postupno uvodi u problematiku, objašnjava metode i analizira rezultate. Nakon uvoda, prikazuju se osnovna pravila i teorijske osnove metoda izbjegavanja sudara, s posebnim fokusom na metodu prepreke brzine. Nadalje, opisana je metoda za izbjegavanje sudara autonomnih površinskih plovila temeljena na dubokom podržanom učenju. U daljnjim poglavljima detaljno se opisuju eksperimenti i simulacije provedeni za oba pristupa, dok se zaključna poglavlja bave analizom rezultata, identifikacijom izazova i nedostataka te prijedlozima za budući rad.

1. Metoda prepreka brzine za izbjegavanja sudara na temelju COLREG konvencije

1.1. Opis metode

Uvođenje autonomnih površinskih vozila (ASV) u pomorsku navigaciju zahtijeva učinkovite metode za izbjegavanje sudara, posebno u skladu s Međunarodnim pravilima za sprječavanje sudara na moru (COLREGs). Metoda za rješavanje problema izbjegavanja sudara, opisana u ovom radu, naziva se metoda prepreke brzine (Velocity Obstacle - VO). Metoda omogućuje autonomnim površinskim plovilima da predviđaju potencijalne sudare i poduzmu odgovarajuće manevarske radnje kako bi ih izbjegli.

Metoda VO koristi matematičke i geometrijske pristupe za modeliranje prepreka kao konusa u prostoru brzina. Svaki konus predstavlja skup brzina koje dovode do sudara između vlastitog plovila (Own Ship - OS) i ciljnog plovila (Target Ship - TS). Izbor brzine izvan tih konusa osigurava sigurnu i kolizijski slobodnu putanju. Uz osnovni princip VO metode, dodatna ograničenja se implementiraju kako bi plovila poštivala pravila COLREG i osigurala predvidljive i sigurne manevre u stvarnim pomorskim uvjetima.

Svrha metode

Glavna svrha metode je omogućiti autonomnim plovilima da, koristeći unaprijed definirane matematičke modele, predviđaju potencijalne kolizije i generiraju sigurne putanje kretanja. To se postiže analizom ulaznih podataka (kao što su pozicija, brzina, orijentacija) te dinamičkim prilagođivanjem kretanja u realnom vremenu. Metoda je osmišljena kako bi se zadovoljili zahtjevi u pogledu sigurnosti i pouzdanosti, osobito u scenarijima gdje je brzina donošenja odluka presudna.

Osnovni pojmovi

Prije detaljnog opisa algoritma VO, važno je definirati ključne pojmove koji će se koristiti:

- **COLREG:** Međunarodna pravila za sprječavanje sudara na moru.

- **Glavno plovilo (Own Ship - OS):** Autonomno površinsko vozilo koje koristi metoda za izbjegavanje sudara.
- **Ciljno plovilo (Target Ship - TS):** Prepreka koju OS mora izbjeći, poput drugih plovila ili statičkih objekata.
- **Brzina (Velocity):** Vektor koji opisuje brzinu i smjer kretanja plovila.
- **Željena brzina (Desired Velocity):** Brzina koju OS želi postići kako bi stigao do sljedeće točke na planiranoj ruti.
- **Kurs (Course):** Planirani smjer kretanja plovila.
- **Kolizijski konus (Collision Cone - CC):** Skup relativnih brzina između OS i TS koje vode do sudara u nekom trenutku.
- **Prepreka brzine (Velocity Obstacle - VO):** Skup apsolutnih brzina OS koje će rezultirati sudarom s TS u nekom trenutku.

Primjena COLREG pravila

Ova pravila definiraju „pravila prednosti“ za plovila kako bi se spriječili sudari. Posebno su važna pravila iz dijela B – Pravila kormilarenja i plovidbe koja se koriste u ovom algoritmu:

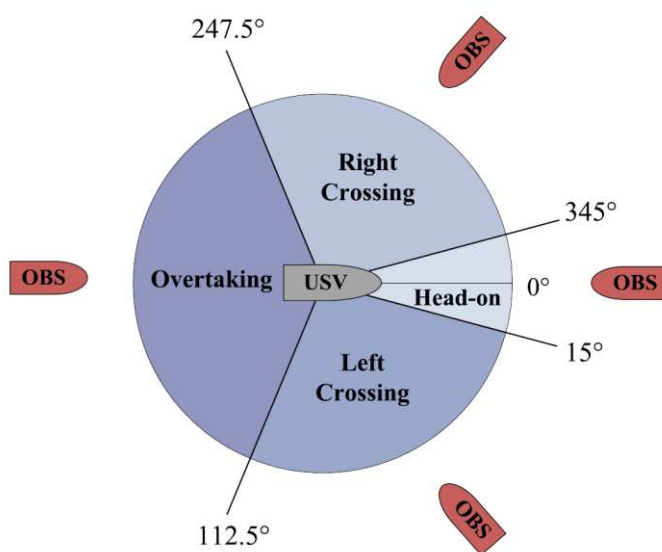
- **Pravilo 8 (Radnja za izbjegavanje sudara):** Radnje trebaju biti pravovremene i jasno vidljive drugim plovilima. Promjena kursa je preferirani način izbjegavanja sudara, pri čemu manevar mora rezultirati prolaskom na sigurnoj udaljenosti.
- **Pravilo 13 (Pretjecanje):** Ako drugo plovilo dolazi iz smjera više od $22,5^\circ$ iza krme OS-a, OS je pretjecano plovilo, a TS je plovilo koje pretječe.
- **Pravilo 14 (Head-on situacija):** Plovila koja se susreću na recipročnim ili gotovo recipročnim kursovima moraju oba skrenuti udesno.
- **Pravilo 15 (Crossing situacija):** Plovilo koje ima drugo plovilo s desne strane mora se maknuti i smatra se plovilom koje se izmiče.
- **Pravilo 16 (Radnja plovila koje se izmiče):** Plovilo koje treba izbjeći sudar mora poduzeti pravovremene radnje kako bi se izbjegao sudar.

- **Pravilo 17 (Radnja plovila koje ima pravo puta):** Plovilo koje ima pravo puta treba zadržati kurs i brzinu, osim ako postane očito da drugo plovilo ne poduzima nikakve radnje za izbjegavanje sudara.

COLREG pravila u praksi su često neprecizna i teška za implementaciju u autonomne sustave. Kako bi se pravila kvantificirala, u VO metodi definirano je nekoliko područja za donošenje odluka na temelju kursa plovila:

- Za head-on susrete definirano je područje $\pm 15^\circ$ od pramca OS-a
- Promjena kursa smatra se „znatnom“ ako iznosi barem 30° od početnog kursa
- Sigurna udaljenost prolaska definirana je kao višekratnik veličine OS-a, ovisan o brzini i veličini plovila

Kombiniranjem ovih pravila stvoreni su sektori koji olakšavaju izbor odgovarajućih pravila u različitim situacijama. Prikaz sektora je vidljiv na slici ispod.



Slika 1 – Prikaz sektora

Sektori su definirani kako bi razlikovali head-on situacije, crossing situacije i ostale manevre na temelju relativnog položaja TS-a u odnosu na OS. Ova klasifikacija omogućuje algoritmu VO da odabere odgovarajuće manevarske radnje usklađene s COLREG pravilima, povećavajući sigurnost i predvidljivost navigacije.

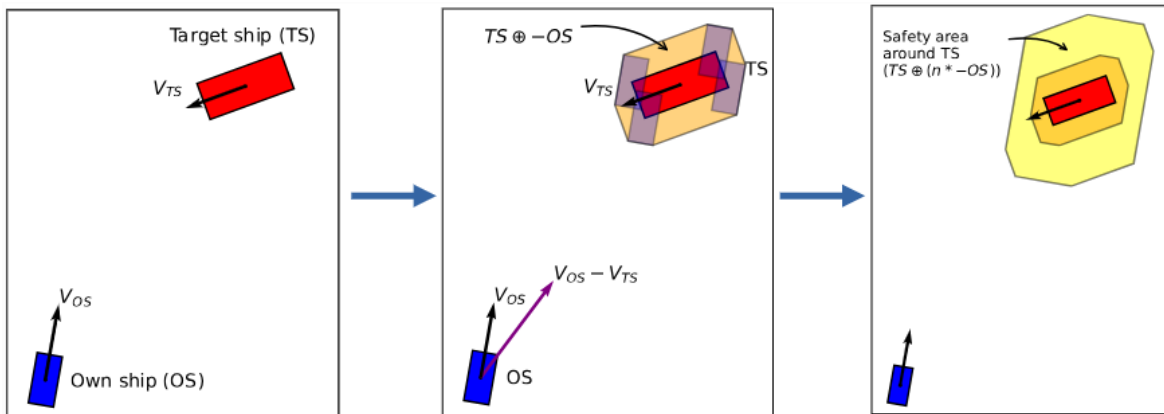
Kriteriji za izračun izbjegavanja sudara

Nakon što je VO definiran, potrebno je identificirati sigurnu brzinu za OS. Ovaj proces uključuje:

- Identifikaciju kolizijskih brzina
 - Brzine OS-a unutar VO konusa smatraju se kolizijskim brzinama.
 - Brzine izvan VO konusa smatraju se sigurnima.
- Odabir optimalne brzine
 - OS odabire brzinu izvan VO koja je najbliža željenoj brzini (v_{des}).
 - Cilj je minimizirati razliku između željene i odabrane brzine, dok se osigurava izbjegavanje sudara.
- Usklađenost s COLREG pravilima
 - Ako postoji više sigurnih brzina, OS odabire onu koja je u skladu s odgovarajućim COLREG pravilom (npr. Skretanje udesno za head-on susret prema pravilu 14).

Osnovni princip metode VO

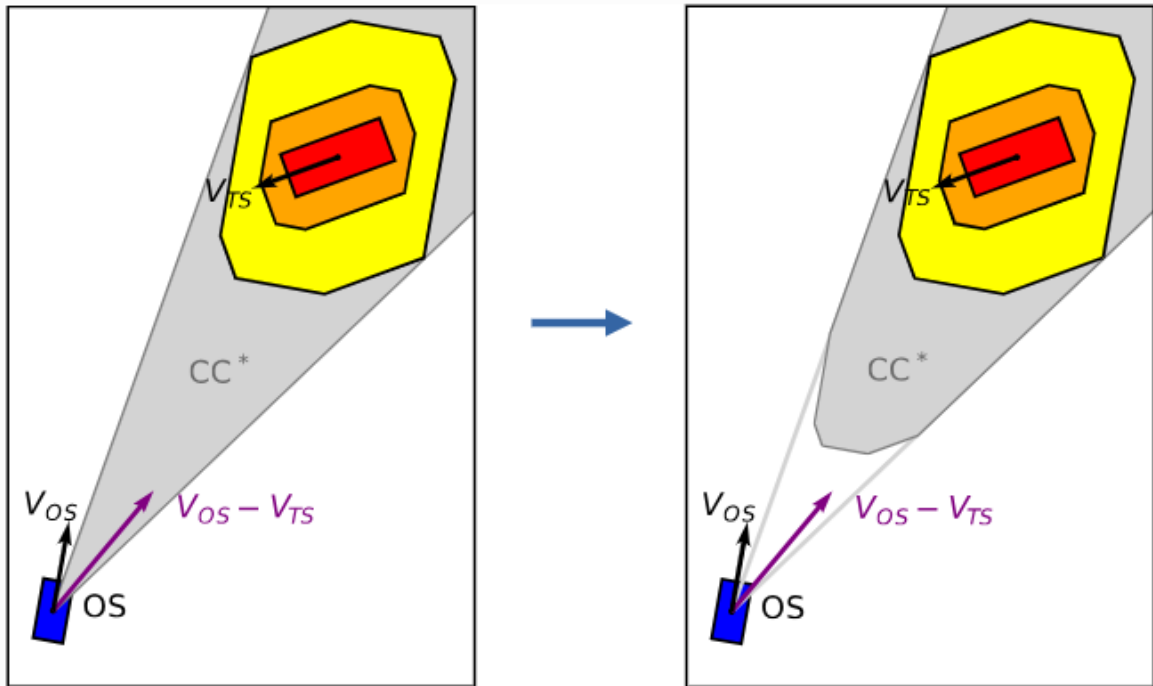
Algoritam VO započinje proširenjem veličine TS-a uključivanjem dimenzija OS-a. Ovo proširenje omogućuje tretiranje OS-a kao točke tijekom proračuna, što pojednostavljuje modeliranje. Proširenje se izvodi primjenom Minkowskijeve sume, matematičke operacije koja kombinira sve točke jednog oblika s oblikom drugog. Uz oblik OS-a, proširenje uključuje i sigurnosnu zonu definiranu kao višekratnik dimenzija OS-a, što osigurava prolaz na sigurnoj udaljenosti, u skladu s COLREG pravilima.



Slika 2 – Proširenje TS-a

Nakon proširenja TS-a, algoritam izračunava kolizijski konus (Collision Cone - CC). Kolizijski konus predstavlja područje relativnih brzina OS-a koje rezultiraju sudarom s TS-om u bilo kojem trenutku u budućnosti. Ovo područje formiraju tangencijalni pravci povučeni od centra OS-a prema proširenom obliku TS-a. Međutim, kako bi se spriječilo nepotrebno izbjegavanje sudara u dalekoj budućnosti, algoritam prilagođava CC isključivanjem brzina čiji je vrijeme do sudara (Time to Collision - TTC) veće od unaprijed definiranog praga. Ova prilagodba smanjuje rizik od zauzimanja cijelog prostora brzina CC-om u situacijama s mnogo prepreka.

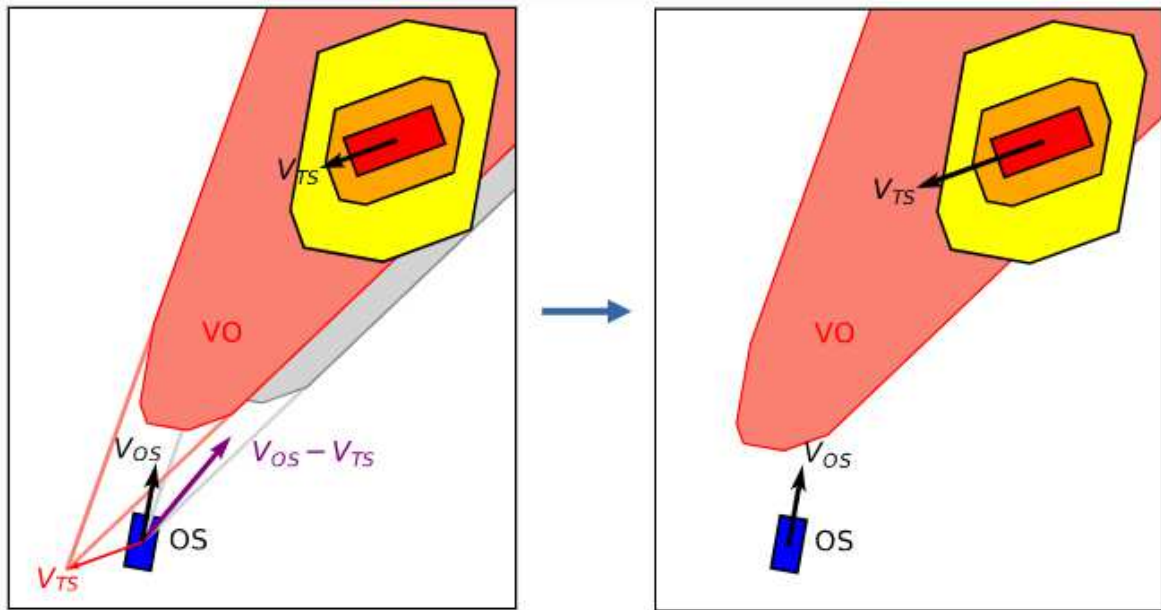
$$CC = \{v_r \mid v_r t \cap T \neq \emptyset, \forall t \geq TTC\}$$



Slika 3 – Kolizijski konus

Sljedeći korak uključuje transformaciju CC-a u prepreku brzine (Velocity Obstacle - VO). VO predstavlja skup apsolutnih brzina OS-a koje rezultiraju sudarom s TS-om, uzimajući u obzir brzinu TS-a. Ovo se postiže translacijom CC-a za vektor brzine TS-a. Ova transformacija omogućuje algoritmu da uzme u obzir stvarne apsolutne brzine oba plovila u procesu donošenja odluka.

$$VO = CC \oplus V_{TS}$$



Slika 4 – VO vizualizacija

Uključivanje nesigurnosti i COLREG pravila

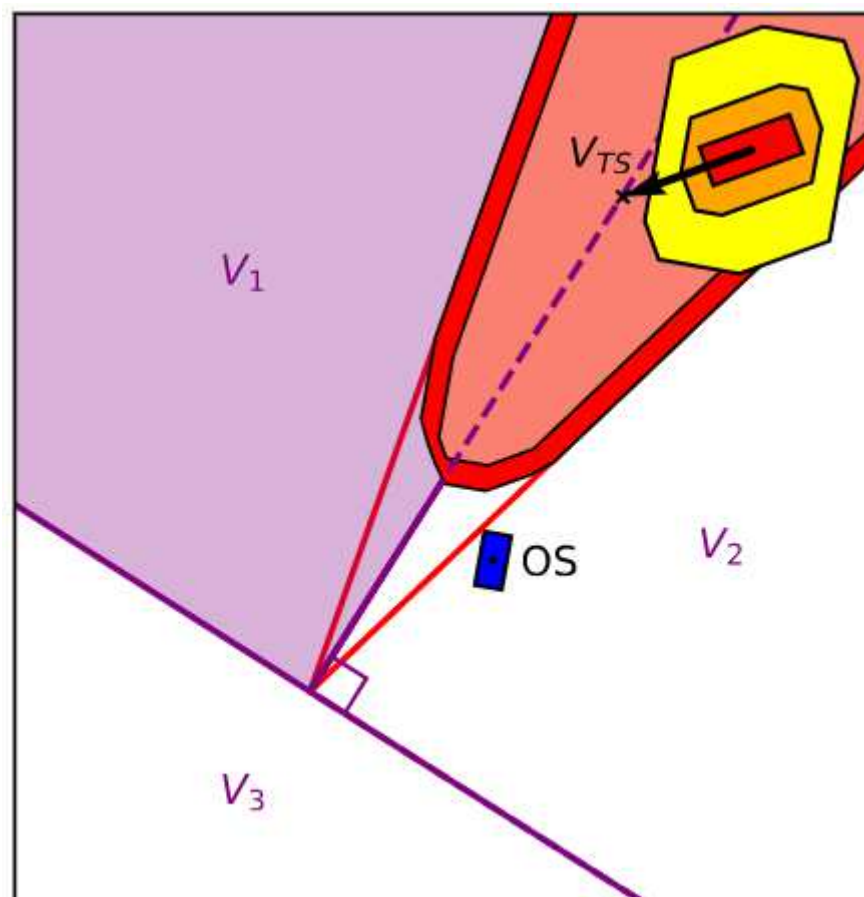
U stvarnim uvjetima, brzina TS-a može biti podložna nesigurnostima zbog ograničenja senzorskih mjerenja ili nepredvidivog ponašanja. Kako bi se te nesigurnosti uzele u obzir, VO se dodatno proširuje na način sličan proširenju TS-a s OS-om, koristeći Minkowskijevu sumu. Na taj način povećava se sigurnosni prostor algoritma.

Osim sigurnosnih aspekata, VO metoda uključuje ograničenja vezana za COLREG pravila. Ova pravila definiraju strane na kojima OS treba izbjegavati TS u različitim scenarijima, kao što su head-on susreti, crossing situacije i pretjecanja. VO algoritam prirodno dijeli prostor brzina u tri područja:

- **V1:** Brzine koje rezultiraju prolaskom TS-a s desne strane OS-a.
- **V2:** Brzine koje rezultiraju prolaskom TS-a s lijeve strane OS-a.
- **V3:** Brzine koje vode udaljavanju OS-a od TS-a.

COLREG pravila se aktiviraju samo u situacijama gdje trenutna brzina OS-a ulazi u VO ciljnog plovila (TS). Ovo ukazuje na postojanje rizika od sudara, što zahtijeva prilagodbu brzine OS-a kako bi se izbjegao kontakt.

Primjer situacije u kojoj COLREG ograničenja nisu potrebna je pretjecanje (Pravilo 13). U takvim situacijama, pravilo ne specificira stranu s koje TS treba biti pretjecan, pa algoritam ne uvodi dodatna ograničenja u prostor brzina.



Slika 5 – VO podjela prostora brzina

Kako bi odabrao optimalnu brzinu, algoritam diskretizira prostor brzina na osnovi intenziteta (od 0 do maksimalne brzine OS-a) i kursa (od 0° do 360°). Za svaku diskretnu brzinu provjerava se nalazi li se unutar VO-a ili ograničenja COLREG pravila. Brzine izvan ovih ograničenja smatraju se sigurnima, a optimalna brzina odabire se na temelju funkcije troška, koja penalizira veća odstupanja od željene brzine i kursa.

Diskretizacija prostora brzina i izbor optimalne brzine

Jednom kada su definirane prepreke brzine (VO) i primijenjena ograničenja COLREG pravila, algoritam prelazi na diskretizaciju prostora brzina kako bi identificirao optimalnu brzinu za vlastito plovilo (OS). Prostor brzina diskretizira se u dvije dimenzije:

- **Intenzitet brzine:** U rasponu od 0 do maksimalne brzine OS-a.
- **Kurs:** U rasponu od 0° do 360°, s definiranom rezolucijom

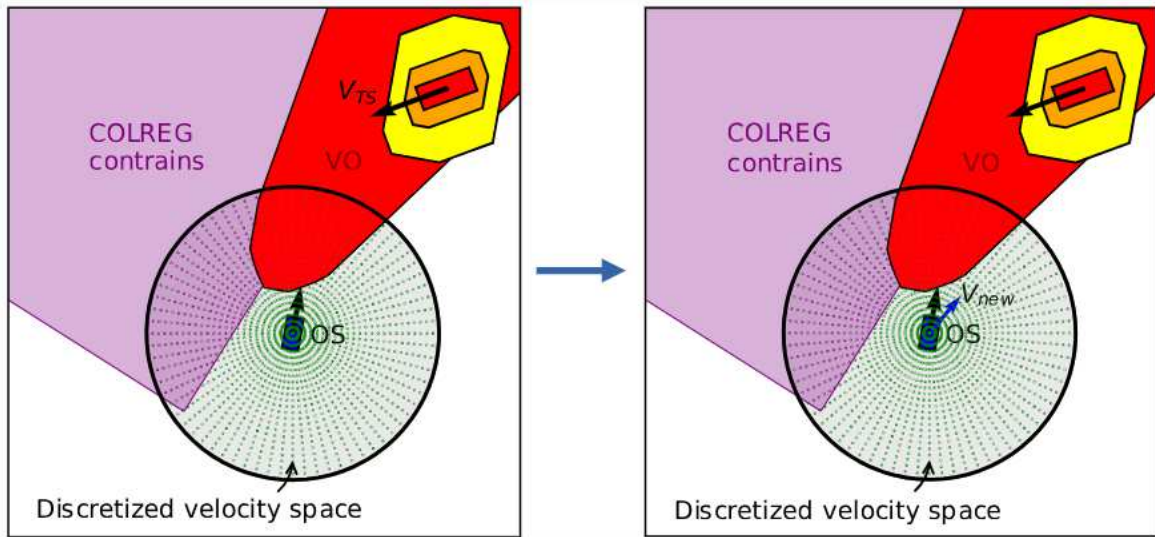
Za svaki diskretni par brzine i kursa provjerava se nalaze li se unutar VO i COLREG ograničenja. Sve brzine izvan tih ograničenja smatraju se sigurnima za navigaciju. Međutim, među sigurnim brzinama treba odabrati optimalnu, što se postiže primjenom funkcije troška. Funkcija troška uzima u obzir tri glavna čimbenika:

1. **Odstupanje od trenutne brzine:** Penalizira se velika promjena brzine kako bi se očuvala stabilnost.
2. **Odstupanje od željene brzine:** Brzina koja vodi prema sljedećoj točki na planiranoj ruti dobiva prioritet.
3. **Vidljiva promjena kursa:** Manevar mora biti jasno vidljiv drugim plovilima, pri čemu je definirano minimalno odstupanje kursa od 30°.

Matematički, funkcija troška može se zapisati kao:

$$J = w_1 \cdot \Delta(\theta_{des} - \theta_{free}) + w_2 \cdot \Delta(v_{des} - v_{free}) + w_3 \cdot \Delta(\theta_{OS+30} - \theta_{free})$$

Gdje su težinski faktori w_1 , w_2 i w_3 određeni eksperimentalno.



Slika 6 – Diskretizirani prostor brzine

Primjena funkcije troška omogućuje izbor optimalne brzine koja zadovoljava sigurnosne i operativne zahtjeve navigacije. Manevar završava kada trenutna brzina OS-a i željena brzina više nisu unutar VO za bilo koje ciljno plovilo. Kada OS plovi u okruženju s više ciljnih plovila (TS), metoda VO integrira sve individualne prepreke brzine i pripadajuća COLREG ograničenja. Na taj način stvara se zajednički prostor brzina unutar kojeg se biraju sigurni smjerovi kretanja. Algoritam računa zajednički prostor brzina i odabire brzinu koja izbjegava sudar sa svim TS-ovima, poštujući ograničenja. Ova sposobnost ključna je za navigaciju u složenim i dinamičnim okruženjima.

1.1.1. Matematička formulacija

Za definiranje VO potrebno je uzeti u obzir položaje, brzine i veličine oba plovila. Pretpostavimo da OS ima položaj x_A i brzinu v_A , dok TS ima položaj x_B i brzinu v_B . Radijusi plovila su r_A i r_B . VO se može definirati kao skup svih brzina v OS-a koje će rezultirati sudarom s TS-om u nekom trenutku $t > 0$:

$$VO_{A|B} = \{v \mid \exists t > 0: (v - v_B)t \in D(x_B - x_A, r_A + r_B)\}$$

Gdje $D(x, r)$ predstavlja disk s centrom u x i radijusom r . Ova jednadžba opisuje skup svih brzina OS-a koje će, u kombinaciji s brzinom TS-a, dovesti do sudara unutar radijusa $r_A + r_B$.

1.1.2. Geometrijska interpretacija

Geometrijski, VO se može vizualizirati kao konus u prostoru brzina OS-a. Vrh konusa je u trenutnoj brzini TS-a (v_B), a širina konusa ovisi o relativnom položaju i veličinama plovila. Ako OS odabere brzinu unutar ovog konusa, sudar je neizbježan ako oba plovila zadrže svoje trenutne brzine. Stoga, za izbjegavanje sudara, OS treba odabrati brzinu izvan ovog konusa. Za geometrijsku konstrukciju prepreke brzine (VO) koristi se Minkowskijeva suma. Ova metoda omogućuje precizno definiranje prostora u kojem se sudar može dogoditi, uzimajući u obzir veličine i položaje plovila.

Minkowskijeva suma dvaju skupova A i B definirana je kao:

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$

U kontekstu VO algoritma:

- A predstavlja trenutni položaj OS-a.
- B predstavlja prostor oko TS-a definiran njegovim radijusom sigurnosti r_B .

Koristeći Minkowskijevu sumu, prepreka brzine $VO_{A|B}$ može se vizualizirati kao konus u prostoru brzina, gdje je rub konusa određen svim potencijalnim sudarima OS-a i TS-a.

1.1.3. Implementacija

Metoda prepreka brzine je implementiran u programskom jeziku Python, poznatom po svojoj jednostavnoj sintaksi i bogatoj standardnoj biblioteci, što osigurava fleksibilnost i jednostavnost integracije s drugim softverskim komponentama. Ova implementacija omogućuje visoku razinu prilagodljivosti za eksperimentiranje i integraciju s drugim tehnologijama.

NumPy je ključna biblioteka za numeričko računanje u Pythonu. Omogućava rad s velikim, višedimenzionalnim nizovima i matricama te pruža niz matematičkih funkcija koje su optimizirane za performanse. U ovom projektu NumPy je korišten za reprezentaciju i manipulaciju podataka u obliku nizova, uključujući pozicije, brzine i smjerove plovila te izračune potrebne za geometrijske operacije, poput **Minkowskijeve sume**, koja je osnova za

proširenje dimenzija ciljnog plovila (TS) uključivanjem dimenzija vlastitog plovila (OS) i sigurnosne zone.

SciPy je nadogradnja na NumPy koja pruža dodatne funkcionalnosti za znanstveno i tehničko računanje. U ovom projektu, SciPy je korišten za optimizacijske rutine pri odabiru optimalne brzine na temelju definiranih funkcija troška te integraciju i interpolaciju podataka za glatke prijelaze između diskretiziranih brzina i stvarnih manevara plovila. SciPy-ove optimizacijske funkcije omogućile su određivanje brzine koja minimizira funkciju troška, uzimajući u obzir ograničenja definirana VO i COLREG pravilima.

ROS2 (Robot Operating System 2) je modernija verzija prvotne inačice ROS (Robot Operating System) koja donosi poboljšanja u pogledu performansi, fleksibilnosti i podrške za distribuirane sustave. U ovom projektu se koristi za:

- **Upravljanje komunikacijom:** ROS2 koristi DDS (Data Distribution Service) za pouzdanu i skalabilnu komunikaciju između različitih modula, uključujući senzore i kontrolne module.
- **Integraciju senzora:** Prikupljanje podataka o položaju, brzini i kursu plovila iz različitih izvora, poput GPS-a i mjerača brzine
- **Izvršavanje manevara:** Slanje kontrolnih naredbi za promjenu kursa i brzine prema izlazima VO metode

ROS2 je odabran zbog svoje kompatibilnosti s višeslojnim sustavima i podrške za stvarno vremensku obradu podataka, što je ključno za navigaciju u dinamičnim okruženjima.

Samba je softver koji omogućuje integraciju između Windows i Linux okruženja, pružajući podršku za dijeljenje direktorija i resursa. U ovom projektu, Samba je korištena za dijeljenje direktorija između sustava. Direktorij na Linux sustavu, gdje se pokreće ROS2, postavljen je kao zajednički direktorij dostupan na Windows sustavu. Ovakva konfiguracija omogućila je izravnu razmjenu datoteka između operativnih sustava. Promjene u kodu i konfiguraciji, napravljene u Visual Studio na Windows računalu, automatski su bile dostupne na Linux sustavu. Na taj način, ROS2 sustav na Linuxu mogao je odmah pokrenuti ažurirani kod, omogućujući brzu iteraciju između razvoja i testiranja.

1.1.4. Arhitektura sustava

Implementacija metode VO organizirana je u nekoliko jasno odvojenih modula, što omogućuje modularnost i jednostavno održavanje.

Ključni moduli sustava

1. ROS Node modul (RosScriptNode):

Ovaj modul predstavlja ulaznu točku sustava i ima sljedeće funkcije:

- **Inicijalizacija i komunikacija**

Modul inicijalizira ROS čvor, čime se omogućuje komunikacija između različitih komponenti sustava. U njemu se postavljaju pretplatnici (subscribers) na senzorske podatke, kao što su GPS podaci, te se osigurava pravovremeno primanje i distribucija podataka.

- **Obrada ulaznih podataka**

Funkcije poput `gps_callback_os()`, `gps_callback_tp()`, i dodatne varijante (`gps_callback_ts_1()`, `gps_callback_ts_2()`, itd.) služe za primanje i obradu podataka iz različitih izvora. Na temelju ovih funkcija, sustav ažurira poziciju, brzinu i smjer plovila u realnom vremenu.

- **Implementacija kontrolera:**

Unutar ovog modula nalaze se funkcije koje implementiraju PD/PID kontrolere. Na primjer, funkcije `compute_pd()` i `compute_pid()` (s njihovim varijantama poput `compute_pid_1()`, `compute_pid_2()`, itd.) omogućuju fino podešavanje putanje plovila. Ovi kontroleri koriste ulazne podatke kako bi izračunali potrebne korekcije u brzini i kutu, čime se postiže optimizacija kretanja.

Za podršku glavnim funkcijama sustava korištene su i dodatne pomoćne funkcije koje značajno doprinose preciznosti i pouzdanosti rada. Primjerice, funkcija `angle_diff()` izračunava razliku između dva kuta, što je ključno za pravilno usmjeravanje plovila u realnom vremenu. Nadalje, funkcija `param_callback()` omogućava dinamičko ažuriranje parametara tijekom izvođenja, čime se osigurava prilagodba sustava promjenjivim

uvjetima, dok *exit_handler()* garantira uredno i kontrolirano zaustavljanje svih procesa u slučaju prekida rada.

2. **Target Ship (TS) modul:**

Modul za simulaciju ciljnog plovila (TS) koristi se za generiranje podataka o kretanju i položaju drugih plovila u simulacijskom okruženju. Ovaj modul osigurava da se metoda VO testira u uvjetima koji repliciraju stvarne situacije na moru, omogućujući detaljnu evaluaciju performansi algoritma u scenarijima poput head-on susreta, crossing situacija i pretjecanja.

- **Simulacija dinamike**

Modul modelira kretanje ciljnog plovila, uključujući promjene brzine, smjera i položaja. Ovi podaci se koriste za testiranje interakcije između glavnog plovila i ostalih subjekata simulacije.

- **Integracija sa VO metodom**

Generirani podaci iz TS modula šalju se u VO metodu, gdje se koriste za izračunavanje potencijalnih kolizijskih zona i određivanje sigurnih putanja.

3. **Velocity Obstacle (VO) metoda:**

Središnji dio sustava je VO metoda, koji sadrži ključne funkcije za izračun nove, sigurne, putanje plovila. Najvažnije funkcionalnosti metode su:

- **Izračun Minkowski sume**

Funkcija `calc_minkowski_sum()` koristi matematičku operaciju Minkowski sume kako bi kombinirala geometrijske oblike plovila i prepreka. Rezultat ove operacije definira kolizijski prostor, odnosno područje u kojem je plovilo potencijalno opasno.

- **Izračun slobodnog prostora**

Funkcija `calc_free_vel_space()` analizira prostor oko plovila i određuje siguran raspon brzina i smjerova za izbjegavanje sudara.

- **Određivanje nove brzine**

Funkcija `calc_new_vel()` na temelju ulaznih podataka (trenutne brzine, pozicije i kuta) i informacija o slobodnom prostoru izračunava optimalnu novu brzinu i smjer kretanja. Time se osigurava da plovilo izbjegne potencijalne kolizije.

- **Provjera COLREG kompatibilnosti**

Funkcija `calc_colreg_con()` provjerava je li predložena putanja u skladu s COLREG pravilima, osiguravajući da se donesene odluke usklade s međunarodnim standardima za sprječavanje sudara.

Uz glavne funkcionalnosti, metoda je obogaćena nizom pomoćnih funkcija koje dodatno povećavaju preciznost analize i donošenja odluka. Primjerice, funkcija `calc_rel_vel()` koristi se za izračunavanje relativne brzine između plovila, dok `calc_vel_final()` omogućuje precizno određivanje konačne brzine nakon primjene korekcija. Nadalje, funkcija `ang_betw_vect()` izračunava kut između vektora, što je od ključne važnosti za određivanje optimalnog smjera kretanja. Zajedno, ove funkcije osiguravaju da se situacija u realnom vremenu analizira s visokim stupnjem točnosti, omogućujući metodi donošenje optimalne odluke u skladu s promjenjivim uvjetima.

1.2. Simulacijski eksperiment

U ovom poglavlju detaljno je opisan postupak testiranja sustava za izbjegavanje sudara kod autonomnih površinskih plovila (ASV) temeljenog na COLREG pravilima. Testiranje je provedeno u simulacijskom okruženju pomoću ROS2 sustava, pri čemu su korišteni različiti scenariji susreta između glavnog broda (OS – Own Ship) i virtualnog cilja (TS – Target Ship). Podaci su prikupljeni u stvarnom vremenu, obrađeni i spremljeni u CSV datoteke, a rezultati su vizualno prezentirani pomoću grafova.

1.2.1. Sadržaj eksperimenta

Testni scenariji su osmišljeni tako da pokriju sve relevantne slučajeve susreta između glavnog broda i virtualnog cilja. Konkretno, testirani su sljedeći slučajevi:

- **Frontalni susret**
 - **Opis:** Glavni brod kreće iz ishodišta (0,0) u pozitivnom smjeru x osi, dok se virtualni cilj približava iz suprotnog smjera.
 - **Očekivani rezultat:** Algoritam treba izračunati i predložiti evazivnu manevru (npr. Pomak prema lijevoj ili desnoj strani) u skladu s COLREGs pravilima.
- **Susret pod kutom (dijagonalni susret)**
 - **Opis:** Glavni brod kreće iz ishodišta (0,0) u pozitivnom smjeru x osi, dok se virtualni cilj približava glavnom brodu pod različitim kutovima (45°, 135°, 225°, 315°).
 - **Očekivani rezultat:** Sustav treba prilagoditi putanju glavnog broda tako da se sigurno izbjegne virtualni cilj, pri čemu se detaljno prati promjena brzine i smjera.
- **Pretjecanje**
 - **Opis:** Glavni brod kreće iz ishodišta (0,0) u pozitivnom smjeru x osi, dok virtualni cilj kreće iz pozicije (20,0) u pozitivnom smjeru x osi ali sporije, što dovodi do situacije pretjecanja.
 - **Očekivani rezultat:**
- **Dodatni varijabilni slučajevi**
 - Testirane su i situacije u kojima se uvjeti plovidbe mijenjaju tijekom trajanja susreta (npr. Promjena brzine, nagli zakreti i promjene smjera virtualnog cilja). Ovi slučajevi osiguravaju da sustav može reagirati na dinamične promjene u okruženju.

1.2.2. Prikupljanje i obrada podataka

Podaci se prikupljaju pomoću ROS2 sustava, gdje se:

- **Odometrijski podaci glavnog broda** dobivaju preko ROS teme /main_ship/odom i koriste se za ažuriranje trenutne pozicije, brzine, orijentacije i puta glavnog broda

- **Podaci o virtualnom cilju** (VirtualTargetTracks) dobivaju se preko ROS teme /virtual_target_tracks, a s njima se bilježe položaj, brzina, orijentacija i dimenzije virtualnog cilja
- **Metoda za izbjegavanje sudara** (baziran na metodi velocity obstacles) koristi podatke obje strane (OS i TS) za izračun nove brzine i orijentacije glavnog broda
- **CSV datoteka** korištena za bilježenje podataka u stvarnom vremenu za svaki interval. Podaci koji se bilježe su:
 - Vrijeme (timestamp)
 - Podatke o glavnom brodu: jedinstveni identifikator (UUID), pozicija (x, y), brzina, orijentacija, širina, duljina
 - Podatke o virtualnom cilju: jedinstveni identifikator (UUID), pozicija (x, y), brzina, orijentacija, širina, duljina
 - Parametre korištene u VO metodi: dimenzije OS, maksimalna brzina, vremena do sudara, sigurnosni faktor, nesigurnosti u brzini i orijentaciji TS, te rezolucije brzine i orijentacije

Ovi podaci omogućuju detaljnu analizu performansi metode i usporedbu stvarnog ponašanja s očekivanim rezultatima.

1.2.3. Analiza podataka

Za analizu prikupljenih podataka korištene su sljedeće metode:

- **Grafička analiza**

Putanje glavnog broda i virtualnog cilja prikazane su grafički, pri čemu se na grafovima dodatno označavaju smjerovi kretanja (strelicama) te ključne točke gdje je došlo do promjene putanje. Ovi grafovi omogućuju vizualnu verifikaciju pravilnosti izračuna i izvedbe evazivnih manevra.

- **Statistička analiza**

Iz CSV datoteka se dobivaju statistički podaci poput prosječne brzine obrade, odstupanja u izračunatim putanjama te frekvencije i vrste odabranih manevra. Time se može kvantificirati točnost planiranja putanje i stabilnost sustava u različitim uvjetima.

1.3. Prikaz testiranih scenarija

Za svaki testni scenarij uključujemo opis scenarija, što se testira te kako se scenarij odvija. Grafičkim prikazom je prikazana putanja brodova, vremensko kretanje, udaljenost te kritične točke i analiza manevra. Analizom rezultata je ocijenjena uspješnost manevra, očekivano ponašanje te potencijalna odstupanja od očekivanog ponašanja.

Zajedničke postavke za sve scenarije

Parametri glavnog broda (OS)

- uuid_main: 000
- velocity_main_y: 0.0

Parametri virtualnog cilja 1 (TS)

- uuid_virtual_target_0: 111
- width_virtual_target_0: 10.0
- length_virtual_target_0: 20.0
- initial_position_virtual_target_z_0: 0.0
- final_position_virtual_target_x_0: 0.0
- final_position_virtual_target_y_0: 0.0
- final_position_virtual_target_z_0: 0.0
- current_position_virtual_target_z_0: 0.0

Parametri metode prepreka brzine

- length of OS: 8.0
- width of OS: 2.0
- maximum speed of OS: 5.0
- time to collision: 120
- time threshold to collision: 60
- safety factor: 5
- uncertainties in speed of TS: 0.15
- uncertainties in orientation of TS: 1.5

- resolution of speed of the velocity space: 0.25
- resolution of orientation of the velocity space: 3

U scenarijima koji nemaju navedene zajedničke postavke istih vrijednosti navedene su vrijednosti koje su izmijenjene.

Scenarij 1 – Frontalni susret

Glavni brod se nalazi na koordinati (0, 0) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj kreće se iz suprotnog smjera, u negativnom x-smjeru, također po osi $y=0$ s konstantnom brzinom. Situacija odgovara frontalnom susretu gdje je relativni kut orijentacije virtualnog cilja 180° .

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 80.0
- initial_position_virtual_target_y_0: 0.0
- current_position_virtual_target_x_0: 80.0
- current_position_virtual_target_y_0: 0.0
- velocity_virtual_target_0: 13.0
- orientation_virtual_target_0: 3.141592

Parametri glavnog broda:

- initial_position_main_x: -40.0
- initial_position_main_y: 0.0
- desired_velocity_main: 14.0
- velocity_main_x: 13.9

Izmijenjeni parametri metode prepreka brzine

- maximum speed of OS: 50.0

Scenarij 2 – Virtualni cilj relativnog kuta orijentacije 90 stupnjeva (prolaz s prednje strane)

Glavni brod se nalazi na koordinati (0, 23) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj kreće iz početne pozicije (50, -30) konstantnom brzinom u pozitivnom y-smjeru, s desne strane glavnog broda. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 90°.

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 50.0
- initial_position_virtual_target_y_0: -30.0
- current_position_virtual_target_x_0: 50.0
- current_position_virtual_target_y_0: -30.0
- velocity_virtual_target_0: 3.0
- orientation_virtual_target_0: 1.570796

Parametri glavnog broda:

- initial_position_main_x: 0.0
- initial_position_main_y: 23.0
- desired_velocity_main: 5.0
- velocity_main_x: 4.9

Scenarij 3 – Virtualni cilj relativnog kuta orijentacije 90 stupnjeva (prolaz sa stražnje strane)

Glavni brod se nalazi na koordinati (0, 23) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj kreće iz početne pozicije (50, 0) okomito konstantnom brzinom u pozitivnom y-smjeru, s desne strane glavnog broda. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 90°.

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 50.0
- initial_position_virtual_target_y_0: 0.0
- current_position_virtual_target_x_0: 50.0
- current_position_virtual_target_y_0: 0.0
- velocity_virtual_target_0: 5.0
- orientation_virtual_target_0: 1.570796

Parametri glavnog broda:

- initial_position_main_x: 0.0
- initial_position_main_y: 23.0
- desired_velocity_main: 5.0
- velocity_main_x: 4.9

Scenarij 4 – Virtualni cilj relativnog kuta orijentacije 270 stupnjeva

Glavni brod se nalazi na koordinati (0,13) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (50, 35) te se kreće okomito u negativnom y-smjeru uz konstantnu brzinu. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 270°.

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 50.0
- initial_position_virtual_target_y_0: 35.0
- current_position_virtual_target_x_0: 50.0
- current_position_virtual_target_y_0: 35.0
- velocity_virtual_target_0: 2.0
- orientation_virtual_target_0: 4.712389

Parametri glavnog broda:

- initial_position_main_x: 0.0

- initial_position_main_y: 13.0
- desired_velocity_main: 4.0
- velocity_main_x: 3.9

Scenarij 5 – Virtualni cilj relativnog kuta orijentacije 270 stupnjeva (prolazak sa stražnje strane)

Glavni brod se nalazi na koordinati (0,13) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (50, 35) te se kreće okomito u negativnom y-smjeru uz konstantnu brzinu. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 270°.

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 50.0
- initial_position_virtual_target_y_0: 35.0
- current_position_virtual_target_x_0: 50.0
- current_position_virtual_target_y_0: 35.0
- velocity_virtual_target_0: 2.0
- orientation_virtual_target_0: 4.712389

Parametri glavnog broda:

- initial_position_main_x: 0.0
- initial_position_main_y: 13.0
- desired_velocity_main: 4.0
- velocity_main_x: 3.9

Izmijenjeni parametri metode prepreka brzine

- length of OS: 20.0
- width of OS: 10.0

Scenarij 6 – Pretjecanje

Glavni brod se nalazi na koordinati (-100, 0) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (40, 0) te se kreće horizontalno konstantnom brzinom u pozitivnom x-smjeru. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 0° .

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 40.0
- initial_position_virtual_target_y_0: 0.0
- current_position_virtual_target_x_0: 40.0
- current_position_virtual_target_y_0: 0.0
- velocity_virtual_target_0: 0.4
- orientation_virtual_target_0: 0.0

Parametri glavnog broda:

- initial_position_main_x: -100.0
- initial_position_main_y: 0.0
- desired_velocity_main: 5.0
- velocity_main_x: 4.9

Scenarij 7 – Virtualni cilj relativnog kuta orijentacije 135 stupnjeva

Glavni brod se nalazi na koordinati (0, 23) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (60, 20) te se kreće u pozitivnom y-smjeru konstantnom brzinom. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 135° .

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 60.0
- initial_position_virtual_target_y_0: 20.0
- current_position_virtual_target_x_0: 60.0

- `current_position_virtual_target_y_0`: 20.0
- `velocity_virtual_target_0`: 2.0
- `orientation_virtual_target_0`: 2.356194

Parametri glavnog broda:

- `initial_position_main_x`: 0.0
- `initial_position_main_y`: 23.0
- `desired_velocity_main`: 4.0
- `velocity_main_x`: 3.9

Scenarij 8 – Virtualni cilj relativnog kuta orijentacije 135 stupnjeva (lijevi zaobilazak)

Glavni brod se nalazi na koordinati (0, 23) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (70, 10) te se kreće u pozitivnom y-smjeru konstantnom brzinom. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 135°.

Parametri virtualnog cilja

- `initial_position_virtual_target_x_0`: 70.0
- `initial_position_virtual_target_y_0`: 10.0
- `current_position_virtual_target_x_0`: 70.0
- `current_position_virtual_target_y_0`: 10.0
- `velocity_virtual_target_0`: 1.0
- `orientation_virtual_target_0`: 2.356194

Parametri glavnog broda:

- `initial_position_main_x`: 0.0
- `initial_position_main_y`: 23.0
- `desired_velocity_main`: 5.0
- `velocity_main_x`: 4.9

Scenarij 9 – Virtualni cilj relativnog kuta orijentacije 45 stupnjeva

Glavni brod se nalazi na koordinati (-50, 13) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (40, 0) te se kreće u pozitivnom y-smjeru konstantnom brzinom. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 45°.

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: -50.0
- initial_position_virtual_target_y_0: 13.0
- current_position_virtual_target_x_0: 50.0
- current_position_virtual_target_y_0: 13.0
- velocity_virtual_target_0: 1.0
- orientation_virtual_target_0: 0.785398

Parametri glavnog broda:

- initial_position_main_x: -50.0
- initial_position_main_y: 13.0
- desired_velocity_main: 5.0
- velocity_main_x: 4.9

Izmijenjeni parametri metode prepreka brzine

- maximum speed of OS: 50.0

Scenarij 10 – Virtualni cilj relativnog kuta orijentacije 315 stupnjeva

Glavni brod se nalazi na koordinati (0, 23) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (50, 50) te se kreće u negativnom y-smjeru konstantnom brzinom. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 315°.

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 50.0

- initial_position_virtual_target_y_0: 50.0
- current_position_virtual_target_x_0: 50.0
- current_position_virtual_target_y_0: 50.0
- velocity_virtual_target_0: 1.5
- orientation_virtual_target_0: 5.497787

Parametri glavnog broda:

- initial_position_main_x: 0.0
- initial_position_main_y: 23.0
- desired_velocity_main: 4.0
- velocity_main_x: 3.9

Izmijenjeni parametri metode prepreka brzine

- maximum speed of OS: 50.0

Scenarij 11 – Virtualni cilj relativnog kuta orijentacije 315 stupnjeva (lijevi zaobilazak)

Glavni brod se nalazi na koordinati (0, 23) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj se nalazi na koordinati (50, 50) te se kreće u negativnom y-smjeru konstantnom brzinom. Situacija odgovara susretu gdje je relativni kut orijentacije virtualnog cilja 315°.

Parametri virtualnog cilja

- initial_position_virtual_target_x_0: 50.0
- initial_position_virtual_target_y_0: 50.0
- current_position_virtual_target_x_0: 50.0
- current_position_virtual_target_y_0: 50.0
- velocity_virtual_target_0: 3.0
- orientation_virtual_target_0: 1.570796

Parametri glavnog broda:

- initial_position_main_x: 0.0

- initial_position_main_y: 23.0
- desired_velocity_main: 4.0
- velocity_main_x: 3.9

Izmijenjeni parametri metode prepreka brzine

- maximum speed of OS: 50.0

Scenarij 12 – Frontalni susret s dva virtualna cilja

Glavni brod se nalazi na koordinati (-100, 13) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj 1 se nalazi na koordinati (60, 13) te se kreće konstantnom brzinom u negativnom x-smjeru pod kutom 180°. Virtualni cilj 2 se nalazi na koordinati (60, 20) te se kreće konstantnom brzinom u negativnom x-smjeru pod kutom 180°.

Parametri virtualnog cilja 1

- initial_position_virtual_target_x_0: 60.0
- initial_position_virtual_target_y_0: 13.0
- current_position_virtual_target_x_0: 60.0
- current_position_virtual_target_y_0: 13.0
- velocity_virtual_target_0: 8.0
- orientation_virtual_target_0: 3.141592

Parametri virtualnog cilja 2

- initial_position_virtual_target_x_0: 60.0
- initial_position_virtual_target_y_0: 20.0
- current_position_virtual_target_x_0: 60.0
- current_position_virtual_target_y_0: 20.0
- velocity_virtual_target_0: 6.9
- orientation_virtual_target_0: 3.141592

Parametri glavnog broda:

- initial_position_main_x: -100.0
- initial_position_main_y: 13.0
- desired_velocity_main: 14.0
- velocity_main_x: 13.9

Izmijenjeni parametri metode prepreka brzine

- maximum speed of OS: 50.0

Scenarij 13 – Frontalni susret s jednim virtualnim ciljem, drugi virtualni cilj relativnog kuta orijentacije 90 stupnjeva

Glavni brod se nalazi na koordinati (-100, 13) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj 1 se nalazi na koordinati (80, 35) te se kreće konstantnom brzinom u negativnom x-smjeru pod kutom 180°. Virtualni cilj 2 se nalazi na koordinati (-60, -20) te se kreće konstantnom brzinom u pozitivnom y-smjeru pod kutom 90°.

Parametri virtualnog cilja 1

- initial_position_virtual_target_x_0: 80.0
- initial_position_virtual_target_y_0: 35.0
- current_position_virtual_target_x_0: 80.0
- current_position_virtual_target_y_0: 35.0
- velocity_virtual_target_0: 8.0
- orientation_virtual_target_0: 3.141592

Parametri virtualnog cilja 2

- initial_position_virtual_target_x_0: -60.0
- initial_position_virtual_target_y_0: -20.0
- current_position_virtual_target_x_0: -60.0
- current_position_virtual_target_y_0: -20.0

- velocity_virtual_target_0: 6.9
- orientation_virtual_target_0: 1.570796

Parametri glavnog broda:

- initial_position_main_x: -100.0
- initial_position_main_y: 13.0
- desired_velocity_main: 14.0
- velocity_main_x: 13.9

Izmijenjeni parametri metode prepreka brzine

- maximum speed of OS: 25.0

Scenarij 14 – Jedan virtualni cilj relativnog kuta orijentacije 270 stupnjeva, drugi virtualni cilj relativnog kuta orijentacije 315 stupnjeva

Glavni brod se nalazi na koordinati (-100, 13) i kreće se vodoravno u pozitivnom x-smjeru uz konstantnu brzinu do ulaska u potencijalnu zonu sudara. Virtualni cilj 1 se nalazi na koordinati (-10, 75) te se kreće konstantnom brzinom u negativnom y-smjeru pod kutom 270°. Virtualni cilj 2 se nalazi na koordinati (-30, 55) te se kreće konstantnom brzinom u negativnom y-smjeru pod kutom 315°.

Parametri virtualnog cilja 1

- initial_position_virtual_target_x_0: -10.0
- initial_position_virtual_target_y_0: 75.0
- current_position_virtual_target_x_0: -10.0
- current_position_virtual_target_y_0: 75.0
- velocity_virtual_target_0: 8.0
- orientation_virtual_target_0: 4.712389

Parametri virtualnog cilja 2

- initial_position_virtual_target_x_0: -30.0
- initial_position_virtual_target_y_0: 55.0

- current_position_virtual_target_x_0: -30.0
- current_position_virtual_target_y_0: 55.0
- velocity_virtual_target_0: 6.9
- orientation_virtual_target_0: 5.497787

Parametri glavnog broda:

- initial_position_main_x: -100.0
- initial_position_main_y: 13.0
- desired_velocity_main: 14.0
- velocity_main_x: 13.9

Izmijenjeni parametri metode prepreka brzine

- maximum speed of OS: 25.0

2. Metoda izbjegavanja sudara na temelju dubokog podržanog učenja

Kroz metodu izbjegavanja sudara na temelju dubokog podržanog učenja istražujemo kako duboko podržano učenje (Deep Reinforcement Learning - DRL) može omogućiti plovilu prilagodljivo donošenje odluka bez potrebe za ručnim podešavanjima. Treniramo RL agenta koji je sposoban samostalno balansirati između praćenja putanje i izbjegavanja prepreka, donoseći optimalne odluke u realnom vremenu. Iako postoji veliki broj istraživanja u području autonomne navigacije, većina metoda pati od određenih ograničenja, uključujući pojednostavljene pretpostavke o dinamici plovila (ili potpuno zanemarivanje istih), nedostatak skalabilnosti s obzirom na složenost okruženja (veliki broj prepreka, varijabilna kretanja), veliki računalni zahtjevi, što otežava primjenu na fizičkim sustavima s ograničenim resursima i nedostatak determinističkih odluka, što može biti problematično u sigurnosno kritičnim aplikacijama. Duboko podržano učenje nudi mogućnost modeliranja optimalnog ponašanja agenta u nepoznatim i djelomično opaženim okruženjima. Agenti koriste pristup istraživanja i evaluacije kako bi postupno poboljšali svoje strategije, učeći kako maksimizirati dugoročnu nagradu kroz niz interakcija sa simulacijskim okruženjem. Kroz promatrani primjer istraženo je kako se RL može primijeniti na vođenje i upravljanje autonomnim površinskim plovilima s posebnim naglaskom na praćenje unaprijed definirane putanje te izbjegavanje prepreka u realnom vremenu. Pristup se temelji na end-to-end učenju, gdje RL agent prima očitavanja senzora i nagrade iz okruženja te na temelju tih podataka razvija inteligentnu strategiju upravljanja za izazovne scenarije izbjegavanja prepreka.

2.1. Opis metode

2.1.1. Navigacija i upravljanje pomorskim plovilima

Upravljanje autonomnim površinskim plovilima zahtijeva precizno definirane matematičke modele koji opisuju njihovu dinamiku. Ti modeli uključuju koordinatne sustave, varijable stanja i dinamičke jednadžbe kretanja, što omogućuje točnu simulaciju i kontrolu plovila u realnom okruženju.

Koordinatni sustavi

Za modeliranje dinamike plovila koriste se dva glavna koordinatna sustava:

- Inercijski (NED) koordinatni sustav $\{n\} = (x_n, y_n, z_n)$: Ovaj referentni sustav koristi se za navigaciju i predstavlja tangentnu ravninu Zemljine površine. Njegove osi definirane su kao:
 - x_n – usmjeren prema sjeveru
 - y_n – usmjeren prema istoku
 - z_n – usmjeren prema središtu Zemlje
- Tijelu pridruženi koordinatni sustav $\{b\} = (x_b, y_b, z_b)$: Ovaj sustav je fiksiran na plovilo i okreće se zajedno s njim. Njegove osi definirane su kako slijedi:
 - x_b – uzdužna os plovila
 - y_b – poprečna os
 - z_b – okomita os

Varijable stanja

Prema standardima Society of Naval Architects and Marine Engineers (SNAME), stanje plovila opisuje se pomoću dvanaest varijabli, koje omogućuju precizno praćenje gibanja plovila i njegovih promjena orijentacije. Varijable su podijeljene u dvije skupine:

- Generalizirane koordinate položaja:

$$\eta = [x_n, y_n, z_n, \phi, \theta, \psi]^T$$

- x_n, y_n, z_n – položaj plovila u NED sustavu,

- ϕ, θ, ψ – kutovi rotacije plovila (roll, pitch i yaw), koji definiraju njegovu orijentaciju u odnosu na inercijski sustav
- Generalizirane brzine

$$v = [u, v, w, p, q, r]^T$$

- u, v, w – komponente linearne brzine u smjerovima x_b, y_b, z_b
- p, q, r – kutne brzine (rotacije oko vlastitih osi plovila)

Dinamika kretanja plovila

Kretanje plovila u vodi može se opisati sustavom nelinearnih diferencijalnih jednadžbi koje uključuju utjecaj mase plovila, hidrodinamičkih sila, gravitacije i aktuatorskih sila. Ako zanemarimo vanjske utjecaje poput struja, vjetra i valova, dinamika plovila može se zapisati u matičnom obliku:

$$\dot{\eta} = J_2(\eta)v$$

$$B_f = M_{RB} \dot{v} + C_{RB}(v)v + g(\eta) + M_A \dot{v} + C_A(v)v + D(v)v$$

- $J_2(\eta)$ – matrica transformacije iz tijelu pridruženog sustava u inercijski sustav,
- M_{RB} i M_A – mase plovila i dodane mase (učinak fluida),
- $C_{RB}(v)$ i $C_A(v)$ – Coriolisove sile i učinak fluida,
- $D(v)$ – matrica prigušenja,
- $g(\eta)$ – sile gravitacije i plovnosti,
- B – matrica aktuatora
- f – vektor upravljačkih sila

Model upravljanja s tri stupnja slobode (3 – DOF)

U mnogim primjenama koristi se pojednostavljeni model s tri stupnja slobode (3-DOF), koji uzima u obzir samo horizontalne komponente gibanja. Ovaj model ignorira vertikalno gibanje te nagib i valjanje plovila (roll i pitch), što je realistična pretpostavka za plovila koja plove na mirnoj vodi.

Pretpostavke modela 3-DOF:

- Bez vanjskih sila

- Ograničenja prostora stanja – plovilo uvijek na površini vode
- Simetrija plovila
- Koordinatni sustav u središnjoj liniji
- Nedostatak lateralne sile

Matematički, pojednostavljeni model 3-DOF izražava se kao:

$$\dot{\eta} = R_z(\psi)v$$

$$M \dot{v} + C(v)v + D(v)v = B_f$$

- R_z – matrica rotacije oko vertikalne osi,
- M – kombinirana matrica mase i dodane mase,
- $C(v)$ – matrica Coriolisovih sila,
- $D(v)$ – matrica hidrodinamičkog prigušenja.

Matrica rotacije definirana je kao:

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

Glavne upravljačke sile u ovom modelu su potisak u smjeru kretanja i moment zakretanja. Ovaj model omogućava simulaciju kretanja plovila s obzirom na brzinu naprijed te zakretanje oko vertikalne osi.

2.1.2. Podržano učenje (Reinforcement Learning – RL)

RL omogućava autonomnim agentima samostalno učenje optimalnog ponašanja u nepoznatim okruženjima. Agent ne prima upute o tome koje akcije su najbolje, već koristi tehniku pokušaja i pogrešaka, nalik načinu na koji ljudi uče iz iskustva. U svakom diskretnom vremenskom koraku, RL agent opaža trenutno stanje s okruženja, na temelju kojeg odabire akciju a . Strategija kojom agent bira akcije naziva se politika i označava se kao FORMULA $\pi(s)$. Drugim riječima, politika π definira pravilo donošenja odluka, odnosno mapiranje iz skupa stanja S u skup akcija A . Nakon svake izvedene akcije, agent

prima numeričku nagradu r koja reflektira kvalitetu te akcije. Cilj RL-a je maksimizirati ukupnu dugoročnu nagradu, što se često naziva i povrat (R_t). Optimizacija politike vrši se kroz iterativne prilagodbe, pri čemu agent uči kako poboljšati svoju strategiju donošenja odluka.

Osnovni elementi RL-a uključuju:

- Agent – entitet koji donosi odluke,
- Okruženje – svijet s kojim agent koegzistira,
- Stanja (s) – informacija o trenutnom stanju okruženja,
- Akcije (a) – skup mogućih odluka koje agent može donijeti,
- Politika (π) – pravilo prema kojem agent bira akcije,
- Nagrada (r) – povratna informacija o kvaliteti odabrane akcije.

Ključni aspekti RL-a

Nagrada ne mora biti izravno povezana s posljednjom akcijom agenta. Na primjer, akcija koja kratkoročno izgleda nepovoljno može dugoročno donijeti veću ukupnu nagradu. Ovaj aspekt RL-a omogućava agentima da razvijaju strategije dugoročnog planiranja.

Politika može biti deterministička ili stohastička. Deterministička politika uvijek bira istu akciju za isto stanje, dok stohastička politika bira prema vjerojatnosnoj distribuciji, što može biti korisno u neizvjesnim okruženjima.

Akcije mogu biti kontinuirane ili diskretne. Tradicionalni RL algoritmi radili su s diskretnim skupovima akcija, no napredni algoritmi poput PPO i DDPG omogućuju rad s kontinuiranim prostorima akcija, što je osobito korisno za upravljanje autonomnim plovilima.

Markovljev proces odlučivanja (MDP)

Mnogi RL problemi modeliraju se pomoću Markovljevog proces odlučivanja (MDP), koji se definira sa:

- S – skup svih mogućih stanja okruženja,
- A – skup svih mogućih akcija,
- $R(s, a)$ – nagradna funkcija koja dodjeljuje nagradu nakon svake akcije,

- $P(s' | s, a)$ – funkcija prijelaza između stanja (vjerojatnost da će sustav prijeći iz s u s' nakon akcije a),
- γ – faktor diskontiranja koji modelira sklonost agenta prema kratkoročnim ili dugoročnim nagradama.

Agent uči strategiju donošenja odluka kroz uzorkovanje sekvenci (trajektorija) interakcija s okruženjem. Prvo se inicijalno stanje s_0 uzorkuje prema distribuciji $p(s_0)$, zatim agent bira akciju a_t prema politici $\pi(s_t)$, nakon čega okruženje generira novo stanje s_{t+1} i nagradu r_t .

Ukupna nagrada definirana je kao:

$$R_t = \sum_{i=t}^{\infty} r(s_i, a_i)$$

Funkcije vrijednosti stanja i akcija

Kako bi RL agent mogao procijeniti vrijednost svojih odluka, definiraju se funkcije vrijednosti stanja $V^\pi(s)$ i funkcija vrijednosti akcije $Q^\pi(s, a)$. Funkcija vrijednosti stanja $V^\pi(s)$ predstavlja očekivanu ukupnu nagradu ako agent započne u stanju s i nastavi se ponašati prema politici π :

$$V^\pi(s_t) = E_{s_{i \geq t}, a_{i \geq t} \sim \pi} [R_t | s_t]$$

Funkcija vrijednosti akcije $Q^\pi(s, a)$ predstavlja očekivanu ukupnu nagradu ako agent započne u stanju s i nastavi se ponašati prema politici π :

$$Q^\pi(s_t, a_t) = E_{s_{i \geq t}, a_{i \geq t} \sim \pi} [R_t | s_t, a_t]$$

Funkcija $Q^\pi(s, a)$ omogućava agentu procjenu kvalitete svake moguće akcije, što je ključno za donošenje optimalnih odluka. RL omogućava agentima samostalno istraživanje i optimizaciju strategija upravljanja, što ga čini idealnim za autonomna plovila. Korištenjem funkcija vrijednosti i MDP modela, agent može učiti optimalnu politiku bez prethodnog znanja o dinamičkom okruženju. Ovi koncepti temelj su naprednih RL algoritama poput PPO (Proximal Policy Optimization), koji se često koriste za navigaciju autonomnih sustava.

Gradijenti politike

U učenju putem pojačanja (RL), metode mogu biti temeljene na vrijednosti (value-based) ili temeljene na politici (policy-based). Dok se metode temeljene na vrijednosti oslanjaju na procjenu funkcije vrijednosti stanja i zatim iz nje izvode optimalnu politiku, metode temeljene na politici izravno optimiziraju politiku. Za probleme s kontinuiranim ili visokodimenzionalnim prostorom akcija, metode temeljene na politici često su učinkovitije jer izbjegavaju potrebu za diskretizacijom prostora akcija. Umjesto da agent mora odabrati akciju iz konačnog skupa opcija, on može naučiti distribuciju optimalnih akcija.

Optimizacija politike pomoću gradijenata

U ovom pristupu politika $\pi(\theta)$ je stohastička, što znači da opisuje vjerojatnost izvođenja određene akcije a u danom stanju s :

$$\pi(\theta) : S \times A \rightarrow [0,1]$$

Ova politika je definirana kao diferencijalna funkcija s parametrima θ , što omogućava njezinu optimizaciju pomoću gradijentnih metoda. Cilj je pronaći optimalne parametre θ tako da se maksimizira ukupna očekivana nagrada, odnosno funkcija cilja:

$$J(\theta) = E_{s_i, a_i \sim \pi(\theta)} [R_0]$$

Metode gradijenta politike koriste gradijentni uspon kako bi iterativno ažurirale parametre politike:

$$\theta_{t+1} = \theta_t + \alpha \nabla_{\theta} J(\theta)$$

gdje je $\nabla_{\theta} J(\theta)$ procjena gradijenta funkcije cilja, a α je faktor učenja.

Teorem gradijenta politike

Jedan od ključnih rezultata RL teorije je teorem gradijenta politike, koji pokazuje da se gradijent funkcije cilja može zapisati kao:

$$\nabla_{\theta} J(\theta) \propto \sum_s \mu(s) \sum_a \nabla_{\theta} \pi(a | s) Q^{\pi}(s, a)$$

gdje je:

- $\mu(s)$ – distribucija stanja pod politikom π ,

- $Q^\pi(s, a)$ – očekivana ukupna nagrada ako agent u stanju s izvede akciju a i zatim slijedi politiku π .

Ono što je važno primijetiti jest da izraz ne sadrži derivaciju distribucije stanja $\nabla_{\theta}\mu(s)$, što znači da možemo procijeniti gradijent uzorkovanjem iz politike, bez potrebe za eksplicitnim modeliranjem promjena distribucije stanja.

Funkcija prednosti

Kako bi se smanjila varijabilnost u procjeni gradijenta, često se koristi funkcija prednosti:

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$$

Ova funkcija mjeri koliko je određena akcija bolja od prosječne akcije u tom stanju. Umjesto $Q^\pi(s, a)$, često koristimo $A^\pi(s, a)$, čime poboljšavamo stabilnost i efikasnost učenja.

Tada procjena gradijenta postaje:

$$\nabla_{\theta}J(\theta) \propto E_{\pi}[\nabla_{\theta} \log \pi (A_t | S_t) A^\pi(S_t, A_t)]$$

Empirijska procjena gradijenta politike

U praksi, gradijent se procjenjuje iz više epizoda interakcije agenta s okruženjem. Ako agent izvrši N epizoda, možemo aproksimirati gradijent kao:

$$\nabla_{\theta}J(\theta) = \frac{1}{N} \sum_{n=1}^N \sum_{t=0}^{\infty} A_t^n \nabla_{\theta} \log \pi (a_t^n | s_t^n)$$

gdje su:

- A_t^n – procjena funkcije prednosti u trenutku t za n -tu epizodu,
- $\log \pi (a_t^n | s_t^n)$ – logaritamska vjerojatnost akcije.

Ova tehnika koristi politikom vođeno uzorkovanje, pri čemu agent prikuplja podatke iz vlastitog iskustva i koristi ih za poboljšanje strategije.

Procjena funkcije prednosti

Budući da su i funkcija vrijednosti akcije $Q^\pi(s, a)$ i funkcija vrijednosti stanja $V^\pi(s)$ općenito nepoznate, proizlazi da je i funkcija prednosti $A^\pi(s, a)$ nepoznata. Stoga se u praksi koristi njezina procijenjena vrijednost $A^\pi(s, a)$.

Razvijene su različite metode za procjenu funkcije prednosti, a jedna od najčešće korištenih je generalizirana procjena prednosti (GAE). Ova metoda koristi diskontirane vremenske razlike funkcije vrijednosti stanja kao osnovne gradivne elemente.

Diskontirani TD reziduali i osnovna procjena funkcije prednosti

Diskontirani TD reziduali predstavljaju osnovnu aproksimaciju prednosti i definiraju se izrazom:

$$\delta_t^V = r_t + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t)$$

gdje je \hat{V} aproksimacija funkcije vrijednosti stanja. Kada bi se koristila točna funkcija vrijednosti V^π , ova procjena bila bi nepristrana. Međutim, u praksi aproksimacija rijetko odgovara stvarnoj funkciji vrijednosti, pa je potrebno primijeniti metode koje smanjuju pristranost procjene. Jedan od načina poboljšanja točnosti procjene prednosti je gledanje više vremenskih koraka unaprijed kako bi se smanjio utjecaj pojedinačnih pogrešnih procjena:

$$\widehat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V$$

gdje veći k omogućuje uzimanje u obzir budućih nagrada, ali istovremeno povećava varijabilnost procjene.

Generalizirana procjena prednosti

Umjesto odabira fiksnog broja koraka unaprijed k , metoda GAE koristi eksponencijalno ponderiranu sumu TD reziduala, pri čemu se faktor težine λ koristi za kontrolu balansa između varijance i pristranosti:

$$\widehat{A}_t^{\text{GAE}(\gamma, \lambda)} = \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{t+l}^V$$

gdje je:

- Mali λ – smanjuje varijancu jer su procjene prednosti uglavnom temeljene na trenutnim TD rezidualima.
- Veliki λ – smanjuje pristranost jer uzima u obzir više budućih nagrada.

Vrijednost λ obično se postavlja između 0.9 i 0.99 kako bi se postigao optimalan kompromis između stabilnosti i točnosti.

Procjena funkcije vrijednosti pomoću dubokih neuronskih mreža

Kako bi se poboljšala preciznost procjene vrijednosti stanja $V^\pi(s)$, često se koristi duboka neuronska mreža (DNN - Deep Neural Network). Ova mreža se trenira na temelju diskontiranih empirijskih povrata koristeći funkciju gubitka:

$$L_t^{\text{VF}}(\theta) = E_t \left[\left(\widehat{V}_\theta(s_t) - \sum_{i=t}^{\infty} \gamma^{i-t} r(s_i, a_i) \right)^2 \right]$$

gdje:

- $\widehat{V}_\theta(s_t)$ je aproksimacija vrijednosti stanja na temelju mreže s parametrima θ ,
- Empirijska očekivanja se procjenjuju na temelju uzorka epizoda iz okruženja.

Primjena neuronskih mreža omogućuje fleksibilnu i preciznu procjenu vrijednosti stanja, čime se poboljšava stabilnost RL algoritama poput Proximal Policy Optimization (PPO).

2.2. Implementacija

PPO se implementira kroz iterativni proces ažuriranja politike. Za svaku iteraciju $i = 1, 2, n$ potrebno je:

- Pokrenuti N istovremenih agenata
- Svaki agent izvodi trenutnu politiku π_θ kroz T vremenskih koraka
- Izračunati procjene funkcije prednosti
- Ažurirati politiku koristeći stohastički gradijent uspon (SGD) nad mini-batch uzorcima
- Ponoviti SGD ažuriranja kroz N epoha

PPO pruža ravnotežu između jednostavnosti implementacije i efikasnosti uzorkovanja, čime se osigurava stabilna konvergencija bez potrebe za opsežnim podešavanjem hiperparametara. Ograničava prevelike promjene u politici, čime sprječava degradaciju performansi te omogućava višestruku ponovnu upotrebu podataka prije ažuriranja politike.

2.2.1. Alati i biblioteke

Za implementaciju koda korišten je OpenAI Gym, popularni Python okvir koji omogućava standardizaciju simulacijskih okruženja za učenje putem pojačanja (RL). OpenAI Gym pruža intuitivno sučelje za razvoj i testiranje RL agenata u različitim scenarijima uz minimalan kodni overhead, što ga čini idealnim za istraživanja i primjene u stvarnom vremenu. Pored OpenAI Gym-a, koristi se Stable Baselines, još jedan Python paket koji nudi skup naprednih RL algoritama kompatibilnih s OpenAI Gym frameworkom. Stable Baselines sadrži unaprijeđene verzije algoritama iz OpenAI Baselines sa standardizacijom algoritma i paralelizacijom učenja. Stable Baselines uključuje Proximal Policy Optimization (PPO) i omogućuje bržu iteraciju modela te lako eksperimentiranje s različitim hiperparametrima, čime se ubrzava proces treniranja i evaluacije RL agenata.

2.2.2. Okruženje

Eksperimentalno okruženje dizajnirano za treniranje agenta predstavlja oceansku površinu s preprekama. U ovom scenariju, plovilo se mora pridržavati unaprijed definirane putanje dok izbjegava sudare s nasumično generiranim preprekama. Okruženja su generirana tako da predstavljaju širok spektar scenarija kako bi postigli generalizaciju agenta i izbjegavanje pretreniranosti modela. Iz tog razloga koriste se i nasumično raspoređene prepreke različite veličine i oblika. Dinamički model plovila smatra se dijelom okruženja jer nije pod izravnom kontrolom agenta. Ako su prepreke uvijek smještene blizu planirane putanje, agent može razviti pretjerano konzervativno ponašanje i nepotrebno izbjegavati prepreke. S druge strane, ako su prepreke rijetko raspoređene, agent možda neće naučiti kako se nositi s visokom gustoćom prepreka. Zbog toga se koristi nasumično generirani tereni kako bi agent naučio prilagodljivo donošenje odluka.

2.2.3. Agent

U kontekstu učenja putem pojačanja (RL), agent se može smatrati sustavom za upravljanje plovilom, budući da se njegovo djelovanje svodi na generiranje upravljačkih signala za aktuatorski sustav plovila. Model koristi dvije neuronske mreže:

- Akcijska mreža – generira upravljačke signale na temelju trenutnog stanja okruženja
- Mreža vrijednosti – procjenjuje funkciju vrijednosti stanja što omogućava izračunavanje prednosti akcija (GAE)

Agent za praćenje putanje mora znati trenutni položaj i orijentaciju plovila u odnosu na zadanu putanju. Za izbjegavanje prepreka koristi se sustav senzora udaljenosti temeljen na lidar, radaru ili sonaru. Simulirani model koristi 225 senzora koji omogućuju široko vidno polje. Kako bi se optimizirala obrada podataka senzora implementirane su min pooling, max pooling i feasibility pooling. Min pooling koristi najmanju izmjerenu udaljenost u sektoru dok max pooling najveću. Metoda feasibility pooling, pokazana kao najstabilnija, izračunava maksimalnu udaljenost unutar koje postoji dovoljno širok prolaz za plovilo.

2.2.4. Nagradna funkcija

Svaki RL agent motiviran je maksimizacijom ukupne nagrade. Idealno, agent bi trebao primiti nagradu tek na kraju epizode, kada dosegne cilj ili doživi sudar. Međutim, takva rijetka nagrada značajno otežava proces učenja, jer agent nema jasne smjernice tijekom navigacije. Zbog toga se koristi kontinuirana nagradna funkcija koja pruža konstantan signal o kvaliteti trenutnih odluka. U ovom radu, s obzirom na dvostruki cilj navigacije (praćenje putanje i izbjegavanje prepreka), agent prima nagradu za praćenje putanje i nagradu za izbjegavanje prepreka. Ukupna nagrada definirana je pomoću težinskog faktora λ koji omogućava prilagodbu prioriteta između ova dva cilja.

$$r(t) = \lambda r_{pf}(t) + (1 - \lambda)r_{oa}(t)$$

Težinski faktor $\lambda \in [0, 1]$ parametar koji balansira između preciznog praćenja putanje ($\lambda \rightarrow 1$) i agresivnog izbjegavanja prepreka ($\lambda \rightarrow 0$).

Agent treba biti nagrađen za smanjenje transverzalne pogreške $e(t)$ u odnosu na zadanu putanju. U nekim radovima koristi se Gaussova funkcija nagrade, no umjesto toga ovdje je odabrana eksponencijalna funkcija jer pruža bolje performanse:

$$r_{pf}(t) = -1 + \left[\frac{\sqrt{u^2 + v^2}}{U_{max}} \cos \tilde{\chi}(t) + 1 \right] [e^{-\gamma_e |e(t)|} + 1]$$

gdje su:

- U_{max} – maksimalna brzina plovila,
- $\tilde{\chi}(t)$ – pogreška kursa,
- γ_e – faktor koji kontrolira osjetljivost na pogrešku.

Ako je pogreška velika ili je brzina u smjeru okomito na putanju, agent dobiva nagradu nula, dok ako se kreće u smjeru putanje i održava nizak $e(t)$, nagrada raste.

Kako bi se spriječili sudari, agent je kažnjen za blizinu prepreka. Koriste se senzorska očitavanja udaljenosti kako bi se dodijelile kazne, pri čemu se težina kazne povećava što je plovilo bliže prepreci. Model nagrade uzima u obzir kutnu orijentaciju prepreka i definira se kao:

$$r_{oa}(t) = -\frac{\sum_{i=1}^N (1 + |\gamma_\theta \theta_i|)^{-1} (\gamma_x \max(x_i, \epsilon_x)^2)^{-1}}{\sum_{i=1}^N (1 + |\gamma_\theta \theta_i|)^{-1}}$$

gdje su:

- θ_i – kut pod kojim se nalazi prepreka u odnosu na smjer plovila (prepreke ispred plovila su važnije od onih iza),
- x_i – udaljenost do prepreke izmjerena sensorima,
- γ_θ – faktor koji povećava težinu prepreka u vidnom polju plovila,
- γ_x – faktor koji prilagođava kaznu prema udaljenosti.

Funkcijom je osigurano da agent bude strogo kažnjen za bliske prepreke.

Kako bismo spriječili da agent ostane stajati na sigurnom mjestu (što bi mu omogućilo izbjegavanje kazni), uvodi se konstantna kazna za mirovanje:

$$r_{exists} = -\lambda(2\alpha_r + 1)$$

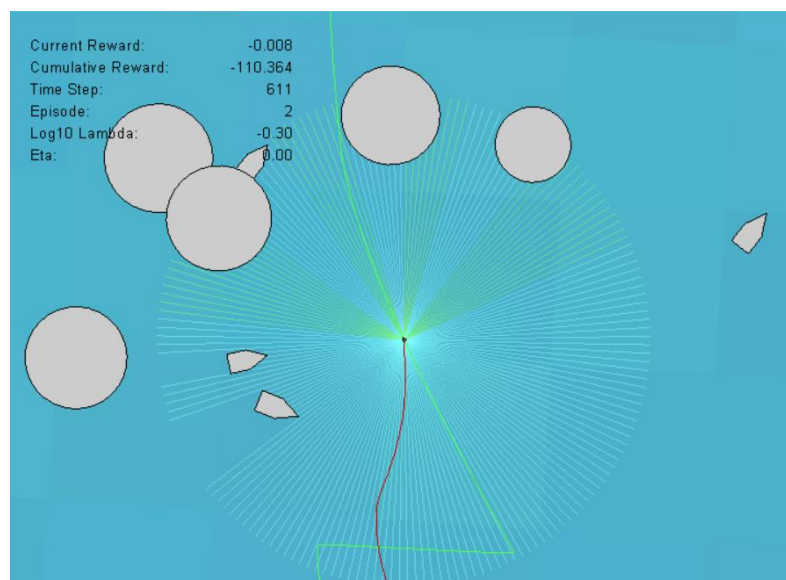
2.3. Simulacijski eksperiment

2.3.1. Trening i evaluacija

Agent se trenira koristeći Proximal Policy Optimization (PPO), s postavkama hiperparametara:

- Diskontni faktor $\gamma = 0.999$
- Veličina batch-a $T = 1024$
- Broj istovremenih agenata $N_A = 8$
- Brzina učenja $\eta = 0.0002$
- Veličina mini-batch-a $N_{MB} = 32$
- Broj epoha N_E
- Koeficijent klipinga $\epsilon = 0.2$

Neuronske mreže za politiku i vrijednost imaju dvije potpuno povezane (fully connected) skrivenih slojeva s 64 neurona i aktivacijsku funkciju $\tanh()$. Resetiranje okruženja događa se kada plovilo ili dođe do cilja ili se sudari s preprekom ili dosegne kumulativnu negativnu nagradu ispod -5000.



Slika 7 – Primjer izvođenja DRL metode

Tijekom testiranja agenta, okruženje sadrži i statičke i dinamičke objekte kako bi se osigurala sigurnost navigacijskih odluka. Statičke prepreke su fiksno postavljene u sceni, dok se dinamički objekti kreću prema unaprijed definiranim ili stohastičkim pravilima, simulirajući realne uvjete pomorske navigacije. Agent opaža i reagira na obje vrste prepreka, prilagođavajući svoju putanju u skladu s njihovim položajem i kretanjem.

3. Rezultati i diskusija

Ovaj odjeljak prikazuje rezultate testiranja kroz niz scenarija koji simuliraju različite uvjete navigacije autonomnog plovila. Rezultati su prikazani u odgovarajućim grafovima, gdje su analizirane putanje plovila u odnosu na virtualne ciljeve i prepreke, udaljenosti te promjene brzine i orijentacije glavnog broda.

Prikazani su rezultati za 14 različitih scenarija, koji uključuju razne vrste susreta, prolazaka i pretjecanja:

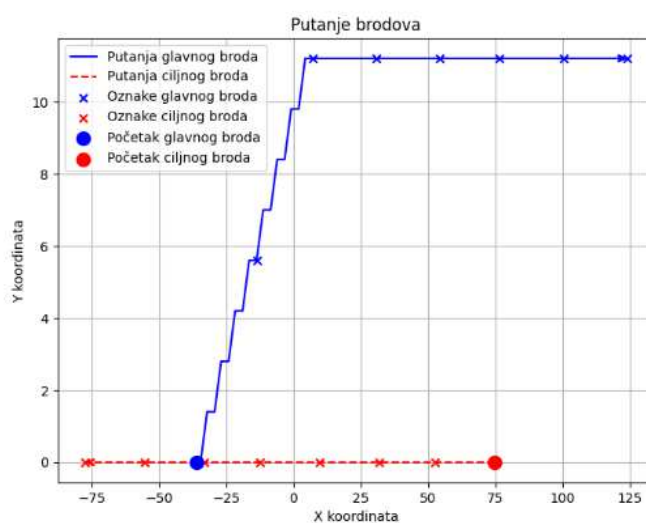
- **Scenarij 1** – Frontalni susret, gdje plovilo mora donijeti odluku o izbjegavanju plovila koje se kreće ususret njemu.
- **Scenariji 2 i 3** – Virtualni cilj pod relativnim kutom od 90° , gdje plovilo prolazi s prednje ili stražnje strane prepreke.
- **Scenariji 4 i 5** – Virtualni cilj pod kutom od 270° , s različitim strategijama prolaska (prednja ili stražnja strana).
- **Scenarij 6** – Pretjecanje, gdje plovilo mora odlučiti kako sigurno zaobići virtualni cilj.
- **Scenariji 7 i 8** – Virtualni cilj pod kutom od 135° , uključujući stražnji zaobilazak.
- **Scenariji 9 i 10** – Virtualni cilj pod kutom od 45° i 315° , simulirajući situacije bočnih susreta.
- **Scenariji 11 i 12** – Napredni slučajevi s dvama virtualnim ciljevima, uključujući frontalni susret s višestrukim opcijama navigacije.
- **Scenariji 13 i 14** – Kombinacija frontalnog susreta i dodatnih ciljeva pod kutovima od 90° , 260° i 315° , što simulira složenije navigacijske odluke.

Rezultati su prikazani grafički, gdje se vidi putanja plovila, položaj virtualnih ciljeva i prepreka te za metodu temeljenu na dubokom podržanom učenju, odluke koje agent donosi u stvarnom vremenu. Svaka slika predstavlja pojedini scenarij, omogućujući usporedbu različitih strategija navigacije. Nakon prikazanih rezultata slijedi diskusija, gdje ćemo

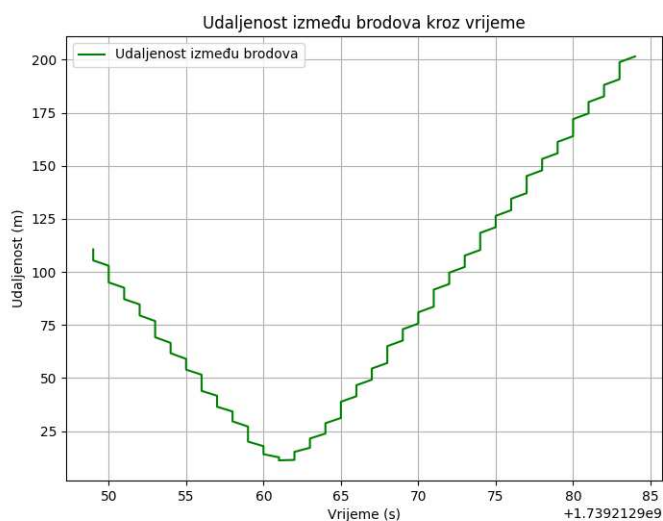
analizirati obrasce ponašanja VO metode i agenta te usporediti različite scenarije kako bismo razumjeli utjecaj postavljenih parametara na konačne odluke navigacije.

3.1. Rezultati VO metode

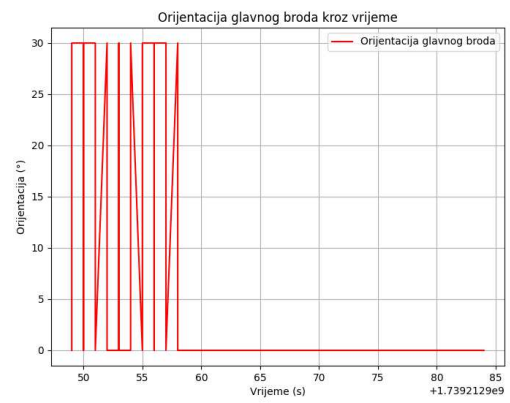
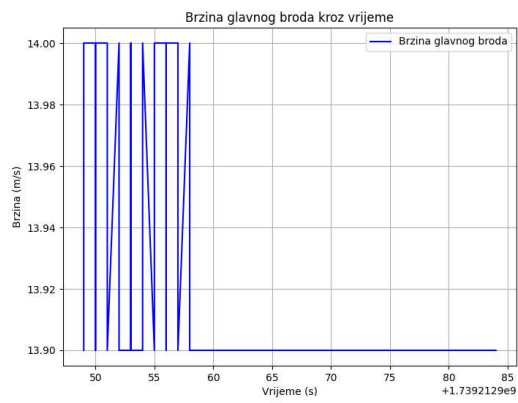
Scenarij 1 – Frontalni susret



Slika 8 – Scenarij 1: graf putanje

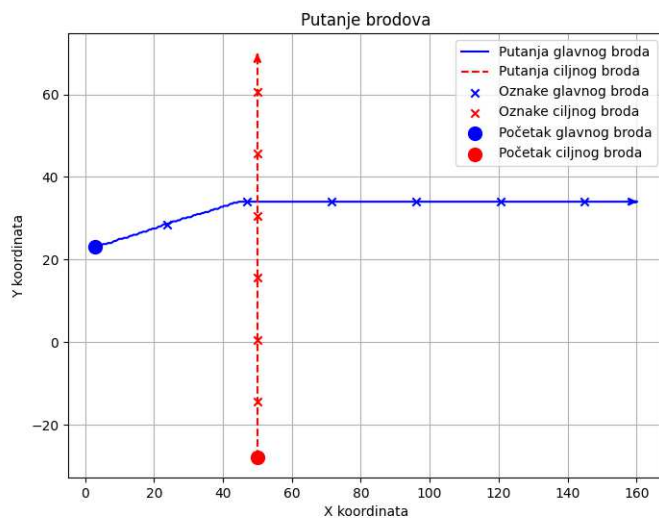


Slika 9 – Scenarij 1: graf udaljenosti

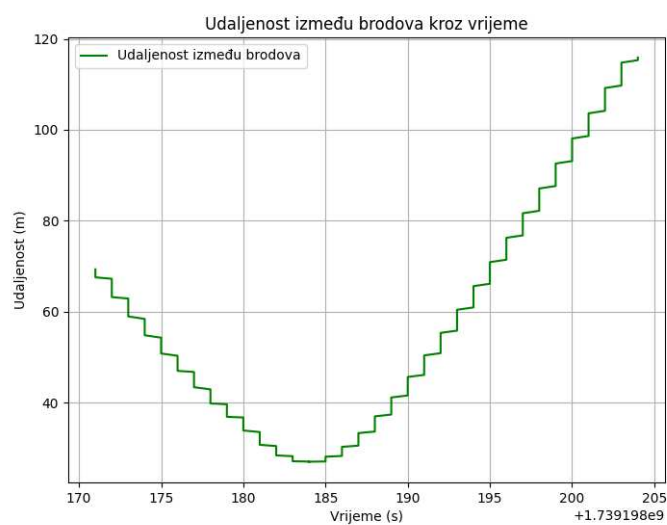


Slika 10 – Scenarij 1: grafovi brzine i orijentacije

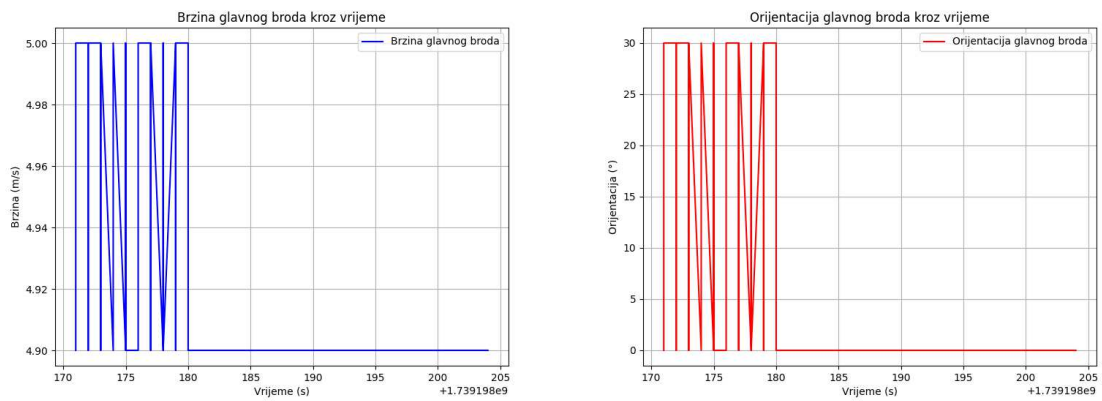
Scenarij 2 – Virtualni cilj relativnog kuta orijentacije 90 stupnjeva (prolaz s prednje strane)



Slika 11 – Scenarij 2: graf putanje

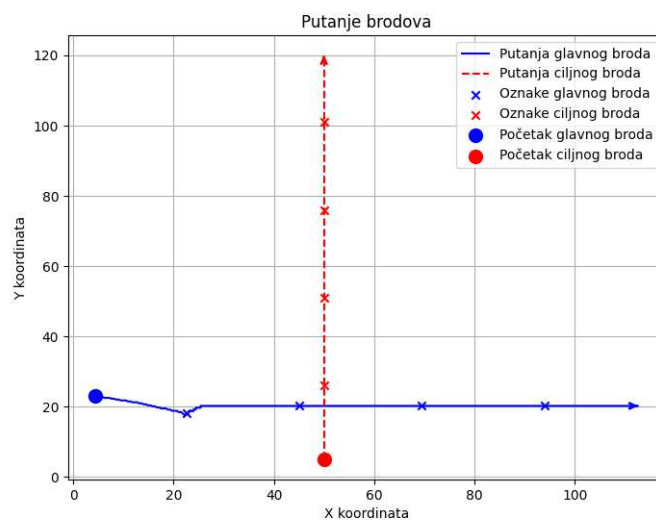


Slika 12 – Scenarij 2: graf udaljenosti

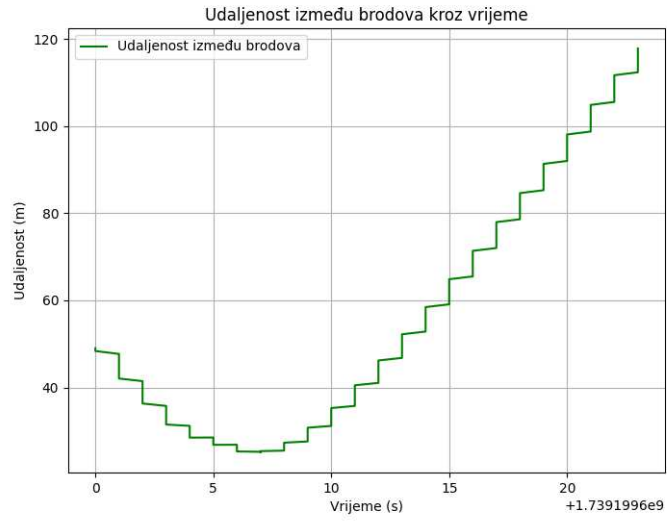


Slika 13 – Scenarij 2: grafovi brzine i orijentacije

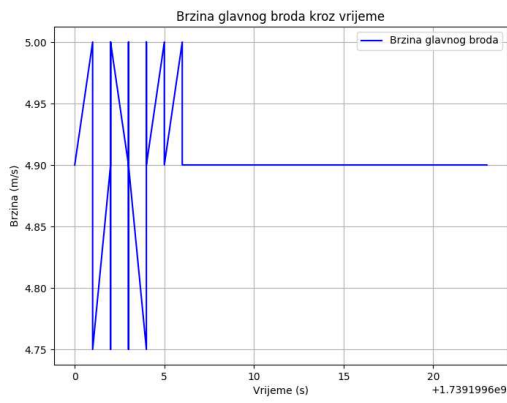
Scenarij 3 – Virtualni cilj relativnog kuta orijentacije 90 stupnjeva (prolaz sa stražnje strane)



Slika 14 – Scenarij 3: graf putanje

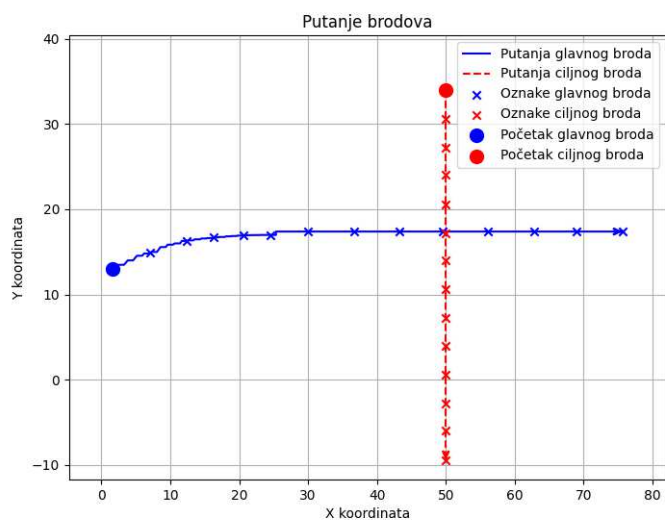


Slika 15 - Scenarij 3: graf udaljenosti

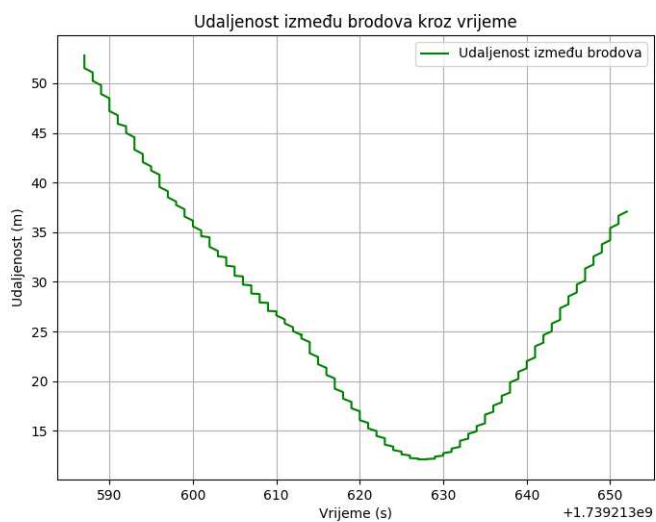


Slika 16 – Scenarij 3: grafovi brzine i orijentacije

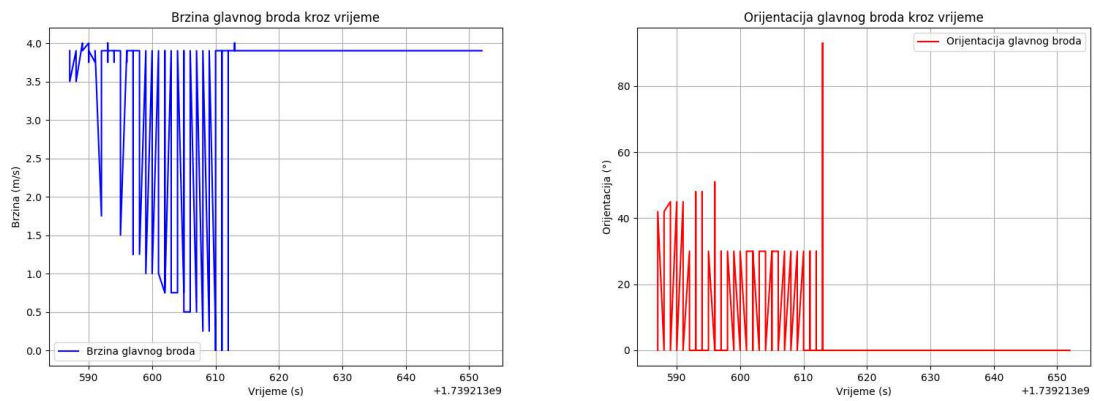
Scenarij 4 – Virtualni cilj relativnog kuta orijentacije 270 stupnjeva



Slika 17 – Scenarij 4: graf putanje

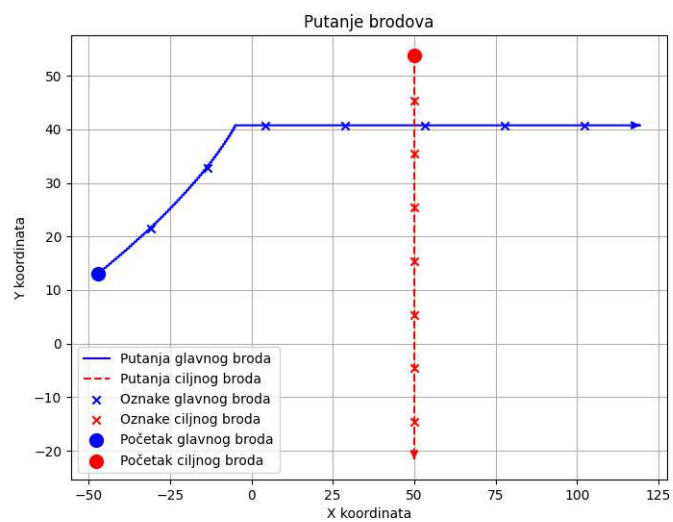


Slika 18 – Scenarij 4: graf udaljenosti

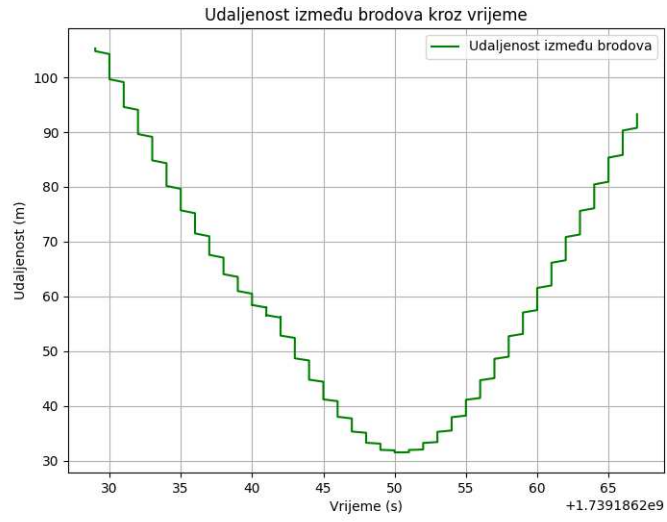


Slika 19 – Scenarij 4: grafovi brzine i orijentacije

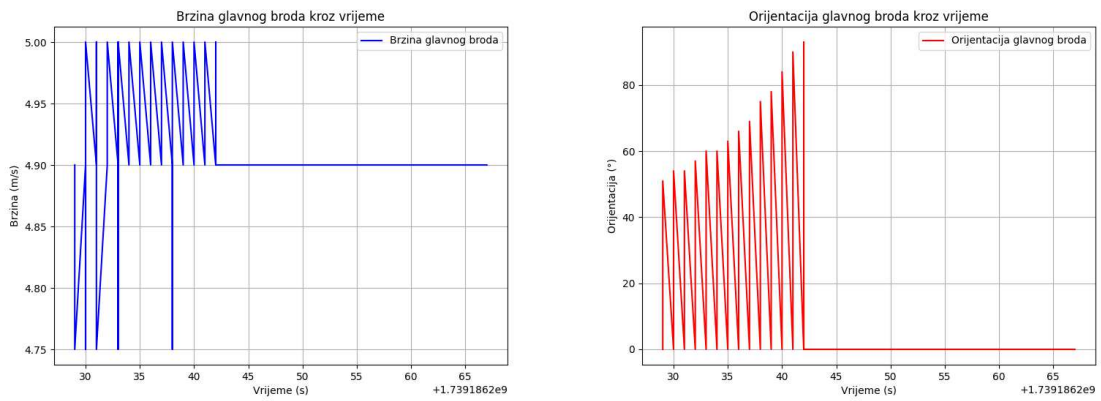
Scenarij 5 – Virtualni cilj relativnog kuta orijentacije 270 stupnjeva (prolazak sa stražnje strane)



Slika 20 – Scenarij 5: graf putanje

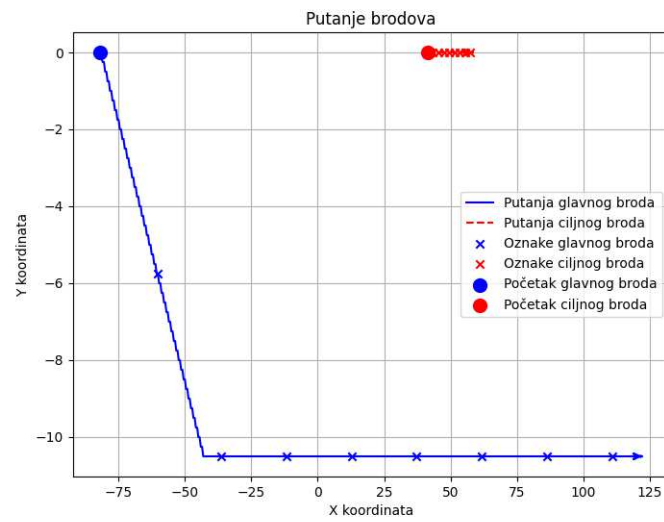


Slika 21 – Scenarij 5: graf udaljenosti

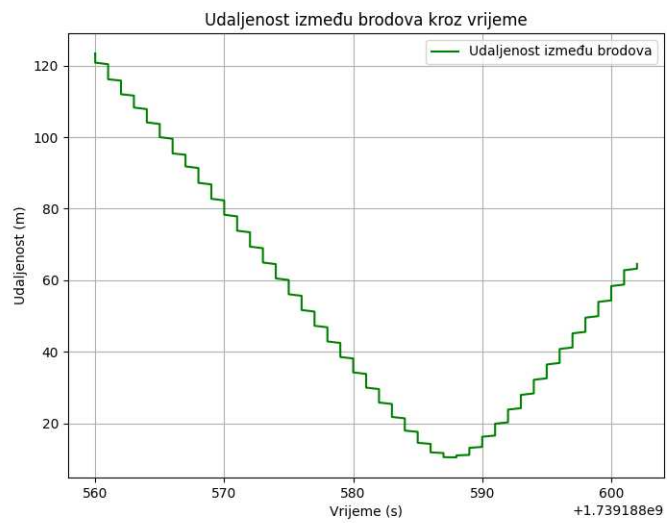


Slika 22 – Scenarij 5: grafovi brzine i orijentacije

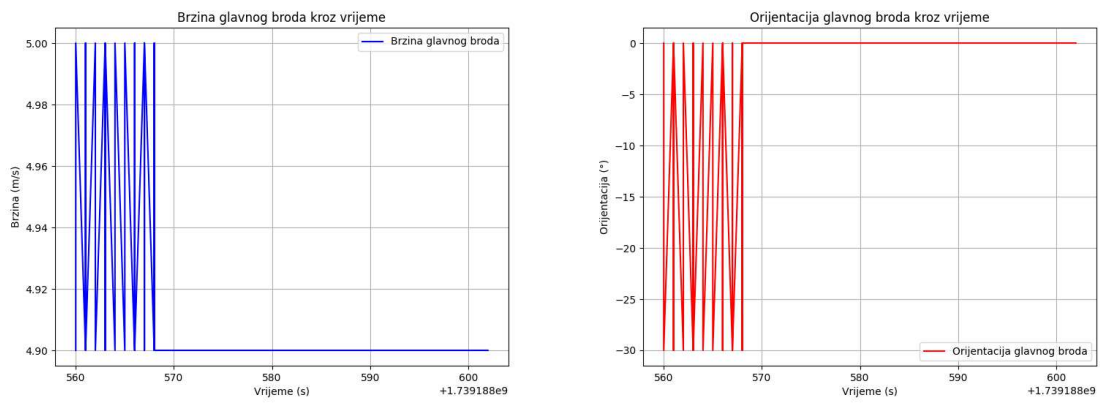
Scenarij 6 – Pretjecanje



Slika 23 – Scenarij 6: graf putanje

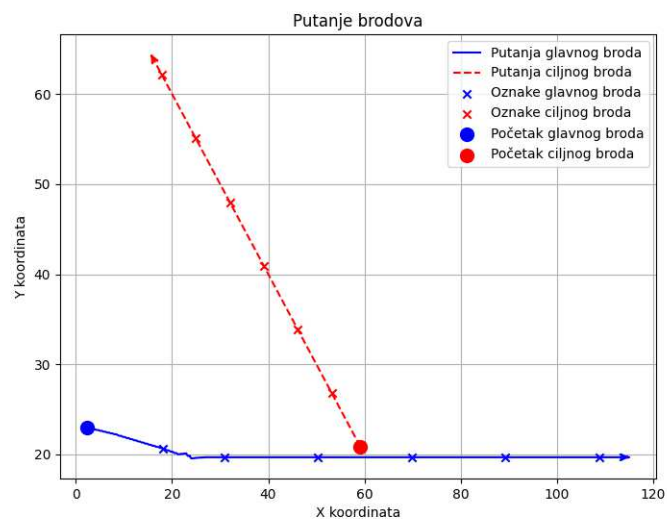


Slika 24 – Scenarij 6: graf udaljenosti

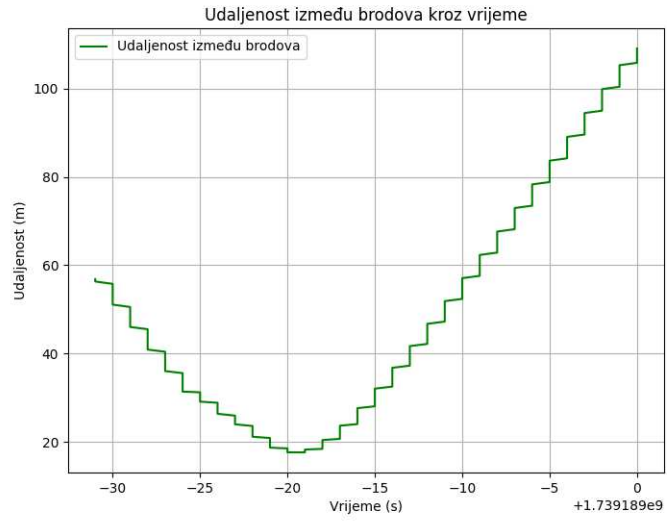


Slika 25 – Scenarij 6: grafovi brzine i orijentacije

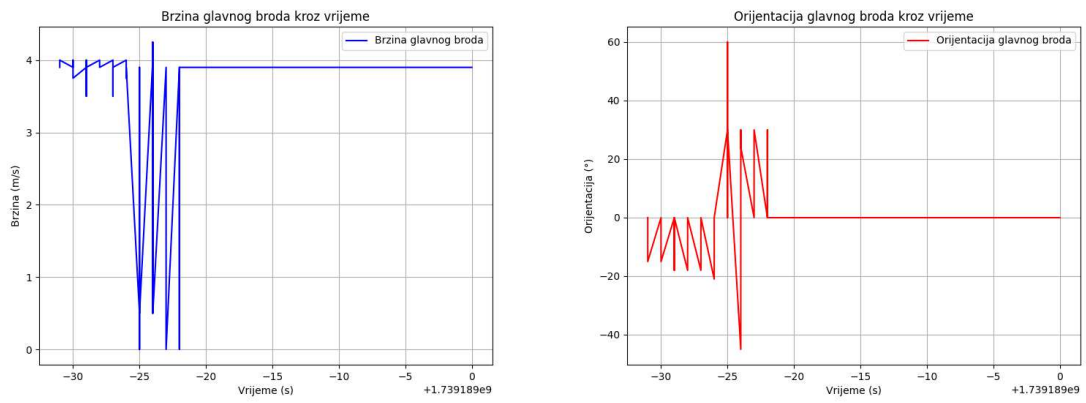
Scenarij 7 – Virtualni cilj relativnog kuta orijentacije 135 stupnjeva



Slika 26 – Scenarij 7: graf putanje

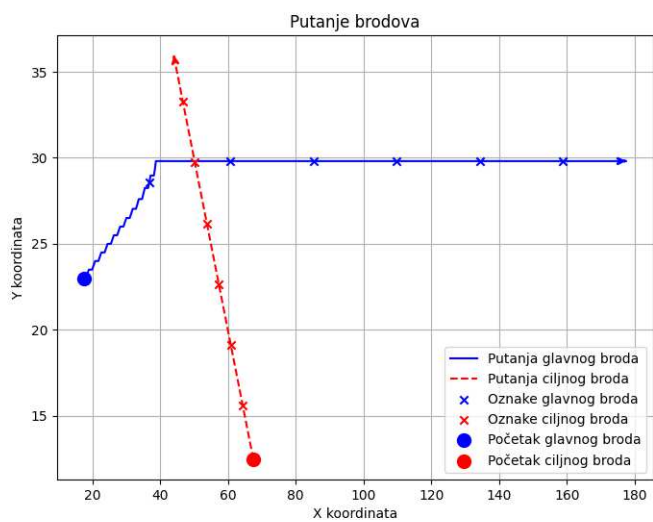


Slika 27 – Scenarij 7: graf udaljenosti

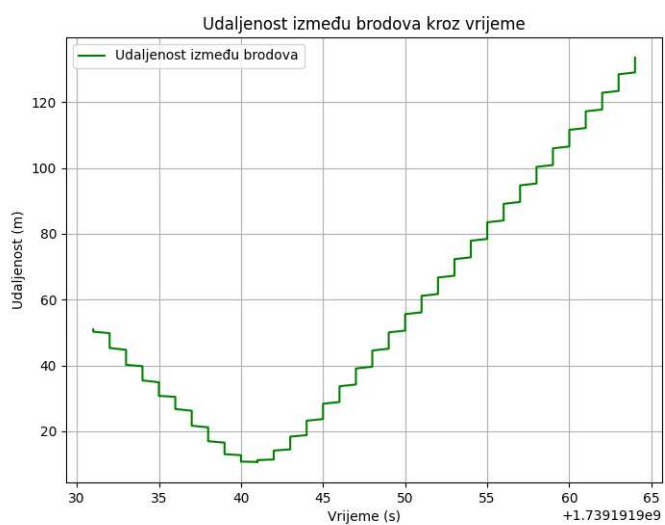


Slika 28 – Scenarij 7: grafovi brzine i orijentacije

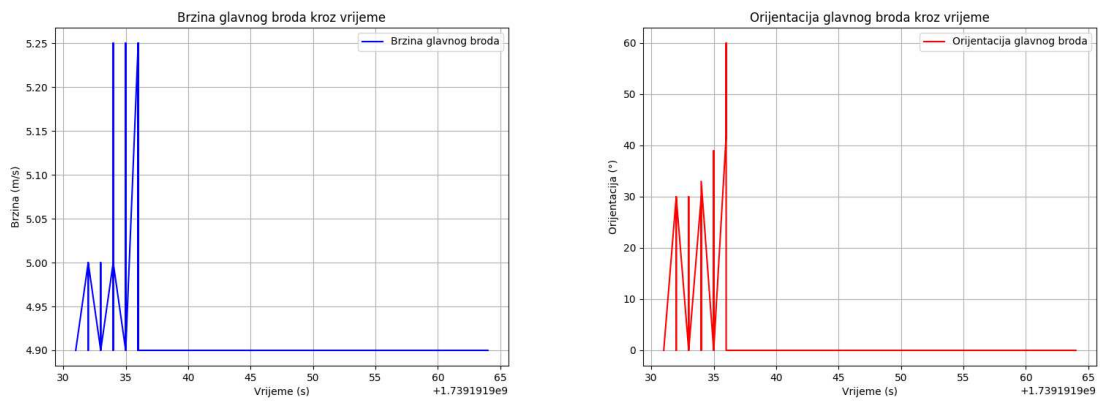
Scenarij 8 – Virtualni cilj relativnog kuta orijentacije 135 stupnjeva (prednji zaobilazak)



Slika 29 – Scenarij 8: graf putanje

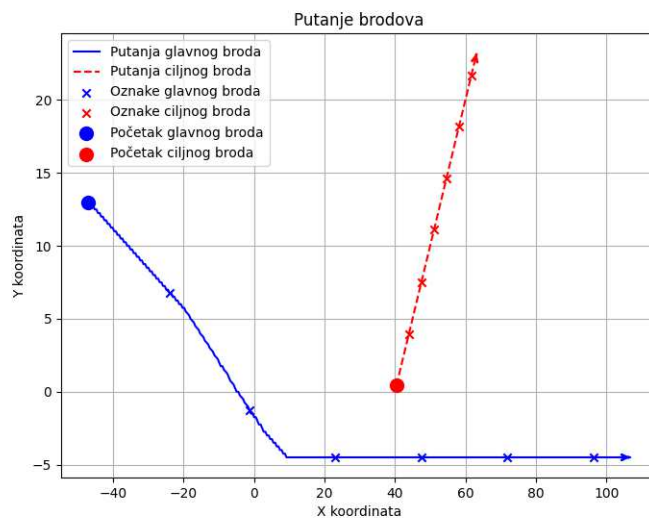


Slika 30 – Scenarij 8: graf udaljenosti

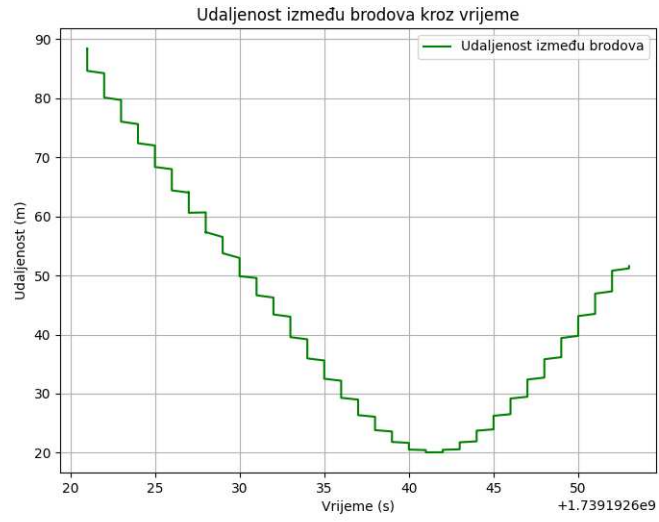


Slika 31 – Scenarij 8: grafovi brzine i orijentacije

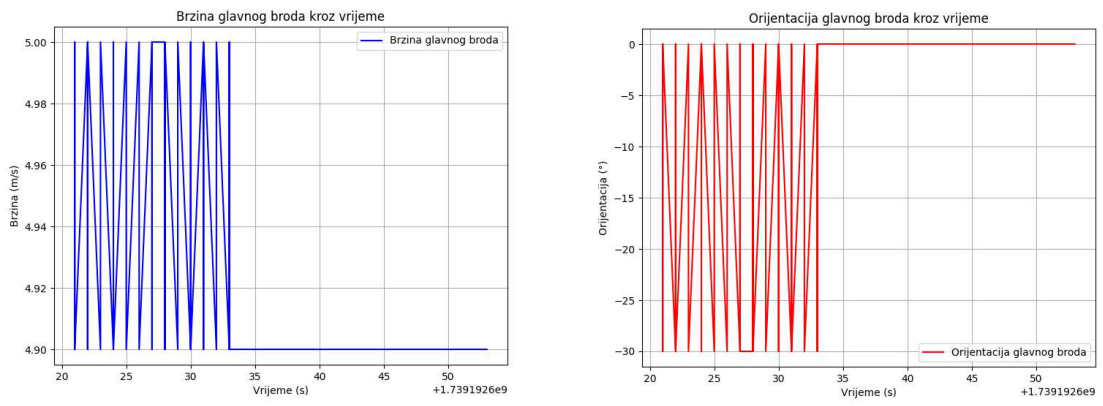
Scenarij 9 – Virtualni cilj relativnog kuta orijentacije 45 stupnjeva



Slika 32 – Scenarij 9: graf putanje

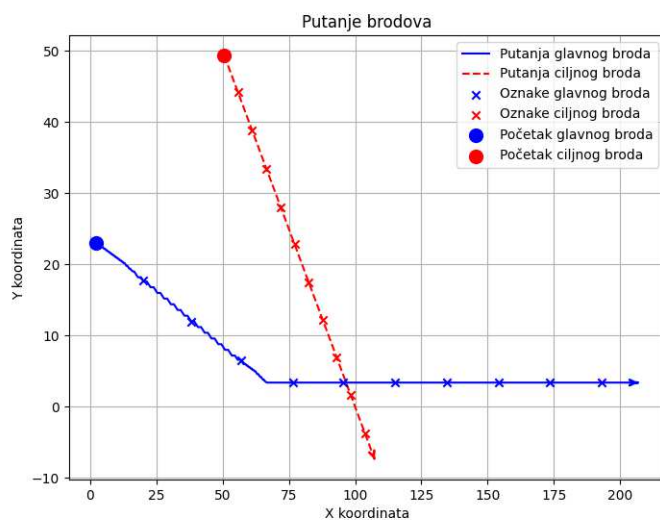


Slika 33 – Scenarij 9: graf udaljenosti

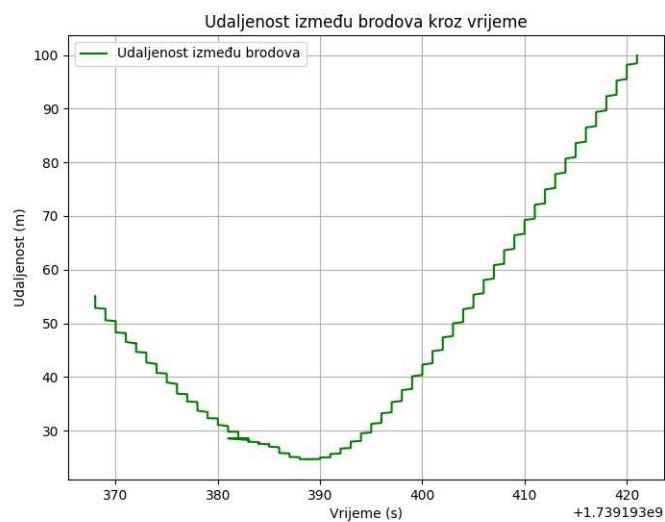


Slika 34 – Scenarij 9: grafovi brzine i orijentacije

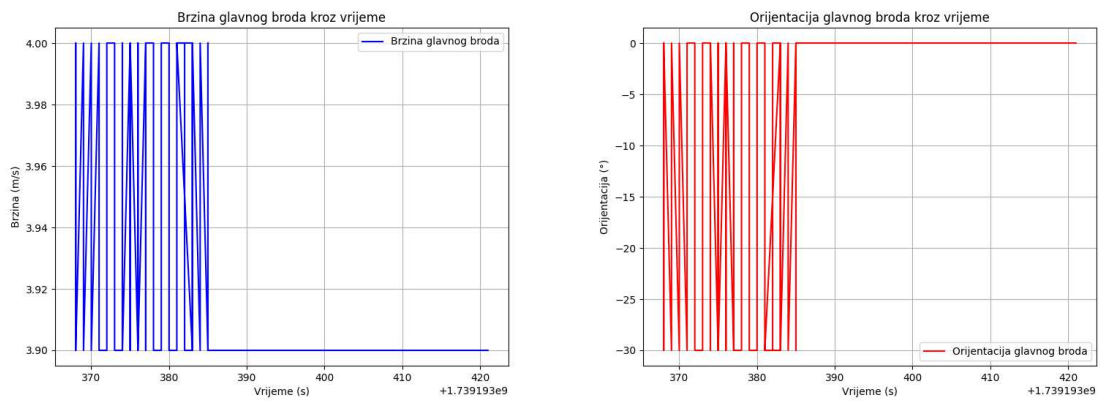
Scenarij 10 – Virtualni cilj relativnog kuta orijentacije 315 stupnjeva



Slika 35 – Scenarij 10: graf putanje

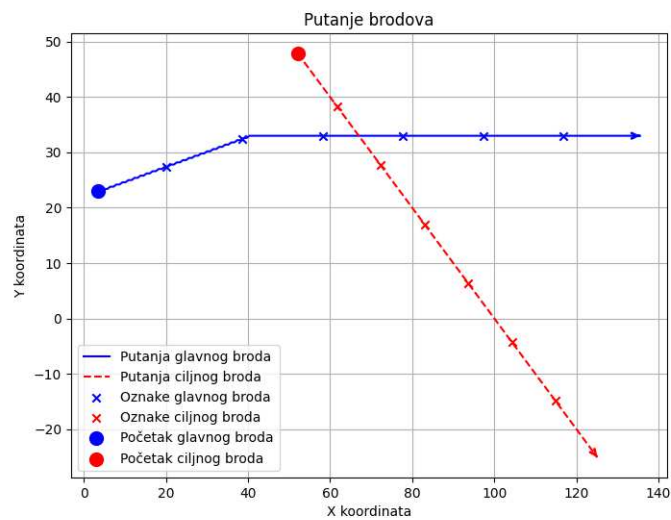


Slika 36 – Scenarij 10: graf udaljenosti

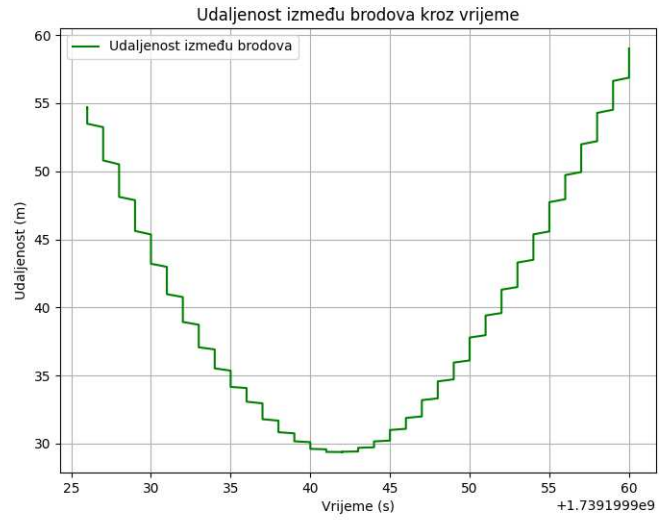


Slika 37 – Scenarij 10: grafovi brzine i orijentacije

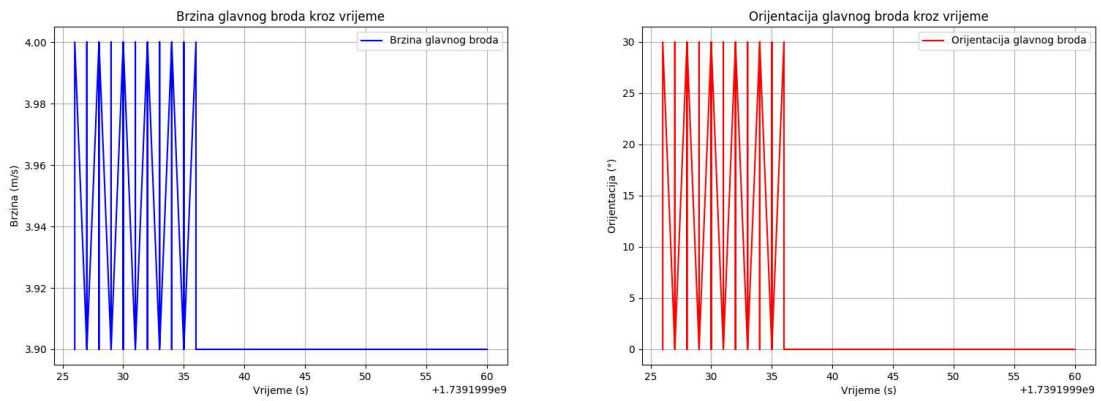
Scenarij 11 – Virtualni cilj relativnog kuta orijentacije 315 stupnjeva (lijevi zaobilazak)



Slika 38 – Scenarij 11: graf putanje

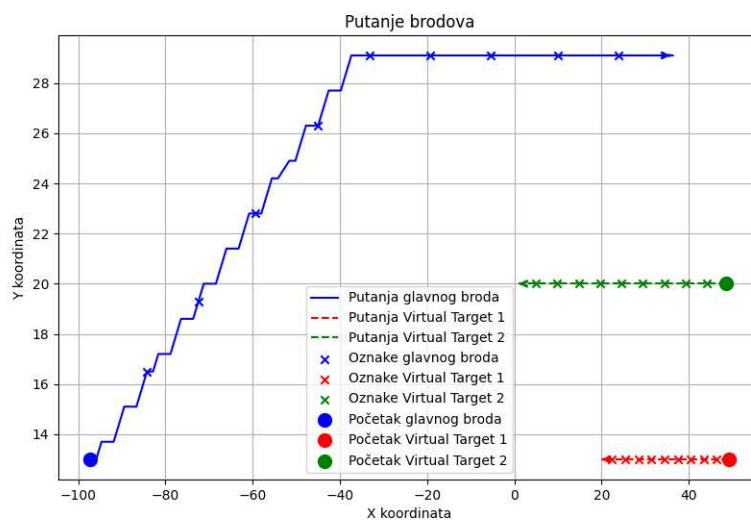


Slika 39 – Scenarij 11: graf udaljenosti

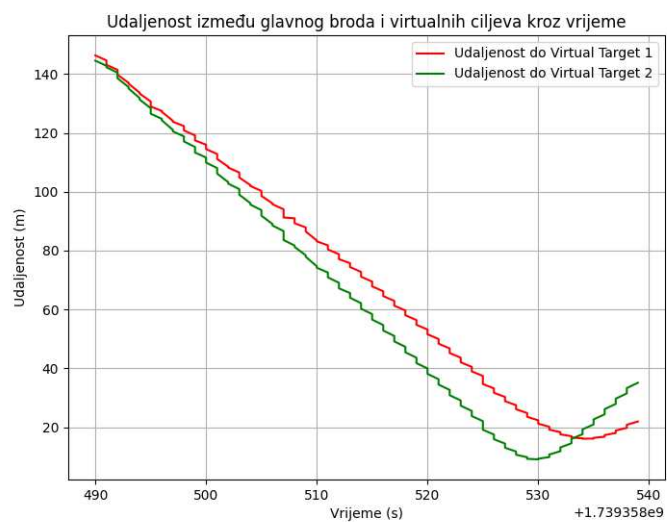


Slika 40 – Scenarij 11: grafovi brzine i orijentacije

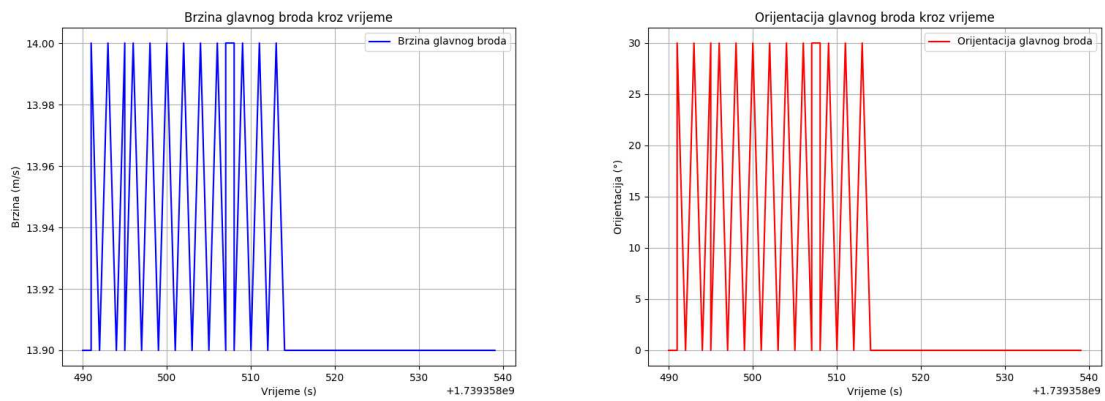
Scenarij 12 – Frontalni susret s dva virtualna cilja



Slika 41 – Scenarij 12: graf putanje

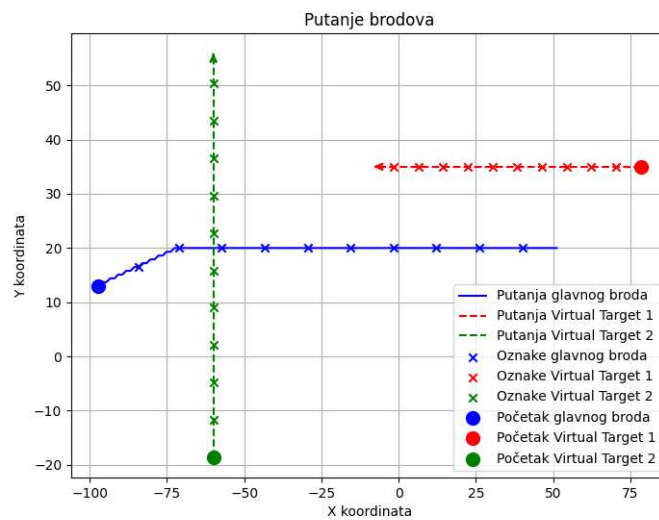


Slika 42 – Scenarij 12: graf udaljenosti

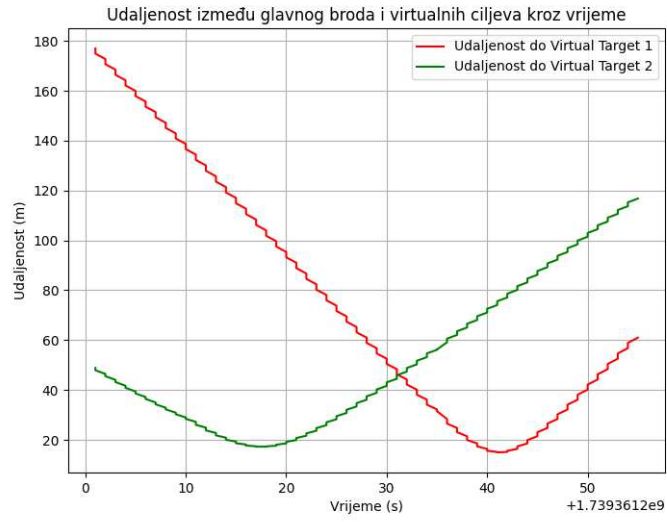


Slika 43 – Scenarij 12: grafovi brzine i orijentacije

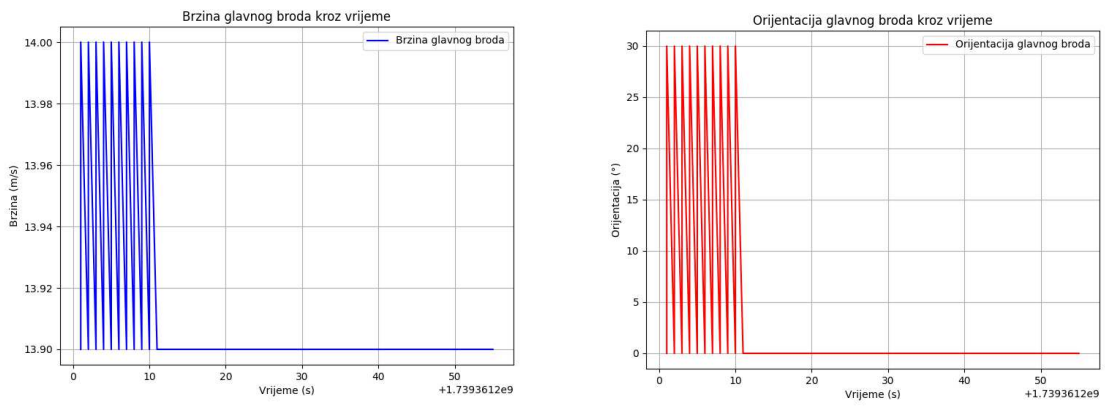
Scenarij 13 – Frontalni susret s jednim virtualnim ciljem, drugi virtualni cilj relativnog kuta orijentacije 90 stupnjeva



Slika 44 – Scenarij 13: graf putanje

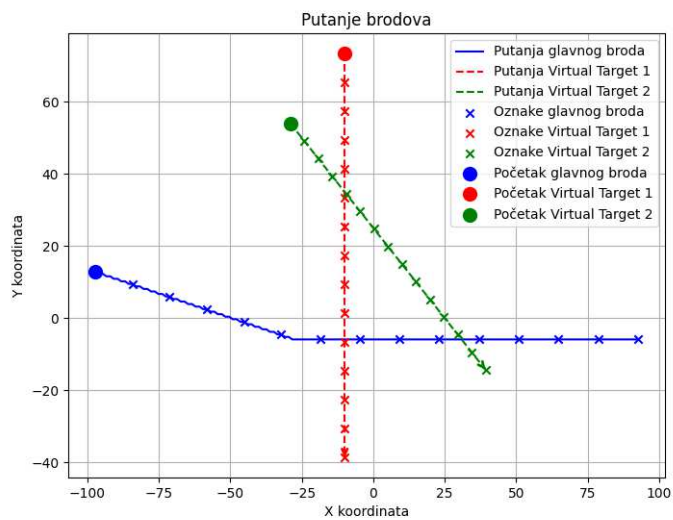


Slika 45 – Scenarij 13: graf udaljenosti

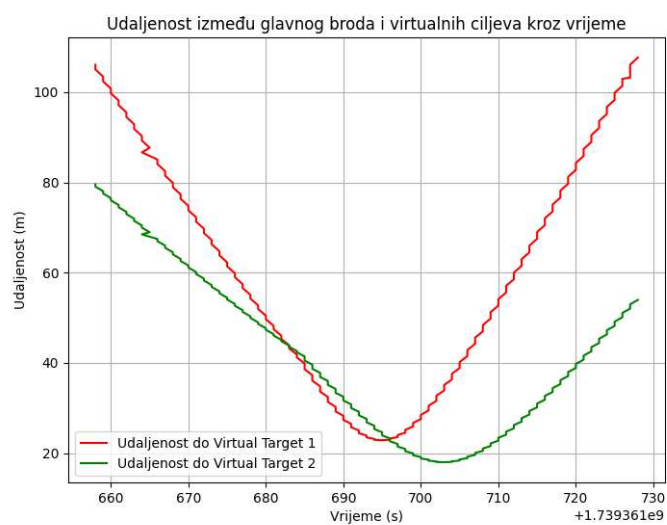


Slika 46 – Scenarij 13: grafovi brzine i udaljenosti

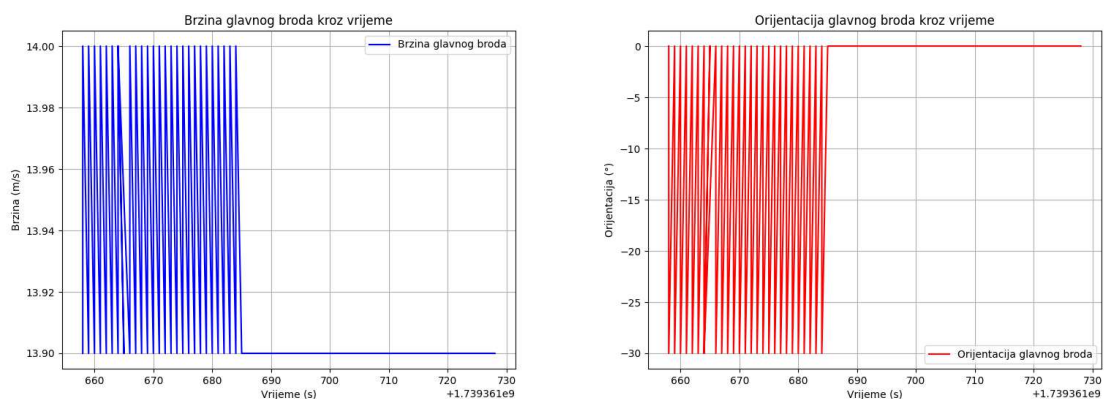
Scenarij 14 – Jedan virtualni cilj relativnog kuta orijentacije 270 stupnjeva, drugi virtualni cilj relativnog kuta orijentacije 315 stupnjeva



Slika 47 – Scenarij 14: graf putanje



Slika 48 – Scenarij 14: graf udaljenosti



Slika 49 – Scenarij 14: grafovi brzine i orijentacije

R.BR.	MAIN	VIRTUAL 1	SMJER	OBILAZAK
1	Na 0,0 ide horizontalno po y=0	Ide u -x smjeru po y=0	Frontalno (180)	Lijevo ✓
2	Na 0,0 ide horizontalno po y=0	Od gore prema dolje	90 stupnjeva (od gore dolje)	Lijevo ✓
3	Na 0,0 ide horizontalno po y=0	Od gore prema dolje	90 stupnjeva (od gore dolje)	Desno ✓
4	Na 0,0 ide horizontalno po y=0	Od dolje prema gore	270 stupnjeva (od dolje gore)	Lijevo ✓
5	Na 0,0 ide horizontalno po y=0	Ide u x smjeru po y=0	Preticanje	Desno ✓
6	Na 0,0 ide horizontalno po y=0	Od gore prema dolje	45 stupnjeva (od gore dolje)	Lijevo ✓
7	Na 0,0 ide horizontalno po y=0	Od gore prema dolje	135 stupnjeva (od gore dolje)	Lijevo ✓
8	Na 0,0 ide horizontalno po y=0	Od gore prema dolje	135 stupnjeva (od gore dolje)	Desno ✓
9	Na 0,0 ide horizontalno po y=0	Od dolje prema gore	225 stupnjeva (od dolje gore)	Lijevo ✓
10	Na 0,0 ide horizontalno po y=0	Od dolje prema gore	315 stupnjeva (od dolje gore)	Lijevo ✓
11	Na 0,0 ide horizontalno po y=0	Od dolje prema gore	315 stupnjeva (od dolje gore)	Desno ✓

Slika 50 – Popis isprobanih scenarija s jednim virtualnim ciljem

3.2. Diskusija VO metode

U ovoj diskusiji analiziramo ponašanje glavnog broda (OS) pri korištenju Velocity Obstacle (VO) metode za izbjegavanje sudara u različitim scenarijima, uspoređujemo rezultate s pravilima COLREG-a te procjenjujemo efikasnost manevriranja. Analiza se temelji na slikama i podacima iz scenarija.

Scenarij 1 – Frontalni susret

U prvom scenariju analizira se situacija u kojoj se glavni brod (OS) i ciljni brod (TS) nalaze na kolizijskom kursu. Iz prikazanih putanja jasno je vidljivo da OS skreće ulijevo kako bi izbjegao sudar, što nije u skladu s COLREG pravilima, prema kojima bi brod u ovakvoj situaciji trebao skrenuti udesno. Ovakav manevar pokazuje da VO metoda u ovom slučaju ne uzima u obzir formalna navigacijska pravila, što bi u stvarnim uvjetima moglo dovesti do nesporazuma među kapetanima brodova. Graf udaljenosti pokazuje da se brodovi u početnoj fazi postupno približavaju te dostižu minimalni međusobni razmak prije nego što OS započne manevar izbjegavanja. Stabilnost brzine OS-a sugerira da metoda ne mijenja značajno dinamiku kretanja osim u trenutku skretanja, dok graf orijentacije ukazuje na naglu promjenu kuta koja potvrđuje brzu reakciju, ali u suprotnom smjeru od očekivanog. Ovaj scenarij ukazuje na nekoliko ključnih problema VO metode. Prvo, iako je sudar izbjegnut, manevar nije u skladu s COLREG pravilima, što može biti opasno u stvarnim uvjetima s drugim plovilima u blizini. Odluka proizlazi zbog oduzimanja 180° za inverziju vektora. Važno je osigurati da se to radi dosljedno kroz sve funkcije koje obrađuju kutove. Ako se ne pazi, to može dovesti do konfuzije pri pretvorbi između referentnih sustava (npr. iz sjeverno-kutnog u istočno-kutni sustav) i može rezultirati neočekivanim obratima kao u ovom slučaju.

Scenarij 2 – Virtualni cilj relativnog kuta orijentacije 90 stupnjeva (prolaz s prednje strane)

Na temelju pažljive analize prikazanih grafova za treći scenarij, jasno je da se glavni brod (OS) inicijalno kreće u smjeru prema desno, dok se ciljni brod (TS) kreće vertikalno prema gore. Uočljivo je da OS u određenom trenutku započinje manevar skretanja ulijevo, što dovodi do sigurne separacije između plovila. Prema pravilima COLREG-a, kada se brodovi nalaze na križajućim putanjama, brod koji vidi drugog na svojoj desnoj strani (OS u ovom slučaju) bi trebao dati prednost i skrenuti udesno, što se ovdje ne događa. Umjesto toga, OS se odlučuje na skretanje ulijevo, što nije u skladu s pravilima, ali ipak omogućuje izbjegavanje sudara. Ovaj potez može biti rezultat optimizacijskog procesa VO metode, koja bira najbrži ili energetski najpovoljniji manevar umjesto da striktno prati formalna

pravila plovidbe. Graf udaljenosti pokazuje tipičan obrazac približavanja i udaljavanja, s najbližom točkom razdvajanja koja se čini sigurnom. Brzina glavnog broda ostaje relativno stabilna tijekom manevra, s blagim oscilacijama u početnim trenucima donošenja odluke. Slično tome, graf orijentacije prikazuje brzu promjenu kursa na početku manevra, nakon čega dolazi do stabilizacije kada brod zauzme novi pravac. Zaključno, iako je sudar izbjegnut, odluka glavnog broda da skrene ulijevo umjesto udesno predstavlja odstupanje od standardnih navigacijskih pravila. Ovo može ukazivati na potrebu za poboljšanjem algoritma kako bi se osiguralo da manevri budu ne samo sigurni, već i u skladu s međunarodnim propisima plovidbe.

Scenarij 3 – Virtualni cilj relativnog kuta orijentacije 90 stupnjeva (prolaz sa stražnje strane)

U ovom scenariju glavni brod u početnoj fazi detektira prisutnost ciljnog broda koji se kreće okomito na njegovu putanju i inicijalno započinje manevar skretanjem udesno. Ovaj početni manevar sugerira da VO metoda pokušava izvršiti izbjegavanje putem promjene smjera. Dolazi i do smanjenja brzine glavnog broda što ukazuje na to da je brod prilagodio svoju dinamiku kako bi omogućio siguran prolazak ciljnog broda. Istovremeno, graf orijentacije potvrđuje da je početno skretanje udesno bilo kratkotrajno te da je kasnija putanja ravna, čime se dodatno osigurava da ne dođe do sudara bez potrebe za naglim i rizičnim manevrima. Gledajući graf udaljenosti između brodova, može se primijetiti da dolazi do smanjenja razmaka u inicijalnoj fazi, nakon čega brodovi prolaze na sigurnoj udaljenosti. Ovaj scenarij pokazuje kako VO metoda može kombinirati manevre skretanja i regulacije brzine kako bi optimizirala izbjegavanje sudara. Za razliku od nekih ranijih scenarija, ovdje se jasno vidi kako algoritam u početku procjenjuje potrebu za skretanjem, ali se na kraju odlučuje za strategiju minimalnih promjena kursa uz prilagođavanje brzine. Ovo ponašanje je u skladu s COLREG pravilima, budući da glavni brod prepoznaje potrebu da omogući prolazak ciljnog broda koji mu je s desne strane te pravilno reagira bez nepotrebnog ugrožavanja sigurnosti plovidbe.

Scenarij 4 – Virtualni cilj relativnog kuta orijentacije 270 stupnjeva

U ovom scenariju ciljni brod dolazi s lijeve strane glavnog broda. Prema pravilima COLREG-a, glavni brod ima pravo prolaska, dok bi ciljni brod trebao poduzeti manevar izbjegavanja. Međutim, prema vizualizaciji putanja, može se primijetiti da glavni brod ipak prilagođava svoj kurs i brzinu umjesto da jednostavno nastavi bez promjena. Graf putanje jasno pokazuje da glavni brod u početku skreće ulijevo, a zatim nastavlja ravnom linijom. Ova odluka je rezultat rada VO metode, koja procjenjuje sigurnost prolaska i pokušava optimizirati putanju kako bi minimalizirala potrebne promjene u brzini i kursu. Unatoč tome što prema pravilima glavni brod nije obavezan mijenjati svoj kurs, algoritam ipak implementira mali korektivni manevar kako bi povećao sigurnost izbjegavanja. Grafikon udaljenosti kroz vrijeme pokazuje da brodovi dolaze relativno blizu jedan drugom, no

udaljenost se ne smanjuje ispod sigurnosne granice. Ovo sugerira da VO metoda radi s određenim pragom sigurnosti koji osigurava dovoljan razmak prilikom križanja putanja. Orijentacija glavnog broda dodatno potvrđuje ovu analizu, jer vidimo da postoji kratkotrajno povećanje kuta, nakon čega se vrijednosti stabiliziraju. To znači da algoritam nije donio drastične odluke, već je koristio minimalne prilagodbe kako bi postigao sigurno izbjegavanje bez velikih odstupanja od optimalne putanje. Ovaj scenarij pokazuje da VO metoda nije striktno vođena pravilima COLREG-a, već u obzir uzima i druge parametre kao što su brzina brodova, vremenska predikcija sudara i optimalna putanja s najmanjim promjenama kursa i brzine. Iako je sudar izbjegnut, moglo bi se raspraviti o tome treba li algoritam eksplicitnije uzimati u obzir pravo prolaska i potpuno eliminirati potrebu za manevrima u situacijama gdje glavni brod ima prednost.

Scenarij 5 – Virtualni cilj relativnog kuta orijentacije 270 stupnjeva (prolazak sa stražnje strane)

U ovom scenariju glavni brod inicijalno započinje manevar skretanjem ulijevo. Prema COLREG pravilima, u situacijama kada se brodovi križaju, brod koji vidi drugog broda s desne strane dužan je ustupiti prolaz i po pravilu bi trebao skrenuti udesno kako bi izbjegao koliziju. Međutim, analiza koda VO metode otkriva da optimizacijski parametri i način na koji se izračunava sigurna putanja dovode do odluke o skretanju ulijevo. Ova odluka može je posljedica toga što VO algoritam minimizira promjene kursa i brzine, tražeći najkraću moguću sigurnu putanju. Dodatno, može se primijetiti da dolazi do povećanja orijentacije glavnog broda na dijagramu kuta kroz vrijeme, što potvrđuje inicijalno skretanje. Također, analiza brzine pokazuje blagu promjenu, što može ukazivati na prilagodbu algoritma u pokušaju pronalaska optimalnog rješenja. Udaljenost između brodova smanjuje se do određene točke, nakon čega ponovno raste, što ukazuje na uspješno izbjegavanje sudara. Zaključno, iako je sudar izbjegnut, ponašanje glavnog broda nije potpuno u skladu s COLREG pravilima. Uvođenje dodatnih ograničenja ili preferencijalnog kursa udesno unutar VO metode moglo bi poboljšati pridržavanje pomorskih navigacijskih pravila, čime bi se izbjegli potencijalni problemi u stvarnim uvjetima.

Scenarij 6 – Pretjecanje

Kako se udaljenost između brodova smanjuje, glavni brod započinje značajno skretanje prema desno, što se može vidjeti na prvoj slici putanja. Ova odluka u skladu je s COLREG pravilima jer glavni brod u ovom slučaju tretira ciljni brod kao prepreku ispred sebe te skretanjem udesno osigurava sigurno zaobilazanje. Graf udaljenosti pokazuje klasičan V-oblik, gdje brodovi dolaze na najmanju međusobnu udaljenost oko sredine grafikona prije nego što se razdvoje. Ova minimalna udaljenost ukazuje na efikasnost VO metode. Analiza brzine ukazuje na ubrzanje glavnog broda u početku kako bi stigao zaobići ciljni brod. Orijehtacija glavnog broda jasno pokazuje naglo povećanje kuta u početnoj fazi, što potvrđuje nagli manevar udesno. Nakon toga, orijentacija ostaje stabilna, što znači da se brod uspješno prilagodio novom kursu bez daljnjih korekcija.

Scenarij 7 – Virtualni cilj relativnog kuta orijentacije 135 stupnjeva

Glavni brod inicijalno kreće ravno, dok se ciljni brod približava s donje desne strane, što predstavlja situaciju križanja. Prema pravilima COLREG-a, brod koji vidi drugo plovilo s desne strane mora mu dati prednost, što znači da bi trebao skrenuti udesno ili usporiti. Graf putanje pokazuje da glavni brod ispravno skreće udesno kako bi propustio ciljni brod, što je u skladu s pravilima. Graf udaljenosti prikazuje postupno smanjenje razmaka između brodova dok glavni brod ne započne manevar izbjegavanja, nakon čega se razmak povećava. Graf orijentacije prikazuje promjenu kuta kada brod započinje skretanje. Ovaj scenarij demonstrira ispravno funkcioniranje VO metode u skladu s COLREG pravilima. Manevar je blag i pravovremen, što omogućuje sigurno izbjegavanje bez naglih promjena kursa ili brzine. Ovakvo ponašanje pokazuje da je VO metoda u ovom slučaju uspješno integrirala navigacijska pravila u donošenje odluka.

Scenarij 8 – Virtualni cilj relativnog kuta orijentacije 135 stupnjeva (prednji zaobilazak)

U ovom scenariju glavni brod ima odluku podići se prema gore i proći ispred ciljnog broda koji dolazi iz donjeg desnog dijela. Ovakvo ponašanje objašnjeno je načinom na koji VO metoda procjenjuje sigurnosne zone i određuje optimalan manevar za izbjegavanje sudara. Odluka se temelji na analizi vremena do sudara (TTC), sigurnosne zone te dostupnih

brzinskih rješenja u skupu slobodnih brzina. Glavni brod počinje skretati prema lijevo čim prepozna potencijalni sudar s ciljnim brodom koji dolazi iz donjeg desnog smjera. Umjesto da smanji brzinu i prođe iza ciljnog broda, odlučuje za strategiju prolaska ispred njega. Grafikon udaljenosti kroz vrijeme pokazuje kako se udaljenost između brodova smanjuje do određene točke, nakon čega se ponovno povećava, što potvrđuje da je manevar bio uspješan u izbjegavanju sudara. Brzina glavnog broda se povećala kako bi ubrzao i stigao proći prije dolaska ciljnog broda. Grafikon orijentacije glavnog broda pokazuje naglu promjenu kuta, što ukazuje na oštar manevar podizanja putanje, ali zatim stabilizaciju kada je brod došao na sigurnu udaljenost ispred ciljnog broda. Ovakav manevar može biti rizičan ako ciljni brod ubrza ili ne održi stabilnu putanju, jer bi tada moglo doći do mogućeg sudara ili potrebe za dodatnim prilagođavanjem kursa. Ovakva odluka je posljedica algoritma koji optimizira putanju u skladu s trenutnim uvjetima, ali može odstupati od standardnih COLREG pravila, gdje bi sigurnija opcija često bila smanjenje brzine i prolazak iza ciljnog broda umjesto pokušaja prolaska ispred njega.

Scenarij 9 – Virtualni cilj relativnog kuta orijentacije 45 stupnjeva

Na temelju prikazanih grafova, glavni brod i virtualni cilj se nalaze u situaciji križanja, pri čemu se virtualni cilj nalazi desno od glavnog broda. U skladu s pravilima izbjegavanja sudara na moru, brod koji vidi drugog s desne strane mora mu prepustiti pravo prolaska. U ovom slučaju, glavni brod prepoznaje situaciju i pravovremeno skreće udesno, čime osigurava sigurno izbjegavanje sudara. Graf putanja brodova jasno pokazuje ovu promjenu smjera, gdje se glavni brod zakreće kako bi prošao iza virtualnog cilja. Udaljenost između brodova kroz vrijeme potvrđuje da je minimalna udaljenost sigurno održana, dok podaci o brzini i orijentaciji pokazuju stabilizaciju nakon manevara. Ovakav postupak je u skladu s pravilima i omogućava sigurno i učinkovito izbjegavanje prepreke.

Scenarij 10 – Virtualni cilj relativnog kuta orijentacije 315 stupnjeva

U ovom scenariju glavni brod dolazi s desne strane ciljnog brodu, što mu prema COLREG pravilima daje pravo prolaska. Međutim, budući da ne upravljamo virtualnim brodom, glavni brod mora donijeti odluku o sigurnom izbjegavanju potencijalne kolizije. Iz prikazanih grafova vidljivo je da glavni brod ubrzava i skreće udesno, čime osigurava da na vrijeme prođe ispred ciljanja broda. Ovo je jasno prikazano na grafu brzine, gdje dolazi do blagog

povećanja brzine, te na grafu orijentacije, gdje se vidi promjena smjera udesno. Odluka o ubrzanju i prolasku ispred temelji se na činjenici da je glavni brod već imao prednost te je procijenjeno da mu je to najsigurnija strategija za izbjegavanje sudara, umjesto usporavanja i prolaska iza ciljanog broda. Graf udaljenosti između brodova pokazuje kako se udaljenost smanjuje do kritične točke, nakon čega ponovno raste, što potvrđuje uspješnost odluke o ubrzanju i izbjegavanju sudara. Ovaj scenarij pokazuje uspješnost poštivanja COLREG pravila.

Scenarij 11 – Virtualni cilj relativnog kuta orijentacije 315 stupnjeva (lijevi zaobilazak)

U ovom scenariju, glavni brod se inicijalno kreće ravno dok mu se virtualni brod približava s njegove lijeve strane i dolazi u zonu potencijalnog sudara. Prema COLREG pravilima, glavni brod koji se nalazi s desne strane u situaciji križanja ima prednost, što bi značilo da bi virtualni brod trebao poduzeti manevar izbjegavanja. Međutim, s obzirom da nemamo kontrolu nad virtualnim brodom, glavni brod donosi odluku da ne iskoristi svoju prednost, već umjesto toga skreće ulijevo kako bi propustio virtualni brod. Ovaj manevar je vidljiv na putanji gdje glavni brod privremeno mijenja kurs prije nego što se vrati na svoj originalni smjer nakon što virtualni brod prođe. Brzina glavnog broda se poveća dok ne dođe u sigurnu zonu za nastavak u originalnom smjeru.

Scenarij 12 – Frontalni susret s dva virtualna cilja

U ovom scenariju glavni brod se nalazi u situaciji frontalnog susreta s virtualnim brodom 1, dok se iznad njega, s lijeve strane, paralelno kreće virtualni brod 2. Prema pravilima COLREG-a, glavni brod bi u ovoj situaciji trebao skrenuti udesno kako bi izbjegao sudar s virtualnim brodom 1, posebno zato što bi mu dodatni manevar ulijevo približio i virtualni brod 2, čime bi potencijalno povećao rizik od kolizije. Međutim, putanja pokazuje da glavni brod donosi suprotnu odluku i umjesto da skrene udesno, on skreće ulijevo i tek nakon što prođe oba virtualna broda, vraća se na svoju originalnu putanju. Analizom koda u *velobst_class.py* vidljivo je da sustav pravilno prepoznaje situaciju kao "Head-on" prema COLREG pravilu 14. Funkcija `check_colreg_rule_heading()` to ispravno klasificira i prosljeđuje ograničenja funkciji `calc_colreg_con()`, koja generira kolizijska ograničenja za optimizaciju putanje. Međutim, odluka o konačnom manevru donosi se u funkciji `calc_new_vel()`, gdje postoji faktor težine koji favorizira što manju promjenu originalnog

kursa. U ovom slučaju, može se vidjeti da algoritam daje prednost skretanju ulijevo, vjerojatno zato što procjenjuje da bi to omogućilo stabilniju i sigurniju putanju bez naglih promjena brzine ili orijentacije. Problem u ovakvoj odluci je što ona nije optimalna prema COLREG pravilima, budući da bi pravilo frontalnog susreta trebalo favorizirati izbjegavanje udesno. Dodatno faktor u kodu, histereznog ograničenja (`hyst_flag`), sprječava promjenu smjera izbjegavanja ako se ranije već detektirao neki manevar. Također, u funkciji `calc_new_vel()` nedostaje eksplicitno favoriziranje desnog skretanja u ovakvim slučajevima, što bi moglo riješiti problem nepravilnog izbjegavanja. Zaključno, sustav je u ovom slučaju donio odluku koja nije u skladu s pravilima COLREG-a, ali je svejedno izbjegao koliziju.

Scenarij 13 – Frontalni susret s jednim virtualnim ciljem, drugi virtualni cilj relativnog kuta orijentacije 90 stupnjeva

U ovom scenariju glavni brod se kreće prema istoku dok mu se virtualni cilj 2 približava s njegove desne strane. Prema pravilima COLREG-a, glavni brod bi trebao propustiti virtualni cilj 1 jer mu dolazi s desne strane, što bi značilo da bi trebao smanjiti brzinu ili skrenuti udesno kako bi omogućio prolaz virtualnom cilju. Međutim, umjesto toga, glavni brod odlučuje zadržati svoj pravac i oduzeti prednost virtualnom cilju 1 jer je procjena da mu je to najsigurniji manevar s najmanjom promjenom smjera. Analizirajući podatke sa grafikona, primjećujemo da je udaljenost između glavnog broda i virtualnog cilja 2 u početku opadajuća, što znači da se približavaju. Međutim, nakon određenog trenutka, udaljenost se počinje povećavati, što potvrđuje da je glavni brod uspješno prošao ispred virtualnog cilja 2 bez kolizije. Odluka o oduzimanju prednosti može se objasniti algoritamskim rješenjem koje procjenjuje da je sigurno nastaviti pravocrtno bez potrebe za izbjegavanjem. Virtualni cilj 1, koji se nalazi na većoj udaljenosti iznad glavnog broda, ne utječe na odluku o kursu, jer je metoda procijenila da je njegov utjecaj na navigaciju zanemariv. Nakon prolaska virtualnog cilja 2, glavni brod nastavlja svoju putanju bez daljnjih prilagodbi, što potvrđuje da je procjena sigurnosti bila točna i da nije došlo do sudara.

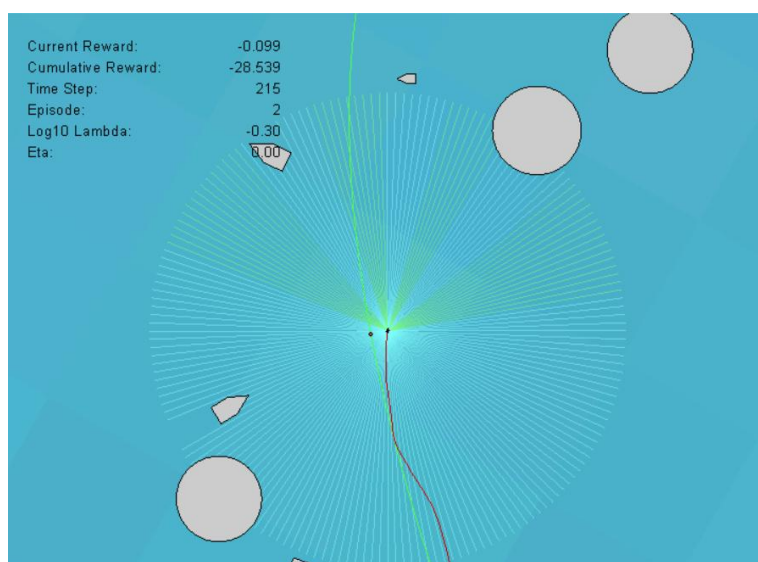
Scenarij 14 – Jedan virtualni cilj relativnog kuta orijentacije 270 stupnjeva, drugi virtualni cilj relativnog kuta orijentacije 315 stupnjeva

U ovoj situaciji, glavni brod kreće se prema istoku, dok mu se virtualni cilj 1 približava izravno sjeverno i virtualni cilj 2 dolazi dijagonalno sjeverozapadno. Kada sustav detektira

da virtualni cilj 1 predstavlja izravnu prijetnju, glavni brod donosi odluku o skretanju udesno, čime izbjegava sudar. Ovaj manevar omogućuje mu da se sigurno odvoji od oba virtualna cilja. Dakle, ispravno je zaključiti da algoritam u ovom slučaju donosi odluku u skladu s pravilima COLREG-a, birajući sigurniji put i smanjujući rizik od potencijalnog konflikta s oba virtualna cilja.

3.3. Rezultati DRL metode

Tijekom treniranja, posebna pažnja posvećena je izazovima poput konvergencije modela, prilagodbe na različite scenarije i dinamičke promjene u okruženju. Svaki agent prošao je niz epizoda u kojima je morao razviti strategije za donošenje odluka u realnim uvjetima, koristeći senzorske podatke i predviđanja gibanja drugih objekata. Za testiranje DRL metode istrenirano je 159 agenata. Rezultati performansi postignuti su treniranjem kroz 100 epizoda. Prosječni napredak doseže 94.87%, a prosječni broj sudara je izuzetno nizak (0.07), uz 93% epizoda bez sudara. Prosječni poprečni odstup (cross-track error) je 27.49, dok agenti u prosjeku provedu 898.51 koraka, što odgovara i prosječnom trajanju epizode. Dodatno, prosječna prijedena putanja iznosi 981.01, uz prosječnu brzinu od 1.28 i maksimalnu brzinu do 13.08. Ovi rezultati potvrđuju da, unatoč izazovima, primijenjena DRL metoda uspješno prilagođava agente za rad u dinamičnom okruženju uz visoku sigurnost kretanja.

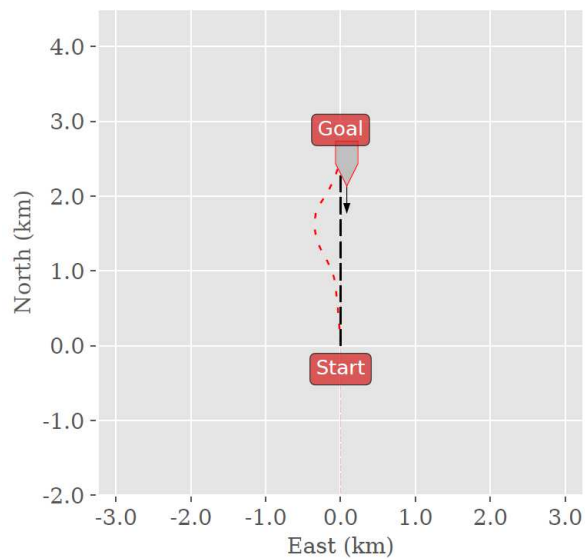


Slika 51 – Primjer scenarija

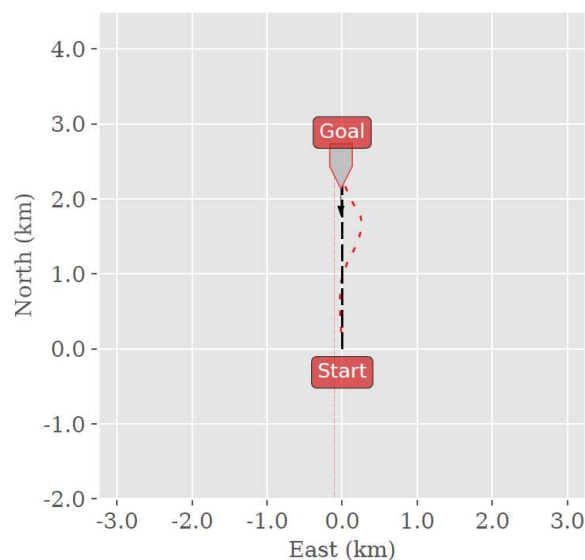
3.4. Diskusija DRL metode

Scenarij 1 – Frontalni susret

Glavni brod kreće iz početne točke (oznaka "Start") i kreće se prema svojoj ciljnoj lokaciji (oznaka "Goal"). Crna isprekidana linija predstavlja idealnu putanju broda, dok crvena isprekidana linija označava stvarnu putanju koju je brod poduzeo kao rezultat svoje strategije izbjegavanja sudara. U ovom scenariju, virtualni cilj nalazi se izravno na ruti glavnog broda i kreće se prema njemu. Umjesto da se kreće ravno prema cilju, glavni brod mijenja kurs blago ulijevo, izbjegavajući frontalni sudar. Nakon prolaska prepreke, vraća se na svoju originalnu putanju i nastavlja prema cilju. U DRL metodi agent je skretanje ulijevo naučio kroz trening i nagradne funkcije, što znači da je kroz iterativni proces učenja zaključio da mu je ovo najbolje rješenje za izbjegavanje sudara. Međutim, taj odabir možda nije vezan uz pravila COLREG-a, već je rezultat optimizacije učenja na temelju dobivenih nagrada. U oba pristupa brod je donio istu odluku – skrenuo ulijevo. To sugerira da su uvjeti scenarija, poput pozicije prepreke i dinamike kretanja, vodili brod prema istom rješenju neovisno o metodi odlučivanja. Međutim, u DRL metodi ovo je rezultat modela koji je kroz trening prepoznao ovo kao optimalno rješenje, dok u VO metodi to dolazi iz algoritamskog izračuna mogućih sudarnih pravaca i raspoloživih sigurnih putanja. Promjenom pozicije virtualnog broda udesno, glavni brod će skrenuti udesno te zaobići virtualni brod što dokazuje nepostojanje implementacije COLREG pravila.



Slika 52 – Frontalni susret (lijevi zaobilazak)

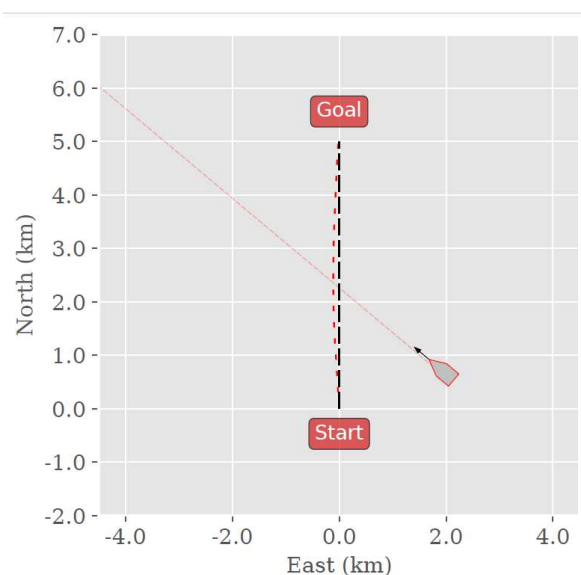


Slika 53 – Scenarij 1: Frontalni susret (desni zaobilazak)

Scenarij 2 – Prilaz virtualnog broda s desne strane glavnog broda

U ovom scenariju, glavni brod kreće se prema sjeveru, slijedeći svoju zadanu putanju. Virtualni brod dolazi s desne strane glavnog broda krećući se prema sjeverozapadu presijecajući putanju glavnog broda. Glavni brod pravovremeno skrene ulijevo te ubrza kako bi prošao prije virtualnog cilja. Ovaj scenarij u DRL metodi možemo usporediti s istim scenarijem 8 u VO metodi. Dobivamo jednako ponašanje ali u VO metodi glavni brod skreće

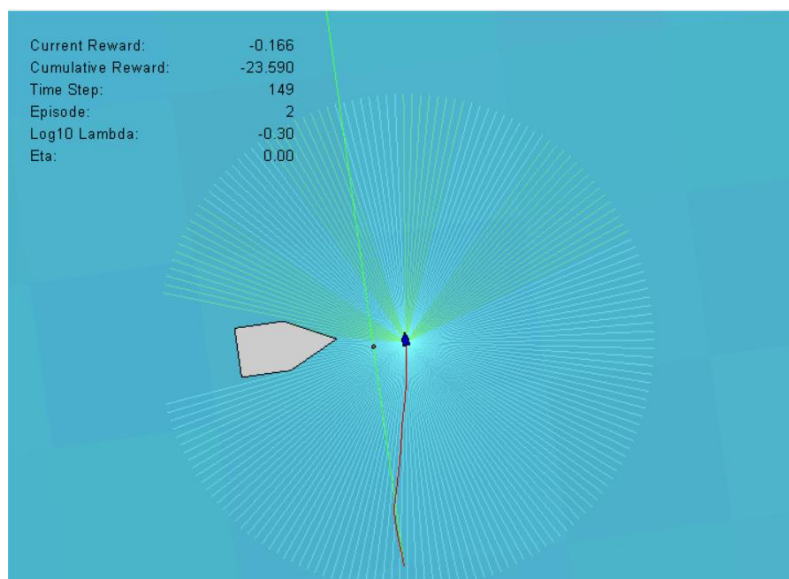
ulijevo pod većim kutom, što rezultira duljom putanjom zaobilaznja te suprotnost u ponašanju sa COLREG pravilima koji su implementirani u VO metodu.



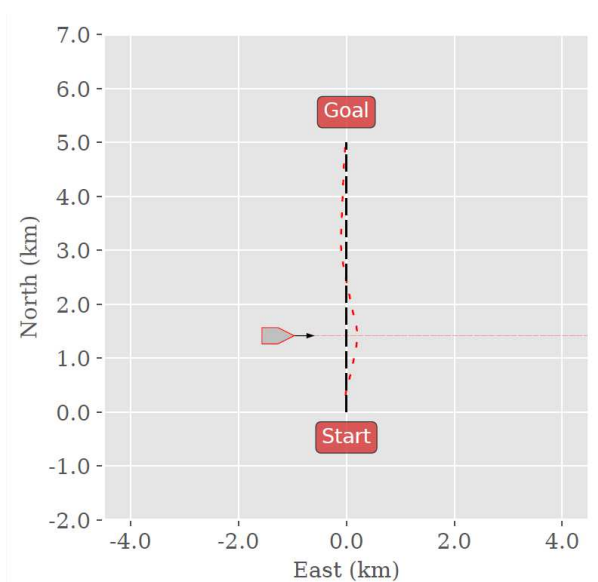
Slika 54 – Scenarij 2: Prilaz s desne strane

Scenarij 3 – Prilaz virtualnog broda s lijeve strane glavnog broda

Uspoređujući Scenarij 10 iz VO metode s odgovarajućim scenarijem u DRL metodi, analizirano je da oba pristupa prikazuju slična ponašanja glavnog broda u situacijama izbjegavanja sudara. Glavnom brodu virtualni brod prilazi s lijeve strane te presijeca putanju kretanja glavnog broda. Glavni brod skreće udesno kako bi izbjegao sudar što je i u skladu sa COLREG pravilima makar nisu ni implementirana u postojeću metodu.



Slika 55 – Scenarij 3: Izvođenje programa

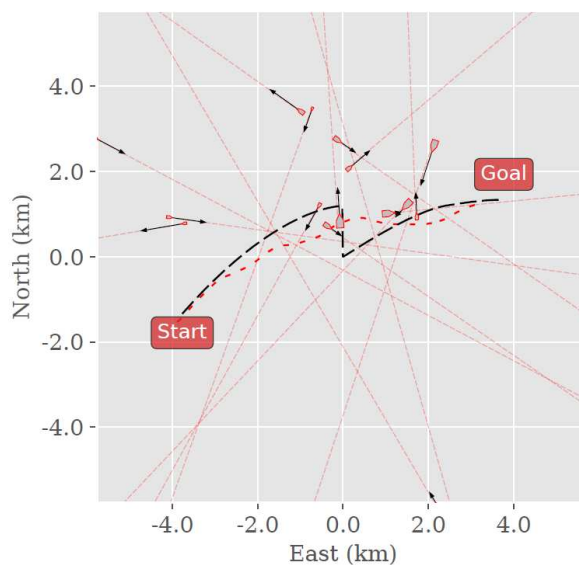


Slika 56 – Scenarij 3: Prilaz s lijeve strane

Scenarij 4 – Složeni scenarij

U složenim scenarijima s nepredvidivim kretanjima te prisutnošću statičkih i dinamičkih objekata, primijećeno je da DRL metoda pokazuje superiorne performanse u usporedbi s VO metodom. DRL agent uspješno navigira kroz kompleksno okruženje, izbjegavajući prepreke na učinkovit i prilagodljiv način, dok VO metoda rezultira neobjašnjivim ponašanjem glavnog broda te izazivanjem sudara. DRL agenti treniraju se kroz simulacije koje uključuju

raznovrsne scenarije, što im omogućuje da nauče optimalne strategije za izbjegavanje sudara u dinamičnim i nepredvidivim okruženjima. Kroz proces učenja, DRL agenti razvijaju sposobnost donošenja odluka u realnom vremenu, prilagođavajući se promjenama u okruženju i ponašanju drugih objekata. VO metoda se oslanja na predeterminirana pravila i geometrijske proračune za izbjegavanje sudara, što može biti nedovoljno u složenim i dinamičnim situacijama. U scenarijima gdje objekti mijenjaju brzinu ili smjer na nepredvidive načine, VO metoda može generirati neoptimalne ili čak neobjašnjive manevre glavnog broda. VO metoda koristi trenutne brzine i smjerove za predviđanje sudara, bez uzimanja u obzir mogućih promjena u kretanju drugih objekata. Pravila za izbjegavanje sudara u VO metodi mogu biti previše rigidna, ne dopuštajući fleksibilnost u donošenju odluka u dinamičnim situacijama. Za razliku od DRL metode, VO metoda ne koristi mehanizme učenja, što znači da ne može poboljšati svoje performanse kroz iskustvo ili prilagoditi se novim scenarijima.



Slika 57 – Scenarij 4: Složeni scenarij

Zaključak

U ovom radu provedena je detaljna analiza performansi algoritma za izbjegavanje sudara temeljenog na COLREG-u, s posebnim naglaskom na metodu prepreke brzine te dodatno na pristup temeljen na dubokom podržanom učenju. Rezultati pokazuju da VO metoda, iako matematički precizna i intuitivna, može biti osjetljiva na fiksne parametre, što može dovesti do neočekivanih manevara u dinamičnim i kompleksnim scenarijima, ali i u jednostavnim scenarijima malim promjenama parametara. DRL pristup omogućuje sustavu adaptaciju u realnom vremenu i učenje optimalnih strategija kroz interakciju sa simulacijskim okruženjem, ali zahtijeva opsežno treniranje te je računalno zahtjevan. Usporedbom obje metode, primijetili smo da VO metoda pruža brz i transparentan izračun sigurnih putanja, što je pogodno za situacije sa stabilnim uvjetima, dok njena rigidnost predstavlja ograničenje u okruženjima s visokom varijabilnošću. DRL sustav, unatoč svojoj računalnoj zahtjevnosti i osjetljivosti na pravilno definiranje nagradne funkcije, pokazuje značajnu prilagodljivost te je sposoban samostalno optimizirati svoje strategije bez potrebe za ručnim podešavanjem parametara.

Budući rad

Daljnje istraživanje trebalo bi se usredotočiti na integraciju ovih dvaju pristupa kako bi se iskoristile prednosti svakog – kombiniranjem preciznosti VO metode s adaptivnošću DRL agenta moglo bi se razviti hibridno rješenje koje je robusnije u stvarnim uvjetima. Poseban naglasak treba biti stavljen na ispravke postojećih grešaka u VO metodi. Naime, trenutni sustav za prijenos podataka o virtualnim ciljevima oslanja se na objavljivanje u intervalu od 1 Hz, zbog čega se podaci ostalih virtualnih brodova osim prvog ne primaju, što se privremeno rješava povećanjem frekvencije na 5 Hz. Idealno bi bilo preoblikovati sustav subscribera i publishera kako bi se osiguralo da svi podaci budu dostupni i pravilno prikazani. Također, potrebno je riješiti probleme u konverzijama između NED koordinatnog sustava i sustava korištenog u matematičkim operacijama te ispraviti pogrešne inverzije koje mogu rezultirati pogrešnim manevrima. Takve promjene omogućile bi VO metodi da preciznije modelira kolizijski prostor i donosi pouzdanije odluke u dinamičnim okruženjima, čime bi se u konačnici povećala sigurnost i pouzdanost autonomnih plovila. Sve popularniji

veliki jezični modeli (LLM). Prvenstveno su razvijeni za obradu prirodnog jezika te imaju sposobnost prepoznavanja i generiranja kompleksnih tekstualnih uzoraka, dok se DRL fokusira na učenje optimalnih akcija kroz nagrade u nekom okruženju. LLM modeli primaju ogromne količine podataka i mogu rješavati raznolike problemske zadatke, a ne samo one specifične za jednu domenu. Sposobni su „razumjeti“ kontekst, generirati scenarije i predviđati moguće ishode na temelju velikog znanja naučenog iz tekstova. Veliki jezični modeli mogu pružiti objašnjenje, prepoznati složene obrasce iz tekstualnih ili drugih nestrukturiranih izvora te generirati same scenarije za testiranje.

Literatura

- [1] Python Software Foundation. Python. Web-stranica. Poveznica: <https://www.python.org/>; pristupljeno 3. studenog 2024.
- [2] SciPy Community. SciPy. Web-stranica. Poveznica: <https://scipy.org/>; pristupljeno 3. studenog 2024.
- [3] NumPy Developers. NumPy. Web-stranica. Poveznica: <https://numpy.org/>; pristupljeno 3. studenog 2024.
- [4] Matplotlib Development Team. Matplotlib. Web-stranica. Poveznica: <https://matplotlib.org/>; pristupljeno 13. studenog 2024.
- [5] International Maritime Organization. COLREG. Web-stranica. Poveznica: <https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx>; pristupljeno 25. listopada 2024.
- [6] International Maritime Organization. Preventing Collisions. Web-stranica. Poveznica: <https://www.imo.org/en/OurWork/Safety/Pages/Preventing-Collisions.aspx>; pristupljeno 25. listopada 2024.
- [7] Egreuel. ASV Path Planner. Web-stranica. Poveznica: https://egreuel.github.io/asv_path_planner/index.html; pristupljeno 25. listopada 2024.
- [8] LABUST. ASV Path Planning. GitHub repozitorij. Poveznica: https://github.com/labust/asv_path_planning/tree/main; pristupljeno 25. listopada 2024.
- [9] ThomasNLarsen. Gym AUV. GitHub repozitorij. Poveznica: <https://github.com/ThomasNLarsen/gym-auv>; pristupljeno 15. siječnja 2025.
- [10] Meyer, E., Robinson, H., Rasheed, A. i San, O. Taming an Autonomous Surface Vehicle for Path Following and Collision Avoidance Using Deep Reinforcement Learning, IEEE Access, 8 (2020), str. 41466-41481, doi: 10.1109/ACCESS.2020.2976586..
- [11] Amazon Web Services. What is a Large Language Model? Web-stranica. Poveznica: <https://aws.amazon.com/what-is/large-language-model/>; pristupljeno 13. veljače 2025.

- [12] Liu, W., Zhang, B., Liu, P., Pan, J. i Chen, S. Velocity obstacle guided motion planning method in dynamic environments, *Journal of King Saud University - Computer and Information Sciences*, 36,1 (2024), 101889, ISSN 1319-1578.
- [13] Cebollada, S., Payá, L., Flores, M., Peidró, A. i Reinoso, O. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data, *Expert Systems with Applications*, 167 (2021). pristupljeno 13. veljače 2025.
- [14] Huang, Y., van Gelder, P.H.A.J.M. i Wen, Y. Velocity obstacle algorithms for collision prevention at sea, *Ocean Engineering*, 151 (2018), str. 308-321, ISSN 0029-8018.
- [15] A Beginner's Guide to Deep Reinforcement Learning. ScienceDirect, (n.d.).
Poveznica: <https://www.sciencedirect.com/science/article/abs/pii/S0029801818300015>;
pristupljeno 16. siječnja 2025.

Sažetak

Naslov: Analiza performansi algoritma za izbjegavanje sudara na moru temeljenog na COLREG-u

Sažetak:

Ovaj rad predstavlja analizu performansi algoritma za izbjegavanje sudara u pomorskoj navigaciji, s posebnim naglaskom na metodu prepreke brzine i pristup temeljen na dubokom podržanom učenju. Istraživanje obuhvaća matematičku i geometrijsku formulaciju VO metode, njezinu implementaciju u Pythonu, integraciju sa ROS2 sustavom te simulacijske eksperimente s različitim scenarijima susreta. Rezultati pokazuju da, iako je VO metoda brza i transparentna, njena rigidnost i osjetljivost na nesigurnosti senzorskih podataka mogu dovesti do neočekivanih manevara. S druge strane, DRL pristup omogućuje adaptaciju u realnom vremenu bez potrebe za ručnim podešavanjem parametara, no zahtijeva opsežno treniranje. Rad također raspravlja o mogućnostima hibridnog sustava koji bi kombinirao prednosti oba pristupa uz dodatne ispravke, poput poboljšanja sustava prijenosa podataka i dosljedne konverzije koordinata.

Ključne riječi: COLREG, prepreka brzine, duboko podržano učenje, autonomna navigacija, izbjegavanje sudara, ROS2, simulacijski eksperimenti

Summary

Title: Performance analysis of a COLREG-based marine collision avoidance algorithm

Abstract: This paper presents an analysis of the performance of a collision avoidance algorithm in marine navigation, with a special emphasis on the speed barrier method and an approach based on deep support learning. The research includes the mathematical and geometric formulation of the VO method, its implementation in Python, integration with the ROS2 system, and simulation experiments with different meeting scenarios. The results show that, although the VO method is fast and transparent, its rigidity and sensitivity to sensor data uncertainties can lead to unexpected maneuvers. On the other hand, the DRL approach enables real-time adaptation without the need for manual parameter adjustment but requires extensive training. The paper also discusses the possibilities of a hybrid system that would combine the advantages of both approaches with additional improvements, such as improving the data transfer system and consistent coordinate conversion.

Keywords: COLREG, Velocity Obstacle, Deep Reinforcement Learning, autonomous navigation, collision avoidance, ROS2, simulation experiments

Skraćenice

RL	<i>Reinforced Learning</i>	podržano učenje
DRL	<i>Deep Reinforced Learning</i>	duboko podržano učenje
VO	<i>Velocity Obstacle</i>	brzina prepreka
OS	<i>Own Ship</i>	glavni brod
TS	<i>Target Ship</i>	virtualni brod/cilj
LLM	<i>Large Language Models</i>	veliki jezični modeli

Tablica slika

Slika 1 – Prikaz sektora	5
Slika 2 – Proširenje TS-a.....	7
Slika 3 – Kolizijski konus.....	8
Slika 4 – VO vizualizacija	9
Slika 5 – VO podjela prostora brzina	10
Slika 6 – Diskretizirani prostor brzine.....	12
Slika 7 – Primjer izvođenja DRL metode.....	46
Slika 8 – Scenarij 1: graf putanje.....	49
Slika 9 – Scenarij 1: graf udaljenosti.....	49
Slika 10 – Scenarij 1: grafovi brzine i orijentacije	50
Slika 11 – Scenarij 2: graf putanje	51
Slika 12 – Scenarij 2: graf udaljenosti.....	51
Slika 13 – Scenarij 2: grafovi brzine i orijentacije	52
Slika 14 – Scenarij 3: graf putanje	52
Slika 15 - Scenarij 3: graf udaljenosti	53
Slika 16 – Scenarij 3: grafovi brzine i orijentacije	53
Slika 17 – Scenarij 4: graf putanje	54
Slika 18 – Scenarij 4: graf udaljenosti.....	54
Slika 19 – Scenarij 4: grafovi brzine i orijentacije	55
Slika 20 – Scenarij 5: graf putanje	55
Slika 21 – Scenarij 5: graf udaljenosti.....	56
Slika 22 – Scenarij 5: grafovi brzine i orijentacije	56

Slika 23 – Scenarij 6: graf putanje	57
Slika 24 – Scenarij 6: graf udaljenosti.....	57
Slika 25 – Scenarij 6: grafovi brzine i orijentacije	58
Slika 26 – Scenarij 7: graf putanje	58
Slika 27 – Scenarij 7: graf udaljenosti.....	59
Slika 28 – Scenarij 7: grafovi brzine i orijentacije	59
Slika 29 – Scenarij 8: graf putanje	60
Slika 30 – Scenarij 8: graf udaljenosti.....	60
Slika 31 – Scenarij 8: grafovi brzine i orijentacije	61
Slika 32 – Scenarij 9: graf putanje	61
Slika 33 – Scenarij 9: graf udaljenosti.....	62
Slika 34 – Scenarij 9: grafovi brzine i orijentacije	62
Slika 35 – Scenarij 10: graf putanje.....	63
Slika 36 – Scenarij 10: graf udaljenosti.....	63
Slika 37 – Scenarij 10: grafovi brzine i orijentacije	64
Slika 38 – Scenarij 11: graf putanje.....	64
Slika 39 – Scenarij 11: graf udaljenosti.....	65
Slika 40 – Scenarij 11: grafovi brzine i orijentacije	65
Slika 41 – Scenarij 12: graf putanje.....	66
Slika 42 – Scenarij 12: graf udaljenosti.....	66
Slika 43 – Scenarij 12: grafovi brzine i orijentacije	67
Slika 44 – Scenarij 13: graf putanje.....	67
Slika 45 – Scenarij 13: graf udaljenosti.....	68
Slika 46 – Scenarij 13: grafovi brzine i udaljenosti.....	68

Slika 47 – Scenarij 14: graf putanje.....	69
Slika 48 – Scenarij 14: graf udaljenosti.....	69
Slika 49 – Scenarij 14: grafovi brzine i orijentacije	70
Slika 50 – Popis isprobanih scenarija s jednim virtualnim ciljem.....	70
Slika 51 – Primjer scenarija.....	79
Slika 52 – Frontalni susret (lijevi zaobilazak).....	81
Slika 53 – Scenarij 1: Frontalni susret (desni zaobilazak).....	81
Slika 54 – Scenarij 2: Prilaz s desne strane	82
Slika 55 – Scenarij 3: Izvođenje programa.....	83
Slika 56 – Scenarij 3: Prilaz s lijeve strane	83
Slika 57 – Scenarij 4: Složeni scenarij	84