

Fazi logika u upravljačkim sistemima za bespilotna vozila

Tamara Baranin

Milica Kužet

Septembar 2024

Sadržaj

1	Uvod	2
2	Fazi logika	2
2.1	Opis kontrolnog sistema	2
3	Formulacija problema	3
3.1	Ulazne funkcije pripadnosti	4
3.2	Izlazne funkcije pripadnosti	4
4	Osnovna pravila donošenja odluka za fazi kontroler	4
4.1	Praktična implementacija	5
4.1.1	Slučaj bez prepreka	5
4.1.2	Slučaj sa detektovanjem prepreka u dometu senzora	5
4.1.3	Slučaj sa preprekama na ekstremnim uglovima	6
4.2	Grafici	8
4.2.1	Slučaj bez prepreka	8
4.2.2	Slučaj sa detektovanjem prepreka u dometu senzora	8
4.2.3	Slučaj sa preprekama na ekstremnim uglovima	9
5	Zaključak	10
6	Literatura	10

1 Uvod

U poslednje vreme govori se o sve većoj upotrebi autonomnih vozila u misijama koje su opasne, monotone ili rizične, poput operacija u nuklearnim postrojenjima, istraživanja Marsa, vojne špijunaže, nadzora šumskih požara i vremenskog prognoziranja. Trenutni problem sa bespilotnim letelicama je da nisu sposobne da promene ugao ili zadatak tokom leta, što je ključno kada se ciljevi misije promene ili se pojave nove pretnje. Zbog toga je postalo jasno da je ključni izazov za razvoj autonomna i inteligentna kontrola. Jedna od najvažnijih sposobnosti je planiranje putanja, posebno u dinamičnim okruženjima ili onima gde nema prethodnih informacija o preprekama. Ova prezentacija ima za cilj da predloži algoritam za planiranje putanja u dve dimenzije, koji bi omogućio letelici da funkcioniše u realnom vremenu i nepoznatom okruženju sa preprekama. Algoritam koristi senzore za detekciju prepreka, a informacije se procesiraju kako bi mogao da izbegne prepreke i planira put ka cilju, uzimajući u obzir dinamička i kinematička ograničenja.

2 Fazi logika

Fazi logika nudi značajne prednosti u situacijama gde se okruženje neprekidno menja i gde su dostupne nepotpune informacije. Zato je fazi logika osnova kontrolne strategije za planiranje kretanja koja je predstavljena u ovom delu.

2.1 Opis kontrolnog sistema

- Fazifikacija:

Ovaj prvi korak uzima analogne ulaze i pretvara ih u kontinuirane vrednosti između 0 i 1, na osnovu stepena pripadnosti svakoj funkciji članstva.

- Pravila Odluka:

Koristi se baza znanja za formiranje skupa pravila koja povezuju ulaze i izlaze u obliku "ako-onda" izjava. Ova pravila definišu kako se ulazi transformišu u izlaze.

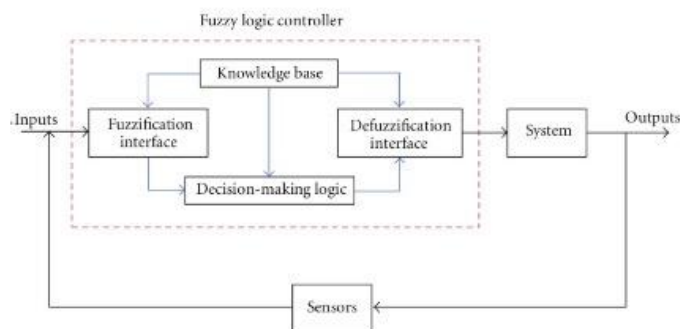
- Defazifikacija:

Na kraju, izlazi se pretvaraju u jasne brojeve koristeći metodu defazifikacije. Ovaj proces konvertuje fazi rezultate u konkretne, primenljive vrednosti koje se koriste za kontrolu.

- Praktična implementacija:

Kontroler fazi logike može se vizualizovati kao sistem koji koristi heurističko znanje za manipulaciju ulazima, obrađuje ih kroz logiku odlučivanja, i zatim generiše precizne izlaze putem defazifikacije.

Ukratko, fazi logika omogućava upravljanje u dinamičnim i neizvesnim okruženjima tako što koristi fleksibilan i adaptivan pristup za donošenje odluka, koji je prikladan za planiranje kretanja autonomnih vozila.



Slika 1: Fazi kontroler

3 Formulacija problema

Problem planiranja putanje u dvodimenzionalnom prostoru formuliše se generisanjem putanje između poznatog početnog stanja, $O(x_0, y_0)$, i krajnjeg stanja, $T(x_t, y_t)$. Kinematičke jednačine za UAV u ovoj situaciji zavise od inercijalne pozicije (x, y) , brzine kretanja (v) i ugla pravca (θ) , kao što je prikazano na slici 1. Kako se bavimo planiranjem putanje u dvodimenzionalnom prostoru, odgovor na promenu visine (visinsku razliku) je izostavljen. Dakle, fokusiramo se samo na kontrolu brzine i ugla pravca, gde su v_{c_i} i θ_{c_i} ulazni kontrolni parametri. Kretanje UAV-a može se opisati kao što je prikazano u jednačini gde su τ_v i τ_θ vremenska kašnjenja povezana sa kontrolom brzine i ugla pravca.

Jednačina:

$$\begin{aligned}\dot{v} &= \frac{1}{\tau_v}(v_c - v) \\ \dot{\theta} &= \frac{1}{\tau_\theta}(\theta_c - \theta)\end{aligned}\tag{1}$$

Pozicija UAV-a može definisati pomoću ugla pravca (θ) i udaljenosti od početne tačke (l) na sledeći način:

$$X = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} l \cos \theta \\ l \sin \theta \end{pmatrix}\tag{2}$$

$$\dot{X} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} v \cos \theta \\ v \sin \theta \end{pmatrix}\tag{3}$$

$$\ddot{X} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \dot{v} \cos \theta - v \dot{\theta} \sin \theta \\ \dot{v} \sin \theta + v \dot{\theta} \cos \theta \end{pmatrix}\tag{4}$$

Kombinovanjem (1) i (4), kinematička jednačina može se zapisati kao:

$$\ddot{X} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \frac{1}{\tau_v}(v_c - v) \cos \theta - v \frac{1}{\tau_\theta}(\theta_c - \theta) \sin \theta \\ \frac{1}{\tau_v}(v_c - v) \sin \theta + v \frac{1}{\tau_\theta}(\theta_c - \theta) \cos \theta \end{pmatrix}\tag{5}$$

gde je v_c brzina, a θ_c ugao nagiba. Dodatna ograničenja:

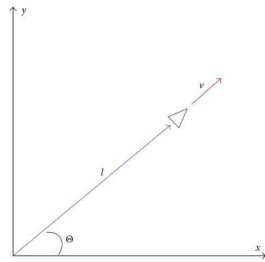
$$v_{\min} \leq v \leq v_{\max}\tag{6}$$

$$-a_{\max} \leq \dot{v} \leq a_{\max}\tag{7}$$

$$-\theta_{\max} \leq \theta \leq \theta_{\max}\tag{8}$$

$$-\omega_{\max} \leq \dot{\theta} \leq \omega_{\max}\tag{9}$$

U ovom opisu problema, agent ima opseg senzora koji može da detektuje prepreke u rasponu od $\pm 90^\circ$ i unutar određenog radijusa senzora.



Slika 2: Kretanje bespilotne letelice

3.1 Ulazne funkcije pripadnosti

Za ovaj sistem se koriste četiri ulaza i dva izlaza. Ulazi su:

- Udaljenost od prepreke:
Opisuje se sa četiri funkcije pripadnosti: Blizu, Srednje, Daleko i Veoma Daleko, (Close, Medium, Far, Very far).
- Ugao između bespilotne letelice (UAV) i prepreke:
Opisuje se sa šest funkcija pripadnosti: Veliki Negativan, Srednji Negativan, Mali Negativan, Mali Pozitivan, Srednji Pozitivan, i Veliki Pozitivan, (Neg big, Neg med, Neg small, Pos small, Pos med, Pos big).
- Udaljenost do cilja:
Opisuje se sa tri funkcije pripadnosti: Na Vrh, Srednje, i Daleko, (On top, Medium, Far).
- Greška između trenutnog ugla putanje i ugla cilja:
Opisuje se sa sedam funkcija pripadnosti: Veliki Negativan, Srednji Negativan, Mali Negativan, Nula, Mali Pozitivan, Srednji Pozitivan, i Veliki Pozitivan, (Neg big, Neg med, Neg small, Zero, Pos small, Pos med, Pos big).

3.2 Izlazne funkcije pripadnosti

Na osnovu ulaza, koristi se baza pravila da se odrede izlazi kontrolnog sistema.

- Promena brzine:
Izlazna brzina može se klasifikovati u četiri funkcije pripadnosti: Veoma Sporo, Sporo, Brzo, i Veoma Brzo, (Very slow, Slow, Fast, Very fast).
- Promena ugla putanje:
Promena ugla putanje je opisana sa sedam funkcija pripadnosti: Veliki Negativan, Srednji Negativan, Mali Negativan, Nula, Mali Pozitivan, Srednji Pozitivan, i Veliki Pozitivan, (Neg big, Pos small, Neg med, Pos med, Neg small, Pos big, Zero).

4 Osnovna pravila donošenja odluka za fazi kontroler

- Baza pravila:
Pravila koja povezuju ulaze i izlaze za fazi logiku su postavljena u obliku "ako-onda" izjava i bazirana su na heuristikama i ljudskom iskustvu u navigaciji kroz okruženje (slično vožnji automobila). Postoji ukupno 40 pravila koja se mogu podeliti u dva glavna slučaja: kada postoji prepreka u dometu senzora i kada ne postoji.
- Planiranje direktnog puta:
Kada nema prepreka u dometu senzora, glavni cilj kontrolera je planiranje direktnog puta ka glavnom cilju. U ovom slučaju, udaljenost do prepreka se postavlja na "veoma daleko" (dalje od dometa senzora). Planiranje puta se vrši promenom ugla putanje UAV-a kako bi se uskladio sa uglom cilja u inernom referentnom okviru, ili smanjenjem greške između dva ugla na nulu.
- Brzina i ugao:
Kada nema prepreka i cilj je veoma daleko, UAV teži maksimalnoj operativnoj brzini. Kada UAV dostigne cilj, smanjuje brzinu kako bi završio misiju i nastavlja da smanjuje grešku u uglu na nulu.

4.1 Praktična implementacija

4.1.1 Slučaj bez prepreka

Kada nema prepreka u dometu senzora, pravila su usmerena na postizanje maksimalne brzine i tačno usklađivanje ugla putanje sa ciljem.

- Prilagođavanje brzine:

Kada se približava cilju, brzina se smanjuje kako bi se postigla preciznost u završavanju misije, a greška u uglu se dovodi do nule.

- Zaključak:

Ova pravila su osmišljena da efikasno upravljaju UAV-om u uslovima kada nema prepreka, bazirajući se na principima koji su slični onima korišćenim u vožnji automobila. Funkcija tih pravila je da omogućava direktno i efikasno kretanje ka cilju dok se minimizira greška u uglu i optimizuje brzina UAV-a.

- Pravila:

- (1) If $D_o = \text{Very Far}$ and $D_t = \text{Far}$, then $v = \text{Very Fast}$.
- (2) If $D_o = \text{Very Far}$ and $D_t = \text{Med.}$, then $v = \text{Slow}$.
- (3) If $D_o = \text{Very Far}$ and $D_t = \text{On Top}$, then $v = \text{Very Slow}$.
- (4) If $D_o = \text{Very Far}$ and $\theta_t = \text{Neg. Big}$, then $\theta_c = \text{Neg. Big}$.
- (5) If $D_o = \text{Very Far}$ and $\theta_t = \text{Neg. Med.}$, then $\theta_c = \text{Neg. Med.}$.
- (6) If $D_o = \text{Very Far}$ and $\theta_t = \text{Neg. Small}$, then $\theta_c = \text{Neg. Small}$.
- (7) If $D_o = \text{Very Far}$ and $\theta_t = \text{Zero}$, then $\theta_c = \text{Zero}$.
- (8) If $D_o = \text{Very Far}$ and $\theta_t = \text{Pos. Small}$, then $\theta_c = \text{Pos. Small}$.
- (9) If $D_o = \text{Very Far}$ and $\theta_t = \text{Pos. Med.}$, then $\theta_c = \text{Pos. Med.}$.
- (10) If $D_o = \text{Very Far}$ and $\theta_t = \text{Pos. Big}$, then $\theta_c = \text{Pos. Big}$.

4.1.2 Slučaj sa detektovanjem prepreka u dometu senzora

Kada se prepreke detektuju u dometu senzora, autonomno vozilo (UAV) treba da izvede sledeće:

- Smanji brzinu:

UAV treba da uspori kako bi sigurno manevrisao oko prepreke.

- Promeni ugao putanje:

Da bi izbegao prepreku, UAV menja svoj pravac tako da greška u uglu bude oko 90 stepeni u odnosu na prepreku.

- Povratak na putanju:

Nakon što zaobide prepreku, UAV se vraća na prvobitni put ka cilju, postupno se usklađujući sa početnim putem i brzinom.

- Smanji brzinu i promeni pravac:

Kada se prepreka otkrije, smanji brzinu i prilagodi pravac tako da se greška u uglu pomeri ka 90 stepeni.

- Nakon izbegavanja prepreke:

Kada je prepreka zaobidena, povrati brzinu i pravac kako bi se nastavio put ka cilju.

- Pravila:

Ova pravila omogućavaju vozilu da bezbedno izbegne prepreke i ponovo nastavi putanju ka cilju.

- (11) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Med.}$, then $v = \text{Very Fast}$ and $\theta_c = \text{Zero}$.

- (12) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Small}$, then $v = \text{Fast}$ and $\theta_c = \text{Neg. Small}$.
- (13) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Small}$, then $v = \text{Fast}$ and $\theta_c = \text{Pos. Small}$.
- (14) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Med}$, then $v = \text{Very Fast}$ and $\theta_c = \text{Zero}$.
- (15) If $D_o = \text{Medium}$ and $\theta_o = \text{Pos. Big}$, then $v = \text{Fast}$ and $\theta_c = \text{Zero}$.
- (16) If $D_o = \text{Medium}$ and $\theta_o = \text{Pos. Med.}$, then $v = \text{Slow}$ and $\theta_c = \text{Neg. Small}$.
- (17) If $D_o = \text{Medium}$ and $\theta_o = \text{Pos. Small}$, then $v = \text{Slow}$ and $\theta_c = \text{Neg. Med}$.
- (18) If $D_o = \text{Medium}$ and $\theta_o = \text{Neg. Small}$, then $v = \text{Slow}$ and $\theta_c = \text{Pos. Med}$.
- (19) If $D_o = \text{Medium}$ and $\theta_o = \text{NM}$, then $v = \text{Slow}$ and $\theta_c = \text{Pos. Small}$.
- (20) If $D_o = \text{Medium}$ and $\theta_o = \text{NB}$, then $v = \text{Fast}$ and $\theta_c = \text{Zero}$.
- (21) If $D_o = \text{Close}$ and $\theta_o = \text{PB}$, then $v = \text{Slow}$ and $\theta_c = \text{Neg. Small}$.
- (22) If $D_o = \text{Close}$ and $\theta_o = \text{PM}$, then $v = \text{Very Slow}$ and $\theta_c = \text{Neg. Med}$.
- (23) If $D_o = \text{Close}$ and $\theta_o = \text{PS}$, then $v = \text{Very Slow}$ and $\theta_c = \text{Neg. Big}$.
- (24) If $D_o = \text{Close}$ and $\theta_o = \text{NS}$, then $v = \text{Very Slow}$ and $\theta_c = \text{Pos. Big}$.
- (25) If $D_o = \text{Close}$ and $\theta_o = \text{NM}$, then $v = \text{Very Slow}$ and $\theta_c = \text{Pos. Med}$.
- (26) If $D_o = \text{Close}$ and $\theta_o = \text{NB}$, then $v = \text{Slow}$ and $\theta_c = \text{Pos. Small}$.

4.1.3 Slučaj sa preprekama na ekstremnim uglovima

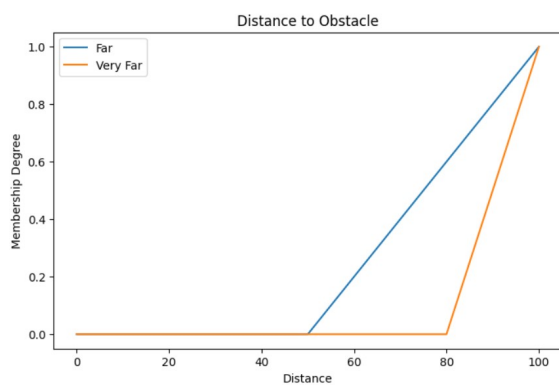
Kada su prepreke prisutne na ekstremnim uglovima (daleko od putanje) i ne predstavljaju pretnju za sudar, UAV treba da postupa na sledeći način:

- Pratiti putanju ka cilju:
Ako su prepreke daleko i ne predstavljaju neposrednu pretnju, UAV treba da se usmeri ka cilju.
- Smanjena brzina:
Da bi se izbeglo oštro skretanje ka potencijalnim preprekama, brzina treba da bude smanjena. Ovo omogućava lakše i sigurnije manevrisanje u okolini sa preprekama koje su van dometa senzora.
- Prioriteti:
 - Izbegavanje prepreka:
Kada su prepreke u dometu senzora, prioritet je da se izbegnu. UAV će prilagoditi brzinu i pravac kako bi zaobišao prepreke.
 - Planiranje putanje ka cilju:
Kada prepreke nisu u dometu senzora, prioritet je da se UAV usmeri ka cilju, uz manju brzinu kako bi se izbeglo brzo skretanje ka nepredviđenim preprekama.
- Pravila:
Ova pravila omogućavaju UAV-u da efikasno balansira između izbegavanja prepreka i praćenja optimalne putanje ka cilju.
- (27) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Big}$ and $\theta_t = \text{Neg. Big}$, then $v = \text{Fast}$ and $\theta_c = \text{Neg. Big}$.
- (28) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Big}$ and $\theta_t = \text{Neg. Med}$, then $v = \text{Fast}$ and $\theta_c = \text{Neg. Med}$.
- (29) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Big}$ and $\theta_t = \text{Neg. Small}$, then $v = \text{Fast}$ and $\theta_c = \text{Neg. Small}$.
- (30) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Big}$ and $\theta_t = \text{Zero}$, then $V = \text{Fast}$ and $\theta_c = \text{Zero}$.
- (31) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Big}$ and $\theta_t = \text{Pos. Small}$, then $v = \text{Fast}$ and $\theta_c = \text{Pos. Small}$.
- (32) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Big}$ and $\theta_t = \text{Pos. Med}$, then $v = \text{Fast}$ and $\theta_c = \text{Pos. Med}$.
- (33) If $D_o = \text{Far}$ and $\theta_o = \text{Pos. Big}$ and $\theta_t = \text{Pos. Big}$, then $v = \text{Fast}$ and $\theta_c = \text{Pos. Big}$.

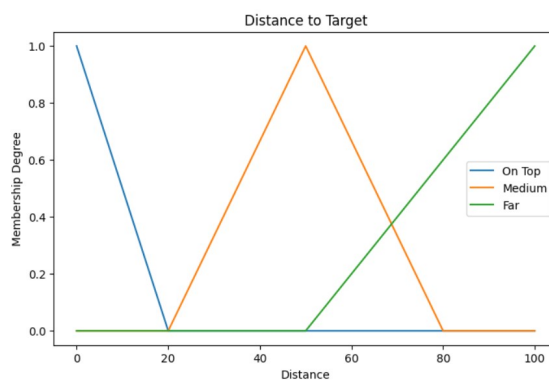
- (34) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Big}$ and $\theta_t = \text{Neg. Big}$, then $v = \text{Fast}$ and $\theta_c = \text{Neg. Big}$.
- (35) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Big}$ and $\theta_t = \text{Neg. Med}$, then $v = \text{Fast}$ and $\theta_c = \text{Neg. Med}$.
- (36) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Big}$ and $\theta_t = \text{Neg. Small}$, then $v = \text{Fast}$ and $\theta_c = \text{Neg. Small}$.
- (37) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Big}$ and $\theta_t = \text{Zero}$, then $v = \text{Fast}$ and $\theta_c = \text{Zero}$.
- (38) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Big}$ and $\theta_t = \text{Pos. Small}$, then $v = \text{Fast}$ and $\theta_c = \text{Pos. Small}$.
- (39) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Big}$ and $\theta_t = \text{Pos. Med}$, then $v = \text{Fast}$ and $\theta_c = \text{Pos. Med}$.
- (40) If $D_o = \text{Far}$ and $\theta_o = \text{Neg. Big}$ and $\theta_t = \text{Pos. Big}$, then $v = \text{Fast}$ and $\theta_c = \text{Pos. Big}$.

4.2 Grafici

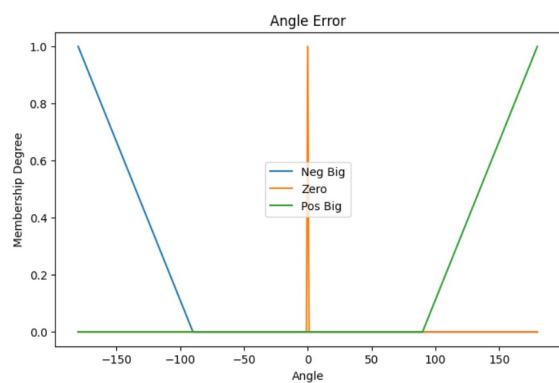
4.2.1 Slučaj bez prepreka



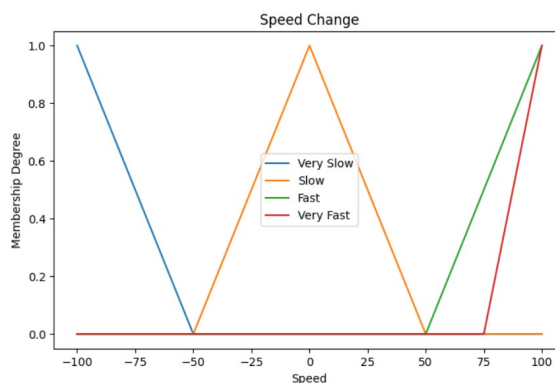
Slika 3: Rastojanje do prepreke



Slika 4: Rastojanje do cilja

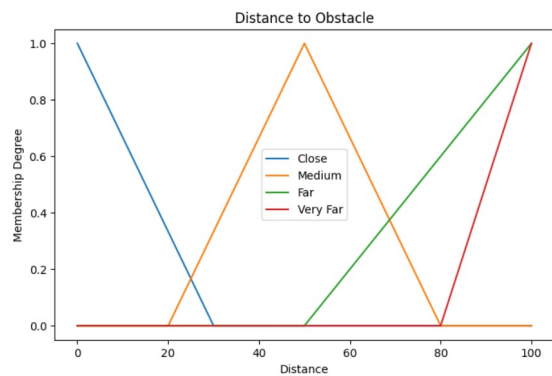


Slika 5: Greska ugla

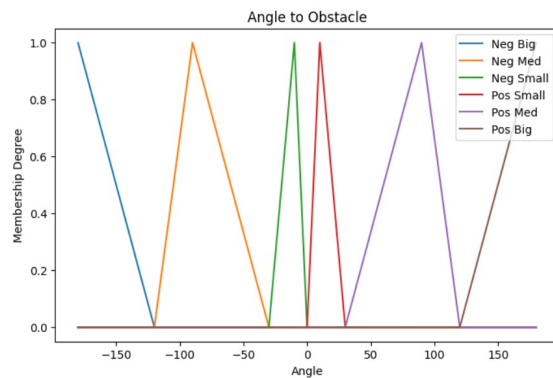


Slika 6: Promena brzine

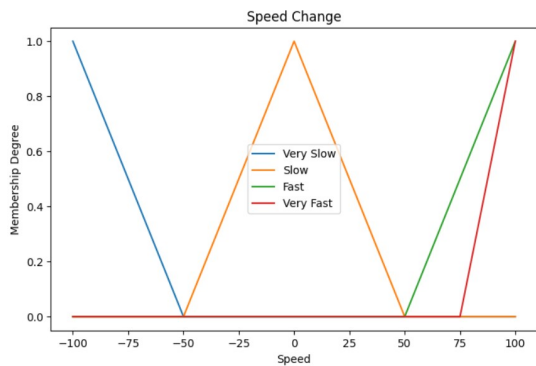
4.2.2 Slučaj sa detektovanjem prepreka u dometu senzora



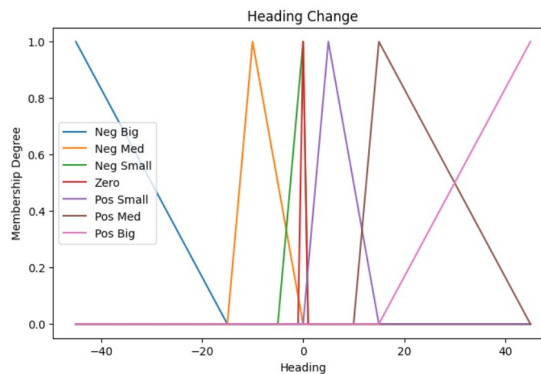
Slika 7: Rastojanje do prepreke



Slika 8: Ugao do prepreke

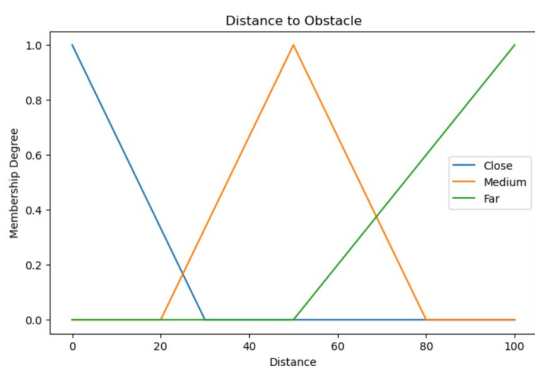


Slika 9: Promena brzine

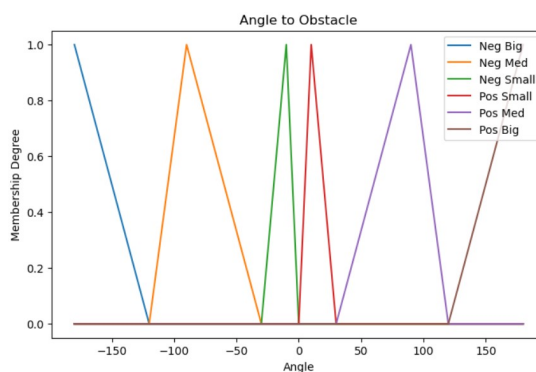


Slika 10: Promena ugla

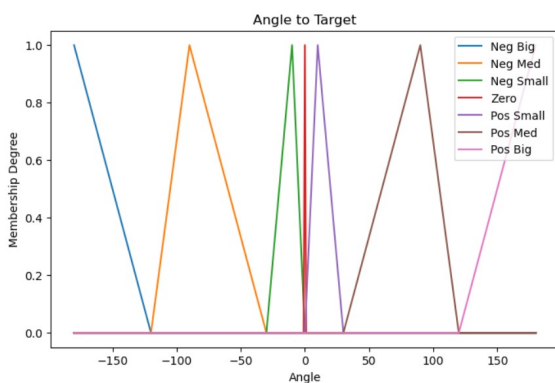
4.2.3 Slučaj sa preprekama na ekstremnim uglovima



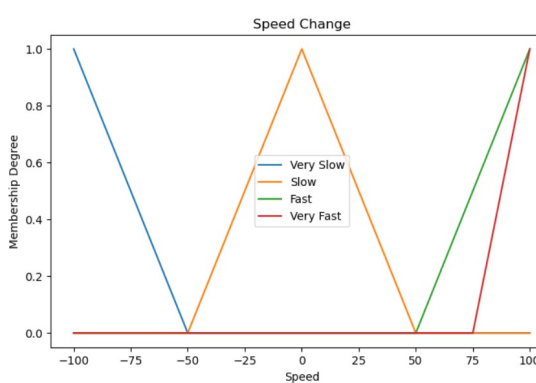
Slika 11: Rastojanje do prepreke



Slika 12: Ugao do prepreke



Slika 13: Ugao izmedju trenutne putanje i cilja



Slika 14: Promena brzine

5 Zaključak

Zaključak ovog istraživanja je da fuzzy logika predstavlja efikasno rešenje za planiranje kretanja bespilotnih letelica u nepoznatom i dinamičnom okruženju. Algoritam baziran na fuzzy logici omogućava letelici da izbegne prepreke i pronađe put ka cilju, dok se prilagođava promenama u okruženju. Testiranje je pokazalo da ovaj pristup ima visoku stopu uspeha i omogućava bolje upravljanje resursima letelice u realnom vremenu, što ga čini pouzdanim za primenu u stvarnim situacijama.

6 Literatura

- [1] Chelsea Sabo, Kelly Cohen. Fuzzy Logic Unmanned Air Vehicle Motion Planning
- [2] Justyna Walotek, Jagoda Oleksiak, Pawel Cebula, Adam Stanek, Mateusz Szczypinski. A Fuzzy Logic Based Autonomous Car Simulation in Unity