



Universitatea Politehnica din București
Facultatea de Inginerie Aerospațială
Ingineria Sistemelor Aeronautice „Nicolae Tîpei”

**Planificarea
Traectoriei de Zbor
pentru o Formație de
UAV-uri cu
Minimizarea
Consumului Energetic**

Teză de doctorat

Răzvan-Viorel Mihai

Conducător științific:

Prof. univ. dr. ing. Adrian-Mihail Stoica

București, Aprilie 2021

Cuprins

Cuprins	iii
Convenții Matematice Adoptate	v
1 Introducere în Problematika Planificării Traiectoriei de Zbor	1
1.1 Considerații generale privind metodele de planificare a traiectoriei de zbor pentru o formație de UAV-uri	2
1.2 Planificarea Cooperativă și Distributivă a Traiectoriei de Zbor	2
1.3 Scopul și obiectivele principale ale tezei de doctorat	3
2 Metode pentru Generarea și Planificarea Traiectoriei de Zbor	5
2.1 Metoda „Cell Decomposition”	5
2.2 Metoda „Road Map”	6
2.2.1 Graficul de vizibilitate	6
2.2.2 Diagrama Voronoi	6
2.3 Metoda câmpului de potențiale	6
2.4 Metodele RRT, RRT*, IRRT*	7
2.5 Metoda bazată pe interpolare de curbe	8
2.5.1 Curbe Bézier	8
2.5.2 Curbe B-Spline	8
2.6 Planificare pe bază de optimizare numerică	9
2.7 Concluzii	10
3 Modelarea unui Quadcopter	11
3.1 Forțe și momente ce acționează asupra quadcopterului	11
3.2 Reprezentarea atitudinii quadcopterului	12
3.3 Reprezentarea vitezelor unghiulare	12
3.4 Modelarea quadcopterului utilizând ecuațiile Newton-Euler	12
3.5 Concluzii	12
4 Controlul unui Quadcopter	13
4.1 Controlul altitudinii de zbor	14
4.2 Controlul mișcării 2D a quadcopterului	14
4.3 Controlul mișcării 3D	15
4.4 Concluzii	18

5	Proiectarea, Realizarea și Testarea Experimentală a Pilotului Automat pentru Quadcopter	19
5.1	Componente utilizate	19
5.2	Proiectarea, realizarea și programarea sistemului de comandă automată . .	20
5.3	Analiza datelor experimentale	21
5.3.1	Analiza rezultatelor zborului după urmărirea unei traiectorii 2D . .	23
5.3.2	Analiza rezultatelor zborului după urmărirea unei traiectorii 3D . .	25
5.4	Concluzii	26
6	Planificarea Traiectoriei de Zbor cu Minimizarea Consumului Energetic, în Absența Obstacolelor	27
6.1	Elemente pregătitoare pentru calculul traiectoriei de zbor folosind ecuațiile Euler-Lagrange	27
6.2	Derularea unei misiuni cu traiectorie de zbor planificată, în condițiile minimizării accelerației de translație/ forței de tracțiune care acționează asupra structurii quadcopterului	28
6.3	Derularea unei misiuni cu traiectorie de zbor planificată, în condițiile minimizării variației accelerației de translație/ forței de tracțiune sau a momentelor care acționează asupra structurii quadcopterului	31
6.4	Estimarea consumului energetic la derularea unor misiuni cu traiectorie de zbor planificată, în diferite scenarii de minimizare a variabilelor ce țin de dinamica quadcopterului	31
6.5	Concluzii	32
7	Planificarea Traiectoriei de Zbor cu Minimizarea Consumului Energetic, în Prezența Obstacolelor	33
7.1	Considerații teoretice privind optimizarea convexă	34
7.2	Planificarea traiectoriei cu evitarea obstacolelor statice , pentru un UAV care execută zbor solitar	34
7.3	Planificarea traiectoriei pentru zbor în formație, cu evitarea coliziunilor între membrii formației	40
7.4	Planificarea traiectoriei pentru zbor în formație, cu evitarea coliziunilor atât între membrii formației, cât și cu obstacolele statice	41
7.5	Concluzii	42
8	Rezultate și Concluzii	43
8.1	Concluzii generale	43
8.2	Contribuții	43
8.3	Perspective de dezvoltare ulterioară	46
	Bibliografie	47

Convenții Matematice Adoptate

Constante scalare: a, b, c, A, B, C etc.

Variabile scalare: a, b, c etc.

Constante vectoriale: $\mathbf{a}, \mathbf{b}, \mathbf{c}$, etc.

Variabile vectoriale: $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{A}, \mathbf{B}, \mathbf{C}$ etc.

Matrice: $\mathbf{A}, \mathbf{B}, \mathbf{C}$ etc.

Sisteme de coordonate: A, B, C , etc.

Matrice de transformare de la sistemul de coordonate B la A, ${}^A\mathbf{R}_B$

Timp continuu: $t \in \mathbb{R}$

Timp discret: $n \in \mathbb{Z}$ sau $k \in \mathbb{Z}$

Argument de timp continuu: (t)

Argument de timp discret: $[n]$ sau $[k]$

Funcții: f, \mathbf{g}

Capitolul 1

Introducere în Problematika Planificării Traiectoriei de Zbor

Literatura de specialitate prezintă o multitudine de soluții la problema de planificare a traiectoriei de zbor pentru Unmanned Aerial Vehicle (UAV)-uri, fiecare abordare prezentând propriile avantaje și dezavantaje, în funcție de particularitatea misiunii pe care acestea trebuie să o execute. Majoritatea abordărilor urmează schema bloc din fig. 1.1. Cele

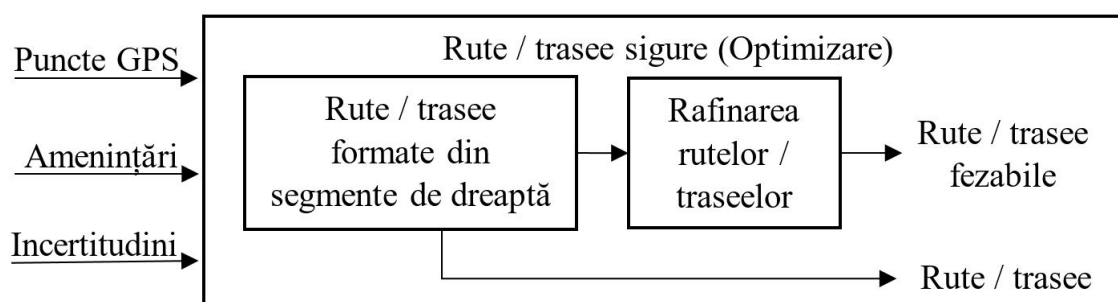


Figura 1.1: Diagramă cu abordarea problemei de planificare a traiectoriei de zbor

trei săgeți din partea stângă reprezintă datele de intrare ale componentei de planificare, respectiv punctele Global Positioning System (GPS) (puncte de navigație), amenințările care pot fi reprezentate de coliziunile cu obstacolele statice și dinamice și incertitudinile. Structura blocului „Rute/trasee sigure” sugerează faptul că obținerea traiectoriei de zbor are loc, de cele mai multe ori, în două faze:

- 1) Obținerea de rute /trasee simple, sub formă de linii drepte, obținute prin interconectarea directă a punctelor de navigație;
- 2) Rafinarea rutelor/traseelor simple în ideea obținerii unei traiectorii fezabile, de formă curbilinie.

1.1 Considerații generale privind metodele de planificare a traiectoriei de zbor pentru o formație de UAV-uri

Tabelul 1.1 [1] prezintă o sinteză cu lucrările din literatura de specialitate privind cei mai importanți algoritmi de planificare a traiectoriei.

Tabel 1.1: Revizuirea algoritmilor de planificare a traiectoriei [1]

Categorii de metode	Denumire metodă	Algoritmi asociați	Referințe
Planificare pe bază de căutare grafică	Metoda „Cell Decomposition”	Dijkstra	[2], [3], [4], [5], [6], [7]
	Metoda „Road Map”	A*	[8], [9], [10], [11], [12]
Planificare pe bază de funcții continue	Metoda câmpul de potențiale		[13], [14], [15], [16]
Planificare pe bază de eșantionare	Metoda Rapidly exploring Random Tree (RRT)		[17], [18]
	Metoda Rapidly Exploring Random Tree star (RRT*)		[19], [20], [21]
	Metoda Informed Rapidly Exploring Random Tree star (IRRT*)		[22], [23], [24]
Planificare pe bază de interpolare de curbe	Curbe Bézier		[25], [26]
	Curbe B-Spline		[27], [28], [29]
Planificare pe bază de optimizare numerică	Optimizare numerică		[12], [30], [31], [32]

1.2 Planificarea Cooperativă și Distribuțivă a Traiectoriei de Zbor

Planificarea traiectoriei de zbor pentru zborul în formație se poate realiza urmând două proceduri distincte: i) Planificarea Cooperativă; ii) Planificarea Distribuțivă.

1.3 Scopul și obiectivele principale ale tezei de doctorat

Scopul tezei de doctorat este acela de a studia problematica planificării traiectoriei de zbor plecând de la elementele teoretice de bază și trecând succesiv prin toate etapele necesare dezvoltării unei aplicații experimentale de planificare pentru o formație de UAV-uri: a) Studiul teoretic și numeric al metodelor de planificare a traiectoriei de zbor; b) Planificarea optimizată a traiectoriei de zbor pentru un singur UAV (elaborare model matematic UAV; proiectare, simulare numerică și validare experimentală sistem de comandă automată UAV; proiectare algoritmi de planificare a traiectoriei cu optimizarea consumului energetic în diferite scenarii de zbor, simularea numerică a acestora și implementarea rezultatelor de planificare obținute pe sistemul experimental de comandă automată a UAV-ului); c) Planificarea cooperativă optimizată a traiectoriei de zbor pentru o formație de UAV-uri (proiectare algoritmi de planificare a traiectoriei cu optimizarea consumului energetic în diferite scenarii de zbor și validarea acestora prin simulare numerică). Pentru proiectarea și validarea algoritmilor de planificare a traiectoriei am conceput o misiune de zbor care să permită parcurgerea tuturor etapelor anterior menționate. În cadrul acestei misiuni de zbor scenariul de lucru pe care l-am urmat este prezentat schematic în fig. 1.2. Dintre detaliile

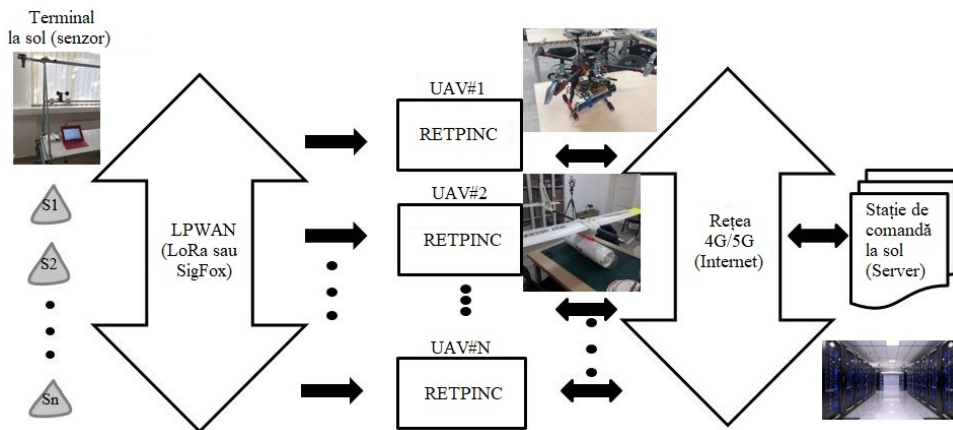


Figura 1.2: Scenariu de lucru adoptat în cadrul misiunii de zbor [32]

proiectării și derulării misiunii de zbor, se pot menționa:

- Utilizarea unei rețele de n senzori S_1, \dots, S_n amplasată pe un teren agricol, care trebuie să furnizeze date utilizatorului uman pentru a monitoriza starea recoltei și pentru a face predicții asupra producției maxime;
- Raza de acțiune pentru transmiterea wireless a informațiilor de stare privind recolta este limitată pentru fiecare senzor în parte, funcție de tehnologia utilizată pentru a realiza transmisia;
- Formația de UAV-uri survolează zona în care sunt amplasați senzorii [33] pentru a culege datele achiziționate de către aceștia, pe baza unui traseu de zbor, generat de

algoritmul de planificare a traiectoriei, care rulează la bordul fiecărui membru. Blocul RETPINC (REal Time Processing and INternet Communication Unit) echipează fiecare dintre membrii formației de zbor [34];

- Prin intermediul formației de UAV-uri, datele de la senzori ajung la stația de comandă, utilizând o legătură wireless de mare viteză 4G/5G;
- Prin intermediul aceleiași rețele de comunicații 4G/5G, Stația de Comandă la Sol (SCS) primește informații despre starea curentă P_t^i (unde t reprezintă momentul de timp la care se face recepția, iar i , reprezintă indexul/numărul UAV-ului) a fiecărui membru al formației și despre obstacolele detectate în zbor. Pe baza acestor informații sunt generate seturi de manevre pentru UAV-urile implicate într-un conflict de coliziune.

În strânsă corelație cu cele prezentate mai sus, pe parcursul tezei de doctorat îmi propun să ating următoarele obiective:

- OB1:** Prezentarea teoretică și simularea numerică a metodelor de generare și planificare a traiectoriei de zbor;
- OB2:** Modelarea unui membru al formației de zbor de tip quadcopter;
- OB3:** Descrierea unui controler de zbor care să permită facil implementarea software;
- OB4:** Validarea modelului controlerului de zbor prin simulări numerice;
- OB5:** Realizarea experimentală a pilotului automat pentru UAV-ul de tip multirotor;
- OB6:** Testarea experimentală a pilotului automat dezvoltat;
- OB7:** Planificarea, calcularea și simularea numerică a unei misiuni de zbor în diferite scenarii de minimizare a variabilelor ce țin de dinamica aparatului de zbor;
- OB8:** Dezvoltarea unei metode de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru un quadcopter în absența obstacolelor;
- OB9:** Evaluarea prin simulări numerice și demonstrarea performanțelor algoritmului de planificare a traiectoriei de zbor cu minimizarea consumului energetic, în absența obstacolelor;
- OB10:** Dezvoltarea unei metode de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru un quadcopter în prezența obstacolelor;
- OB11:** Evaluarea prin simulări numerice și demonstrarea performanțelor algoritmului de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru un quadcopter, în prezența obstacolelor;
- OB12:** Dezvoltarea unei metode de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor, în absența obstacolelor;
- OB13:** Evaluarea prin simulări numerice și demonstrarea performanțelor algoritmului de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor, în absența obstacolelor;
- OB14:** Dezvoltarea unei metode de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor în prezența obstacolelor;
- OB15:** Evaluarea prin simulări numerice și demonstrarea performanțelor algoritmului de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor, în prezența obstacolelor.

Capitolul 2

Metode pentru Generarea și Planificarea Traectoriei de Zbor

Metodele expuse, dezvoltate inițial în cadrul aplicațiilor cu roboți tereștri, au fost preluate și perfecționate în timp în cadrul aplicațiilor cu vehicule aeriene, cu sau fără echipaj uman [35]. Acestea se încadrează în cinci mari categorii, expuse pe scurt în Table 1.1, fiind utilizate în primul rând la generarea hărților pentru algoritmul de planificare a traiectoriei de zbor. Pentru generarea rezultatelor în urma aplicării metodelor studiate în cadrul prezentului capitol am folosit resursele hardware și software din Table 2.1.

Tabel 2.1: Resurse utilizate pentru generarea rezultatelor în urma aplicării metodelor enumerate în [35]

Nr. crt.	Denumire resursa	Tip resursa
1	Procesor i7-8750H@2.20GHz	hardware
2	16 GB RAM	hardware
3	x64bit Ubuntu 18.04	sistem de operare
4	Visual Studio Code	mediu de dezvoltare
5	Python3	limbaj de programare

2.1 Metoda „Cell Decomposition”

În cazul acestei metode, spațiul de lucru este împărțit în celule. La pasul următor sunt generate rutele posibile care trec prin celule adiacente libere. Celulele libere sunt cele care nu sunt ocupate de obstacole. Obstacolele sunt izolate prin identificarea conectivității între celulele libere. Astfel se produce o versiune discretă a mediului. Un algoritm de căutare este utilizat pentru a conecta celulele adiacente libere. Celulele umbrite (încrucșate) sunt eliminate, deoarece sunt ocupate de obstacole (gri).

2.2 Metoda „Road Map”

Aplicarea acestei metode reprezintă prima etapă a componentei de planificare a traiectoriei. Metoda presupune crearea unei rețele bidimensionale de linii drepte care leagă punctele inițiale și finale fără a intersecta obstacolele definite în hartă. Cu alte cuvinte, este o reprezentare a unui spațiu 2D. UAV-ul este tratat ca un punct, iar spațiul de lucru este dimensionat pentru a respecta scara, prin raportare la dimensiunea fizică a vehiculului aerian. Pentru un punct de pornire dat \mathbf{p}_s și punctul de final \mathbf{p}_f , sunt generate toate rutele posibile de conectare evitând obstacolele. Se definește o metrică M^* , care va uni rutele generate. Această metrică poate considera spre exemplu traseul cel mai scurt dintre cele două puncte, numărul minim de rute interconectate, etc.

2.2.1 Graficul de vizibilitate

După cum sugerează și numele, graficul de vizibilitate produce o linie de vizibilitate directă printr-un mediu. Aceasta este una dintre cele mai vechi metode folosite pentru planificarea traiectoriei. Pentru această metodă, numai obstacolele poligonale sunt considerate. Pentru un grafic $G = (\mathbf{V}, \mathbf{E})$, unde \mathbf{V} reprezintă vârfurile obstacolelor, în timp ce \mathbf{E} reprezintă marginile obstacolelor (nodurile conectate cu linii drepte). Prin urmare, numai nodurile care sunt vizibile (în sensul că fiecare vârf poate fi văzut altul) sunt incluse în rețeaua de conectivitate.

2.2.2 Diagrama Voronoi

O diagramă Voronoi este un grafic de conectivitate generat prin formarea de poligoane în jurul obstacolelor. Fiecare margine a poligoanelor este definită construind mai întâi un set de linii care leagă centrele obstacolelor. Setul de poligoane este apoi construit prin desenarea unui set de linii perpendiculare pe liniile care unesc obstacolele. Acestea sunt apoi ajustate să se întâlnească la un set minim de noduri.

2.3 Metoda câmpului de potențiale

Această metodă presupune o descriere specifică a hărții sau a spațiului de lucru, astfel încât obstacolele cunoscute să fie descrise de forma unor poligoane. De data aceasta, spațiul de lucru nu va mai fi descris discret, ca o hartă de celule, ci folosind funcții continue de forma:

$$U(q) = U_{atr}(q) + U_{res}(q) \quad (2.1)$$

în care $U_{atr}(q)$ caracterizează funcția ce descrie „atracția” dintre „potențialul” punctului de start și a celui de final, în timp ce $U_{res}(q)$ caracterizează funcția de „respingere” față de obstacole de-a lungul traiectoriei [13].

2.4 Metodele RRT, RRT*, IRRT*

Această metodă face parte din categoria de metode de planificare a traiectoriei, utilizând puncte din spațiul de lucru, eșantionate aleatoriu. Conform [36], atât metoda RRT cât și formele derivate ale acestui algoritm respectă procedura din algoritmul (1). Algoritmul

Algorithm 1 Algoritm RRT [37]

```

function ALGORITMRRT( $\mathbf{p}_{initial}$ )
     $\mathbf{T} \leftarrow \text{InitializareRetea}()$ ;
     $\mathbf{T} \leftarrow \text{InsereazaPunct}(\emptyset, \mathbf{p}_{initial}, \mathbf{T})$ ;
     $i \leftarrow 0$ ;
     $N \leftarrow \text{FixatDeUtilizator}()$ ;
    loop:
        if  $i \leq N$  then
             $\mathbf{p}_{aleatoriu} \leftarrow \text{Esantioneaza}(i)$ ;
             $\mathbf{p}_{apropiat} \leftarrow \text{DeterminaPunctApropiat}(\mathbf{T}, \mathbf{p}_{aleatoriu})$ ;
             $\mathbf{p}_{nou} \leftarrow \text{AdaugaPunct}(\mathbf{p}_{apropiat}, \mathbf{p}_{aleatoriu})$ ;
            if  $\text{VerificaColiziune}(\mathbf{p}_{nou})$  then
                 $\mathbf{T} \leftarrow \text{InsereazaPunct}(\mathbf{p}_{min}, \mathbf{p}_{nou}, \mathbf{T})$ ;

```

de mai sus debutează prin funcția *InitializareRetea*, cu rol în inițializarea graficului sau a rețelei care va conține ca punct de start, $\mathbf{p}_{initial}$, după execuția funcției *InsereazaPunct*. Variabila i este folosită pentru a contabiliza numărul de iterații curente, număr care va fi comparat cu o valoare de referință fixată de utilizator prin intermediul variabilei N . Cum RRT este un algoritm ce se bazează pe selecția aleatorie a punctelor din spațiul de lucru, funcția *Esantioneaza* îndeplinește următoarea acțiune din cadrul algoritmului (1). Apoi, se determină cel mai apropiat punct din rețeaua existentă \mathbf{T} , față de $\mathbf{p}_{aleatoriu}$. Punctul \mathbf{p}_{nou} care se adaugă la rețeaua existentă rezultă în urma selecției dintre $\mathbf{p}_{apropiat}$ și $\mathbf{p}_{aleatoriu}$, executând funcția *AdaugaPunct*, care evaluează care din cele două puncte este mai apropiat de punctul de destinație al traiectoriei.

În final, se verifică dacă punctul nou obținut se află în afara spațiului ocupat de obstacole, iar în caz afirmativ, acesta este adăugat la rețeaua \mathbf{T} . Plecând de la algoritmul (1), au fost dezvoltate algoritmi RRT* [21], IRRT* [23], etc., iar rezultatele numerice pe care le-am obținut sunt ilustrate în Figure 2.1.

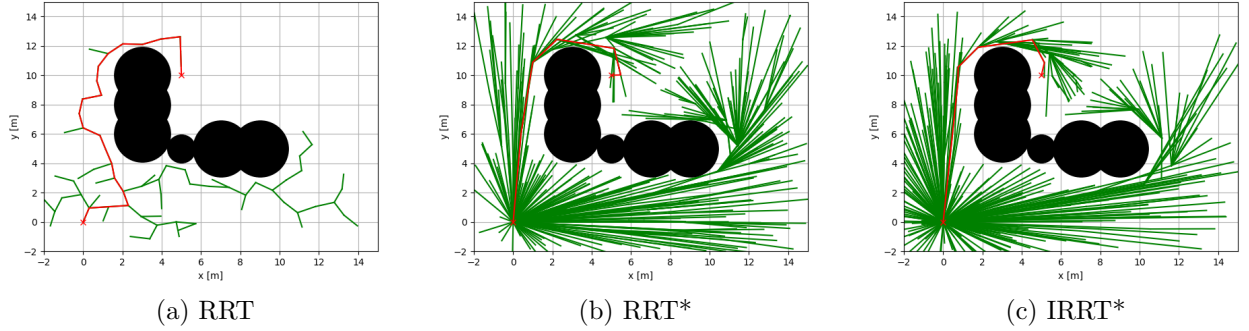


Figura 2.1: Metodele RRT, RRT*, IRRT*

2.5 Metoda bazată pe interpolare de curbe

2.5.1 Curbe Bézier

Conform lucrării [38], obținerea unei traiectorii de zbor de formă curbilinie, poate fi generată folosind curbele Bézier. Forma generală a expresiei pentru o curbă Bézier de ordin n este:

$$P(t) = \sum_{i=0}^n B_i^n(t) \mathbf{P}_i, \quad t \in [0, 1], \quad (2.2)$$

în care \mathbf{P}_i reprezintă punctele de control astfel încât $\mathbf{P}(0) = \mathbf{P}_0$ și $\mathbf{P}(1) = \mathbf{P}_n$, iar $B_i^n(t)$ este un polinom de forma:

$$B_i^n(t) = C_n^i (1-t)^{n-i} t^i, \quad i \in 0, 1, \dots, n. \quad (2.3)$$

Folosirea curbelor Bézier pentru calcularea traiectoriei unui robot prezintă următoarele avantaje:

- Traiectoria va include punctele vizate de la \mathbf{P}_0 către \mathbf{P}_n ;
- Curbele care unesc punctele $\mathbf{P}_0 \rightarrow \mathbf{P}_1 \rightarrow \dots \rightarrow \mathbf{P}_{n-1} \rightarrow \mathbf{P}_n$ sunt tangente la segmentele corespunzătoare, în punctele \mathbf{P}_0 și respectiv \mathbf{P}_n ;

2.5.2 Curbe B-Spline

Dezavantajul pe care îl prezintă traiectoria determinată cu ajutorul algoritmului specific interpolării de curbe Bézier constă în faptul că punctele de control pe baza cărora este generată curba Bézier nu vor fi vizitate de UAV. Chiar dacă planificarea traiectoriei folosind curbe „B-spline” nu rezolvă problema vizitării punctelor de control, avantajul pe care îl oferă această metodă asupra curbei Bézier se referă la eficientizarea timpului de execuție al algoritmului, acesta nefiind afectat de creșterea numărului de puncte de control.

2.6 Planificare pe bază de optimizare numerică

Tehnicile de actualitate bazate pe optimizare numerică, pentru rezolvarea problemei de planificare a traiectoriei sunt reprezentate de metode de căutare în spațiul de lucru precum Mixed Integer Linear Programming (MILP), algoritmi genetici [39], etc. MILP este o metodă de optimizare numerică, denumită programare liniară cu restricții întregi sau binare. Aceste restricții sunt utilizate pentru deciziile logice, cum ar fi spre exemplu „viraj la stânga” sau „viraj la dreapta”. Această metodă are rol în generarea unei traiectorii sigure pentru un vehicul sau pentru o formație de vehicule, în funcție de aplicație [40], [41], [42], [43].

În figura Figure 2.2 este ilustrat un scenariu de zbor pentru un UAV, a cărei traiectorie calculată și simulată este evidențiată prin intermediul curbei colorate cu albastru. Punctele de start și de final ale misiunii de zbor sunt reprezentate folosind simboluri de formă triunghiulară și respectiv circulară, colorate cu magenta. Cele trei obstacole statice cu laturile de culoare neagră au fost adăugate suplimentar pentru a evalua performanțele metodei de planificare a traiectoriei de zbor în condiții de evitare a coliziunilor cu acestea. Condiția de evitare a obstacolelor statice este satisfăcută pe durata întregii simulări, iar acest lucru se poate observa în figura 2.2c.

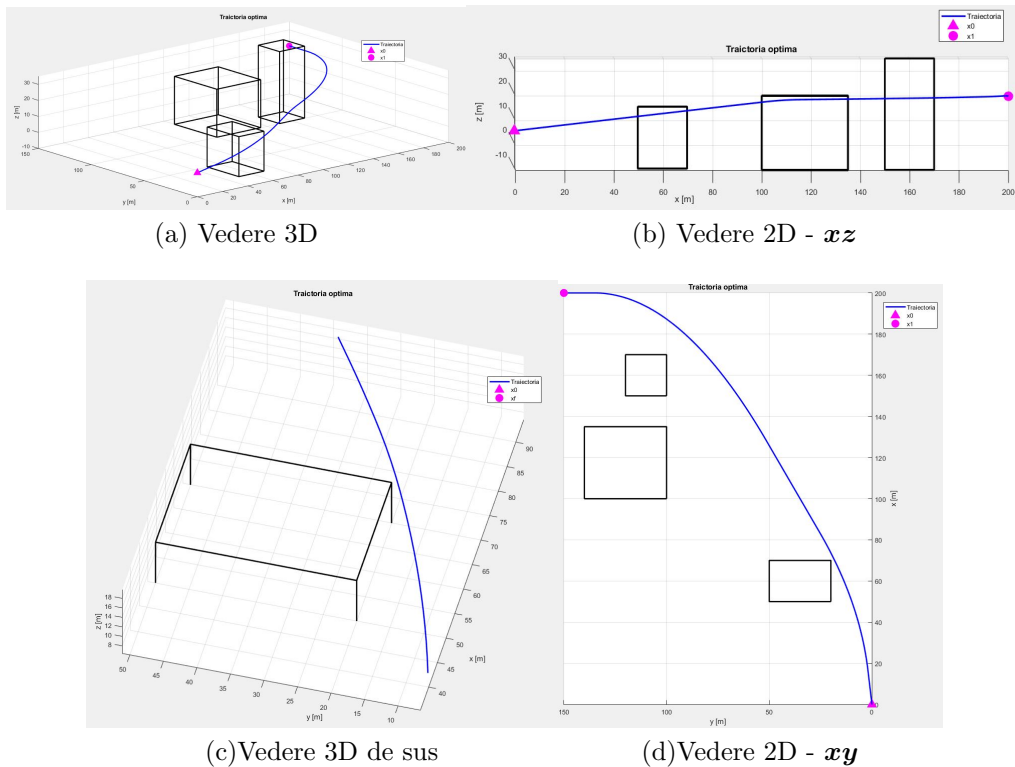


Figura 2.2: Traiectoria de zbor fezabilă pentru un UAV, calculată cu metoda MILP [32]

2.7 Concluzii

Tabel 2.2: Avantajele și dezavantajele algoritmilor de planificare a traiectoriei.

Metodă sau algoritm	Avantaj	Dezavantaj
Dijkstra	Determină traiectoria cea mai scurtă pe baza punctelor sau nodurilor unei hărți.	Căutarea rutelor nu este heuristică. Algoritmul rulează din ce în ce mai lent odată cu creșterea numărului de destinații. Traiectoria obținută nu este continuă.
A^*	Se bazează pe algoritmul Dijkstra. Utilizează o funcție heuristică, ceea ce duce la reducerea timpului de execuție.	Traiectoria obținută nu este continuă. Problema de planificare a traiectoriei de zbor nu este rezolvabilă de fiecare dată.
RRT, RRT*, IRRT*	Generează soluții chiar și în cazul sistemelor multidimensionale. Converge constant către soluție și identifică traiectoria dacă aceasta există și dacă timpul alocat este suficient	Traiectoria obținută nu este continuă. Depinde semnificativ de timpul alocat pentru execuție.
Curbe Polinomiale	Eficiente ca și timp de execuție. Concatenarea mai multor curbe este posibilă.	Determinarea coeficienților optimi, specifici curbelor de ordin ≥ 4 este dificilă.
Curbe Bezier	Eficiente ca și timp de execuție. Curbura se poate manipula intuitiv prin intermediul punctelor de control.	Odată cu creșterea ordinului curbei, crește și timpul de execuție. Utilizat cu precădere în spațiul bidimensional.
Curbe B-Spline	Eficiente ca și timp de execuție. Traiectorie continuă și controlabilă prin intermediul punctelor de control.	Obținerea unei curburi continue poate încălca restricțiile formulate. De evitat la planificarea traiectoriilor în medii cu obstacole.
Optimizare numerică	Consideră toate restricțiile și minimizează funcția cost formulată.	Necesită o resursă de calcul semnificativă deoarece algoritmul rulează la intervale de timp de ordinul milisecundelor.

Rezultatele numerice, obținute pe baza algoritmilor rezultați din metodele de planificare a traiectoriei, specificul misiunii de zbor, dar și modelul UAV-ului ca membru al formației de zbor, au dus la considerarea algoritmului IRRT* în vederea dezvoltării acestuia, adăugând elemente de optimizare numerică.

Capitolul 3

Modelarea unui Quadcopter

În imaginea din Figure 3.1 sunt evidențiate sensurile de rotație ale elicelor, ale căror viteze de rotație sunt notate cu ω_i , forțele generate de ansamblele motor-elice, notate cu \mathbf{F}_i , cuplul rezistent al elicelor, notat cu \mathbf{M}_i , lungimile celor patru brațe, notate cu l_i ; $i = 1 \div 4$. Versorii axelor $\mathbf{x}\mathbf{y}\mathbf{z}$ sunt notați cu $\mathbf{a}_x\mathbf{a}_y\mathbf{a}_z$, iar unghiurile de atitudine, caracterizând mișcările de rulu, tangaj și girație, sunt notate cu ϕ, θ, ψ .

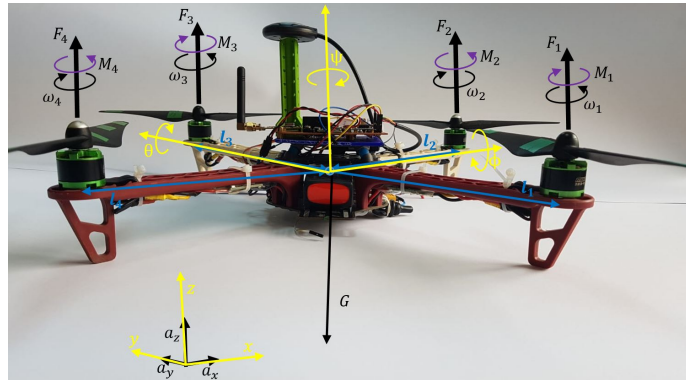


Figura 3.1: Reprezentarea sensurilor de rotație a motoarelor și direcția forțelor de tracțiune rezultate

3.1 Forțe și momente ce acționează asupra quadcopterului

Forța de tracțiune se scrie ca $\mathbf{F} = \sum_{i=1}^4 \mathbf{F}_i - \mathbf{G} = \sum_{i=1}^4 \mathbf{F}_i - m\mathbf{g}$, în care m reprezintă masa quadcopterului, iar \mathbf{g} accelerația gravitațională. Momentele în jurul centrului de greutate al quadcopterului, considerând fiecare braț al structurii, pot fi scrise ca fiind $\mathbf{M} = \sum_{i=1}^4 (r_i \mathbf{F}_i + k_M \omega_i^2)$. La echilibru (situația zborului la punct fix și în absența mișcării de girație), rezultanta forțelor și momentul rezultat care acționează asupra structurii quadcopterului sunt zero.

3.2 Reprezentarea atitudinii quadcopterului

Pentru a descrie mișcarea tridimensională, caracterizată atât de poziția cât și de orientarea UAV-ului, am utilizat matricea de transformare

$$\begin{bmatrix} \mathbf{p}_1' \\ \mathbf{p}_2' \\ \mathbf{p}_3' \end{bmatrix} = \begin{bmatrix} c\psi c\theta - s\phi s\phi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}. \quad (3.1)$$

3.3 Reprezentarea vitezelor unghiulare

În ceea ce privește viteza de rotație a unui sistem de coordonate avem

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{r} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\cos \phi \sin \theta \\ 0 & 1 & \sin \phi \\ \sin \theta & 0 & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad (3.2)$$

în care p, q, r sunt viteze unghiulare în sistemul de coordonate legat de aparat.

3.4 Modelarea quadcopterului utilizând ecuațiile Newton-Euler

Modelul neliniar al quadcopterului este

$$\begin{cases} \ddot{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^4 \mathbf{F}_i (\cos \psi \sin \theta + \cos \theta \sin \phi \sin \psi) \\ \ddot{\mathbf{y}} = \frac{1}{m} \sum_{i=1}^4 \mathbf{F}_i (\sin \psi \sin \theta - \cos \psi \cos \theta \sin \phi) \\ \ddot{\mathbf{z}} = \frac{1}{m} \sum_{i=1}^4 \mathbf{F}_i (\cos \phi \cos \theta) - \mathbf{g} \\ \dot{\mathbf{p}} = \frac{\ell_{k_F}(\omega_2^2 - \omega_4^2)}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} (\dot{\psi} + \theta \sin \phi) (\dot{\phi} \sin \psi + \theta \cos \phi \cos \psi) \\ \dot{\mathbf{q}} = \frac{\ell_{k_F}(\omega_3^2 - \omega_1^2)}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} (\dot{\psi} + \theta \sin \phi) (\dot{\phi} \cos \phi - \theta \cos \phi \sin \psi) \\ \dot{\mathbf{r}} = \frac{k_M(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} (\dot{\phi} \sin \psi + \theta \cos \phi \cos \psi) (\dot{\phi} \cos \psi - \theta \cos \phi \sin \psi) \end{cases} \quad (3.3)$$

3.5 Concluzii

Capitolul a vizat atingerea obiectivului **OB2** referitor la modelarea quadcopterului, membru al formației de zbor. Rolul modelului obținut este acela de a ajuta la proiectarea sistemului de comandă și control automat ce va echipa aparatul de zbor, permițând descrierea evoluției acestuia în spațiul tridimensional, atât din punct de vedere al mișcării de translație, cât și din punct de vedere al mișcării de rotație.

Capitolul 4

Controlul unui Quadcopter

Metodologia de control utilizată în această lucrare de cercetare, este evidențiată în schema bloc din Figure 4.1, în care bucla interioară desenată cu linie punctată albastră are rol în controlul atitudinii, iar bucla exterioară desenată cu linie punctată verde are rol în controlul poziției. Comanda \mathbf{u}_F este

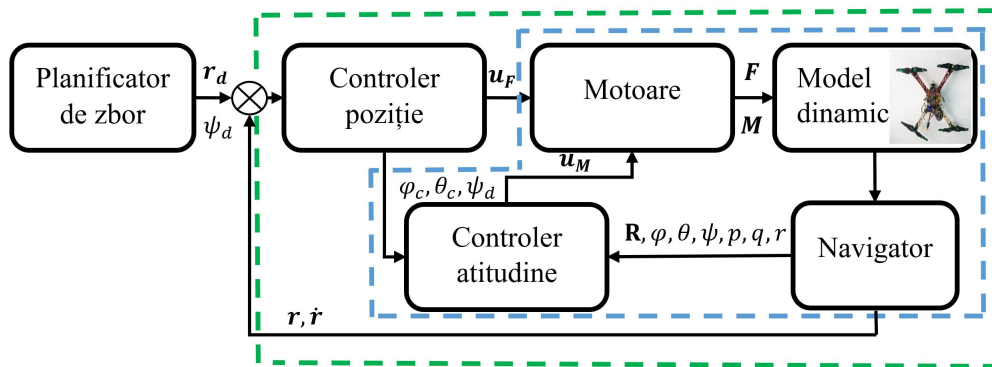


Figura 4.1: Schema bloc a controlului în atitudine și în poziție

$$\mathbf{u}_F = m\mathbf{g} + m(\ddot{\mathbf{z}}_d + K_{dz}(\dot{\mathbf{z}}_d - \dot{\mathbf{z}}) + K_{pz}(\mathbf{z}_d - \mathbf{z})), \quad (4.1)$$

iar \mathbf{u}_M

$$\mathbf{u}_M = \begin{bmatrix} u_{Mx} \\ u_{My} \\ u_{Mz} \end{bmatrix} = \begin{bmatrix} I_{xx}(\ddot{\phi}_c + K_{d\phi}(\dot{\phi}_c - \dot{\phi}) + K_{p\phi}(\phi_c - \phi)) \\ I_{yy}(\ddot{\theta}_c + K_{d\theta}(\dot{\theta}_c - \dot{\theta}) + K_{p\theta}(\theta_c - \theta)) \\ I_{zz}(\ddot{\psi}_c + K_{d\psi}(\dot{\psi}_c - \dot{\psi}) + K_{p\psi}(\psi_c - \psi)) \end{bmatrix}. \quad (4.2)$$

Vom împărți problema controlului quadcopterului în subproblema mișcării pe o direcție (1D), apoi se va adăuga mișcarea de ruliu/tangaj în planul \mathbf{xz} sau \mathbf{yz} (2D) în vederea determinării setului de coeficienți K_{px} , K_{dx} și $K_{p\theta}$, $K_{d\theta}$, sau K_{py} , K_{dy} și $K_{p\phi}$, $K_{d\phi}$. Mișcarea compusă 1D+2D ne va permite să realizăm ajustarea finală a tuturor coeficienților din expresiile (4.1) și (4.2).

4.1 Controlul altitudinii de zbor

Dorim să executăm traiectoria

$$\begin{bmatrix} \mathbf{r}(0) \\ \psi(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{r}_d(T) \\ \psi_d(T) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2.2m \\ 1rad \end{bmatrix}, \quad (4.3)$$

ceea ce semnifică decolarea la momentul $t = 0$, iar apoi deplasarea și rotirea de-a lungul și în jurul axei \mathbf{z} până la $t = T$, moment la care $\mathbf{z}_d = 2.2m$ și $\psi_d = 1rad$. În acest caz, legile de control sunt

$$\mathbf{u}_{1D} = \begin{bmatrix} \mathbf{u}_F \\ \mathbf{u}_{M,1D} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} + m(\ddot{\mathbf{z}}_d + K_{dz,1D}(\dot{\mathbf{z}}_d - \dot{\mathbf{z}}) + K_{pz,1D}(\mathbf{z} - \mathbf{z}_d)) \\ 0 \\ 0 \\ \mathbf{I}_{zz}(\ddot{\psi}_d + K_{d\psi,1D}(\dot{\psi}_d - \dot{\psi}) + K_{p\psi,1D}(\psi_d - \psi)) \end{bmatrix}, \quad (4.4)$$

în care $K_{pz}, K_{p\psi} \in \mathbb{R}$, $K_{pz}, K_{p\psi} > 0$ sunt coeficienții proporționali, iar $K_{dz}, K_{d\psi} \in \mathbb{R}$, $K_{dz}, K_{d\psi} > 0$ sunt coeficienții derivativi.

Specificațiile tehnice ale quadcopterului care au fost utilizate în cadrul simulărilor pentru a evalua performanțele legilor de control în fiecare din scenariile 1D, 2D și 3D, sunt evidențiate în Table 4.1. Rezultatele au rămas neschimbate chiar dacă valoarea tensorului de inerție \mathbf{I}_C a fost modificată pentru a utiliza cele trei seturi de valori conform lucrărilor de cercetare menționate în Table 4.1.

Tabel 4.1: Specificații tehnice quadcopter

Parametru	Descriere	Valoare
l_1, l_2, l_3, l_4	Lungimea brațelor	0.225m
m	Masa la decolare	1540g
\mathbf{I}_C	Matricea cu momentele de inerție	[44], [45], [46]
ω_{max}	Viteza maximă de rotație a unei elice	7200 RPM (Rotații pe minut)
\mathbf{F}_{max}	Forța totală maximă de tracțiune	30.8N

4.2 Controlul mișcării 2D a quadcopterului

În cazul mișcării în plan, alegem să determinăm valorile coeficienților legilor de control pentru mișcarea de rulu $K_{p\phi}, K_{d\phi}$. Efectul acestei mișcări va duce la o mișcare de translație de-a lungul axei \mathbf{y} . Adăugarea mișcării în planul \mathbf{xy} la traiectoria descrisă prin (4.3),

implică urmărirea traiectoriei

$$\begin{bmatrix} \mathbf{r}(0) \\ \psi(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{r}_d(\mathbf{T}) \\ \psi_d(T) \end{bmatrix} = \begin{bmatrix} 0 \\ 1m \\ 2.2m \\ 0 \end{bmatrix} \quad (4.5)$$

la care a fost adăugată străbaterea lungimii de 1m de-a lungul axei \mathbf{y} . A fost necesară eliminarea mișcării de rotație pentru a nu influența mișcarea de translație urmărită, iar în acest sens am modificat funcția $\text{linie}(\mathbf{t})$ pentru a fixa $\psi_d = \dot{\psi}_d = \ddot{\psi}_d = 0$. Legile de control pentru acest scenariu sunt

$$\mathbf{u}_{2D} = \begin{bmatrix} \mathbf{u}_F \\ \mathbf{u}_{M,2D} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} + m(\ddot{\mathbf{z}}_d + \mathbf{K}_{dz,1D}(\dot{\mathbf{z}}_d - \dot{\mathbf{z}}) + \mathbf{K}_{pz,1D}(\mathbf{z} - \mathbf{z}_d)) \\ I_{xx}(\mathbf{K}_{d\phi}(\dot{\phi}_c - \dot{\phi}) + \mathbf{K}_{p\phi}(\phi_c - \phi)) \\ 0 \\ I_{zz}(\ddot{\psi}_d + \mathbf{K}_{d\psi,1D}(\dot{\psi}_d - \dot{\psi}) + \mathbf{K}_{p\psi,1D}(\psi_d - \psi)) \end{bmatrix}. \quad (4.6)$$

4.3 Controlul mișcării 3D

Traectoria urmărită a fost completată astfel încât

$$\begin{bmatrix} \mathbf{r}(0) \\ \psi(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{r}_d(\mathbf{T}) \\ \psi_d(T) \end{bmatrix} = \begin{bmatrix} 1m \\ 1m \\ 2.2m \\ 0 \end{bmatrix}, \quad (4.7)$$

iar după cum se poate observa, a fost adăugată și mișcarea de-a lungul axei \mathbf{x} .

$$\mathbf{u}_{3D} = \begin{bmatrix} \mathbf{u}_F \\ \mathbf{u}_{M,3D} \end{bmatrix} = \begin{bmatrix} m\mathbf{g} + m(\ddot{\mathbf{z}}_d + \mathbf{K}_{dz,1D}(\dot{\mathbf{z}}_d - \dot{\mathbf{z}}) + \mathbf{K}_{pz,1D}(\mathbf{z} - \mathbf{z}_d)) \\ I_{xx}(\ddot{\phi}_c + \mathbf{K}_{d\phi}(\dot{\phi}_c - \dot{\phi}) + \mathbf{K}_{p\phi}(\phi_c - \phi)) \\ I_{yy}(\ddot{\theta}_c + \mathbf{K}_{d\theta}(\dot{\theta}_c - \dot{\theta}) + \mathbf{K}_{p\theta}(\theta_c - \theta)) \\ I_{zz}(\ddot{\psi}_d + \mathbf{K}_{d\psi,1D}(\dot{\psi}_d - \dot{\psi}) + \mathbf{K}_{p\psi,1D}(\psi_d - \psi)) \end{bmatrix} \quad (4.8)$$

Rezultatele evidențiate cu ajutorul graficelor din Figure 4.3, Figure 4.4 precum și urmărirea traiectoriei (ilustrată în Figure 4.2) cu o precizie de 0.37% confirmă modul în care a fost abordată și rezolvată problema controlului quadcopterului. În final, s-a adăugat mișcarea de rotație corespunzătoare expresiei (4.3), și în modelul descris de (4.7), ceea ce a dus la scăderea preciziei cu care a fost urmărită traiectoria de zbor cu aproximativ 0.001%. Așadar performanțele legii de control din (4.8) se mențin.

Algoritm 2 Controler de tip PD pentru controlul altitudinii și mișcării de ruliu și tangaaj

```

function CONTROLER3D(stare_curentă, stare_dorită, parametri_quad)
    ( $K_{pz}$ ,  $K_{dz}$ )  $\leftarrow$  (1.4, 0.75);
    ( $K_{p\psi}$ ,  $K_{d\psi}$ )  $\leftarrow$  (4, 0);
    ( $K_{p\phi}$ ,  $K_{d\phi}$ )  $\leftarrow$  (200, 35);
    ( $K_{py}$ ,  $K_{dy}$ )  $\leftarrow$  (17, 8);
    ( $K_{p\theta}$ ,  $K_{d\theta}$ )  $\leftarrow$  (200, 35);
    ( $K_{px}$ ,  $K_{dx}$ )  $\leftarrow$  (17, 8);
     $u_F \leftarrow$  parametri_quad.m · parametri_quad.g +
        parametri_quad.m(stare_dorită.z +
             $K_{dz}(\dot{stare\_dorită.z} - \dot{stare\_curentă.z}) +$ 
             $K_{pz}(stare\_dorită.z - stare\_curentă.z)$ );
     $\phi_c \leftarrow 1/parametri\_quad.g \left( \left( (stare\_dorită.x + K_{dx}(\dot{stare\_dorită.x} - \dot{stare\_curentă.x}) +$ 
         $K_{px}(stare\_dorită.x - stare\_curentă.x)) \sin(stare\_dorită.\psi) \right) -$ 
         $\left( (stare\_dorită.y + K_{dy}(\dot{stare\_dorită.y} - \dot{stare\_curentă.y}) +$ 
         $K_{py}(stare\_dorită.y - stare\_curentă.y)) \cos(stare\_dorită.\psi) \right) \right)$ 
     $\theta_c \leftarrow 1/parametri\_quad.g \left( \left( (stare\_dorită.x + K_{dx}(\dot{stare\_dorită.x} - \dot{stare\_curentă.x}) +$ 
         $K_{px}(stare\_dorită.x - stare\_curentă.x)) \cos(stare\_dorită.\psi) \right) +$ 
         $\left( (stare\_dorită.y + K_{dy}(\dot{stare\_dorită.y} - \dot{stare\_curentă.y}) +$ 
         $K_{py}(stare\_dorită.y - stare\_curentă.y)) \sin(stare\_dorită.\psi) \right) \right)$ 
    ( $\dot{\phi}_c$ ,  $\ddot{\phi}_c$ )  $\leftarrow$  (0, 0);
    ( $\dot{\theta}_c$ ,  $\ddot{\theta}_c$ )  $\leftarrow$  (0, 0);
     $\mathbf{u}_{Mx,3D} \leftarrow K_{d\phi}(\dot{\phi}_c - \dot{stare\_curentă.\phi}) + K_{p\phi}(\phi_c - stare\_curentă.\phi)$ 
     $\mathbf{u}_{My,3D} \leftarrow K_{d\theta}(\dot{\theta}_c - \dot{stare\_curentă.\theta}) + K_{p\theta}(\theta_c - stare\_curentă.\theta)$ 
     $\mathbf{u}_{Mz,3D} \leftarrow parametri\_quad.I_{zz}(stare\_dorită.\psi +$ 
         $K_{d\psi,1D}(stare\_dorită.\psi - stare\_curentă.\psi) +$ 
         $K_{p\psi,1D}(stare\_dorită.\psi - stare\_curentă.\psi))$ ;
     $\mathbf{u}_{M,3D} \leftarrow [\mathbf{u}_{Mx,3D}, \mathbf{u}_{My,3D}, \mathbf{u}_{Mz,3D}]^T$ ;
    returnează  $\mathbf{u}_F, \mathbf{u}_{M,3D}$ ;
    
```

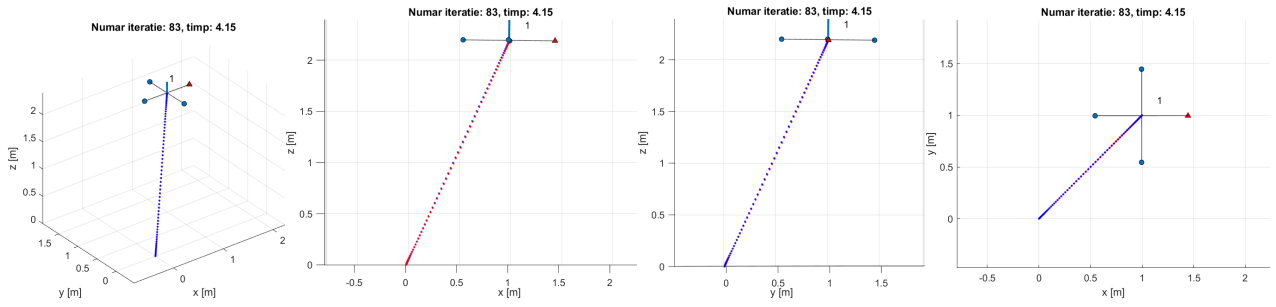


Figura 4.2: Urmărirea unei traiectorii de forma unui segment de dreaptă reprezentat 3D

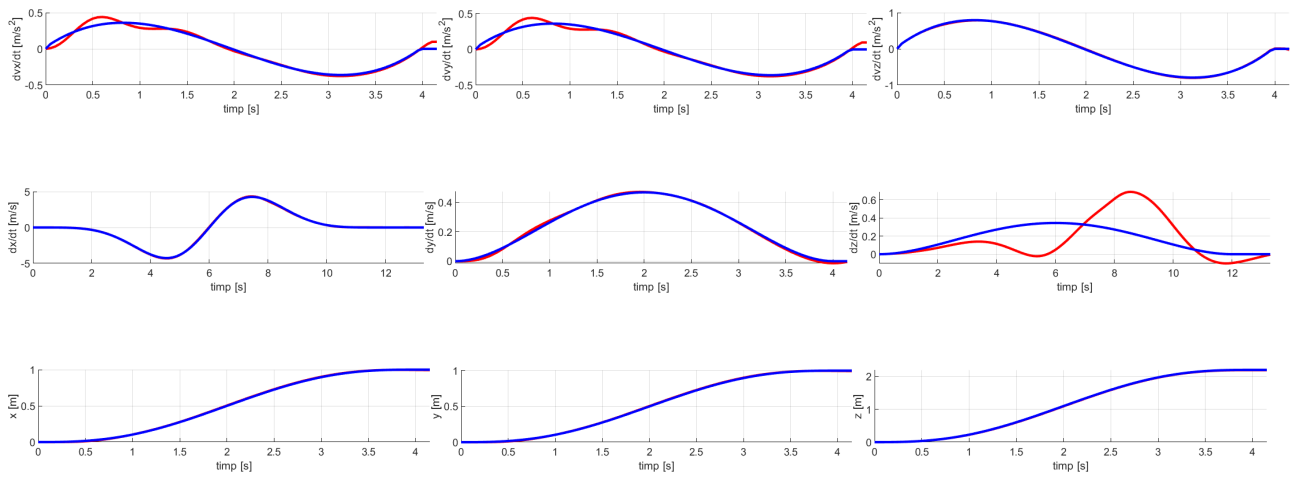


Figura 4.3: Evoluția în timp a poziției, vitezei și accelerației de translație în cazul zborului 3D

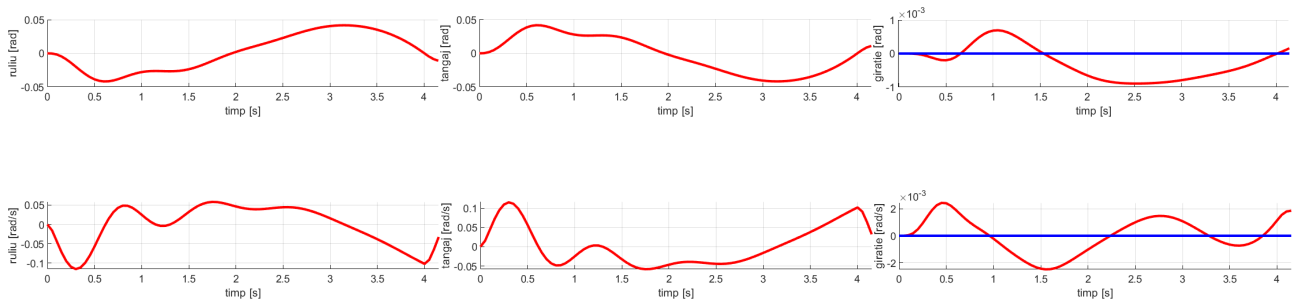


Figura 4.4: Evoluția în timp a poziției și vitezei unghiulare de girație în cazul zborului 3D

Setul cu valorile finale ale coeficienților proporționali și derivativi se află în Table 4.2.

Tabel 4.2: Coeficienții legii de control PD, determinați numeric

	K_{px}	K_{dx}	K_{py}	K_{dy}	K_{pz}	K_{dz}	$K_{p\phi}$	$K_{d\psi}$	$K_{p\theta}$	$K_{d\theta}$	$K_{p\psi}$	$K_{d\psi}$
Valori ”linie(t)”	17	8	17	8	1.4	0.75	200	35	200	35	4	0.1
Valori ”spirală(t)”	17	8	17	8	90	11	200	35	200	35	310	15

4.4 Concluzii

Capitolul a urmărit dezvoltarea unui sistem de control automat care să echipeze fiecare dintre membrii formației, implicați în studii de planificare a traiectoriei de zbor. Elementele prezentate pe parcursul capitolului au condus la îndeplinirea obiectivelor **OB3** și **OB4** propuse pentru realizarea tezei de doctorat. Rezultatele numerice obținute au încurajat realizarea practică a controlerului de zbor pentru testarea experimentală a traiectoriilor impuse.

Capitolul 5

Proiectarea, Realizarea și Testarea Experimentală a Pilotului Automat pentru Quadcopter

Pe parcursul capitolului sunt expuse etapizat informații legate de: componentele utilizate; mecanismele de proiectare și fabricație a modului electronic (stadiile de dezvoltare aferente versiunilor A, B, C și D); implementarea software a modurilor principale de zbor (controlul atitudinii, controlul altitudinii, zborul la punct fix) și a unui mod opțional/de siguranță („Return to Home”); testarea experimentală, procesarea și analiza datelor achiziționate în vederea evaluării performanțelor modurilor de zbor implementate.

5.1 Componente utilizate

Componentele utilizate pentru realizarea părții practice a acestei teze de doctorat sunt următoarele:

1. Structură quadcopter
 - Structură: Dji F450
 - Motor electric fără perii de 1000kV: A2212
 - Controler turatie motor de 30A
 - Elice din plastic de 10 inch Dji
 - Baterie litiu polimer cu trei celule, 3000mAh, 15C: Turnigy Graphene
2. Sistem de comandă automată
 - Microcontroler: STM32F103C8T6
 - Unitate digitală triaxială - accelerometru și giroscop: MPU6050
 - Magnetometru digital triaxial: HMC5883L
 - Modul de poziționare globală: NEO-M8
 - Modul digital pentru măsurarea presiunii atmosferice: MS5611
 - Transmițător radio pentru telecomandă: FS-iA6B

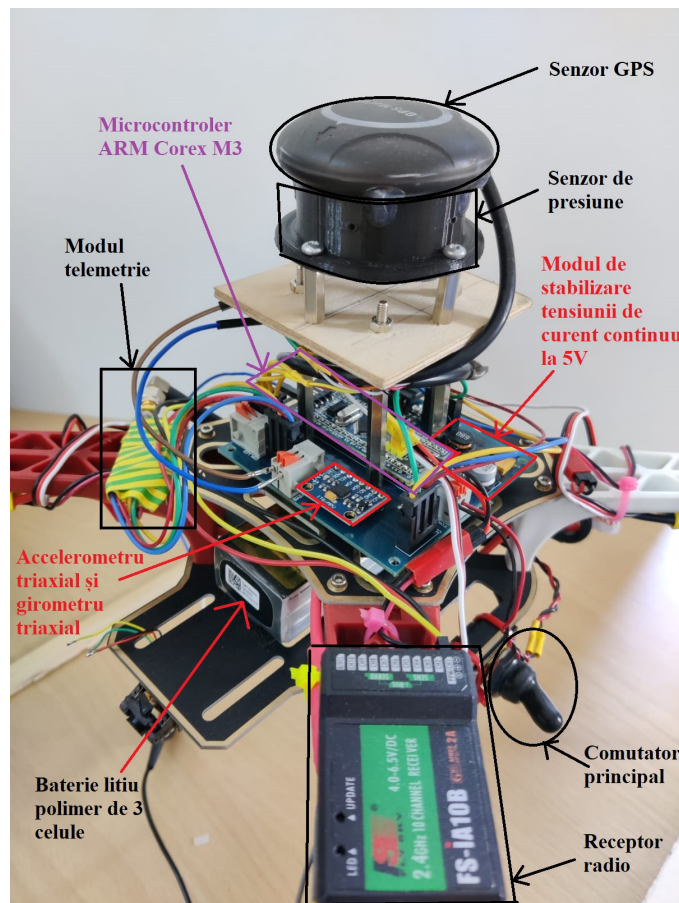


Figura 5.1: Vedere de aproape a componentelor electronice de la bord

- Modul radio pentru telemetrie: APC220

5.2 Proiectarea, realizarea și programarea sistemului de comandă automată

Versiunea finală a pilotului automat este ilustrată în Figure 5.2.

S-a realizat implementarea software a următoarelor moduri de zbor:

1. Controlul atitudinii;
2. Controlul altitudinii;
3. Punct fix.
4. Întoarcerea la primul punct GPS înregistrat la pornire sau „Return To Home”

Implementarea software a algoritmilor de control a fost realizată pe o arhitectură hardware reprezentată de un microcontroler - 32 biți, din clasa Advanced Reduced Instruction Set Computer (RISC) Machine (ARM)-Cortex M3. Am utilizat ca mediu de programare μ Vision® IDE, iar programul software a fost scris în C/C++. Fișierele sursă

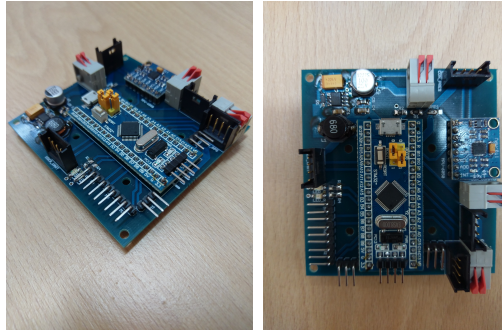


Figura 5.2: Versiunea D a pilotului automat pentru quadcopter

corespunzătoare algoritmilor de control au fost obținute cu ajutorul pachetului din Matlab, Embedded Coder®.

5.3 Analiza datelor experimentale

Realizarea traiectoriilor obținute pentru fiecare caz simulat în Matlab a necesitat dezvoltarea componentei software de programare a misiunii de zbor din Figure 5.3. Cu ajutorul limbajului de programare C#, am realizat o interfață grafică, în mediul de programare VisualStudio. Aceasta se poate folosi pentru planificarea misiunii de zbor selectând manual puncte GPS pe hartă. Analiza datelor achiziționate în urma zborului a fost efectuată cu

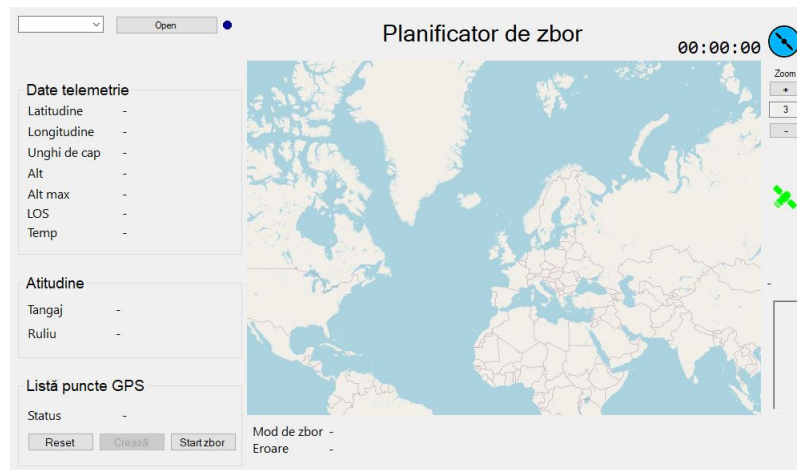


Figura 5.3: Captură de ecran cu interfața grafică

ajutorul Matlab, program în care a fost realizat graficul din Figure 5.4, pe baza măsurării presiunii atmosferice (cu ajutorul MS5611), și a unghiurilor de ruliu și tangaj (cu ajutorul MPU6050). Parametrii înregistrați au fost transmiși la sol prin intermediul modului radio APC220 de emisie-recepție, pe durata desfășurării zborului.

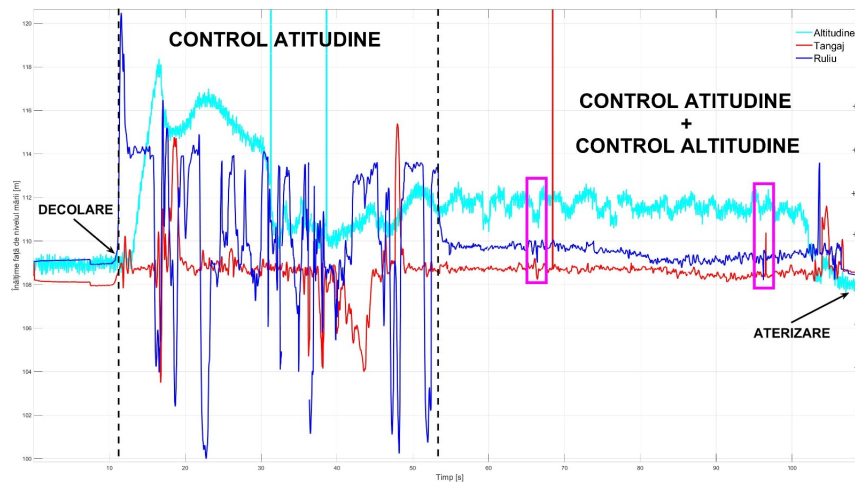


Figura 5.4: Evoluția în timp a unghiurilor de rulu și de tangaj precum și a altitudinii de zbor

La următorul pas aferent testării în zbor, am programat un zbor la punct fix pentru a testa dacă abaterile în latitudine și longitudine specificate de foaia de catalog a senzorului de poziționare globală utilizat sunt conforme. Astfel, în Figure 5.5 este reprezentat zborul



Figura 5.5: Poziția quadcopterului, reprezentată cu ajutorul aplicației Google Earth, în cazul scenariului de zbor la punct fix

la punct fix care a durat circa 33 de secunde de la momentul activării modului de zbor, moment la care au fost înregistrate valorile din Table 5.1. Scopul acestui mod de zbor este de a menține quadcopterul în spațiu la poziția 3D specificată de latitudine, longitudine și altitudinea de zbor față de nivelul mării. Important de menționat este faptul că abaterea

Tabel 5.1: Listă cu coordonatele GPS și altitudinea de zbor absolută, folosite în cadrul modului de zbor GPSfix

Tip zbor (abreviere)	Figură	Latitudine [°]	Longitudine [°]	Altitudine de zbor [m]
GPSfix	Figure 5.5	44.418163	26.086555	3

absolută maximă atât în latitudine cât și în longitudine, nu depășește valoarea maxim admisibilă, consemnată în foaia de catalog a senzorului GPS utilizat.

5.3.1 Analiza rezultatelor zborului după urmărirea unei traiectorii 2D

În cele ce urmează, sunt prezentate rezultatele evaluării performanțelor pilotului automat, efectuând o analiză comparativă între rezultatele obținute experimental și cele obținute numeric pe următoarele scenarii de zbor:

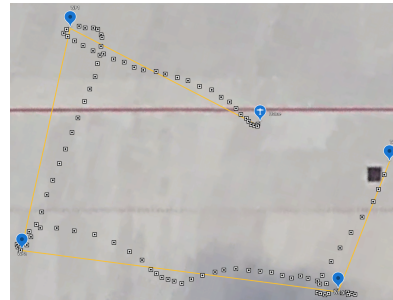
- Zbor după traiectoria din Figure 5.6, impusă de patru puncte GPS
- Zbor după traiectoria din Figure 5.8, impusă de cinci puncte GPS
- Zbor după traiectoria din Figure 5.10, impusă de șase puncte GPS

Cele trei scenarii au crescut gradual în complexitate, în cadrul ultimului scenariu vehiculul fiind obligat să realizeze accelerări și decelerări repetate, uneori până la atingerea unui zbor la punct fix, pentru a schimba direcția de zbor. Aceste evoluții au impuse prin alegerea coordonatelor punctelor GPS pe care aparatul de zbor a fost obligat să le viziteze.

Scenariul 1: Plan de zbor impus de patru puncte GPS



(a) Vedere de departe, din satelit



(b) Vedere de aproape, din satelit

Figura 5.6: Traiectorie formată din patru puncte GPS

Scenariul 2: Plan de zbor impus de cinci puncte GPS

La fel ca în cazul scenariului precedent, traiectoria din 5.8a a fost realizată folosind aplicația Google Earth. În esență, traictoria formată din cele cinci puncte GPS este similară cu cea din Figure 5.6, cu mențiunea că a mai fost adăugat un punct, iar altitudinea de zbor absolută a fost fixată la valoarea 4m.

Scenariul 3: Plan de zbor impus de șase puncte GPS

Scopul realizării acestui scenariu de zbor a fost de a evalua comportamentul pilotului automat pe o traiectorie de formă diferită față de primele două misiuni de zbor, pentru a testa eficacitatea cu care quadcopterul reușește să accelereze și să decelereze, direcția de zbor schimbându-se la aproape 180° la fiecare punct GPS de interes.

Capitolul 5. Proiectarea, Realizarea și Testarea Experimentală a Pilotului Automat pentru Quadcopter

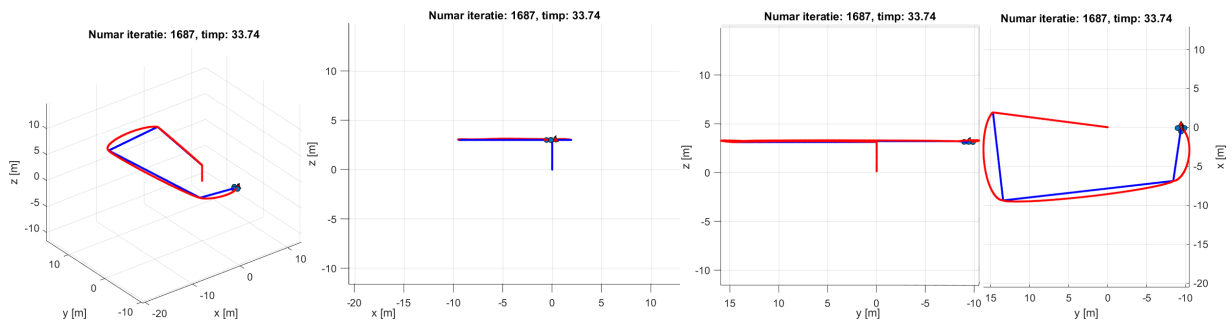
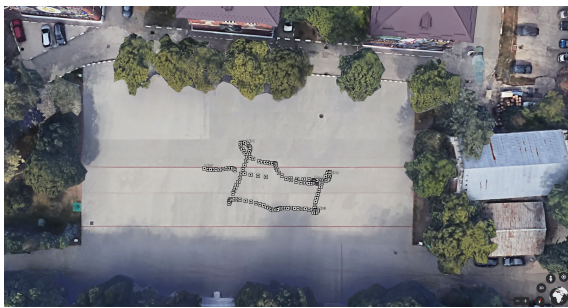
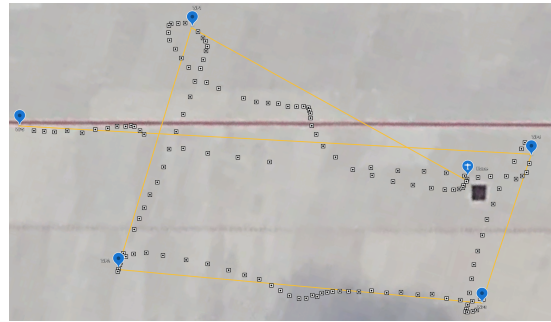


Figura 5.7: Urmărirea traiectoriei 2D (impusă și colorată cu albastru) din 5.6b



(a) Vedere de departe, din satelit



(b) Vedere de aproape, din satelit

Figura 5.8: Traiectoria quadcopterului formată din cinci puncte GPS

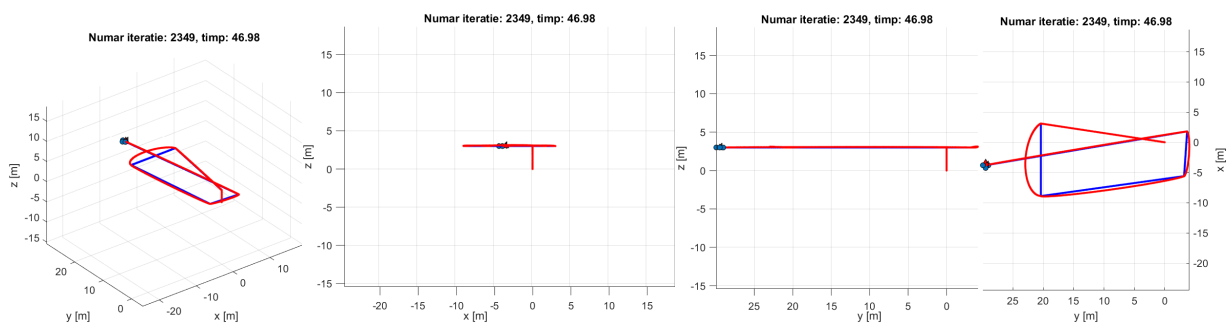
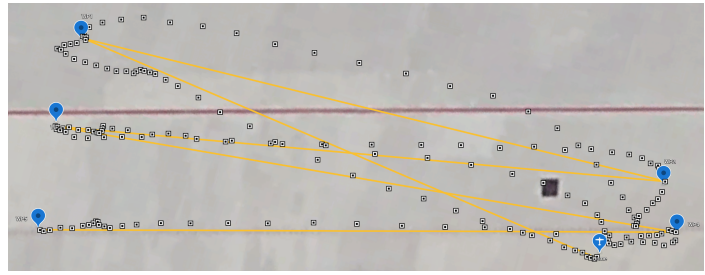


Figura 5.9: Urmărirea unei traiectorii 2D (impusă și colorată cu albastru), formată din cinci coordonate



(a) Vedere de departe, din satelit



(b) Vedere de aproape, din satelit

Figura 5.10: Traiectoria quadcopterului formată din șase puncte GPS

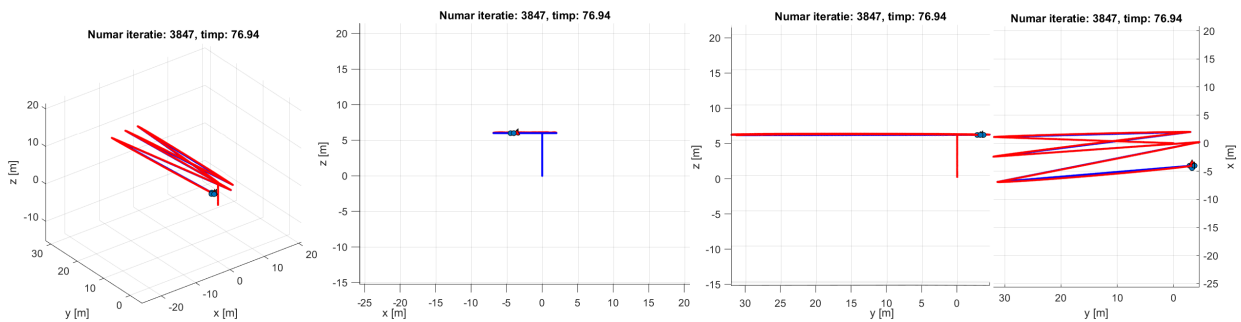


Figura 5.11: Urmărirea unei traiectorii 2D (impusă și colorată cu albastru), formată din șase coordonate

5.3.2 Analiza rezultatelor zborului după urmărirea unei traiectorii 3D

Suplimentar situației anterior testate, pentru acest test am implementat conceptul modului de zbor opțional/de siguranță „Return To Home”, care duce la executarea unei traiectorii 3D după cum urmează:

1. Se compară altitudinea de zbor cu valoarea fixată la +20m față de altitudinea (față de nivelul mării) înregistrată la decolare și primul pas constă în efectuarea deplasării de-a lungul axei z pentru a ajunge la altitudinea repectivă;
2. Se compară poziția GPS curentă cu cea înregistrată la decolare și se determină diferența în latitudine și longitudine, după care se comandă mișcarea în planul xy pentru a întoarce quadcopterul;
3. După ce quadcopterul a ajuns în punctul GPS înregistrat la start, se comandă aterizarea, deci scăderea graduală a poziției corespunzătoare axei z .

Acest mod de zbor (opțional) are rolul de a întoarce quadcopterul la punctul GPS înregistrat la decolare. Această măsură a fost luată din motive de siguranță, în special în cazurile în care se pierde legătura radio dintre aplicația (interfața grafică) de programare

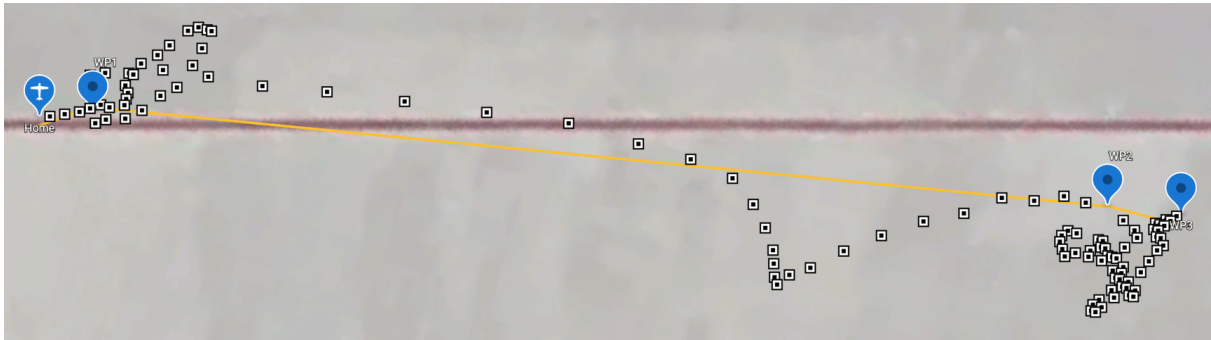


Figura 5.12: Traiectoria quadcopterului formată din trei puncte GPS

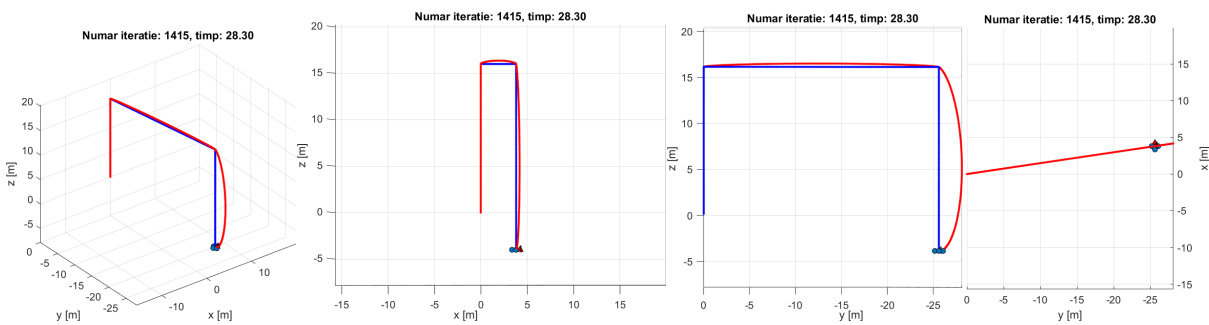


Figura 5.13: Urmărirea traiectoriei 3D (impusă și colorată cu albastru) din Figure 5.12

a traiectoriei de zbor, care rulează la sol, și quadcopter, precum și în situația în care se pierde controlul asupra aparatului de zbor de către cel care îl pilotează. Modul de zbor „Return To Home” sau „Return To Land” a fost activat după efectuarea scenariilor de zbor 5.6b și 5.8b, lucru care poate fi observat foarte ușor prin analiza imaginilor.

5.4 Concluzii

Rezultatele prezentate în acest capitol au condus la îndeplinirea obiectivelor **OB5** și **OB6** ale tezei de doctorat și anume: realizarea experimentală a pilotului automat pentru UAV-ul de tip multirotor și testarea experimentală a pilotului automat dezvoltat. Rezultatele expuse în acest capitol au confirmat faptul că sistemul „quadcopter controlat” pe care l-am dezvoltat este pregătit pentru a fi utilizat în vederea testării algoritmilor de planificare a traiectoriei de zbor descriși în chapter 6 și chapter 7. Performanțele obținute sunt încurajatoare și oferă atât suportul hardware, cât și software prevăzut în structura sistemului RETPINC (din Figure 1.2), care presupune existența componentei de control care va rula împreună cu componenta de planificare a traiectoriei de zbor.

Capitolul 6

Planificarea Traiectoriei de Zbor cu Minimizarea Consumului Energetic, în Absența Obstacolelor

6.1 Elemente pregătitoare pentru calculul traiectoriei de zbor folosind ecuațiile Euler-Lagrange

Rezumăm cele avute în vedere pentru rezolvarea problemei de planificare a traiectoriei de zbor astfel:

- (i) Fixarea coordonatelor spațiale de start, stop, a celor intermediare (punctelor GPS) prin intermediul \mathbf{r}_d ;
- (ii) Fixarea orientărilor corespunzătoare punctelor GPS stabilite prin intermediul ψ_d ;
- (iii) Minimizarea variației intrărilor de comandă $\mathbf{u}_F, \mathbf{u}_M$;
- (iv) Stabilirea ordinului sistemului (n) și impunerea a ($n-1$) restricții pentru formularea și rezolvarea funcției cost a problemei de planificare a traiectoriei de zbor care să minimizeze consumul total de energie electrică.

Rezolvarea acestei probleme, presupune utilizarea unor instrumente din calculul variațional, la care, contribuții esențiale au fost aduse de Joseph Louis Lagrange. Scopul utilizării calculului variațional este de a găsi funcția optimală $x^*(t)$ pentru următoarele obiective:

- Minimizarea vitezei, $n = 1$;
- Minimizarea accelerației, $n = 2$;
- Minimizarea variației accelerației, $n = 3$;
- Minimizarea variației variației accelerației, $n = 4$; etc.

6.2 Derularea unei misiuni cu traiectorie de zbor planificată, în condițiile minimizării accelerației de translație/ forței de tracțiune care acționează asupra structurii quadcopterului

Unul din obiectivele principale ale acestei teze de doctorat îl reprezintă planificarea traiectoriei de zbor, urmărind reducerea consumului energetic. În mod intuitiv, pentru a reuși acest lucru se dorește evitarea manevrelor bruște. Acest lucru se poate traduce prin minimizarea accelerațiilor de translație ale quadcopterului, care, în esență presupune minimizarea forțelor care acționează asupra structurii UAV-ului, așa cum reiese din ecuația (??). Pentru a minimiza accelerația totală de-a lungul traiectoriei de zbor, avem în vedere exemplul scenariului de zbor din figura 4.2a. Rezolvarea problemei de planificare a traiectoriei de zbor în spațiu tridimensional, presupune, ca pe lângă cele trei coordonate \mathbf{x} , \mathbf{y} și \mathbf{z} să identificăm optimul și pentru unghiul de cap, ψ . Rezultă

$$\begin{aligned} \mathbf{x}^*(t), \mathbf{y}^*(t), \mathbf{z}^*(t), \psi^*(t) &= \arg \min_{\mathbf{x}(t)} \int_0^T \mathcal{L}(\dot{\mathbf{x}}, \dot{\mathbf{y}}, \dot{\mathbf{z}}, \dot{\psi}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \psi, t) dt \\ &\int_0^T \mathcal{L}(\dot{\mathbf{x}}, \dot{\mathbf{y}}, \dot{\mathbf{z}}, \dot{\psi}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \psi, t) dt = \int_0^T (\ddot{\mathbf{x}}^2 + \ddot{\mathbf{y}}^2 + \ddot{\mathbf{z}}^2 + \ddot{\psi}^2) dt \quad (6.1) \\ \int_0^T \mathcal{L}(\dot{\mathbf{x}}, \dot{\mathbf{y}}, \dot{\mathbf{z}}, \dot{\psi}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \psi, t) dt &= \int_0^T \ddot{\mathbf{x}}^2 dt + \int_0^T \ddot{\mathbf{y}}^2 dt + \int_0^T \ddot{\mathbf{z}}^2 dt + \int_0^T \ddot{\psi}^2 dt. \end{aligned}$$

Rezolvarea problemei de planificare a traiectoriei a fost realizată în Matlab , pe baza algoritmului (3). Acest algoritm este în esență o funcție sau metodă, apelată de algoritmul (4), care, pe baza timpului curent al misiunii de zbor t și a punctelor GPS ce descriu traiectoria, va furniza poziția, viteza și accelerația de translație (în sistemul de coordonate inerțial) pe care trebuie să le aibă quadcopterul la următoarea iterație a algoritmului de control.

Algorithm 3 Generare traiectorie 3D cu optimizarea accelerației de translație

function GENERARETRAIECTORIE(*timp*, *puncteGPS*)

if *status* = *mod_inicializare* **then**

$[\dot{\mathbf{r}}_{d,3D}, \ddot{\mathbf{r}}_{d,3D}, \psi_{d,3D}, \dot{\psi}_{d,3D}] \leftarrow [0, 0, 0, 0]$
 $c_x \leftarrow \text{CalculeazaCoef}(\text{puncteGPS}_x)$
 $c_y \leftarrow \text{CalculeazaCoef}(\text{puncteGPS}_y)$
 $c_z \leftarrow \text{CalculeazaCoef}(\text{puncteGPS}_z)$
 $\ddot{\mathbf{r}}_{d,3D_{\max}} \leftarrow \text{valoare_stabilita}$
 $d \leftarrow \text{euclid}(\text{puncteGPS}, \ddot{\mathbf{r}}_{d,3D_{\max}})$
 $\text{timp_traiectorie} \leftarrow \text{suma}(d)$
 $\text{status} \leftarrow \text{mod_continuu}$

else

if $\text{timp} > \text{timp_traiectorie}[\text{final}]$ **then**

$\text{timp} = \text{timp_traiectorie}[\text{final}]$

$\text{index_timp} \leftarrow \text{gaseste_index_timp}(\text{timp}, \text{timp_traiectorie})$

if $\text{timp} = 0$ **then**

$\text{index_timp} \leftarrow 1$

$\text{factor} \leftarrow (\text{timp_traiectorie}[\text{index_timp}]) / d[\text{index_timp}]$

$t_0 \leftarrow \text{CalculeazaValoriTimp}(\text{ordin_polinom}_0, \#derivata_0, \text{factor})$

$t_1 \leftarrow \text{CalculeazaValoriTimp}(\text{ordin_polinom}_1, \#derivata_1, \text{factor})$

$t_2 \leftarrow \text{CalculeazaValoriTimp}(\text{ordin_polinom}_2, \#derivata_2, \text{factor})$

$\text{idx} \leftarrow [4(\text{index_timp} - 1) + 1 : 4\text{index_timp}]$

$\mathbf{r}_{d,3D} \leftarrow [c_x[\text{idx}]t_0; c_y[\text{idx}]t_0; c_z[\text{idx}]t_0]$

$\dot{\mathbf{r}}_{d,3D} \leftarrow [c_x[\text{idx}]t_1/d[\text{index_timp}]; c_y[\text{idx}]t_1/d[\text{index_timp}]; c_z[\text{idx}]t_1/d[\text{index_timp}]]$

$\ddot{\mathbf{r}}_{d,3D} \leftarrow [c_x[\text{idx}]t_2/d[\text{index_timp}]^2; c_y[\text{idx}]t_2/d[\text{index_timp}]^2; c_z[\text{idx}]t_2/d[\text{index_timp}]^2]$

$\psi_{d,3D} \leftarrow 0$

$\dot{\psi}_{d,3D} \leftarrow 0$

return stare_dorită: $[\mathbf{r}_{d,3D}, \dot{\mathbf{r}}_{d,3D}, \ddot{\mathbf{r}}_{d,3D}, \psi_{d,3D}, \dot{\psi}_{d,3D}]$.

Algorithm 4 Simulare traiectorie 3D cu optimizarea accelerației de translație

```
function SIMULARE3D(GenerareTraiectorie, Controler3D)
  initializare_grafice();
  timp_alocat_maxim  $\leftarrow$  30s
  inițial:
  control_increment_timp  $\leftarrow$  4ms
  traiectorie_increment_timp gets 10control_increment_timp
  numar_iterații  $\leftarrow$  timp_alocat_maxim / control_increment_timp
  timp_curent  $\leftarrow$  0
  [ $\mathbf{r}_{d,3DSTART}$ ,  $\dot{\mathbf{r}}_{d,3DSTART}$ ,  $\ddot{\mathbf{r}}_{d,3DSTART}$ ]  $\leftarrow$  GenerareTraiectorie(timp_curent)
  [ $\mathbf{r}_{d,3DSTOP}$ ,  $\dot{\mathbf{r}}_{d,3DSTOP}$ ,  $\ddot{\mathbf{r}}_{d,3DSTOP}$ ]  $\leftarrow$  GenerareTraiectorie( $\infty$ )
  abatere_r  $\leftarrow$  0.05
  abatere_ṙ  $\leftarrow$  0.01
  loop:
  if iterație  $\leq$  numar_iterații then

    if iterație =1 then
      ReprezentareGrafică(quad,traiectorie_increment_timp)
    else
      [timp,stări]  $\leftarrow$  ModelQuad(
        EcuatiiMișcare(Controler3D,GenerareTraiectorie,ParamQuad),
        iterație, $\mathbf{r}$ ,  $\dot{\mathbf{r}}$ ,  $\ddot{\mathbf{r}}$ )
      ReprezentareGrafică(quad,traiectorie_increment_timp)
    iterație  $\leftarrow$  iterație+1

    if VerificareStop(
       $\mathbf{r}_{d,3DSTOP}$ ,  $\dot{\mathbf{r}}$ ,  $\ddot{\mathbf{r}}$ , timp_curent,
      abatere_r, abatere_ṙ, numar_iterații)=TRUE then

      close
      goto loop
    close
  ReprezentareGrafice( $\mathbf{r}$ ,  $\dot{\mathbf{r}}$ ,  $\ddot{\mathbf{r}}$ , timp).
```

6.3 Derularea unei misiuni cu traiectorie de zbor planificată, în condițiile minimizării variației accelerației de translație/ forței de tracțiune sau a momentelor care acționează asupra structurii quadcopterului

În continuare, ne propunem ca traiectoria pe care o obținem să fie compusă din rute de zbor de forma unor curbe derivabile de trei ori, pentru a modela și variația accelerației în cadrul rezolvării problemei de planificare a traiectoriei de zbor. În acest sens, vom rezolva problema de planificare a traiectoriei de zbor de forma

$$\begin{aligned} \mathbf{x}^*(t), \mathbf{y}^*(t), \mathbf{z}^*(t) = & \arg \min_{\mathbf{x}(t)} \left(\int_{t_0}^{t_1} \ddot{\mathbf{x}}^2 dt + \dots + \int_{t_3}^{t_4} \ddot{\mathbf{x}}^2 dt \right) + \\ & \arg \min_{\mathbf{y}(t)} \left(\int_{t_0}^{t_1} \ddot{\mathbf{y}}^2 dt + \dots + \int_{t_3}^{t_4} \ddot{\mathbf{y}}^2 dt \right) + \\ & \arg \min_{\mathbf{z}(t)} \left(\int_{t_0}^{t_1} \ddot{\mathbf{z}}^2 dt + \dots + \int_{t_3}^{t_4} \ddot{\mathbf{z}}^2 dt \right), \end{aligned} \quad (6.2)$$

6.4 Estimarea consumului energetic la derularea unor misiuni cu traiectorie de zbor planificată, în diferite scenarii de minimizare a variabilelor ce țin de dinamica quadcopterului

Acest subcapitol urmărește estimarea consumului energetic al quadcopterului atunci când efectuează diferite misiuni de zbor în condițiile de minimizare studiate în subcapitolele anterioare (minimizarea accelerației de translație, respectiv, minimizarea variației acesteia și a momentelor). În situația în care consumul energetic crește atunci când traiectoria de zbor este calculată minimizând ⁽⁴⁾ \mathbf{r} . Acest lucru are loc, deoarece, pentru a menține quadcopterul pe traiectorie (adică abaterea în poziție, viteză și accelerație să fie cât mai mică), valoarea vectorului forță \mathbf{F} care acționează asupra quadcopterului crește. Astfel, chiar dacă necesarul energetic este cel mai mic în cazul minimizării lui $\ddot{\mathbf{r}}$ sau $\ddot{\mathbf{r}}$, precizia cu care este urmărită traiectoria de către quadcopter este mai mare în cazul minimizării lui \mathbf{r} .

Tabel 6.1: Estimarea consumului energetic pentru scenariile de zbor din Figure 5.12, Figure 5.6 și Figure 5.10

Limită	Tip traiectorie	E_{WP3} [Wh]	E_{WP4} [Wh]	E_{WP6} [Wh]
$\ddot{\mathbf{r}}_{d,3D_{\max}} = 0.1m/s^2$	$\min \dot{\mathbf{r}}$	196.12	197.79	-
	$\min \ddot{\mathbf{r}}$	195.18	196.31	196.11
	$\min \ddot{\mathbf{r}}$	195	195.46	195.27
	$\min \mathbf{r}^{(4)}$	194.88	195.23	195.14
$\ddot{\mathbf{r}}_{d,3D_{\max}} = 0.5m/s^2$	$\min \dot{\mathbf{r}}$	-	-	-
	$\min \ddot{\mathbf{r}}$	195.96	197.23	196.42
	$\min \ddot{\mathbf{r}}$	195.79	198.97	196.35
	$\min \mathbf{r}^{(4)}$	195.29	197.07	196.23
$\ddot{\mathbf{r}}_{d,3D_{\max}} = 1m/s^2$	$\min \dot{\mathbf{r}}$	-	-	-
	$\min \ddot{\mathbf{r}}$	198.92	203.24	199.8
	$\min \ddot{\mathbf{r}}$	198.96	206.88	199.35
	$\min \mathbf{r}^{(4)}$	198.16	-	200.47
$\ddot{\mathbf{r}}_{d,3D_{\max}} = 2m/s^2$	$\min \dot{\mathbf{r}}$	-	-	-
	$\min \ddot{\mathbf{r}}$	202.94	222.55	212.62
	$\min \ddot{\mathbf{r}}$	202.24	-	212.59
	$\min \mathbf{r}^{(4)}$	202.27	-	212.79

6.5 Concluzii

În cadrul acestui capitol s-a realizat îndeplinirea obiectivelor **OB7**, **OB8** și **OB9**, ale tezei de doctorat. În baza rezultatelor obținute până în acest punct, capitolul următor va prezenta un studiu asupra extinderii problemei de planificare a traiectoriei de zbor prin adăugarea de obstacole statice (obiecte din spațiul de lucru) și dinamice, reprezentate de ceilalți membri ai formației de zbor.

Capitolul 7

Planificarea Traiectoriei de Zbor cu Minimizarea Consumului Energetic, în Prezența Obstacolelor

Conform concluziilor analizei bibliografice realizate în primul capitol al tezei și obiectivelor pe care mi le-am propus pentru realizarea tezei, acest capitol vizează:

- Dezvoltarea unei metode de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru un quadcopter în prezența obstacolelor;
- Validarea prin simulări numerice și demonstrarea performanțelor algoritmului de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru un quadcopter, în prezența obstacolelor;
- Dezvoltarea unei metode de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor, în absența obstacolelor;
- Validarea prin simulări numerice și demonstrarea performanțelor algoritmului de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor, în absența obstacolelor;
- Dezvoltarea unei metode de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor în prezența obstacolelor;
- Validarea prin simulări numerice și demonstrarea performanțelor algoritmului de planificare a traiectoriei de zbor cu minimizarea consumului energetic pentru doi membri ai formației de zbor, în prezența obstacolelor.

Toate acestea coincid cu îndeplinirea totală a obiectivelor **OB10–OB15**. Subcapitolele următoare tratează succesiv provocările ridicate de:

- Planificare traiectoriei de zbor pentru un UAV cu evitarea obstacolelor statice.
- Planificare traiectoriei de zbor pentru două UAV-uri cu evitarea coliziunilor dintre membrii formației.
- Planificarea traiectoriei de zbor pentru două UAV-uri cu evitarea coliziunilor dintre membrii formației și cu obstacolele statice.

7.1 Considerații teoretice privind optimizarea convexă

Plecând de la forma generală a unei probleme de optimizare scrisă de forma

$$\begin{aligned} \min f_0(x) \\ \text{evaluată la } f_i(x) \leq b_i, \quad i = 1, \dots, m, \end{aligned} \quad (7.1)$$

în care $x = (x_1, \dots, x_n)$ și reprezintă variabila în raport cu care se realizează optimizarea, funcția $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}$ este funcția obiectiv, funcțiile $f_i : \mathbf{R}^n \rightarrow \mathbf{R}$ reprezintă restricțiile, iar b_1, \dots, b_m reprezintă limitele restricțiilor. Forma generală a problemei de optimizare (7.1) devine o problemă liniară dacă funcția obiectiv precum și funcțiile care descriu restricțiile sunt liniare, adică satisfac relația

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y), \quad (7.2)$$

$\forall x, y, \in \mathbf{R}^n$, iar $\alpha, \beta \in \mathbf{R}$.

Problema planificării traiectoriei de zbor a fost tratată până în acest moment ca o problemă liniară. Problemele de optimizare liniare fac parte dintr-o clasă importantă de probleme de optimizare convexă [47]. Plecând de la expresia (7.2), o problemă de optimizare convexă impune ca funcția obiectiv sau funcționala precum și restricțiile să fie convexe, ceea ce matematic semnifică:

$$\begin{aligned} f_i(\alpha x + \beta y) &\leq \alpha f_i(x) + \beta f_i(y) \\ \alpha + \beta &= 1 \\ \alpha \geq 0, \beta \geq 0 \quad \alpha, \beta \in \mathbf{R}, \end{aligned} \quad (7.3)$$

$\forall x, y, \in \mathbf{R}^n$. Se observă din (7.2) și (7.3) că putem considera optimizarea convexă, o generalizare a problemei liniară de optimizare.

7.2 Planificarea traiectoriei cu evitarea obstacolelor statice, pentru un UAV care execută zbor solitar

Aproximăm dinamica membrilor formației de UAV-uri folosind modelul integratorului dublu, iar în conformitate cu [48], expresia generală a sistemului liniar de ordin II invariant în timp pentru un model discret poate fi aproximată ca fiind

$$\begin{aligned} \mathbf{x}_{t+\delta t} &= \mathbf{A}_d \mathbf{x}_t + \mathbf{B}_d \mathbf{u}_t \\ \mathbf{x}_0 &= \begin{bmatrix} \mathbf{r}_0 \\ \dot{\mathbf{r}}_0 \end{bmatrix}, \end{aligned} \quad (7.4)$$

în care am notat cu $\mathbf{u}_t = \ddot{\mathbf{r}}_t$ vectorul coloană cu componentele accelerației de translație la momentul de timp $t \forall t \in [0, T]$. Condiția inițială este specificată prin \mathbf{x}_0 . În formă extinsă,

relația (7.4) devine

$$\begin{bmatrix} \mathbf{x}_{t+\delta t} \\ \mathbf{y}_{t+\delta t} \\ \dot{\mathbf{x}}_{t+\delta t} \\ \dot{\mathbf{y}}_{t+\delta t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \delta t & 0 \\ 0 & 1 & 0 & \delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{y}_t \\ \dot{\mathbf{x}}_t \\ \dot{\mathbf{y}}_t \end{bmatrix} + \begin{bmatrix} \frac{\delta t^2}{2} & 0 \\ 0 & \frac{\delta t^2}{2} \\ \delta t & 0 \\ 0 & \delta t \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_t \\ \ddot{\mathbf{y}}_t \end{bmatrix}. \quad (7.5)$$

Astfel, în contextul consumului redus de energie, așa cum s-a demonstrat în cadrul capitolului anterior, urmărim ca magnitudinea forțelor (care vor genera accelerațiile de translație) care acționează asupra structurii UAV-ului trebuie să fie cât mai mică sau chiar minimă. Urmărirea acestui obiectiv are drept soluție minimizarea funcției cost, scrisă sub formă discretă pentru cazul 2D

$$J = \sum_{t=0}^T (\|\ddot{\mathbf{x}}_t\|^2 + \|\ddot{\mathbf{y}}_t\|^2). \quad (7.6)$$

Scriem forma explicită a relației (7.1) de forma

$$\begin{aligned} & \min_{\ddot{\mathbf{u}}} \sum_{t=0}^T \|\ddot{\mathbf{u}}_t\|^2 \\ & \mathbf{x}_0 = [\mathbf{r}_0 \quad \dot{\mathbf{r}}_0]^T \\ & \begin{bmatrix} \mathbf{r}_{t+\delta t} \\ \dot{\mathbf{r}}_{t+\delta t} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & \delta t \mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{r}_t \\ \dot{\mathbf{r}}_t \end{bmatrix} + \begin{bmatrix} \frac{\delta t^2}{2} \mathbf{I}_2 \\ \delta t \mathbf{I}_2 \end{bmatrix} \ddot{\mathbf{u}}_t \\ & \begin{cases} x_t - x_{l,min} \leq R b_{t,1} - d_l \\ y_t - y_{l,min} \leq R b_{t,2} - d_l \\ x_{l,max} - x_t \leq R b_{t,3} - d_l \\ y_{l,max} - y_t \leq R b_{t,4} - d_l \\ \sum_{o=1}^4 b_{t,o} \leq 3 \end{cases}, \quad \forall l = [1, L] \\ & \|\dot{\mathbf{r}}_t\| \leq \|\dot{\mathbf{r}}_{\max}\| \\ & \|\ddot{\mathbf{r}}_t\| \leq \|\ddot{\mathbf{r}}_{\max}\| \\ & \mathbf{x}_T = [\mathbf{r}_T \quad \dot{\mathbf{r}}_T]^T, \end{aligned} \quad (7.7)$$

în care am notat cu d_l distanța de siguranță d din jurul obstacolului l , iar cu L numărul de obstacole. Urmărim minimizarea funcției satisfăcând restricțiile de egalitate și de inegalitate din interiorul acoladei. Restricțiile de inegalitate din interiorul acoladei mai mici se satisfac la fiecare iterație δt de L ori. Particularizăm problema descrisă de (7.7), denumită în continuare OPTimization ACCeleration (OPTACC), în raport cu condițiile reale de lucru descrise de scenariul din Figure 7.1:

- δt a fost fixat pe rând la valorile 1s, 0.5s, 0.2s, acestea fiind intervalele de timp utilizate cel mai des în cadrul recepției de coordonate GPS;

- $T = 7s$, $L = 1$;
- Coordonate punct de start: $(x, y) = (0m, 0m)$; UAV-ul realizează punct fix la o înălțime oarecare z ;
- Coordonate punct de final: $(x, y) = (5m, 10m)$. UAV-ul menține altitudinea z constantă;
- Coordonate obstacol: $(x_{1,min}, y_{1,min}, x_{1,max}, y_{1,max}) = (2m, 5m, 3m, 6m)$;
- $R = 1000000m$
- $\|\dot{\mathbf{r}}_{\max}\| = 2m/s$, $\|\ddot{\mathbf{r}}_{\max}\| = 2m/s^2$;
- d_{max} se obține când $\gamma = \pi/4$.

Rezolvarea numerică a problemei

$$\begin{aligned}
 & \min_{\ddot{\mathbf{u}}} \sum_{t=0}^{7s} \|\ddot{\mathbf{u}}_t\|^2 \\
 & \mathbf{x}_0 = [0 \ 0 \ 0 \ 0]^T \\
 & \begin{bmatrix} \mathbf{r}_{t+\delta t} \\ \dot{\mathbf{r}}_{t+\delta t} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & \delta t \mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{r}_t \\ \dot{\mathbf{r}}_t \end{bmatrix} + \begin{bmatrix} \frac{\delta t^2}{2} \mathbf{I}_2 \\ \delta t \mathbf{I}_2 \end{bmatrix} \ddot{\mathbf{u}}_t \\
 & \begin{cases} x_t - 2 \leq 1000000b_{t,1} - \frac{\dot{r}_{t,\max}\delta t}{2} \sin(\pi/4) \\ y_t - 5 \leq 1000000b_{t,2} - \frac{\dot{r}_{t,\max}\delta t}{2} \sin(\pi/4) \\ 3 - x_t \leq 1000000b_{t,3} - \frac{\dot{r}_{t,\max}\delta t}{2} \sin(\pi/4) \\ 6 - y_t \leq 1000000b_{t,4} - \frac{\dot{r}_{t,\max}\delta t}{2} \sin(\pi/4) \\ \sum_{o=1}^4 b_{t,o} \leq 3 \end{cases} \quad (7.8) \\
 & \|\dot{\mathbf{r}}_{t,\max}\| \leq 2 \\
 & \|\ddot{\mathbf{r}}_{t,\max}\| \leq 2 \\
 & \mathbf{x}_T = [5 \ 10 \ 0 \ 0]^T,
 \end{aligned}$$

a avut loc prin intermediul resurselor hardware și software din Table 2.1. Dorim să comparăm aceste rezultate cu o nouă tehnică de rezolvare a problemei de planificare a traiectoriei de zbor, folosind metoda IRRT* și expresia (7.7), specifică metodei MILP. Această idee rezultă din faptul că se dorește „informarea” componentei MOSEK cu privire la existența traiectoriei IRRT*, pentru a evita căutarea tuturor soluțiilor la problema de optimizare din spațiul 2D. Matematic, spunem că dorim să rezolvăm problema de optimizare

$$\begin{aligned}
 & \min_{\ddot{\mathbf{u}}} \sum_{t=0}^T \|\ddot{\mathbf{u}}_t\|^2 \\
 & \mathbf{x}_0 = [\mathbf{r}_0 \ \dot{\mathbf{r}}_0]^T \\
 & \begin{bmatrix} \mathbf{r}_{t+\delta t} \\ \dot{\mathbf{r}}_{t+\delta t} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & \delta t \mathbf{I}_2 \\ \mathbf{O}_2 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{r}_t \\ \dot{\mathbf{r}}_t \end{bmatrix} + \begin{bmatrix} \frac{\delta t^2}{2} \mathbf{I}_2 \\ \delta t \mathbf{I}_2 \end{bmatrix} \ddot{\mathbf{u}}_t \\
 & \|\mathbf{r}_t - \mathbf{r}_{t,irrt}\| \leq d_{l,irrt} - d_l, \forall l = [1, L] \quad (7.9)
 \end{aligned}$$

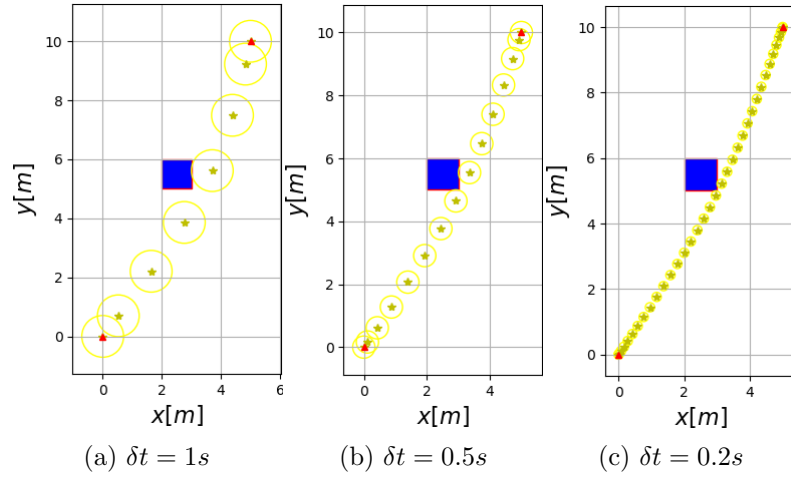
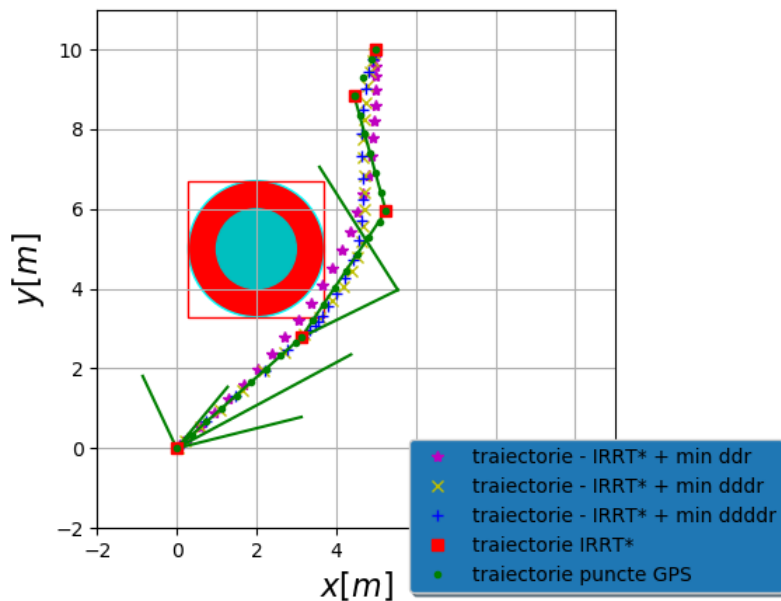


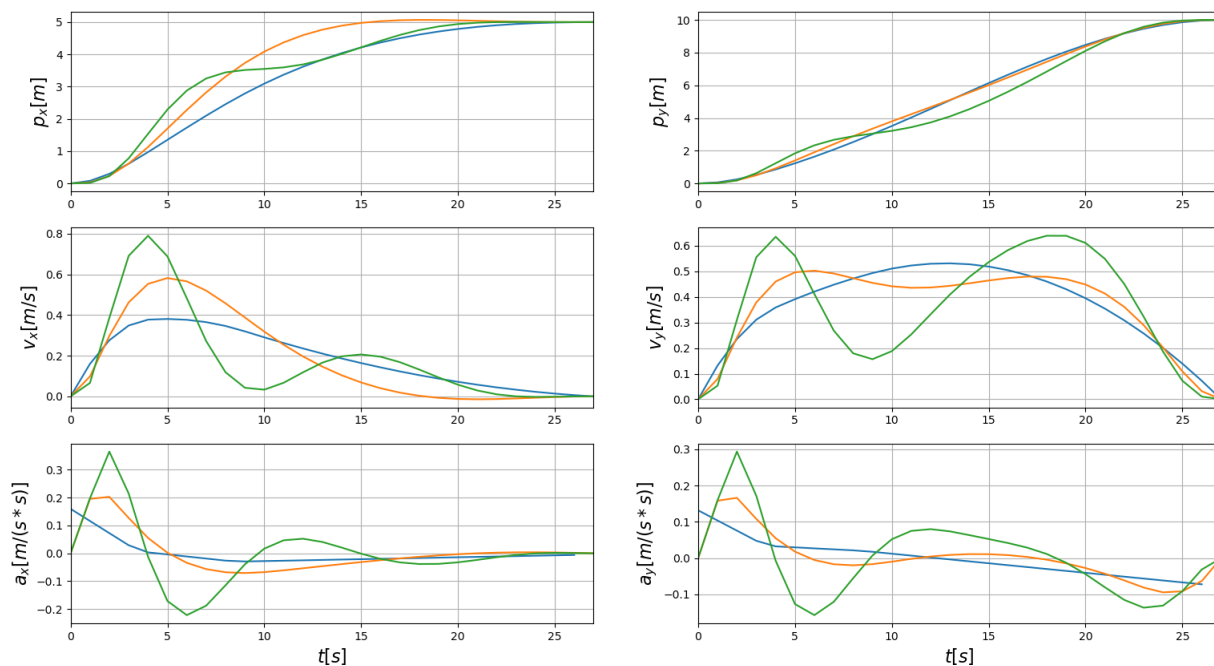
Figura 7.1: Traiectoria 2D a quadcopterului evitând un obstacol, obținută după aplicarea OPTACC cu pasul de timp δt

$$\begin{aligned} \|\dot{\mathbf{r}}_t\| &\leq \|\dot{\mathbf{r}}_{\max}\| \\ \|\ddot{\mathbf{r}}_t\| &\leq \|\ddot{\mathbf{r}}_{\max}\| \\ \mathbf{x}_T &= [\mathbf{r}_T \quad \dot{\mathbf{r}}_T]^T, \end{aligned}$$

denumită în continuare IRRT star OPTACC (OPTDACC), în care $d_{l,irrt}$ este distanța față de quadcopter, la care sunt generate eșantioane în cadrul algoritmului IRRT* și a fost fixată la valoarea $d_{l,irrt} = \mathbf{r}_l + s_l/2 + \dot{\mathbf{r}}_{\max}\delta t$ (\mathbf{r}_l reprezintă poziția obstacolului l , iar s lungimea laturii acestuia). În cadrul capitolului anterior, am demonstrat că minimizarea $\ddot{\mathbf{r}}$ și $\dot{\mathbf{r}}$ pentru execuția unui traseu de zbor descris prin intermediul punctelor GPS, duce la reducerea consumului energetic. Această motivație a presupus particularizarea expresiei (7.7), proces în urma căruia s-a obținut Derivative OPTACC (OPTDACC), IRRT star Derivative OPTACC (IRRTOPTDACC), și respectiv Double Derivative OPTACC (OPTDDACC), IRRT star Double Derivative OPTACC (IRRTOPTDDACC).

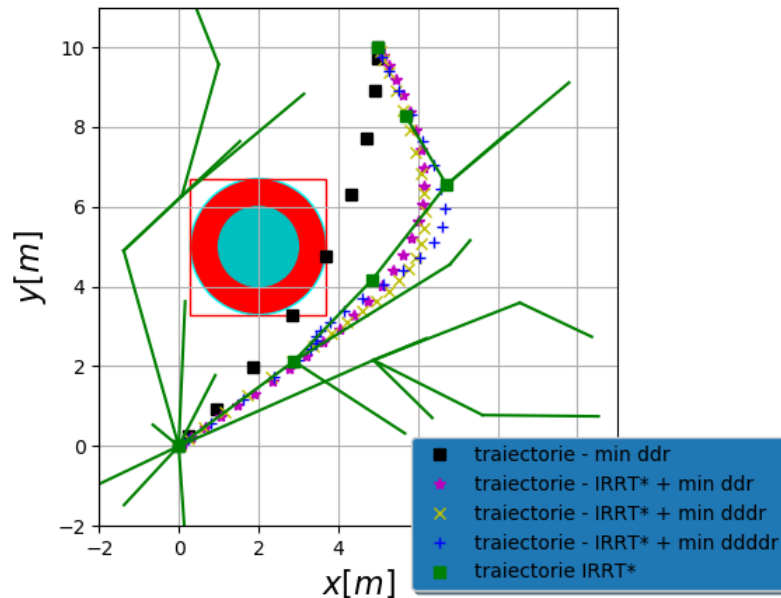


(a) Traiectorii IRRTOPT*

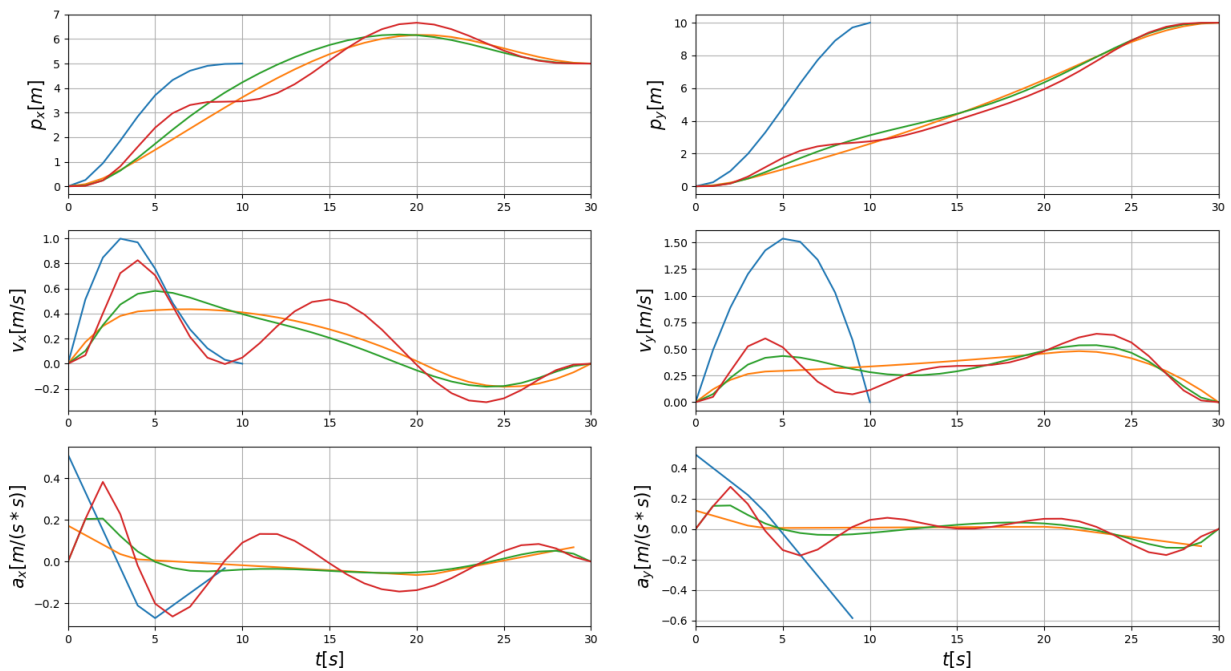


(b) Rezultatele traiectoriilor IRRTOPT*

Figura 7.2: Traiectoria 2D și evoluția în timp a poziției, vitezei și accelerației de translație a quadcopterului evitând un obstacol în formă de cerc, obținută după aplicarea IRRTOPTACC (albastru), IRRTOPTDACC (portocaliu) și IRRTOPTDDACC (verde) cu pasul de timp $\delta t = 1s$



(a) Traiectoriile OPTACC și IRRTOPT*



(b) Rezultatele traiectoriilor OPT și IRRTOPT*

Figura 7.3: Traiectoria 2D și evoluția în timp a poziției, vitezei și accelerației de translație a quadcopterului evitând un obstacol în formă de cerc, obținută după aplicarea OPTACC (albastru), IRRTOPTACC (portocaliu), IRRTOPTDACC (verde) și IRRTOPTDACC (roșu) cu pasul de timp $\delta t = 1s$

7.3 Planificarea traiectoriei pentru zbor în formație, cu evitarea coliziunilor între membrii formației

Rezolvarea problemei de planificare a traiectoriei de zbor pentru o formație de UAV-uri cu evitarea coliziunilor dintre membrii formației presupune modificarea/completarea setului de probleme OPT* și IRRTOPT* în mod similar cu modul în care a fost rezolvată problema de evitarea a obstacolelor. Considerăm că fiecare UAV este înconjurat cu ajutorul unui cerc, acesta situându-se pe timpul zborului în centrul cercului. Astfel, raza cercului h va stabili suprafața din jurul obstacolului ca zonă de siguranță, în care este restricționat accesul celorlalți membri ai formației de zbor. Matematic, putem scrie că

$$\begin{aligned}
 & x_t^i - x_t^j \geq dx_{ij} \\
 \text{sau} & x_t^j - x_t^i \geq dx_{ij} \\
 \text{sau} & y_t^i - y_t^j \geq dy_{ij} \\
 \text{sau} & y_t^j - y_t^i \geq dy_{ij},
 \end{aligned} \tag{7.10}$$

expresie în care, prin utilizarea superscriptului i și j , specificăm poziția atât pe axa x cât și pe y a UAV-ului i la momentul de timp t , respectiv a UAV-ului j , $\forall i, j \in [1, N]$, $i \neq j$, iar N reprezintă numărul total de UAV-uri. Valoarea distanței de siguranță pe axele x și y este

$$d_{ij} = (||\dot{\mathbf{r}}_{t,\max}^i|| + ||\dot{\mathbf{r}}_{t,\max}^j||)\delta t. \tag{7.11}$$

În cazul a două UAV-uri de tip quadcopter construite identic, $||\dot{\mathbf{r}}_{t,\max}^i|| = ||\dot{\mathbf{r}}_{t,\max}^j||$, iar (7.11) se simplifică astfel:

$$d_{ij} = 2||\dot{\mathbf{r}}_{t,\max}^i||\delta t, \forall i \in [1, N]. \tag{7.12}$$

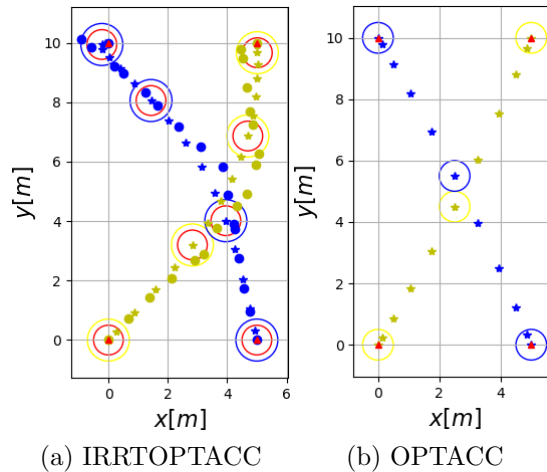


Figura 7.4: Traiectoria 2D a două quadcoptere în absența obstacolelor, folosind IRRTOPTACC și OPTACC

7.4 Planificarea traiectoriei pentru zbor în formație, cu evitarea coliziunilor atât între membrii formației, cât și cu obstacolele statice

În cadrul acestui subcapitol vom pune cap la cap cele studiate cu privire la evitarea coliziunilor atât cu obstacolele statice cât și cu cele dinamice. Astfel, urmărim formularea și rezolvarea problemei de planificare a traiectoriei de zbor care să corespundă aplicațiilor cât mai aproape de realitate. Zborul în formație presupune atât evitarea coliziunilor dintre membrii formației cât și a acestora cu obstacolele existente. Atingerea acestui obiectiv presupune integrarea tuturor restricțiilor formulate în cadrul capitolelor anterioare în cadrul problemei de planificare a traiectoriei de zbor, pentru a forma setul OPT* precum și setul IRRTOPT*. Natura probabilistică a algoritmului IRRT* implicat în metoda IRRTOPTACC poate duce și la rezultate nefavorabile cu privire la consumul energetic. Un astfel de exemplu este captat în Figure 7.5. Chiar dacă traiectoriile celor două UAV-uri din figura 7.5b au fost determinate în aproximativ 0.84s, cea colorată cu galben este departe de soluția optimă, lungimea acesteia fiind cu aproximativ 24.63% mai mare față de cea calculată și evidențiată în figura 7.5a, care este cu 6.6% mai lungă față de cea optimă.

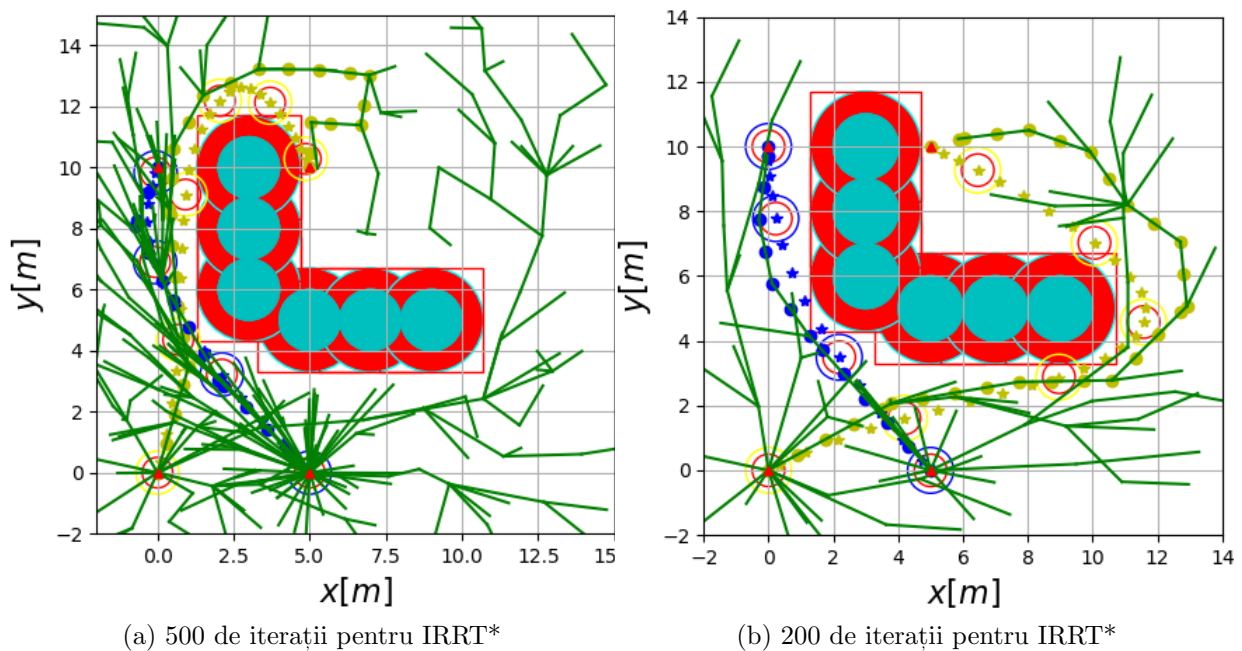


Figura 7.5: Traiectoria 2D a două quadcoptere în prezența obstacolelor, folosind IRRTOPTACC

7.5 Concluzii

În cadrul acestui capitol a fost rezolvată problema planificării traiectoriei de zbor în prezența obstacolelor statice pentru o formație de zbor formată din doi membri, fiind îndeplinite următoarele obiective:

- **OB10, OB11**, în subcapitolul 7.2;
- **OB12, OB13**, în subcapitolul 7.3;
- **OB14, OB15**, în subcapitolul 7.4.

Rezolvarea numerică a problemei de planificare a traiectoriei prin intermediul metodelor IRRTOPT* și OPT* a scos în evidență următoarele aspecte:

- Folosind OPTACC s-a obținut de fiecare dată soluția optimă sub forma traiectoriei de lungime minimă;
- Timpul de execuție al algoritmului corespunzător metodei OPTACC crește exponențial cu numărul de membri ai formației de zbor și cu numărul de obstacole, iar acest lucru nu convine în cazul aplicațiilor ce rulează în timp real, cum este, de exemplu, scenariul de zbor ilustrat în Figure 1.2, propus pentru atingerea obiectivelor tezei;
- Familia de metode IRRTOPT* a condus la obținerea unui consum energetic mai mic comparativ cu OPT*, atât timp cât numărul de iterații corespunzător componentei IRRT* din compunerea metodei dezvoltate este fixat la o valoare care să asigure un timp de execuție $< 1s$;
- Este necesar să se dezvolte o relație de calcul care să genereze numărul de iterații pentru metoda IRRT*, în funcție de dispunerea obstacolelor precum și de raportul $C_{obst}/C_{spațiu}$.

Capitolul 8

Rezultate și Concluzii

8.1 Concluzii generale

Pentru îndeplinirea celor cincisprezece obiective propuse, teza este structurată pe opt capitole incluzând un capitol introductiv pentru familiarizare cu problematica planificării traiectoriei de zbor, și un capitol final, rezervat rezultatelor și concluziilor, în care sunt punctate și principalele contribuții ale autorului.

8.2 Contribuții

Contribuțiile sunt prezentate sintetic, în cele ce urmează:

Capitolul 1:

- Sintetizarea din literatura de specialitate a informațiilor legate de implicațiile privind planificarea traiectoriei de zbor pentru o formație de UAV-uri;
- Prezentarea conceptelor de planificare cooperativă și distributivă a traiectoriei de zbor pentru o formație de vehicule;
- Propunerea spre rezolvare a unei misiuni de zbor și a scenariului aferent pentru planificarea cooperativă a traiectoriei de zbor pentru o formație de UAV-uri.

Capitolul 2:

- Prezentarea categoriilor importante de metode de planificare a traiectoriei, utilizate atât în cadrul aplicațiilor cu vehicule terestre, cât și cu vehicule aeriene;
- Realizare de software în limbajul Python pentru simularea numerică a metodelor/algoritmilor folosiți în cadrul categoriilor de planificare a traiectoriei;
- Evidențierea și analiza avantajelor și a dezavantajelor metodelor de planificare a traiectoriei pe baza rezultatelor simulărilor numerice a acestora, dar și pe baza consultării literaturii de specialitate;
- Formularea unei propuneri de îmbunătățire a algoritmului IRRT* prin adăugarea de elemente de optimizare numerică.

Capitolul 3:

- Modelarea unui membru al formației de zbor (aparat de zbor de tip aripă rotativă) folosind ecuațiile Newton-Euler;
- Derivarea modelului neliniar al quadcopterului.

Capitolul 4:

- Prezentarea succintă a avantajelor și a dezavantajelor aferente diferiților algoritmi de control, pentru tipurile de regulatoare studiate în vederea realizării controlului zborului;
- Formularea și utilizarea ipotezelor specifice „zborului la punct fix” pentru liniarizarea modelului neliniar propus al quadcopterului;
- Determinarea legilor de control pentru zborul pe o traiectorie 3D folosind regulatorul liniar de tip PD;
- Realizarea de software în limbajul Matlab pentru controlul altitudinii, a mișcării 2D și a mișcării 3D a quadcopterului, urmărind două profile de traiectorie impuse, prima formată din segmente de dreaptă interconectate, iar cea de-a doua de forma unei spirale;
- Identificarea coeficienților din legile de control al zborului prin metoda „trial and error”;
- Evidențierea și discutarea influenței valorilor coeficienților din din legile de control asupra preciziei de urmărire a traiectoriei de zbor de către quadcopter.

Capitolul 5:

- Proiectare platformă aeriană de tip quadcopter și prezentarea componentelor utilizate pentru construirea acesteia;
- Proiectare arhitectură sistem hardware ce echipează quadcopterul (modul electronic de achiziție a datelor de la senzori, modul electronic de comandă și control, modul electronic de comunicație radio, modul electronic de stabilizare a tensiunii de curent continuu);
- Realizarea hardware a sistemului de comandă automată și prezentarea etapizată a versiunilor dezvoltate;
- Implementarea software a legilor de control PD folosind pachetul software „Embedded Coder” din Matlab pentru generarea fișierelor C/C++, necesare compilării și scrierii executabilului pe o arhitectura de microcontroler ARM, STM32F103;
- Implementarea software în limbajul C a driverelor de interfață cu senzorii din compunerea sistemului de comandă automată și a driverelor de comunicație serială, pentru comunicația cu aceștia, precum și cu operatorul uman;
- Realizare de software în limbajul C# a unei interfețe grafice de monitorizare a parametrilor de zbor și de planificare a traiectoriei de zbor (prin fixarea de puncte GPS) pentru quadcopterul construit;
- Derularea de proceduri de testare la sol a sistemului „quadcopter controlat” pentru definitivarea platformei controlate și pregătirea de zbor;

- Folosirea aplicației GoogleEarth și realizarea de software în limbajele Python și Matlab pentru analiza la sol a datelor transmise de la bordul quadcopterului, după urmărirea unor traiectorii 2D (menținerea constantă a altitudinii de zbor) și 3D (modul de zbor implementat software „Return to Home”);
- Analiza numerică și grafică a datelor experimentale în vederea validării funcționării corespunzătoare a algoritmului de control proiectat.

Capitolul 6:

- Particularizarea ecuațiilor Euler-Lagrange pentru rezolvarea problemei de planificare a traiectoriei de zbor pentru quadcopterul construit;
- Elaborare metodă de planificare a traiectoriei de zbor pentru un quadcopter folosind ecuațiile Euler-Lagrange, în condițiile minimizării accelerației de translație;
- Realizare de software în limbajul Matlab pentru implementarea metodei de rezolvare a problemei de planificare a traiectoriei de zbor pentru un quadcopter în condițiile minimizării accelerației de translație;
- Validarea prin simulări numerice și demonstrarea performanțelor metodei de planificare a traiectoriei de zbor cu privire la precizia cu care această traiectorie este urmărită de quadcopter în condițiile de minimizare a accelerației de translație;
- Demonstrarea influenței variației forțelor și a momentelor care acționează asupra structurii quadcopterului, asupra derivatelor de ordin I și II ale accelerației de translație;
- Adaptarea metodei de planificare a traiectoriei de zbor pentru un quadcopter folosind ecuațiile Euler-Lagrange, la condițiile impuse de minimizarea variației accelerației de translație și a momentelor care acționează asupra structurii quadcopterului;
- Realizare de software în limbajul Matlab pentru calcularea traiectoriilor de zbor impuse în condițiile minimizării accelerației de translație, a variației acesteia și a momentelor care acționează asupra structurii quadcopterului;
- Validarea prin simulări numerice și demonstrarea performanțelor metodelor de planificare a traiectoriilor de zbor cu privire la precizia cu care aceste traiectorii sunt urmărite de quadcopter, în condițiile de minimizare a accelerației de translație, a variației acesteia și a momentelor care acționează asupra structurii quadcopterului;
- Deducerea și implementarea software, folosind limbajul Matlab, a relațiilor matematice de aproximare a energiei electrice, consumată de quadcopter folosind valorile forțelor dezvoltate de cele patru ansambluri motor-elice, captate la fiecare $\delta t = 0.004s$ (timpul de execuție al buclei de control a altitudinii);
- Simularea numerică și evaluarea consumului energetic a metodelor de planificare a traiectoriei în diverse scenarii de zbor.

Capitolul 7:

- Introducerea elementelor de optimizare numerică și particularizarea conceptului de optimizare convexă pe cazul planificării traiectoriei de zbor pentru un UAV, simplificând modelul liniarizat al acestuia prin folosirea modelului integratorului dublu;

- Formularea restricțiilor cu privire la evitarea coliziunilor cu un obstacol static de forma unui pătrat sau cerc, precum și cu mai multe obstacole statice de forma unor cercuri, pentru un UAV;
- Adăugarea restricțiilor cu privire la evitarea coliziunilor cu obstacolele statice pentru un UAV și derivarea setului de probleme OPT*, a căror rezolvare implică minimizarea accelerației de translație și a derivatelor de ordin I și II ale acesteia;
- Propunerea setului de probleme IRRTOPT* pentru rezolvarea problemei de planificare a traiectoriei cu minimizarea consumului energetic în condiții de evitare a obstacolelor pentru zborul solitar, ca abordare superioară celei formulate prin setul OPT*;
- Realizare de software în limbajul Python pentru rezolvarea setului de probleme OPT* și IRRTOPT*;
- Evaluarea prin simulare numerică a performanțelor oferite de metodele OPT* și IRRTOPT*, legate atât de precizia urmăririi traiectoriilor impuse, cât și de reducerea consumului energetic, în condițiile de evitare a coliziunilor cu obstacolele statice;
- Introducerea restricțiilor cu privire la evitarea coliziunilor între membrii formației de zbor și particularizarea metodelor OPT* și IRRTOPT* pentru simularea unui zbor pentru două UAV-uri;
- Evaluarea prin simulări numerice a performanțelor oferite de metodele OPT* și IRRTOPT*, legate atât de precizia urmăririi traiectoriilor impuse, cât și de reducerea consumului energetic, în condițiile de evitare a coliziunilor între membrii formației de zbor;
- Derivarea expresiilor specifice metodelor OPTACC și IRRTOPTACC pentru rezolvarea planificării traiectoriei de zbor cu evitarea coliziunilor atât între membrii formației, cât și cu obstacolele statice.
- Evaluarea prin simulări numerice a performanțelor oferite de metoda OPTACC versus IRRTOPTACC, legate atât de precizia urmăririi traiectoriilor impuse, cât și de reducerea consumului energetic, în condițiile de evitare a coliziunilor între membrii formației de zbor.

8.3 Perspective de dezvoltare ulterioară

Printre direcțiile de continuare a cercetării, menționăm studiul asupra prelucrării datelor de navigație prin dezvoltarea de algoritmi complecși care să fie evaluați în ceea ce privește precizia de urmărire a traiectoriilor de zbor calculate prin metodele OPT* și IRRTOPT* dezvoltate. Acest lucru necesită testarea experimentală a metodelor OPT* și IRRTOPT* dezvoltate, pentru planificarea traiectoriei formației de zbor formată din quadcoptere, în condițiile de minimizare a consumului energetic și cu evitarea coliziunilor între membrii formației și cu obstacolele statice. În acest sens, va fi nevoie de integrarea atât hardware, cât și software, a unui senzor GPS diferențial pentru a reduce eroarea în poziție.

Bibliografie

- [1] D. González, J. Pérez, V. Milanés, and F. Nashashibi, “A review of motion planning techniques for automated vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [2] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. D. Lee, A. Stewart, P. Vernaza, J. C. Derenick, J. R. Spletzer, and B. Satterfield, “Little ben: The ben franklin racing team’s entry in the 2007 DARPA urban challenge,” in *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic, George Air Force Base, Victorville, California, USA*, ser. Springer Tracts in Advanced Robotics, M. Buehler, K. Iagnemma, and S. Singh, Eds., vol. 56. Springer, 2009, pp. 231–255. [Online]. Available: https://doi.org/10.1007/978-3-642-03991-1_6
- [3] Y.-L. Chen, V. Sundareswaran, C. Anderson, A. Broggi, P. Grisleri, P. P. Porta, P. Zani, and J. Beck, “Terramax™: Team oshkosh urban robot,” *Journal of Field Robotics*, vol. 25, no. 10, pp. 841–860, 2008. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20267>
- [4] B. Patz, Y. Papeilis, R. Pillat, G. Stein, and D. Harper, “A practical approach to robotic design for the darpa urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 528–566, 2008.
- [5] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, “Constraint-based planning and control for safe, semi-autonomous operation of vehicles,” in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 383–388.
- [6] A. Bacha, “Odin: Team victortango’s entry in the darpa urban challenge,” *J. Field Robot*, vol. 25, no. 8, pp. 467–492, Aug. 2008.
- [7] R. Kala and K. Warwick, “Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization,” *J Intell Robot Syst*, vol. 72, pp. 559–590, 2013.
- [8] C. Urmson, “Autonomous driving in urban environments: Boss and the urban challenge,” *J. Field Robot*, vol. 25, no. 8, pp. 425–466, Aug. 2008.
- [9] D. Ferguson, T. Howard, and M. Likhachev, “Motion planning in urban environments,” *J. Field Robot*, vol. 25, no. 11, pp. 939–960, Nov./Dec. 2008.

- [10] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *Int. J. Robot. Res.*, vol. 28, no. 8, pp. 933–945, Aug. 2009.
- [11] M. Montemerlo, "Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, Sep. 2008.
- [12] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, Apr. 2010.
- [13] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, May 2005.
- [14] J. Ren, K. McIsaac, and R. Patel, "Modified newton's method applied to potential field-based navigation for mobile robots," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 384–391, apr 2006.
- [15] Y. bo Chen, G. chen Luo, Y. song Mei, J. qiao Yu, and X. long Su, "UAV path planning using artificial potential field method updated by optimal control theory," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407–1420, jun 2014.
- [16] B. Di, R. Zhou, and H. Duan, "Potential field based receding horizon motion planning for centrality-aware multiple UAV cooperative surveillance," *Aerospace Science and Technology*, vol. 46, pp. 386–397, oct 2015.
- [17] Y. Kuwata, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Sep. 2009.
- [18] D. Braid, A. Broggi, and G. Schmiedel, "The terramax autonomous vehicle," *J. Field Robot.*, vol. 23, no. 9, pp. 693–708, Sep. 2006.
- [19] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt*," in *Proc. IEEE ICRA*, 2011, pp. 1478–1483.
- [20] J. Jeon, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proc. ACC*, Jun. 2013, pp. 188–193.
- [21] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, no. 2, 2010.
- [22] Jonathan D. Gammell and Siddhartha S. Srinivasa and Timothy D. Barfoot, "Batch informed trees (BITstar): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015.

- [23] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed RRTstar: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014.
- [24] S. Choudhury, J. D. Gammell, T. D. Barfoot, S. S. Srinivasa, and S. A. Scherer, “Regionally accelerated batch informed trees (rabit*): A framework to integrate local information into optimal path planning,” *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4207–4214, 2016.
- [25] L. Han, H. Yashiro, H. Tehrani Nik Nejad, Q. H. Do, and S. Mita, “Bézier curve based path planning for autonomous vehicle in urban environment,” in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 1036–1042.
- [26] Z. Liang, G. Zheng, and J. Li, “Automatic parking path optimization based on bezier curve fitting,” in *2012 IEEE International Conference on Automation and Logistics*, 2012, pp. 583–587.
- [27] S. Thrun, “Stanley: The robot that won the darpa grand challenge,” *J. Field Robot*, vol. 23, no. 9, pp. 661–692, Sep. 2006.
- [28] A. Piazzzi, C. G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, “Quintic g/sup 2/-splines for the iterative steering of vision-based autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27–36, 2002.
- [29] P. Trepagnier, J. Nagel, P. Kinney, C. Koutsougeras, and M. Dooner, “Kat-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain,” *J. Field Robot*, vol. 23, no. 8, pp. 509–526, Aug. 2006.
- [30] T. Gu and J. Dolan, *On-Road Motion Planning for Autonomous Vehicles*. Berlin, Germany: Springer-Verlag, 2012, pp. 588–597.
- [31] J. Ziegler, P. Bender, T. Dang, and C. Stiller, “Trajectory planning for bertha—a local, continuous method,” in *Proc. IEEE Intell. Veh. Symp*, 2014, pp. 450–457.
- [32] R.-V. Mihai and M.-M. Bivolaru, “Cooperative distributed trajectory optimization for a heterogeneous UAV formation.” Author(s), 2018.
- [33] R.-V. Mihai, C. Vidan, A.-D. Radu, and A. Gavril, “Vision based cooperative distributed collision avoidance estimation in a dynamic environment for a heterogeneous UAV formation,” in *CENTRAL EUROPEAN SYMPOSIUM ON THERMOPHYSICS 2019 (CEST)*. AIP Publishing, 2019.
- [34] R.-V. Mihai and A.-M. Stoica, “On the development of an onboard real time system for the members of a heterogeneous UAV formation acting as LPWAN gateways,” in *CENTRAL EUROPEAN SYMPOSIUM ON THERMOPHYSICS 2019 (CEST)*. AIP Publishing, 2019.

- [35] Latombe, *Robot motion planning*. Boston: Kluwer Academic Publishers, 1991.
- [36] I. Noreen, A. Khan, and Z. Habib, “A comparison of rrt, rrt* and rrt*-smart path planning algorithms,” 2016.
- [37] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [38] J. Choi, R. Curry, and G. Elkaim, “Path planning based on bézier curve for autonomous ground vehicles,” in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, 2008, pp. 158–166.
- [39] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, no. 2, jun 1994.
- [40] T. Schouwenaars, A. Richards, E. Feron, and J. How, “Plume avoidance maneuver planning using mixed integer linear programming,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, aug 2001.
- [41] T. Schouwenaars, B. D. Moor, E. Feron, and J. How, “Mixed integer programming for multi-vehicle path planning,” in *2001 European Control Conference (ECC)*. IEEE, sep 2001.
- [42] T. Schouwenaars, E. Feron, and J. How, “Multi-vehicle path planning for non-line of sight communication,” in *2006 American Control Conference*. IEEE, 2006.
- [43] A. Richards and J. How, “Aircraft trajectory planning with collision avoidance using mixed integer linear programming,” in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*. IEEE, 2002.
- [44] N. P. Nguyen and S. K. Hong, “Sliding mode thau observer for actuator fault diagnosis of quadcopter UAVs,” *Applied Sciences*, vol. 8, no. 10, p. 1893, oct 2018.
- [45] P. Rucz, Z. Belso, B. Gati, I. Koller, and A. Turóczy, “Design and implementation of nonlinear control systems for rotary and fixed wing uavs,” 2016.
- [46] I. Grujić and R. Nilsson, “Model-based development and evaluation of control for complex multi-domain systems: Attitude control for a quadrotor uav,” 2016.
- [47] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, mar 2004.
- [48] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer London, 2008.